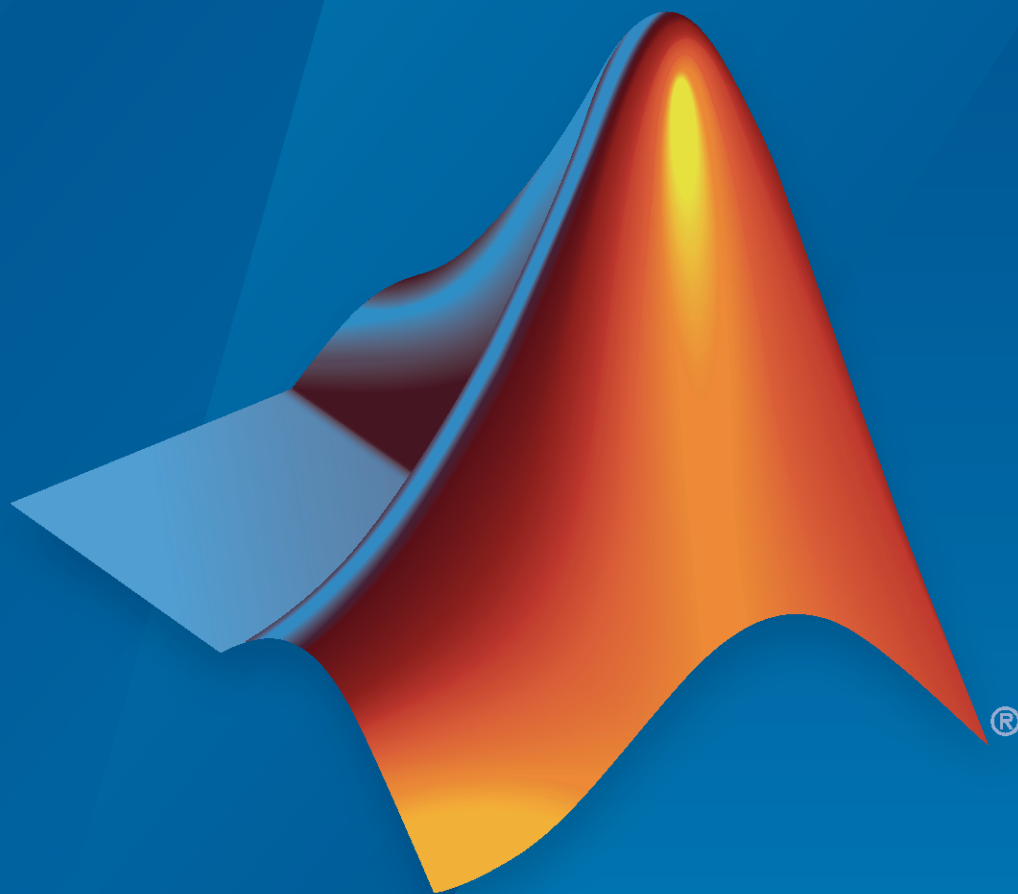


**Simscape™ Electrical™**  
Reference



**MATLAB® & SIMULINK®**

R2022b



## How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

*Simscape™ Electrical™ Reference*

© COPYRIGHT 2013–2022 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](http://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](http://www.mathworks.com/patents) for more information.

## Revision History

September 2013	Online only	New for Version 6.0 (Release 2013b)
March 2014	Online only	Revised for Version 6.1 (Release 2014a) (Renamed from <i>SimPowerSystems™ Reference (Third Generation)</i> )
October 2014	Online only	Revised for Version 6.2 (Release 2014b)
March 2015	Online only	Revised for Version 6.3 (Release 2015a)
September 2015	Online only	Revised for Version 6.4 (Release 2015b)
March 2016	Online only	Revised for Version 6.5 (Release 2016a) (Renamed from <i>SimPowerSystems™ Reference (Simscape™ Components)</i> )
September 2016	Online only	Revised for Version 6.6 (Release 2016b)
March 2017	Online only	Revised for Version 6.7 (Release 2017a)
September 2017	Online only	Revised for Version 6.8 (Release 2017b)
March 2018	Online only	Revised for Version 6.9 (Release 2018a)
September 2018	Online only	Revised for Version 7.0 (Release 2018b) (Renamed from <i>Simscape™ Power Systems™ Reference (Simscape™ Components) and Simscape™ Electronics™ Reference</i> )
March 2019	Online only	Revised for Version 7.1 (Release 2019a) (Renamed from <i>Simscape™ Electrical™ Reference (Electronics, Mechatronics, and Power Systems)</i> )
September 2019	Online only	Revised for Version 7.2 (Release 2019b)
March 2020	Online only	Revised for Version 7.3 (Release 2020a)
September 2020	Online only	Revised for Version 7.4 (Release 2020b)
March 2021	Online only	Revised for Version 7.5 (Release 2021a)
September 2021	Online only	Revised for Version 7.6 (Release 2021b)
March 2022	Online only	Revised for Version 7.7 (Release 2022a)
September 2022	Online only	Revised for Version 7.8 (Release 2022b)



**1** | Blocks

**2** | Functions

**3** | Tools and Apps

**A** | Parameter Dependencies

<b>Parameter Dependencies</b> .....	<b>A-2</b>
Parameter Dependency Tables .....	<b>A-2</b>



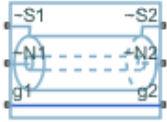
# Blocks

---

## AC Cable (Three-Phase)

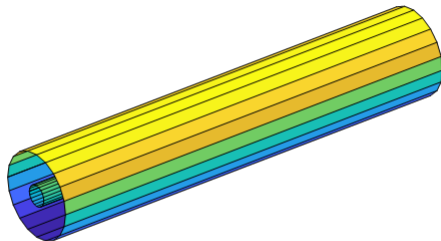
Three-phase AC power cable

**Library:** Simscape / Electrical / Passive / Lines



### Description

The AC Cable (Three-Phase) block represents a three-phase AC power cable with a conducting sheath surrounding each phase. The figure shows a single-phase conductor inside a conducting sheath. The inner cylinder represents the main conductor for the phase, and the outer cylinder represents the conducting sheath.



You can model composite or expanded three-phase ports by setting the **Modeling option** parameter to either:

- **Composite three-phase ports** — Contains three-phase connection ports for the sheaths and phases and a single-phase connection port for each electrical reference node.
- **Expanded three-phase ports** — Contains single-phase connection ports for each sheath, phase, and electrical reference node.

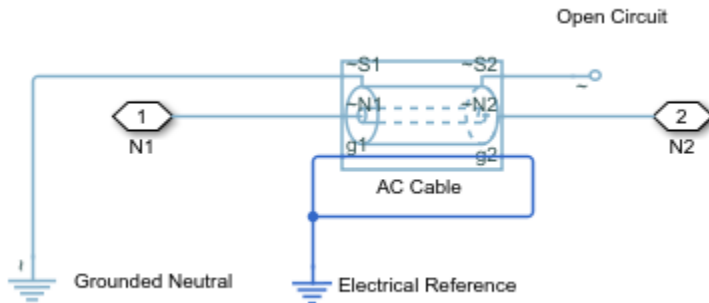
The AC Cable (Three-Phase) block includes inductances and mutual inductances between each phase, sheath, and return path. Therefore, you can connect an ideal electrical reference block to both return ports, **g1** and **g2**, while maintaining loss modeling in the Earth- or neutral-return line.

To facilitate simulation convergence when you connect the AC Cable (Three-Phase) block to a source block, include source impedance using one of these methods:

- Configure the source block to include impedance.
- Insert a block that models impedance between the source block and the AC Cable (Three-Phase) block.

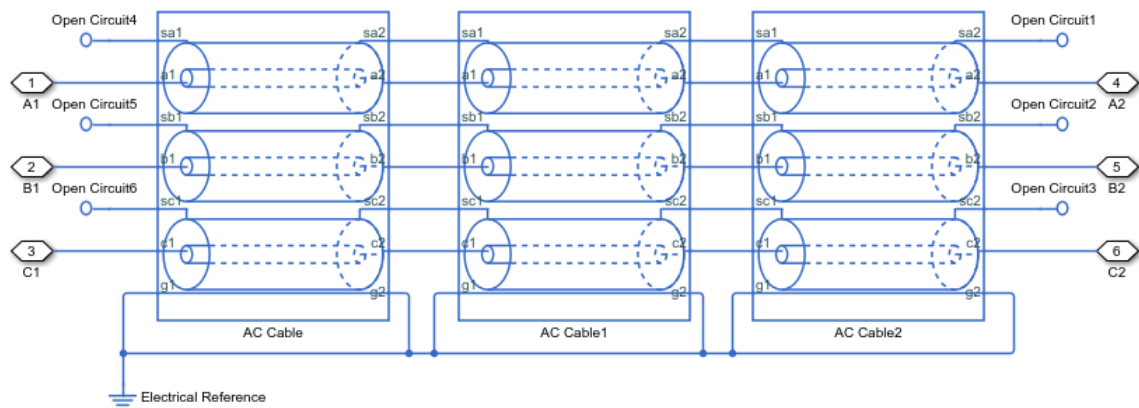


To model unbonded sheaths, connect the unbonded sheaths to an Open Circuit (Three-Phase) block. The figure shows a model of single-point bonding when you set **Modeling option** to Composite three-phase ports.

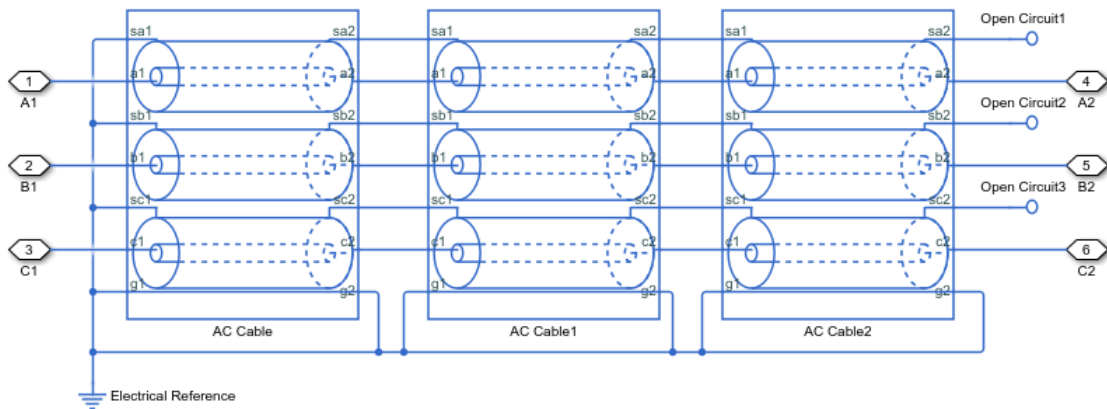


For high performance modeling, in terms of simulation speed, use a single AC Cable (Three-Phase) block. To improve model fidelity in terms of frequency behavior, connect several AC Cable (Three-Phase) blocks in series. For series-connected blocks, the sheaths and main conductors act as coupled transmission lines with perfect transposition of the phases. The number of AC Cable (Three-Phase) blocks that you use to model a particular physical length of cable must be less than the number of transpositions in the physical system that you are modeling. Types of continuous multi-segment cables that you can model include:

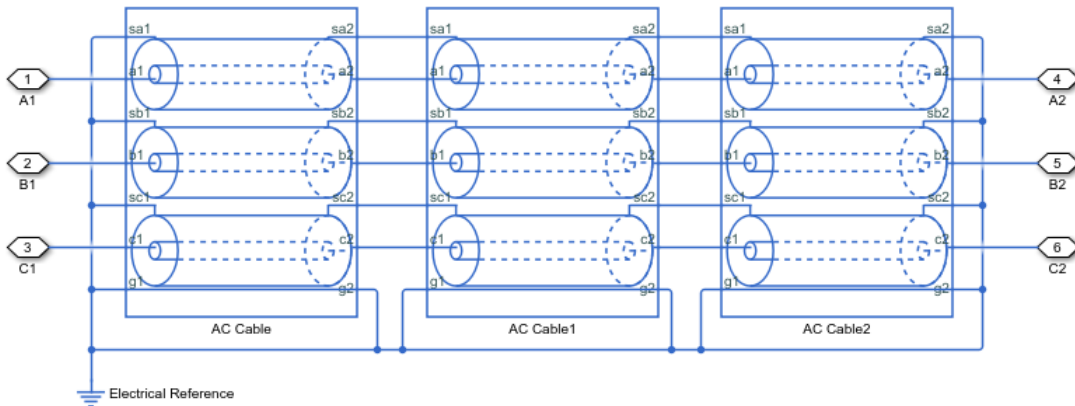
- Unbonded continuous cables



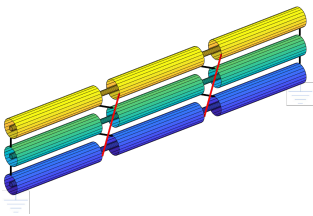
- Single-point bonded continuous cables



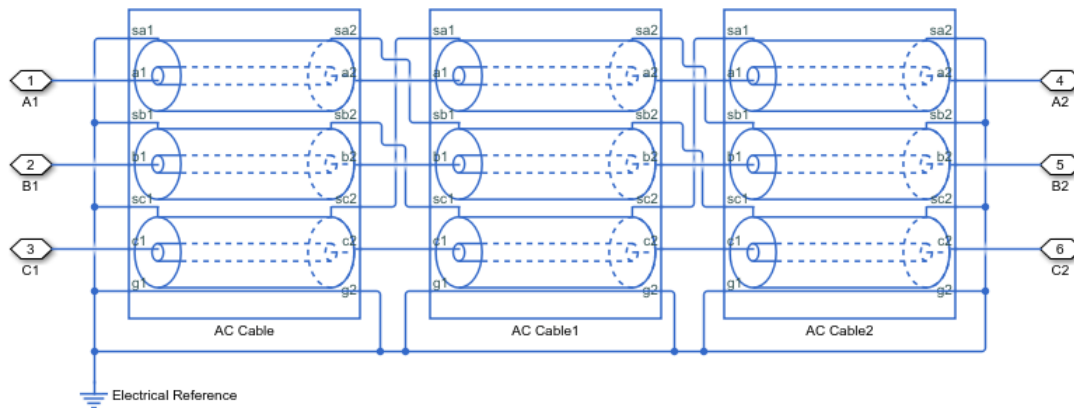
- Double-point bonded continuous cables



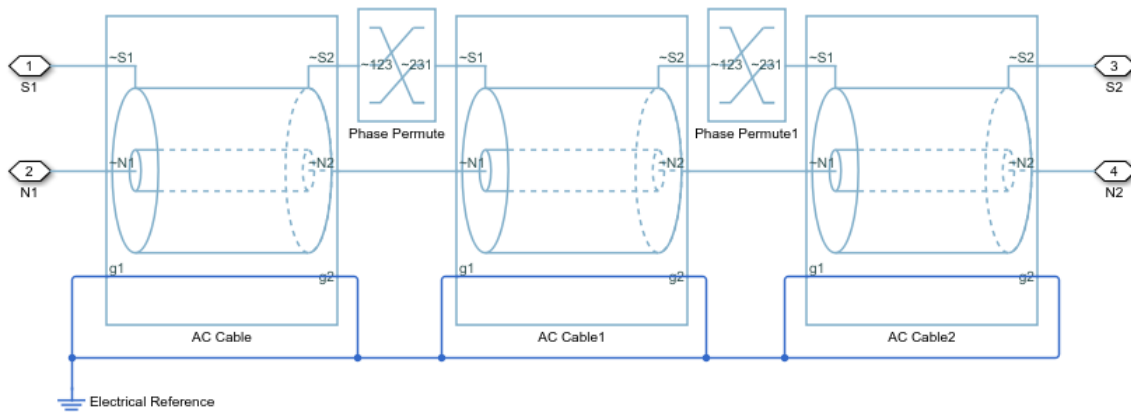
You can also model cross-bonded cables using the AC Cable (Three Phase) block.



This three pi-segment cable model implements cross-bonding using expanded three-phase ports and single-phase connection lines. The sheath in the model is two-point bonded.



This model of blocks with composite three-phase-ports uses Phase Permute blocks to implement cross bonding. The sheath in the model is unbonded.



For an example that allows you to choose the number of segments and type of bonding, see “AC Cable with Bonded Sheaths”.

### Three-Phase AC Cable Model

The AC Cable (Three-Phase) block uses the concept of partial inductances to calculate the inductance values. These values include the partial self-inductance of each phase, sheath, and return path and the partial mutual inductances between each:

- Phase and each other phase
- Phase and the sheath of that phase
- Phase and the sheath of neighboring phases
- Phase and the return
- Sheath and each neighboring sheath
- Sheath and the return

For three equivalent phases, the matrix that defines the resistance relationships for the vector [phase A; sheath A; phase B; sheath B; phase C; sheath C] is

$$R = \begin{bmatrix} R_a + R_g & R_g & R_g & R_g & R_g & R_g \\ R_g & R_s + R_g & R_g & R_g & R_g & R_g \\ R_g & R_g & R_a + R_g & R_g & R_g & R_g \\ R_g & R_g & R_g & R_s + R_g & R_g & R_g \\ R_g & R_g & R_g & R_g & R_a + R_g & R_g \\ R_g & R_g & R_g & R_g & R_g & R_s + R_g \end{bmatrix}$$

$$R_a = R'_a l$$

$$R_g = R'_{\text{return}} l,$$

for which  $R'_{\text{return}}$  depends on the return parameterization method such that:

- For a return parameterization based on distance and resistance  $R'_{\text{return}} = R'_g$ .
- For a return parameterization based on frequency and Earth resistivity  $R'_{\text{return}} = \pi^2 10^{-7} f$

and

$$R_s = R'_s l,$$

where:

- $R$  is the resistance matrix.
- $R_a$  is the resistance of a particular phase.
- $R_s$  is the resistance of a particular sheath.
- $R_g$  is the resistance of the Earth- or neutral-return.
- $R'_a$  is the resistance per unit length for the phase.
- $l$  is the cable length.
- $R'_s$  is the resistance per unit length for the sheath.
- $R'_{\text{return}}$  is the resistance per unit length of the return. The value of  $R'_{\text{return}}$  varies depending on the return parameterization method.
- $R'_g$  is the resistance per unit length for the Earth- or neutral return.
- $f$  is the frequency that the block uses to calculate Earth-return parameters if you parameterize the block using the frequency and Earth resistivity method.

The block uses standard expressions to calculate the capacitances between:

- Concentric or adjacent cylinders
- Each phase and its own sheath
- Each sheath and the return

The matrix that defines these capacitance relationships is

$$C = \begin{bmatrix} C_{as_a} & -C_{as_a} & 0 & 0 & 0 & 0 \\ -C_{as_a} & C_{as_a} + C_{s_ag} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{as_a} & -C_{as_a} & 0 & 0 \\ 0 & 0 & -C_{as_a} & C_{as_a} + C_{s_ag} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{as_a} & -C_{as_a} \\ 0 & 0 & 0 & 0 & -C_{as_a} & C_{as_a} + C_{s_ag} \end{bmatrix}$$

$$C_{as_a} = \frac{2\pi\epsilon_r\epsilon_0 l}{\ln\left(\frac{r_s}{r_a}\right)}$$

$$r_a = GMR \cdot e^{\frac{1}{4}}$$

$$C_{s_ag} = \frac{2\pi\epsilon_{env}\epsilon_0 l}{\ln\left(\frac{r_{cable}}{r_{s,outer}}\right)}$$

where:

- $C$  is the capacitance matrix.
- $C_{as_a}$  is the capacitance between each phase and the sheath of that phase.
- $C_{s_ag}$  is the capacitance between each sheath and return.
- $\epsilon_r$  is the permittivity of the dielectric.
- $\epsilon_0$  is the permittivity of free space.
- $r_s$  is the radius of the sheath.
- $r_a$  is the effective radius of the conductor. For a single-strand conductor,  $r_a$  is the radius of the strand.
- $r_{cable}$  is the cable radius and  $r_{cable}$  is greater than  $r_{s,outer}$ .
- $GMR$  is the geometric mean radius of the conductor. For a single-strand conductor,  $GMR = r_{strand}e^{-\frac{1}{4}}$ , where  $r_{strand}$  is the radius of the strand.
- $\epsilon_{env}$  is the permittivity of the material between the sheathed lines and the return path.

The block uses the concept of partial inductances to calculate inductance values. These values include the partial self-inductance of each phase, sheath, and return path and the partial mutual inductances between each:

- Phase and each other phase
- Phase and the sheath of that phase
- Phase and the sheath of neighboring phases
- Phase and the return
- Sheath and each neighboring sheath
- Sheath and the return

The equations that define these inductance relationships are:

$$L = \begin{bmatrix} D_a & \delta & A & \alpha & A & \alpha \\ \delta & D_s & \alpha & S & \alpha & S \\ A & \alpha & D_a & \delta & A & \alpha \\ \alpha & S & \delta & D_s & \alpha & S \\ A & \alpha & A & \alpha & D_a & \delta \\ \alpha & S & \alpha & S & \delta & D_s \end{bmatrix}$$

$$D_a = L_a - M_{ag}$$

$$L_a = 2 \times 10^{-7} l \left[ \ln\left(\frac{2l}{r_a}\right) - \frac{3}{4} \right]$$

$$M_{ag} = M_{sg} = 2 \times 10^{-7} l \left[ \ln\left(\frac{2l}{D_{\text{return}}}\right) - 1 \right]$$

for which  $D_{\text{return}}$  depends on the return parameterization method such that:

- For a return parameterization based on distance and resistance  $D_{\text{return}} = D_e$ .
- For a return parameterization based on frequency and Earth resistivity  $D_{\text{return}} = 1650 \sqrt{\frac{\rho}{2\pi f}}$ .

$$D_s = L_s - M_{sg}$$

$$L_s = M_{asa} = 2 \times 10^{-7} l \left[ \ln\left(\frac{2l}{r_s}\right) - \frac{3}{4} \right]$$

$$\delta = M_{asa} - M_{ag}$$

$$\alpha = M_{asb} - M_{ag}$$

$$M_{asb} = M_{sasb} = M_{ab} = 2 \times 10^{-7} l \left[ \ln\left(\frac{2l}{d_{ab}}\right) - 1 \right],$$

for which  $d_{ab}$  depends on the line formation parameterization method, such that:

- For a trefoil line formation parameterization  $d_{ab} = D_{ab}$ .
- For a flat line formation parameterization  $d_{ab} = D_{ab} \sqrt[3]{2}$ .

$$A = M_{ab} - M_{ag}$$

$$S = M_{sasb} - M_{sg},$$

where:

- $L$  is the inductance matrix.
- $D_a$  is the self-inductance of a single phase through its entire path and return.
- $L_a$  is the partial self-inductance of each phase.
- $M_{ag}$  is the partial mutual inductance between each phase and the Earth- or neutral-return.
- $M_{sg}$  is the partial mutual inductance between each sheath and the Earth- or neutral-return.
- The factor,  $2 \times 10^{-7}$  is equal to  $\mu_0/2\pi$ , because permeability of free space,  $\mu_0$ , is equal to  $1.257 \times 10^{-6}$  or  $4\pi \times 10^{-7}$  H/m.

- $D_s$  is the self-inductance of a single sheath through its entire path and return.
- $L_s$  is the partial self-inductance of each sheath.
- $M_{as_a}$  is the partial mutual inductance between each phase and the sheath of that phase.
- $\delta$  is the effective mutual inductance between a phase and the sheath of that phase.
- $\alpha$  is the effective mutual inductance between a phase and a neighboring sheath.
- $M_{as_b}$  is the partial mutual inductance between each phase and the sheath of each neighboring phase.
- $M_{s_a s_b}$  is the partial mutual inductance between sheaths of different phases.
- $M_{ab}$  is the partial mutual inductance between each phase and each other phase.
- $D_{return}$  is the effective distance to the return. The value of  $D_{return}$  varies if you use the distance/return parameterization method.
- $D_e$  is the effective distance to the Earth- or neutral-return.
- $\rho$  is the effective Earth resistivity for an Earth-return.
- $f$  is the frequency that is used to determine the return path properties.
- $d_{ab}$  is the effective distance between adjacent phases. The value of  $d_{ab}$  varies depending on the line parameterization method.
- $D_{ab}$  is the center-to-center distance between adjacent phases.
- $A$  is the effective mutual inductance between phases.
- $S$  is the effective mutual inductance between sheaths.

A modal transformation that is related to the Clarke transform simplifies the equivalent circuit. The six-by-six transformation,  $T$ , is

$$T = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & \sqrt{2} & 0 & 0 \\ 1 & 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{\sqrt{3}}{2} \\ 1 & 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{\sqrt{3}}{2} \end{bmatrix}.$$

As  $T^\dagger = T^{-1}$ , applying the  $T$  transform yields the modal resistance matrix,  $R_m$ , the modal capacitance matrix,  $C_m$ , and the modal inductance matrix,  $L_m$ .

The transformed matrices are:

$$R_m = T^\dagger R T = \begin{bmatrix} R_a + 3R_g & 3R_g & 0 & 0 & 0 & 0 \\ 3R_g & R_s + 3R_g & 0 & 0 & 0 & 0 \\ 0 & 0 & R_a & 0 & 0 & 0 \\ 0 & 0 & 0 & R_s & 0 & 0 \\ 0 & 0 & 0 & 0 & R_a & 0 \\ 0 & 0 & 0 & 0 & 0 & R_s \end{bmatrix}$$

$$C_m = T^\dagger C T = \begin{bmatrix} C_{as_a} & -C_{as_a} & 0 & 0 & 0 & 0 \\ -C_{as_a} & C_{as_a} + C_{s_a g} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{as_a} & -C_{as_a} & 0 & 0 \\ 0 & 0 & -C_{as_a} & C_{as_a} + C_{s_a g} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{as_a} & -C_{as_a} \\ 0 & 0 & 0 & 0 & -C_{as_a} & C_{as_a} + C_{s_a g} \end{bmatrix} = C$$

$$L_m = T^\dagger L T = \begin{bmatrix} D_a + 2A & \delta + 2\alpha & 0 & 0 & 0 & 0 \\ \delta + 2\alpha & D_s + 2A & 0 & 0 & 0 & 0 \\ 0 & 0 & D_a - A & \delta - \alpha & 0 & 0 \\ 0 & 0 & \delta - \alpha & D_s - S & 0 & 0 \\ 0 & 0 & 0 & 0 & D_a - A & \delta - \alpha \\ 0 & 0 & 0 & 0 & \delta - \alpha & D_s - S \end{bmatrix}.$$

The transformation changes each six-by-six matrix into three uncoupled two-by-two matrices. The capacitance matrix is invariant under this transformation. The power is invariant in the transformed and untransformed domains because  $T$  is unitary.

### Assumptions and Limitations

- For resistance calculations, the phases are equivalent.
- Relative to the phase-to-sheath capacitance and the sheath-return capacitances all other capacitances, are negligible due to the shielding provided by the conducting sheaths.

## Ports

### Conserving

#### ~S1 – Sheath

electrical

Expandable three-phase electrical conserving port associated with sheath 1.

#### ~N1 – Phase

electrical

Expandable three-phase electrical conserving port associated with  $a$ ,  $b$ , and  $c$  phases 1.

#### g1 – Ground

electrical

Electrical conserving port associated with ground 1.

#### ~S2 – Sheath

electrical

Expandable three-phase electrical conserving port associated with sheath 2.



**~N2 — Phase**

electrical

Expandable three-phase electrical conserving port associated with *a*, *b*, and *c* phases 2.

**g2 — Ground**

electrical

Electrical conserving port associated with ground 2.

**Parameters****Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Cable length — Length**

120 km (default)

Length of the cable.

**Geometric mean radius of conductor — Radius**

5 mm (default)

Geometric mean radius of the conductor, which is a function of the number and type of individual strands in the conductor of the AC cable.

**Sheath radius — Radius**

10 mm (default)

Average radius of the sheath. To ensure that the sheath radius is greater than the physical radius of a single-stranded conductor with a particular GMR, the sheath radius must be greater than  $GMR * e^{\frac{1}{4}}$ .

**Outer cable radius — Outer radius**

20 mm (default)

Outer radius of the cable, in mm. The cable radius must be greater than the **Sheath radius** parameter. This ensures that the sheath and conductor are both enclosed inside an insulating outer cable layer.

**Line-line spacing (center-to-center) — Distance**

25 mm (default)

Distance between the line centers.

**Line formation — Line configuration**

Trefoil (default) | Flat

Cable line formation.

**Conductor resistance per length – Resistance**

1 Ohm/km (default)

Resistance per length of a conductor.

**Sheath resistance per length – Resistance**

10 Ohm/km (default)

Resistance per length of a sheath.

**Insulation relative permittivity – Permittivity**

2.4 (default)

Relative permittivity of the insulation.

**Relative permittivity between lines and return path – Relative permittivity**

1 (default)

Relative permittivity of the circuit.

**Return parameterization – Model**

Use frequency and Earth resistivity (default) | Use distance and resistance

Parameterization method.

**Dependencies**

Enabling either option enables other parameters.

**Frequency for Earth-return impedance – Frequency**

60 Hz (default)

Frequency at which the Earth-return impedance is calculated.

**Dependencies**

Selecting Use frequency and Earth resistivity for the **Return parameterization** parameter enables this parameter.

**Earth resistivity – Resistance**

100 m\*Ohm (default)

Earth-return resistivity.

**Dependencies**

Selecting Use frequency and Earth resistivity for the **Return parameterization** parameter enables this parameter.

**Effective distance to return path – Return path distance**

1 km (default)

Effective distance between the phases and the return path.

**Dependencies**

Selecting Use distance and resistance for the **Return parameterization** parameter enables this parameter.

**Return path resistance per length — Return path unit resistance**

0.1 Ohm/km (default)

Resistance per length of the return path.

**Dependencies**

Selecting Use distance and resistance for the **Return parameterization** parameter enables this parameter.

**Version History**

**Introduced in R2017b**

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

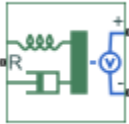
**See Also**

Phase Permute | Open Circuit (Three-Phase)

# Accelerometer

Behavioral model of MEMS accelerometer

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Accelerometer block implements a behavioral model of a MicroElectroMechanical Systems (MEMS) accelerometer. For the default output type `Voltage level`, the accelerometer provides an output voltage that is proportional to the acceleration rate presented at the mechanical translational physical port R. The output voltage is limited according to the values that you provide for maximum and minimum output voltage.

The block also has an alternative output type, `PWM duty cycle`. With this choice, the output of the block is a PWM signal with a duty cycle that is proportional to the measured acceleration. You can limit the variation in duty cycle to a specified range.

Optionally, you can model sensor dynamics by setting the **Dynamics** parameter to `Model sensor bandwidth`. Including dynamics adds a first-order lag between the angular rate presented at port R and the corresponding voltage applied to the electrical + and - ports.

If running your simulation with a fixed-step solver, or generating code for hardware-in-the-loop testing, MathWorks recommends that you set the **Dynamics** parameter to `No dynamics – Suitable for HIL`, because this avoids the need for a small simulation time step if the sensor bandwidth is high.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The **Measured acceleration** variable target specifies the initial output for the sensor.

## Ports

### Conserving

#### R — Translational port

mechanical

Mechanical conserving port associated with the accelerometer translational port.

#### **+ — Positive port**

electrical

Electrical conserving port associated with the accelerometer positive terminal.

#### **- — Negative port**

electrical

Electrical conserving port associated with the accelerometer negative terminal.

## Parameters

### **Output type — Block output type**

Voltage level (default) | PWM duty cycle

Select one of the following options to define the block output type:

- **Voltage level** — The amplitude of the output voltage is proportional to the measured acceleration. This is the default option.
- **PWM duty cycle** — The duty cycle (on time divided by the pulse total time) is proportional to the measured acceleration.

### **Sensitivity — Sensitivity**

1000 mV/gee (default)

Change in output voltage level per unit change in acceleration when the output is not being limited.

#### **Dependencies**

This parameter is visible only when you select **Voltage level** for the **Output type** parameter.

### **Output voltage for zero acceleration — Output voltage for zero acceleration**

2.5 V (default)

The output voltage from the sensor when the acceleration is zero.

#### **Dependencies**

This parameter is visible only when you select **Voltage level** for the **Output type** parameter.

### **Maximum output voltage — Maximum output voltage**

4 V (default)

The maximum output voltage from the sensor, which determines the sensor maximum measured positive acceleration.

#### **Dependencies**

This parameter is visible only when you select **Voltage level** for the **Output type** parameter.

### **Minimum output voltage — Minimum output voltage**

1 V (default)

The minimum output voltage from the sensor, which determines the sensor maximum measured negative acceleration. This parameter is only visible when you select `Voltage` level for the **Output type** parameter. The default value is 1 V.

**Duty cycle sensitivity (percent per unit acceleration) – Duty cycle sensitivity**  
10 1/gee (default)

The change in duty cycle per unit acceleration. Duty cycle is expressed as a percentage of the PWM period.

**Dependencies**

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

**Duty cycle for zero acceleration (percent) – Duty cycle for zero acceleration**  
50 (default)

The duty cycle output by the sensor when the acceleration is zero.

**Dependencies**

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

**Maximum duty cycle (percent) – Maximum duty cycle**  
75 (default)

The maximum duty cycle output by the sensor. Increasing acceleration levels beyond this point will not register an increase in duty cycle.

**Dependencies**

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

**Minimum duty cycle (percent) – Minimum duty cycle**  
25 (default)

The minimum duty cycle output by the sensor. Decreasing acceleration levels beyond this point will not register a decrease in duty cycle.

**Dependencies**

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

**PWM frequency – PWM frequency**  
1000 Hz (default)

The frequency of the output pulse train.

**Dependencies**

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

**Output voltage amplitude – Output voltage amplitude**  
5 V (default)

The amplitude of the output pulse train when high.

### Dependencies

This parameter is visible only when you select `PWM duty cycle` for the **Output type** parameter.

### Dynamics — Dynamics

`No dynamics – Suitable for HIL (default) | Model sensor bandwidth`

Select one of the following options for modeling sensor dynamics:

- `No dynamics – Suitable for HIL` — Do not model sensor dynamics. Use this option when running your simulation fixed step or generating code for hardware-in-the-loop testing, because this avoids the need for a small simulation time step if the sensor bandwidth is high. This is the default option.
- `Model sensor bandwidth` — Model sensor dynamics with a first-order lag approximation, based on the **Bandwidth** parameter value. You can control the initial condition for the lag by specifying the **Measured acceleration** variable target.

### Bandwidth — Bandwidth

`3 kHz (default)`

Specifies the 3dB bandwidth for the measured acceleration assuming a first-order time constant.

### Dependencies

This parameter is visible only when you select `Model sensor bandwidth` for the **Dynamics** parameter.

## Version History

**Introduced in R2012b**

## Extended Capabilities

### C/C++ Code Generation

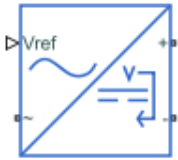
Generate C and C++ code using Simulink® Coder™.

## See Also

## AC-DC Converter (Three-Phase)

Behavioral model of AC-DC converter

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



### Description

This block represents the behavioral model of an AC-DC converter. Use this block to model AC-DC converters without simulating individual switching events. Use the DC voltage reference input to convert the electrical energy between the AC and DC sides.

This block is compatible with time simulation mode and frequency-and-time simulation mode.

### Equations

#### DC Side

At the DC side, if you set the **Voltage dynamics at DC side** parameter to **No dynamics**, the block calculates the output voltage,  $V_{DC}$ , by using this equation:

$$V_{DC} = V_{ref} - DI_{DC},$$

where  $V_{ref}$  is the value of the input signal at the **Vref** port,  $D$  is the value of the **Output DC voltage droop with output current** parameter, and  $I_{DC}$  is the output DC current.

If you set the **Voltage dynamics at DC side** parameter to **Specify voltage regulation time constant**, the block calculates the output voltage,  $V_{DC}$ , by using this equation:

$$\tau_{DC}\dot{V}_{DC} + V_{DC} = V_{ref} - DI_{DC},$$

where  $\tau_{DC}$  is the value of the **Voltage regulation time constant at DC side** parameter.

The power at the DC side is

$$P_{DC} = V_{DC}I_{DC}.$$

#### AC Side

If you set the **Power balance dynamics** parameter to **No dynamics**, the input power at the AC side is equal to the output power at the DC side.

$$P_{AC} = P_{DC}.$$

If you set the **Power balance dynamics** parameter to **Specify power time constant**, the block calculates the input power at the AC side by using this equation:



$$\tau_P \dot{P}_{AC} + P_{AC} = P_{DC},$$

where  $\tau_P$  is the value of the **Power balance time constant** parameter.

To determine the three-phase input currents at the AC side, the block first calculates the  $dq$ -axes voltages,  $v_d$  and  $v_q$ , using the Park transform on the  $abc$ -phase voltages,  $v_a$ ,  $v_b$ , and  $v_c$ :

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin\theta & \sin(\theta - 120^\circ) & \sin(\theta + 120^\circ) \\ \cos\theta & \cos(\theta - 120^\circ) & \cos(\theta + 120^\circ) \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

where  $\theta = 2\pi F_{\text{rated}} \cdot t$  is the angle of the synchronously rotating reference frame.

The block then calculates the three-phase AC currents,  $i_a$ ,  $i_b$ , and  $i_c$ , by using these power equations:

$$P_{AC} = \frac{3}{2}(v_d i_d + v_q i_q)$$

$$Q_{AC} = 0 = \frac{3}{2}(v_d i_q - v_q i_d).$$

From these equations, the  $dq$ -axes currents,  $i_d$  and  $i_q$ , are

$$i_d = \frac{2}{3} \frac{P_{AC} v_{dx}}{v_{dx}^2 + v_{qx}^2}$$

$$i_q = \frac{2}{3} \frac{P_{AC} v_{qx}}{v_{dx}^2 + v_{qx}^2},$$

where  $v_{dx}$  and  $v_{qx}$  are obtained from these equations:

$$\tau_{AC} \dot{v}_{dx} + v_{dx} = v_d$$

$$\tau_{AC} \dot{v}_{qx} + v_{qx} = v_q,$$

where  $\tau_{AC}$  is the value of the **Voltage time constant at AC side** parameter.

Finally, the block calculates the three-phase AC currents that flow into the converter by using this equation:

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ \sin(\theta - 120^\circ) & \cos(\theta - 120^\circ) \\ \sin(\theta + 120^\circ) & \cos(\theta + 120^\circ) \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix}.$$

## Limitations

The AC-DC Converter (Three-Phase) block does not model the power losses.

## Ports

### Input

#### Vref — DC voltage reference

physical

Physical signal input port associated with the external signal of the DC voltage reference.

### **Conserving**

#### **~ — Three-phase port**

electrical

Expandable three-phase port.

#### **Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a — a-phase**

electrical

Electrical conserving port associated with the *a*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b — b-phase**

electrical

Electrical conserving port associated with the *b*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c — c-phase**

electrical

Electrical conserving port associated with the *c*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the positive terminal.

#### **- — Negative terminal**

electrical

Electrical conserving port associated with the negative terminal.

### **Parameters**

#### **Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Option to model composite or expanded three-phase ports.

**Output DC voltage droop with output current — Voltage droop at 1 A**

1V/A (default) | nonnegative scalar

Amount of voltage that the output DC voltage drops from the set point for an output current of 1 A.

**Voltage dynamics at DC side — Dynamics model at DC side**

No dynamics (default) | Specify voltage regulation time constant

Option to include voltage regulation dynamics. If you set this parameter to `No dynamics`, the block does not consider the voltage regulation dynamics. If you set this parameter to `Specify voltage regulation time constant`, the block adds a first-order lag to the equation that defines the value of the DC voltage source.

**Voltage regulation time constant at DC side — Voltage dynamics time constant at DC side**

0.02 s (default) | positive scalar

Time constant associated with the voltage transients at the DC side when the load current at the DC side is stepped.

**Dependencies**

To enable this parameter, set **Voltage dynamics at DC side** to `Specify voltage regulation time constant`.

**Initial output voltage at DC side — Output voltage at DC side at time zero**

100 V (default) | scalar

DC side output voltage at time zero. To initialize the model with no transients when it delivers a steady-state load current, use this parameter to set the initial DC side output voltage.

**Dependencies**

To enable this parameter, set **Voltage dynamics at DC side** to `Specify voltage regulation time constant`.

**Rated electrical frequency at AC side — Rated electrical frequency at AC side**

60 Hz (default) | positive scalar

Rated electrical frequency at AC side.

**Minimum AC side voltage (phase-to-phase RMS) — Minimum AC-side voltage**

80 V (default) | positive scalar

Minimum value of the phase-to-phase RMS AC-side voltage for the converter to produce a DC output voltage.

**Voltage time constant at AC side — Voltage time constant at AC-side**

0.01 s (default) | positive scalar

Time constant associated with the voltage measurement transients at the AC side.

**Initial voltage at AC side (phase-to-phase RMS) — AC-side initial voltage**

100 V (default) | positive scalar

Phase-to-phase RMS voltage at the AC-side at time zero. The value of this parameter must be greater than the value of the **Minimum AC side voltage (phase-to-phase RMS)** parameter.

**Initial voltage angle at AC side — Initial voltage angle of a-phase at AC side**

0 deg (default) | scalar

Initial voltage angle of the *a*-phase at time zero for the AC side.

**Power balance dynamics — Option to include power balance dynamics**

Specify power time constant (default) | No dynamics

Whether to include power balance dynamics. If you set this parameter to `No dynamics`, the block does not consider power balance dynamics. If you set this parameter to `Specify power time constant`, the block adds a first-order lag to the equation that defines the AC-side power. When you enable the dynamics, a DC-side output power change results in a transient change in the input power at the AC side.

**Power balance time constant — Time constant for power balance dynamics**

0.01 s (default) | positive scalar

Time constant associated with the transients of the power balance between the AC and DC side.

**Dependencies**

To enable this parameter, set **Power balance dynamics** to `Specify power time constant`.

**Initial active power — Initial active power at AC side**

1e3 W (default) | scalar

Initial active power at time zero at the AC side.

**Dependencies**

To enable this parameter, set **Power balance dynamics** to `Specify power time constant`.

**Parasitic conductance to electrical reference — Parasitic conductance to electrical reference**

1e-6 S (default) | nonnegative scalar

Parasitic conductance to the electrical reference.

## Version History

Introduced in R2022b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Average-Value Rectifier (Three-Phase) | Average-Value Voltage Source Converter (Three-Phase)

# Average-Value Chopper

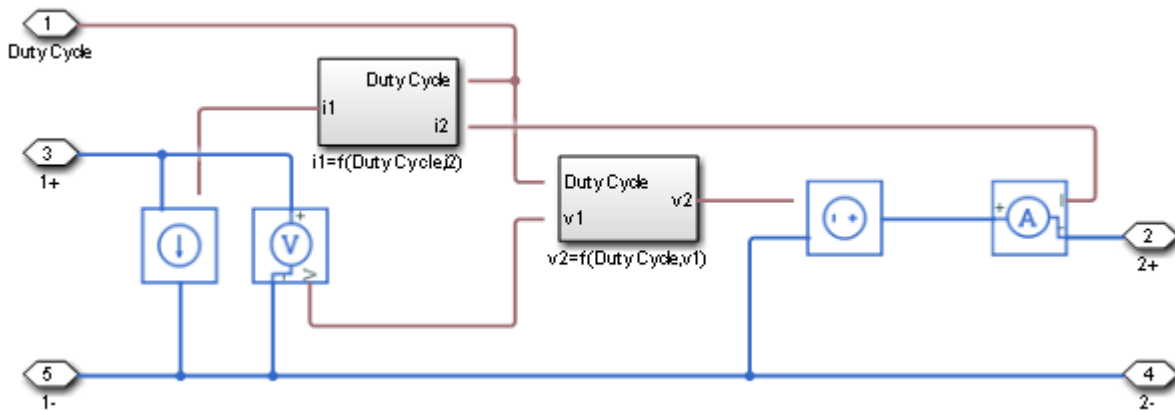
Average-value chopper

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Average-Value Chopper block represents a controlled average-value chopper. Use the duty cycle input to convert the electrical energy between the two sides. The figure shows the equivalent circuit for the block.



## Equations

The input current and output voltage depend on the chopper class that you specify.

### Voltage and Current Equations

Chopper Class	Quadrants	Output Voltage, $v_2$			Input Current, $i_1$		
		$i_2 < 0$	$i_2 = 0$	$i_2 > 0$	$i_2 < 0$	$i_2 = 0$	$i_2 > 0$
<b>A</b>	1 <sup>st</sup>	$v_2 = v_1$	$v_2 = DutyCycle \cdot v_1$		$i_1 = i_2$	$i_1 = DutyCycle \cdot i_2$	
<b>B</b>	2 <sup>nd</sup>	$v_2 = (1 - DutyCycle) \cdot v_1$		$v_2 = 0$	$i_1 = (1 - DutyCycle) \cdot i_2$		$i_1 = 0$
<b>C</b>	1 <sup>st</sup> and 2 <sup>nd</sup>	$v_2 = DutyCycle \cdot v_1$			$i_1 = DutyCycle \cdot i_2$		
<b>D</b>	1 <sup>st</sup> and 4 <sup>th</sup>	$v_2 = v_1$	$v_2 = (2 \cdot DutyCycle - 1) \cdot v_1$		$i_1 = i_2$	$i_1 = (2 \cdot DutyCycle - 1) \cdot i_2$	
<b>E</b>	Four	$v_2 = (2 \cdot DutyCycle - 1) \cdot v_1$			$i_1 = (2 \cdot DutyCycle - 1) \cdot i_2$		

### Limitations and Assumptions

- Input voltage,  $v_1$  is positive.
- Power losses are neglected.
- The Average-Value Chopper block represents an ideal implementation of a controlled average-value chopper with no dynamics, and it is suitable for low fidelity simulations. For higher fidelity models, use the One-Quadrant Chopper, Two-Quadrant Chopper, or Four-Quadrant Chopper blocks.

## Ports

### Conserving

#### DutyCycle – Duty Cycle

electrical | scalar

Electrical conserving port associated with the duty cycle.

Data Types: double

#### 1+ – Positive DC voltage 1

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

#### 1- – Negative DC voltage 1

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

#### 2+ – Positive DC voltage 2

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

**2- – Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

**Parameters****Chopper type – Chopper class**

Class A - first quadrant (default) | Class B - second quadrant | Class C - first and second quadrant | Class D - first and fourth quadrant | Class E - four quadrant

Chopper class.

**Converter efficiency – Converter efficiency parameterization**

Constant (default) | Tabulated

Specify the parameterization of the converter efficiency. If you select Tabulated, conduction losses will depend on the provided output current.

**Efficiency (%) – Converter efficiency**

100 (default) | scalar

Efficiency of the converter, in percentage.

**Dependencies**

To enable this parameter, set **Converter efficiency** to Constant.

**Output current vector – Vector of output current**

[0, 2, 5] A (default) | row vector of at least two elements

Vector of output currents.

**Dependencies**

To enable this parameter, set **Converter efficiency** to Tabulated.

**Efficiency vector (%) – Vector of efficiencies**

[90, 95, 98] (default) | row vector of at least two elements

Vector of efficiencies for each output current specified in **Output current vector**, in percentage. This parameter must have the same size of **Output current vector**.

**Dependencies**

To enable this parameter, set **Converter efficiency** to Tabulated.

**Version History**

Introduced in R2018b

## References

[1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics*. 2nd Ed. Hoboken, NJ: John Wiley & Sons Inc., 2010.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

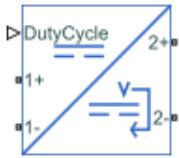
Average-Value DC-DC Converter | Average-Value Inverter (Three-Phase) | Average-Value Rectifier (Three-Phase) | Four-Quadrant Chopper | One-Quadrant Chopper | Two-Quadrant Chopper



# Average-Value DC-DC Converter

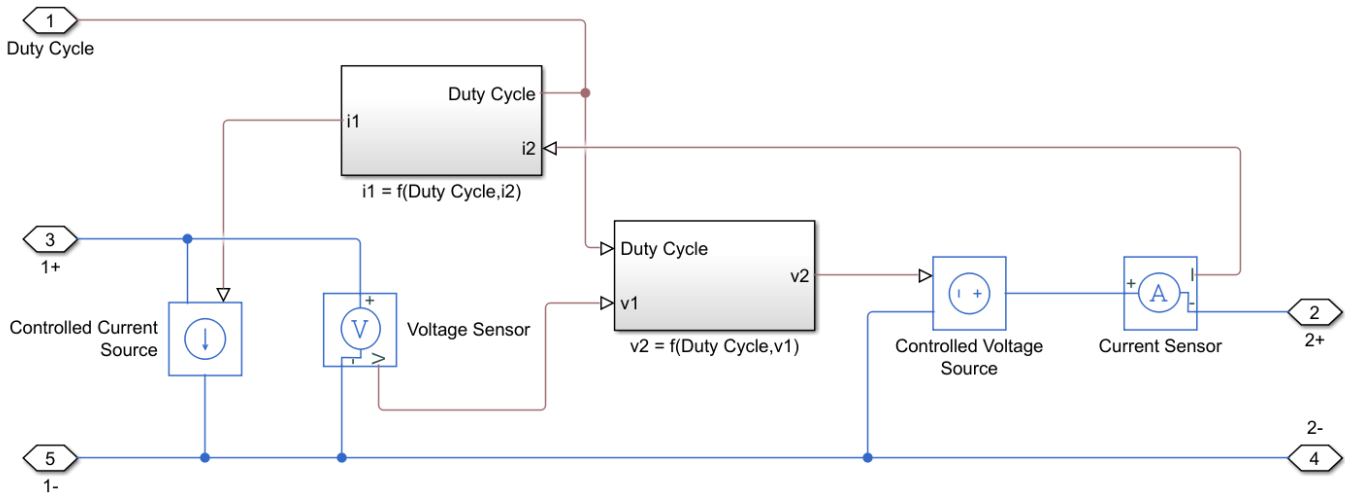
Average-value DC-DC converter

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Average-Value DC-DC Converter block represents a controlled average-value DC-DC converter. You can program the block as a buck converter, boost converter, or buck-boost converter by providing the duty cycle. The diagram shows the equivalent circuit for the block with duty cycle as input. The converter contains a controlled current source and a controlled voltage source. Use the duty cycle, the current reference, or the voltage reference ports as control input to convert the electrical energy between the connected components on either side of the converter.



## Equations

If you set the **Control input** parameter to Duty cycle, the input current and output voltage are a function of the duty cycle and depend on the converter type.

## Voltage and Current Equations

Converter Type	Output Voltage, $v_2$	Input Current, $i_1$
<b>Buck</b>	$v_2 = DutyCycle \cdot v_1$	$i_1 = DutyCycle \cdot i_2$
<b>Boost</b>	$v_2 = \frac{v_1}{1 - DutyCycle}$	$i_1 = \frac{i_2}{1 - DutyCycle}$
<b>Buck-Boost</b>	$v_2 = \frac{DutyCycle \cdot v_1}{1 - DutyCycle}$	$i_1 = \frac{DutyCycle \cdot i_2}{1 - DutyCycle}$

If you set the **Control input** parameter to `Current` reference, the converter sets the output current and it computes the voltage.

Similarly, if you set the **Control input** parameter to `Voltage` reference, the converter sets the output voltage and it computes the current.

### Limitations and Assumptions

- The input voltage is positive.
- All converter types use the same polarity for input and output.

## Ports

### Conserving

#### DutyCycle — Duty cycle

physical | scalar

Electrical conserving port associated with the duty cycle.

#### Dependencies

To enable this port, set **Control Input** to `Duty Cycle`.

Data Types: `double`

#### Iref — Current reference

physical

Electrical conserving port associated with the current reference.

#### Dependencies

To enable this port, set **Control Input** to `Current` reference.

Data Types: `double`

#### Vref — Voltage reference

physical

Electrical conserving port associated with the voltage reference.

#### Dependencies

To enable this port, set **Control Input** to `Voltage` reference.

Data Types: double

### **1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

### **1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

### **2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

### **2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

## **Parameters**

### **Control input — Control input**

Duty cycle (default) | Current reference | Voltage reference

Specify the control input to convert the electrical energy between the two sides.

### **Converter type — Converter type**

Buck converter (default) | Buck-boost converter | Boost converter

Type of converter.

### **Dependencies**

To enable this parameter, set **Control Input** to Duty Cycle.

### **Converter efficiency — Converter efficiency parameterization**

Constant (default) | Tabulated

Specify the parameterization of the converter efficiency. If you select Tabulated, conduction losses will depend on the provided output current.

---

**Note** For tabulated efficiency, the extrapolation method is nearest. Therefore, the efficiency does not go above the highest value or below the lowest value in the **Efficiency vector (%)** parameter.

---

**Efficiency (%) – Converter efficiency**

100 (default) | scalar

Efficiency of the converter, in percentage.

**Dependencies**To enable this parameter, set **Converter efficiency** to Constant.**Output current vector – Vector of output current**

[0, 2, 5] A (default) | row vector of at least two elements

Vector of output currents.

**Dependencies**To enable this parameter, set **Converter efficiency** to Tabulated.**Efficiency vector (%) – Vector of efficiencies**

[90, 95, 98] (default) | row vector of at least two elements

Vector of efficiencies for each output current specified in **Output current vector**, in percentage. This parameter must have the same size of **Output current vector**.**Dependencies**To enable this parameter, set **Converter efficiency** to Tabulated.

## Version History

Introduced in R2018b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

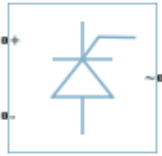
**See Also**

Average-Value Chopper | Average-Value Inverter (Three-Phase) | Average-Value Rectifier (Three-Phase) | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Boost Converter | Converter (Three-Phase) | Rectifier (Three-Phase) | Three-Level Converter (Three-Phase)

## Average-Value Inverter (Three-Phase)

Average-value DC Voltage to three-phase AC voltage converter with fixed power loss

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters

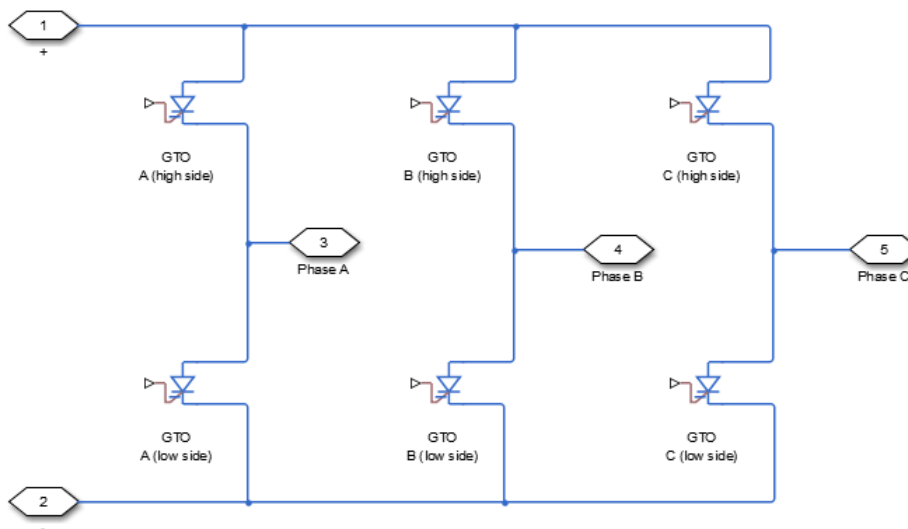


### Description

The Average-Value Inverter (Three-Phase) block models an average-value, full-wave inverter. It converts DC voltage to three-phase AC voltages and converts three-phase AC power demand to DC power demand. The corresponding DC power demand is equal to the sum of the fixed power loss and the AC power demand.

You can use the Average-Value Inverter (Three-Phase) block only as a full-wave inverter. It behaves as a DC-voltage-controlled AC voltage source. The ratio you specify determines the ratio between the DC voltage and the AC voltage.

The figure shows the equivalent circuit for the inverter as a full-wave inverter. The Average-Value Inverter (Three-Phase) block does not yield the harmonics that are typically associated with the detailed representation, however, because it performs an average-value power conversion.



### Electrical Defining Equations

The voltages are defined by

$$v_{DC} = v_p - v_n,$$

$$v_{ref} = \frac{v_p + v_n}{2},$$

$$v_{RMS} = v_{ratio}v_{DC},$$

$$V_0 = \frac{\sqrt{2}}{\sqrt{3}}V_{RMS},$$

$$v_a = V_0\sin(2\pi ft + \varphi) + v_{ref},$$

$$v_b = V_0\sin(2\pi ft + \varphi - 120^\circ) + v_{ref},$$

and

$$v_c = V_0\sin(2\pi ft + \varphi + 120^\circ) + v_{ref},$$

where:

- $v_p$  and  $v_n$  are the voltages at the positive and negative terminals of the inverter.
- $v_{DC}$  is the voltage difference between the positive and negative terminals of the inverter.
- $v_{ref}$  is the DC offset.
- $V_{ratio}$  is the ratio of rated AC voltage to rated DC voltage for the inverter. See the **Ratio of rated AC voltage to rated DC voltage** parameter in “Parameters” on page 1-34 for the  $V_{ratio}$  values for common inverter control modes.
- $V_{RMS}$  is the RMS AC line-line voltage.
- $V_0$  is the peak phase voltage.
- $f$  is the frequency.
- $t$  is the time.
- $\varphi$  is the phase shift.
- $v_a, v_b, v_c$  are the respective AC phase voltages.

The power, resistance, and currents are defined by

$$P_{AC} = -v_a i_a - v_b i_b - v_c i_c,$$

$$R_{DC} = \frac{v_{DC}^2}{P_{AC} + P_{fixed}},$$

and

$$i = \frac{v_{DC}}{R_{DC}},$$

where:

- $i_a, i_b,$  and  $i_c$  are the respective AC phase currents flowing into the inverter.
- $P_{AC}$  is the power output on the AC side.  $P_{AC}$  has a minimum limit of 0 W.
- $P_{fixed}$  is the fixed power loss that you specify on the block.
- $R_{DC}$  is the resistance on the DC side.

- $i$  is the current flowing from the positive to the negative terminals of the inverter.

The inverter starts to create an AC voltage, that is turns on, when the DC supply voltage is above the value that you specify for **DC voltage for turn on** parameter. It stops inverting, that is turns off, when the DC supply voltage falls below the value that you specify for **DC voltage for turn off** parameter. When the inverter turns off, the block sets the output AC current to zero.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the positive terminal

#### - – Negative terminal

electrical

Electrical conserving port associated with the negative terminal

#### ~ – Three-phase port

electrical

Expandable three-phase port

### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### a – a-phase

electrical

Electrical conserving port associated with  $a$ -phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### b – b-phase

electrical

Electrical conserving port associated with  $b$ -phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### c – c-phase

electrical

Electrical conserving port associated with  $c$ -phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

## Parameters

### Electrical connection — Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

### Rated AC frequency — AC frequency in Hz

60Hz (default)

AC frequency, specified in Hz (where Hz is defined as 1/s). For example, kHz and MHz are valid units, but rad/s is not.

### Phase shift — Phase shift

0deg (default)

Phase shift in angular units.

### Ratio of rated AC voltage to rated DC voltage — Ratio of rated AC voltage to rated DC voltage

$\sqrt{6}/\pi$  (default)

The table shows ratios for common three-phase two-level inverter control modes. The default value is  $\sqrt{6}/\pi$ .

For 180° and 120° conduction modes, the listed voltages are the fundamental RMS values of line-line voltages. For other methods, the listed voltages are the maximum fundamental RMS values of line-line voltages.

You can control the output voltage of the inverter according to specific requirements. DPWM includes 30° DPWM, 60° DPWM, and 120° DPWM. For details, see references [3] and [4].

Control Method	$V_{RMS} (line-line)$	Ratio of $V_{RMS} (line-line)$ to $V_{DC}$
180° conduction mode [1]	$\frac{\sqrt{6}}{\pi} V_{DC}$	0.7797
120° conduction mode [1]	$\frac{3}{\sqrt{2}\pi} V_{DC}$	0.6752
Hysteresis current control [2]	$\left(\frac{\sqrt{3}}{\sqrt{2}}\right) \left(\frac{2V_{DC}}{\pi}\right)$	0.7797
Sinusoidal PWM (SPMW) [2]	$\left(\frac{\sqrt{3}}{\sqrt{2}}\right) \left(\frac{V_{DC}}{2}\right)$	0.6124
Space vector modulation (SVM) [2]	$\left(\frac{\sqrt{3}}{\sqrt{2}}\right) \left(\frac{V_{DC}}{\sqrt{3}}\right)$	0.7071
Discontinuous PWM (DPWM) [3], [4]	$\left(\frac{\sqrt{3}}{\sqrt{2}}\right) \left(\frac{V_{DC}}{\sqrt{3}}\right)$	0.7071
Convert to the original AC voltage of the average-value rectifier	$\left(\frac{\pi}{\sqrt{2}}\right) \left(\frac{V_{DC}}{3}\right)$	0.7405



**Fixed power loss — Fixed power loss**

1e3W (default)

Minimum power drawn on the DC side. The default value is 1e3.

**DC voltage for turn on — DC voltage for turn on**

100V (default)

When the DC supply voltage rises above this value, the inverter produces an AC output voltage.

**DC voltage for turn off — DC voltage for turn off**

75V (default)

When the DC supply voltage falls below this value, the inverter turns off and the block sets the output AC currents to zero.

## Version History

**Introduced in R2015a****Electrical connection ports update***Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Rashid, M. H. *Pulse-Width-Modulation Inverters*. Upper Saddle River, NJ: Prentice-Hall, 2004, pp. 237-248.
- [2] Krause, P. C., O. Wasynczuk, and S. D. Sudhoff. *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ: IEEE Press, 2002.
- [3] Chung, D. W., J. S. Kim, and S. K. Kul. "Unified voltage modulation technique for real-time three-phase power conversion." *IEEE Transactions on Industry Applications*. Vol. 34, no. 2, 1998, pp. 374-380.
- [4] Hava, A. M., R. J. Kerkman, and T. A. Lipo. "Simple analytical and graphical methods for carrier-based PWM-VSI drives." *IEEE Transactions on Power Electronics*. Vol. 14, 1999, no. 1, pp. 49-61.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Average-Value Chopper | Average-Value DC-DC Converter | Average-Value Rectifier (Three-Phase) | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Boost Converter | Converter (Three-Phase) | Rectifier (Three-Phase) | Three-Level Converter (Three-Phase)

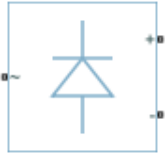
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

## Average-Value Rectifier (Three-Phase)

Average-value three-phase AC voltage to DC voltage converter with fixed power loss

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters

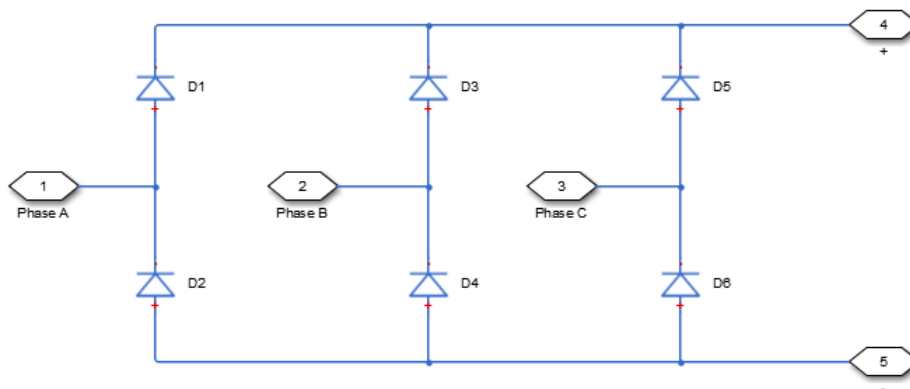


### Description

The Average-Value Rectifier (Three-Phase) block models an average-value, full-wave, six-pulse rectifier. It converts instantaneous three-phase AC voltages to DC voltage and DC power demand to three-phase AC power demand. The corresponding AC power demand is equal to the sum of the fixed power loss and the DC power demand.

You can use the Average-Value Rectifier (Three-Phase) block only as a six-pulse rectifier. You cannot combine two Average-Value Rectifier blocks to represent a twelve-pulse rectifier.

The figure shows the equivalent circuit for the rectifier as a full-wave, six-pulse rectifier. The Average-Value Rectifier (Three-Phase) block does not yield the harmonics that are typically associated with the detailed representation, however, because it performs an average-value power conversion.



This block can work in both time and frequency-and-time simulation modes. If you set the **AC frequency** parameter to **Variable**, this block works only in time simulation mode. If you select **Constant**, this block works in both time and frequency-time simulation modes. For more information, see “Frequency and Time Simulation Mode”.

### Electrical Defining Equations

The voltages are defined by:

$$v_{ref} = \frac{v_a + v_b + v_c}{3},$$

$$V_{RMS} = \sqrt{\frac{(v_a - v_b)^2 + (v_b - v_c)^2 + (v_c - v_a)^2}{3}},$$

$$v_{DC} = 3\frac{\sqrt{2}}{\pi}V_{RMS},$$

$$v_p = v_{ref} + \frac{v_{DC}}{2},$$

and

$$v_n = v_{ref} - \frac{v_{DC}}{2},$$

where:

- $v_a, v_b, v_c$  are the respective AC phase voltages.
- $v_{ref}$  is the DC offset on the AC side. In a balanced AC power system with no DC bias,  $v_{DC}$  is 0 V.
- $V_{RMS}$  is the RMS AC line-line voltage.
- $v_{DC}$  is the voltage difference between the positive and negative terminals of the rectifier.
- $3\sqrt{2}/\pi$  is the  $v_{DC}/V_{RMS}$  ratio for a full-wave, six-pulse rectifier.
- $v_p, v_n$  are the voltages at the positive and negative terminals of the rectifier.

The resistance, power, and currents are defined by

$$R_{fixed} = \frac{V_{Rated}^2}{P_{fixed}},$$

$$P_{DC} = -v_p i_p - v_n i_n,$$

$$R_{AC} = \frac{V_{RMS}^2}{P_{DC} + \frac{V_{RMS}^2}{R_{fixed}}},$$

and

$$[i_a \ i_b \ i_c] = \frac{[v_a \ v_b \ v_c] - v_{ref}}{R_{AC}},$$

where:

- $V_{Rated}$  is the rated AC voltage that you specify on the block mask.
- $P_{fixed}$  is the fixed power loss that you specify on the block mask.
- $R_{fixed}$  is the fixed per-phase series resistance in an equivalent wye-connected load.
- $i_p, i_n$  are the currents flowing into the positive and negative terminals of the rectifier.
- $P_{DC}$  is the power output on the DC side.  $P_{DC}$  has a minimum limit of 0 W.
- $R_{AC}$  is the per-phase series resistance in an equivalent wye-connected load.
- $i_a, i_b, i_c$  are the respective AC phase currents flowing into the rectifier.

## Ports

### Conserving

#### ~ – Three-phase port

electrical

Expandable three-phase port

#### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### a – a-phase

electrical

Electrical conserving port associated with *a*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### b – b-phase

electrical

Electrical conserving port associated with *b*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### c – c-phase

electrical

Electrical conserving port associated with *c*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### + – Positive terminal

electrical

Electrical conserving port associated with the positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the negative terminal.

## Parameters

### Electrical connection – Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**AC frequency — AC frequency**

Variable (default) | Constant

Whether to have a variable or constant AC frequency:

- **Variable** — If the frequency at the AC side varies, this option only works in time simulation mode.
- **Constant** — If the frequency at the AC side is constant, this option works in both time and frequency-and-time simulation modes.

**Rated AC voltage — Rated AC voltage**

4160V (default)

Rated voltage of the AC system.

**Rated electrical frequency — Rated electrical frequency**

60 Hz | scalar

Rated electrical frequency. The value of this parameter must be equal to the value of the rated electrical frequencies of other sinusoidal sources in your model, if present.

**Dependencies**

To enable this parameter, set **AC frequency** to Constant.

**Fixed power loss — Fixed power loss**

1e3W (default)

Minimum power drawn on the AC side at rated AC voltage. When the instantaneous AC voltage is equal to the value you specify for the **Rated AC voltage**, the AC power demand equals the value you specify for the **Fixed power loss** plus DC power demand.

## Version History

**Introduced in R2014b****Electrical connection ports update***Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

[Average-Value Chopper](#) | [Average-Value DC-DC Converter](#) | [Average-Value Inverter \(Three-Phase\)](#) | [Bidirectional DC-DC Converter](#) | [Buck Converter](#) | [Buck-Boost Converter](#) | [Boost Converter](#) | [Converter \(Three-Phase\)](#) | [Rectifier \(Three-Phase\)](#) | [Three-Level Converter \(Three-Phase\)](#)

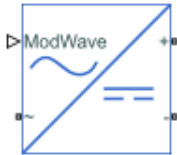
## **Topics**

[“Expand and Collapse Three-Phase Ports on a Block”](#)

# Average-Value Voltage Source Converter (Three-Phase)

Average-value bidirectional AC/DC voltage source converter

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Average-Value Voltage Source Converter (Three-Phase) block converts electrical energy from AC to DC voltage or from DC to AC voltage according to an input three-phase modulation wave. The corresponding input power is equal to the sum of the fixed power loss and the output power.

This block can work in both time and frequency-and-time simulation modes. If you set the **AC frequency** parameter to **Variable**, this block works only in time simulation mode. If you select **Constant**, this block works in both time and frequency-time simulation modes. For more information, see “Frequency and Time Simulation Mode”.

## Losses Parameterization

Switching losses, conduction losses, and quiescent losses are the main heat sources for a converter.

The switching losses are defined by this equation:

$$P_{switching} = k_s v_{dc} I_{rms}$$

where:

- $k_s$  is the proportionality constant that depends on the turn-on and turn-off intervals and switching frequency. Specify this value by setting the **Switching losses coefficient, ks** parameter.

- $v_{dc}$  is the dc-link voltage.

- $I_{rms} = \frac{\sqrt{(i_a - i_{dc})^2 + (i_b - i_{dc})^2 + (i_c - i_{dc})^2}}{\sqrt{3}}$  is the root mean square (RMS) phase current, where

$$i_{dc} = \frac{i_a + i_b + i_c}{3}.$$

The conduction losses are defined by this equation:

$$P_{conduction} = k_{c1} I_{rms} + k_{c2} I_{rms}^2$$

where:



- $k_{c1}$  is the coefficient of the conduction losses that depends on the on-state zero current collector-emitter voltage of the transistor and on the forward voltage drop of the diode. Specify this value by setting the **Conduction losses coefficient, kc1** parameter.
- $k_{c2}$  is the coefficient of the conduction losses that depends on the state resistance of the transistor and on the anti-parallel diode. Specify this value by setting the **Conduction losses coefficient, kc2** parameter.

The quiescent losses are defined by the **Fixed power loss** parameter,  $P_{fixed}$ .

The sum of the switching, conduction, and quiescent losses define the total power losses of the converter:

$$P_{loss} = P_{switching} + P_{conduction} + P_{fixed}.$$

If not available, you can also obtain the  $k_s$ ,  $k_{c1}$ ,  $k_{c2}$  and  $P_{fixed}$  parameters values from the power losses profile, by setting the **Losses parameterization** parameter to **Profile: loss=f(Irms,vdc\_nom)**. The block then solves this equation and calculates the values of the parameters:

$$\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} 1 & v_{dc\_nom} I_{rms,1} & I_{rms,1} & I_{rms,1}^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & v_{dc\_nom} I_{rms,n} & I_{rms,n} & I_{rms,n}^2 \end{bmatrix} \begin{bmatrix} P_{fixed} \\ k_s \\ k_{c1} \\ k_{c2} \end{bmatrix}$$

where  $\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix}$  is the vector of power loss values, **Converter losses**, corresponding to the **RMS current**

for **converter losses** parameter,  $\begin{bmatrix} I_{rms,1} \\ \vdots \\ I_{rms,n} \end{bmatrix}$ , and the **Nominal dc-link voltage**,  $v_{dc\_nom}$ .

### Model Thermal Effects

This block has one optional thermal port. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

## Ports

### Input

#### ModWave — Modulation wave

vector

Physical signal input port associated with the normalized modulation wave.

### Conserving

#### ~ — Voltage

electrical

Expandable electrical conserving port associated with voltage. For more information, see three-phase port.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**+ – Positive terminal**

electrical

Electrical conserving port associated with the positive terminal.

**- – Negative terminal**

electrical

Electrical conserving port associated with the negative terminal.

**H – Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model thermal parameters.

### Electrical connection — Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

### AC frequency — AC frequency

Variable (default) | Constant

Whether to have a variable or constant AC frequency:

- **Variable** — If the frequency of generated AC voltage varies, this option supports both variable-voltage variable-frequency (VVVF) and variable-voltage constant-frequency (VVCF) applications but the block only works in time simulation mode.
- **Constant** — If the frequency of generated AC voltage is constant, this option supports variable-voltage constant-frequency (VVCF) application and the block works in both time and frequency-and-time simulation modes.

### Rated electrical frequency — Rated electrical frequency

60 Hz | scalar

Rated electrical frequency. The value of this parameter must be equal to the value of the rated electrical frequencies of other sinusoidal sources in your model, if present.

#### Dependencies

To enable this parameter, set **AC frequency** to Constant.

### Losses parameterization — Losses parameterization

Fixed (default) | Coefficients:  $loss = P_{fixed} + k_s \cdot v_{dc} \cdot I_{rms} + k_{c1} \cdot I_{rms} + k_{c2} \cdot I_{rms}^2$  | Profile:  $loss = f(I_{rms}, v_{dc\_nom})$

Parameterization option for the power losses. You can choose one of these options:

- **Fixed** — Power loss is fixed and equal to the value of the **Fixed power loss** parameter.
- **Coefficients**:  $loss = P_{fixed} + k_s \cdot v_{dc} \cdot I_{rms} + k_{c1} \cdot I_{rms} + k_{c2} \cdot I_{rms}^2$  — The total power losses are obtained by summing up the switching, the conduction, and quiescent losses.
- **Profile**:  $loss = f(I_{rms}, v_{dc\_nom})$  — The power loss is a function of the RMS current and the nominal dc-link voltage, and is obtained from the power losses profile.

### Fixed power loss — Power loss

1 W (default)

Fixed power loss on semiconductor components, in W. The input power is equal to the fixed power loss plus output power.

#### Dependencies

To enable this parameter, set **Losses parameterization** to either Fixed or Coefficients:  $loss = P_{fixed} + k_s \cdot v_{dc} \cdot I_{rms} + k_{c1} \cdot I_{rms} + k_{c2} \cdot I_{rms}^2$ .

### Switching losses coefficient, $k_s$ — Coefficient of switching losses

0.01 (default)

Proportionality constant that depends on the turn-on and turn-off intervals and switching frequency.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Coefficients:  $\text{loss}=\text{Pfixed}+k_s*v_{dc}*I_{rms}+k_{c1}*I_{rms}+k_{c2}*I_{rms}^2$ .

**Conduction losses coefficient,  $k_{c1}$  — First coefficient of conduction losses**

1.7 V (default)

Coefficient of the conduction losses that depends on the on-state zero current collector-emitter voltage of the transistor and on the forward voltage drop of the diode.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Coefficients:  $\text{loss}=\text{Pfixed}+k_s*v_{dc}*I_{rms}+k_{c1}*I_{rms}+k_{c2}*I_{rms}^2$ .

**Conduction losses coefficient,  $k_{c2}$  — Second coefficient of conduction losses**

0.01 Ohm (default)

Coefficient of the conduction losses that depends on the state resistance of the transistor and on the anti-parallel diode.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Coefficients:  $\text{loss}=\text{Pfixed}+k_s*v_{dc}*I_{rms}+k_{c1}*I_{rms}+k_{c2}*I_{rms}^2$ .

**Converter losses — Converter losses vector**

[100, 800, 1870, 3250, 4900] W (default)

Vector of power losses values.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Profile:  $\text{loss}=f(I_{rms},v_{dc\_nom})$ .

**RMS current for converter losses — RMS currents vector**

[10, 100, 200, 300, 400] A (default)

Vector of root mean square currents associated to the values of the **Converter losses** parameter.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Profile:  $\text{loss}=f(I_{rms},v_{dc\_nom})$ .

**Nominal dc-link voltage — Nominal dc-link voltage**

300 V (default)

Nominal voltage.

**Dependencies**

To enable this parameter, set **Losses parameterization** to Profile:  $\text{loss}=f(I_{rms},v_{dc\_nom})$ .

**Thermal mass — Thermal mass**

30000 J/K (default)

Thermal mass.

### Dependencies

To enable this parameter, set **Modeling option** to Show thermal port.

## Version History

### Introduced in R2018a

### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

[1] Rajput, M. N. *Thermal modeling of permanent magnet synchronous motor and inverter*. 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Average-Value Chopper | Average-Value Inverter (Three-Phase) | Average-Value Rectifier (Three-Phase) | Voltage Source

## Band-Limited Op-Amp

Model band-limited operational amplifier

**Library:** Simscape / Electrical / Integrated Circuits



### Description

The Band-Limited Op-Amp block models a band-limited operational amplifier. If the voltages at the positive and negative ports are  $V_p$  and  $V_m$ , respectively, the output voltage is:

$$V_{out} = \frac{A(V_p - V_m)}{\frac{s}{2\pi f} + 1} - I_{out} * R_{out}$$

where:

- $A$  is the gain.
- $R_{out}$  is the output resistance.
- $I_{out}$  is the output current.
- $s$  is the Laplace operator.
- $f$  is the 3-dB bandwidth.

The input current is:

$$\frac{V_p - V_m}{R_{in}}$$

where  $R_{in}$  is the input resistance.

The block does not use the initial condition you specify using the **Initial output voltage, V0** parameter if you select the **Start simulation from steady state** check box in the SimscapeSolver Configuration block.

### Ports

#### Conserving

##### + – Non-inverting input

electrical

Electrical conserving port associated with the op-amp non-inverting input

##### - – Inverting input

electrical

Electrical conserving port associated with the op-amp inverting input

**OUT – Output voltage**

electrical

Electrical conserving port associated with the op-amp output voltage

**Parameters****Gain, A – Open-loop gain**

1000 (default)

The open-loop gain of the operational amplifier.

**Input resistance, Rin – Input resistance**

1e6 Ohm (default)

The resistance at the input of the operational amplifier that the block uses to calculate the input current.

**Output resistance, Rout – Output resistance**

100 Ohm (default)

The resistance at the output of the operational amplifier that the block uses to calculate the drop in output voltage due to the output current.

**Minimum output, Vmin – Minimum output**

-15 V (default)

The lower limit on the operational amplifier no-load output voltage.

**Maximum output, Vmax – Maximum output**

15 V (default)

The upper limit on the operational amplifier no-load output voltage.

**Maximum slew rate, Vdot – Maximum slew rate**

1000 V/s (default)

The maximum positive or negative rate of change of output voltage magnitude.

**Bandwidth, f – Open-loop bandwidth**

1e5 Hz (default)

The open-loop bandwidth, that is, the frequency at which the gain drops by 3 dB compared to the low-frequency gain, A.

**Initial output voltage, V0 – Initial output voltage**

0 V (default)

The output voltage at the start of the simulation when the output current is zero.

---

**Note** This parameter value does not account for the voltage drop across the output resistor.

---

## **Version History**

**Introduced in R2008a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Op-Amp | Finite-Gain Op-Amp | Fully Differential Op-Amp



# Battery

Behavioral battery model

**Library:** Simscape / Electrical / Sources  
Simscape / Battery / Cells



## Description

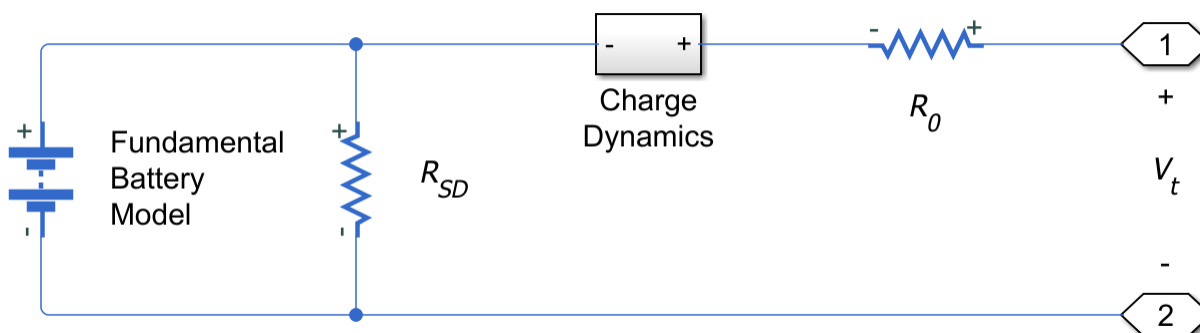
The Battery block represents a simple battery model. The block has four modeling options, accessible by setting the **Modeling option** parameter to either:

- **Uninstrumented | No thermal port** — Basic model that does not output battery charge level or simulate thermal effects.
- **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 modeling options 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 modeling options 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. For more information, see “Modeling Thermal Effects” on page 1-53.

The battery equivalent circuit is made up of the fundamental battery model, the self-discharge resistance  $R_{SD}$ , the charge dynamics model, and the series resistance  $R_0$ .



## Battery Model

If you select **Infinite** for the **Battery charge capacity** parameter, the block models the battery as a series resistor and a constant voltage source. If you select **Finite** for the **Battery charge capacity** parameter, the block models the battery as a series resistor and a charge-dependent voltage source. In the finite case, the voltage is a function of charge and has the following relationship:

$$V = V_0 \left( \frac{\text{SOC}}{1 - \beta(1 - \text{SOC})} \right)$$

where:

- SOC (state-of-charge) is the ratio of current charge to rated battery capacity.
- $V_0$  is the voltage when the battery is fully charged at no load, as defined by the **Nominal voltage**, **Vnom** parameter.
- $\beta$  is a constant that is calculated so that the battery voltage is  $V1$  when the charge is  $AH1$ . Specify the voltage  $V1$  and cell capacity  $AH1$  using block parameters.  $AH1$  is the charge when the no-load (open-circuit) voltage is  $V1$ , and  $V1$  is less than the nominal voltage.

The equation defines an approximate relationship between voltage and remaining charge. This approximation replicates the increasing rate of voltage drop at low charge values, and ensures that the battery voltage becomes zero when the charge level is zero. The advantage of this model is that it requires few parameters, which are readily available on most datasheets.

## Modeling Battery Fade

For battery models with finite battery charge capacity, you can model battery performance deterioration depending on the number of discharge cycles. This deterioration is referred to as battery fade. To enable battery fade, set the **Battery fade** parameter to **Enabled**. This setting exposes additional parameters in the **Fade** section.

The block implements battery fade by scaling certain battery parameter values that you specify in the **Main** section, depending on the number of completed discharge cycles. The block uses multipliers  $\lambda_{AH}$ ,  $\lambda_{R0}$ , and  $\lambda_{V1}$  on the **Cell capacity (Ah rating)**, **Internal resistance**, and **Voltage V1 when charge is AH1** parameter values, respectively. These multipliers, in turn, depend on the number of discharge cycles:

$$\lambda_{AH} = 1 - k_1 N^{0.5}$$

$$\lambda_{R0} = 1 + k_2 N^{0.5}$$

$$\lambda_{V1} = 1 - k_3 N$$

$$N = N_0 + \frac{1}{AH} \int_0^t \frac{i(t) \cdot H(i(t))}{\lambda_{AH}(t)} dt$$

where:

- $\lambda_{AH}$  is the multiplier for battery nominal capacity.
- $\lambda_{R0}$  is the multiplier for battery series resistance.
- $\lambda_{V1}$  is the multiplier for voltage  $V1$ .

- $N$  is the number of discharge cycles completed.
- $N_0$  is the number of full discharge cycles completed before the start of the simulation.
- $AH$  is the rated battery capacity in ampere-hours.
- $i(t)$  is the instantaneous battery output current.
- $H(i(t))$  is the Heaviside function of the instantaneous battery output current. This function returns 0 if the argument is negative, and 1 if the argument is positive.

The block calculates the coefficients  $k_1$ ,  $k_2$ , and  $k_3$  by substituting the parameter values you provide in the **Facte** section into these battery equations. For example, the default set of block parameters corresponds to the following coefficient values:

- $k_1 = 1e-2$
- $k_2 = 1e-3$
- $k_3 = 1e-3$

You can also define a starting point for a simulation based on the previous charge-discharge history by using the high-priority variable **Discharge cycles**. For more information, see “Variables” on page 1-56.

### Modeling Thermal Effects

If you set the **Modeling option** parameter to either `Uninstrumented | Show thermal port` or `Instrumented | Show thermal port`, you provide additional parameters to define battery behavior at a second temperature. The extended equations for the voltage when the thermal port is exposed are:

$$V = V_{0T} \left( \frac{\text{SOC}}{1 - \beta_T(1 - \text{SOC})} \right)$$

$$V_{0T} = V_0(1 + \lambda_V(T - T_1))$$

where:

- $T$  is the battery temperature.
- $T_1$  is the nominal measurement temperature.
- $\lambda_V$  is the parameter temperature dependence coefficient for  $V_0$ .
- $\beta_T = \beta[1 + \lambda_\beta(T - T_1)]$ .
- $\lambda_\beta$  is the parameter temperature dependence coefficient for  $\beta$ .
- $\beta$  is calculated in the same way as “Battery Model” on page 1-52, using the temperature-modified nominal voltage  $V_{0T}$ .

The internal series resistance, self-discharge resistance, and any charge-dynamic resistances are also functions of temperature:

$$R_T = R(1 + \lambda_R(T - T_1))$$

where  $\lambda_R$  is the parameter temperature dependence coefficient.

All the temperature dependence coefficients are determined from the corresponding values you provide at the nominal and second measurement temperatures. If you include charge dynamics in the model, the time constants vary with temperature in the same way.

The battery temperature is determined from a summation of all the ohmic losses included in the model:

$$M_{th}\dot{T} = \sum_i V_{T,i}^2/R_{T,i}$$

where:

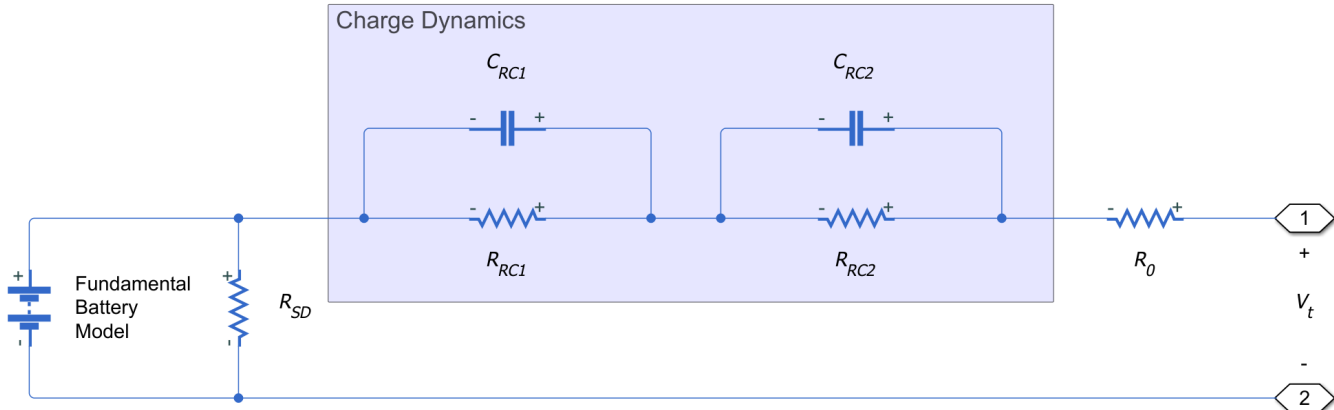
- $M_{th}$  is the battery thermal mass.
- $i$  corresponds to the  $i$ th ohmic loss contributor. Depending on how you have configured the block, the losses include:
  - Series resistance
  - Self-discharge resistance
  - First charge dynamics segment
  - Second charge dynamics segment
  - Third charge dynamics segment
  - Fourth charge dynamics segment
  - Fifth charge dynamics segment
- $V_{T,i}$  is the voltage drop across resistor  $i$ .
- $R_{T,i}$  is resistor  $i$ .

### Modeling Charge Dynamics

You can model battery charge dynamics using the **Charge dynamics** parameter:

- **No dynamics** — The equivalent circuit contains no parallel RC sections. There is no delay between terminal voltage and internal charging voltage of the battery.
- **One time-constant dynamics** — The equivalent circuit contains one parallel RC section. Specify the time constant using the **First time constant** parameter.
- **Two time-constant dynamics** — The equivalent circuit contains two parallel RC sections. Specify the time constants using the **First time constant** and **Second time constant** parameters.
- **Three time-constant dynamics** — The equivalent circuit contains three parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, and **Third time constant** parameters.
- **Four time-constant dynamics** — The equivalent circuit contains four parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, **Third time constant**, and **Fourth time constant** parameters.
- **Five time-constant dynamics** — The equivalent circuit contains five parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, **Third time constant**, **Fourth time constant**, and **Fifth time constant** parameters.

This figure shows the equivalent circuit for the block configured with two time-constant dynamics.



In the diagram:

- $R_{RC1}$  and  $R_{RC2}$  are the parallel RC resistances. Specify these values with the **First polarization resistance** and **Second polarization resistance** parameters, respectively.
- $C_{RC1}$  and  $C_{RC2}$  are the parallel RC capacitances. The time constant  $\tau$  for each parallel section relates the  $R$  and  $C$  values using the relationship  $C = \tau/R$ . Specify  $\tau$  for each section using the **First time constant** and **Second time constant** parameters, respectively.
- $R_0$  is the series resistance. Specify this value with the **Internal resistance** parameter.

### Modeling Battery Aging

For battery models with finite battery charge capacity, you can model the battery performance deterioration that occurs when the battery is not used. Calendar aging affects both the internal resistance and capacity. In particular, the resistance increase depends by various mechanisms such as the creation of Solid Electrolyte Interface (SEI) at both anode and cathode and the corrosion of the current collector. These processes mainly depends on the storage temperature, the storage state of charge, and time.

This equation defines the terminal resistance increase of the battery due to calendar aging:

$$\alpha_r(T, V_{oc}) = (bV_{oc} - c)e^{-\frac{qd}{kT}},$$

$$R = R_0 \left( 1 + \sum_{i=1}^{i=n} \alpha_r(T_i, V_{oc})(t_i^a - t_{i-1}^a - 1) \right),$$

where:

- $V_{oc}$  is the **Normalized open-circuit voltage during storage, V/Vnom**.
- $R_0$  is the **Internal resistance**.
- $t_i$  is the time sample derived from the **Vector of time intervals** parameter.
- $T_i$  is derived from the **Vector of temperatures** parameter.
- $b$  is the **Linear scaling for voltage, b**.
- $c$  is the **Constant offset for voltage, c**.
- $d$  is the **Temperature-dependent exponential increase, d**.
- $a$  is the **Time exponent, a**.

- $q$  is the electron's elementary charge, in C.
- $k$  is the Boltzmann constant, in J/K.

The *R1\_age\_multiplier* variable in the **Simscape Results Explorer** stores the calendar aging of the battery in terms of resistance increase:

$$R1\_age\_multiplier = \frac{R}{R_0}.$$

For thermal modeling options of the block, if you set the **Storage condition** parameter to **Fixed open-circuit voltage**, you must specify the additional **Open-circuit voltage measurement temperature** parameter to convert the storage open-circuit voltage to the temperature-independent state of charge during storage:

$$SOC = \frac{OCV(1 - \beta(T_{OCV\_measurement}))}{1 - \beta(T_{OCV\_measurement})OCV}.$$

The open-circuit voltage according to the storage temperature is then defined by this equation:

$$OCV_{T_{storage}} = \frac{SOC}{1 - \beta(T_{storage})(1 - \beta(T_{storage})SOC)}.$$

Finally, this equation defines the terminal resistance increase of the battery due to calendar aging according to the storage temperature:

$$R = R_0 \left( 1 + \sum_{i=1}^{i=n} \alpha_r(T_{storage_i}, OCV_{T_{storage_i}})(t_i^a - t_{i-1}^a) \right).$$

### Plotting Voltage-Charge Characteristics

A quick plot feature lets you visualize the voltage-charge characteristic for the battery model parameter values. To plot the characteristics, right-click a Battery block in your model and, from the context menu, select **Electrical > Basic characteristic**. The software automatically computes a set of bias conditions, based on the block parameter values, and opens a figure window containing a plot of no-load voltage versus the state-of-charge (SOC) for the block. For more information, see “Plot Basic Characteristics for Battery Blocks”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

When you model battery fade, the **Discharge cycles** variable lets you specify the number of charge-discharge cycles completed prior to the start of simulation. If you disable battery fade modeling, this variable is not used by the block.

## Assumptions and Limitations

- The self-discharge resistance is assumed not to depend strongly on the number of discharge cycles.
- For the thermal modeling option of the battery, you provide fade data only for the reference temperature operation. The block applies the same derived  $\lambda_{AH}$ ,  $\lambda_{RO}$ , and  $\lambda_{V1}$  multipliers to parameter values corresponding to the second temperature.
- When using the thermal block modeling options, use caution when operating at temperatures outside of the temperature range bounded by the **Measurement temperature** and **Second measurement temperature** values. The block uses linear interpolation for the derived equation coefficients, and simulation results can become nonphysical outside of the specified range. The block checks that the internal series resistance, self-discharge resistance, and nominal voltage always remain positive. If there is a violation, the block issues error messages.

## Ports

### Output

#### q — Battery charge level, C

physical signal

Physical signal port that outputs the internal charge, in the units of coulomb (C). Use this output port to change load behavior as a function of charge, without the complexity of building a charge state estimator.

### Dependencies

To enable this port, set **Modeling option** to either Instrumented | No thermal port and Instrumented | Show thermal port.

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the battery positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the battery negative terminal.

#### H — Battery thermal mass

thermal

Thermal conserving port that represents the battery thermal mass. When you expose this port, provide additional parameters to define battery behavior at a second temperature. For more information, see “Modeling Thermal Effects” on page 1-53.

### Dependencies

To enable this port, set **Modeling option** to either Uninstrumented | Show thermal port and Instrumented | Show thermal port.

## Parameters

### **Modeling option — Whether to enable charge level and thermal ports**

Uninstrumented | No thermal port (default) | Uninstrumented | Show thermal port | Instrumented | No thermal port | Instrumented | Show thermal port

Whether to output the internal charge level of the battery and enable the thermal port.

### **Main**

### **Nominal voltage, Vnom — Output voltage when battery is fully charged**

12 V (default) | positive number

The no-load voltage across the battery when it is fully charged.

### **Current directionality — Enable direction of current**

Disabled (default) | Enabled

Whether to enable current directionality. If you set this parameter to Enabled, the terminal resistance depends on the direction of the current.

### **Internal resistance — Battery internal resistance**

2 Ohm (default) | positive number

Internal connection resistance of the battery.

### **Dependencies**

To enable this parameter, set **Current directionality** to Disabled.

### **Internal resistance during charging — Battery internal resistance during charge**

2 Ohm (default) | positive number

Internal connection resistance of the battery during charge phase.

### **Dependencies**

To enable this parameter, set **Current directionality** to Enabled.

### **Internal resistance during discharging — Battery internal resistance during discharge**

2 Ohm (default) | positive number

Internal connection resistance of the battery during discharge phase.

### **Dependencies**

To enable this parameter, set **Current directionality** to Enabled.

### **Battery charge capacity — Select battery model**

Infinite (default) | Finite

Select one of the options for modeling the charge capacity of the battery:

- **Infinite** — The battery voltage is independent of charge drawn from the battery.
- **Finite** — The battery voltage decreases as charge decreases.



**Cell capacity (Ah rating) – Nominal battery capacity when fully charged**

50 hr\*A (default) | positive number

The maximum (nominal) battery charge in ampere-hours. To specify a target value for the initial battery charge at the start of simulation, use the high-priority **Charge** variable. For more information, see “Variables” on page 1-56.

**Dependencies**

Enabled when the **Battery charge capacity** parameter is set to *Finite*.

**Voltage V1 when charge is AH1 – Output voltage at charge level AH1**

11.5 V (default) | positive number

The fundamental battery output voltage when the charge level is AH1, as specified by the **Charge AH1 when no-load voltage is V1** parameter. This parameter must be less than **Nominal voltage, Vnom**.

**Dependencies**

Enabled when the **Battery charge capacity** parameter is set to *Finite*.

**Charge AH1 when no-load voltage is V1 – Charge level when the no-load output voltage is V1**

25 hr\*A (default) | positive number

The battery charge level corresponding to the no-load output voltage specified by the **Voltage V1 when charge is AH1** parameter.

**Dependencies**

Enabled when the **Battery charge capacity** parameter is set to *Finite*.

**Self-discharge – Select whether to model the self-discharge resistance of the battery**

Disabled (default) | Enabled

Select whether to model the self-discharge resistance of the battery. The block models this effect as a resistor across the terminals of the fundamental battery model.

As temperature increases, self-discharge resistance decreases, causing self-discharge to increase. If the decrease in resistance is too fast, thermal runaway of the battery and numerical instability can occur. You can resolve this by doing any of the following:

- Decrease the thermal resistance
- Decrease the gradient of the self-discharge resistance with respect to temperature
- Increase the self-discharge resistance

**Dependencies**

Enabled when the **Battery charge capacity** parameter is set to *Finite*.

**Self-discharge resistance – Resistance that represents battery self-discharge**

2000 Ohm (default) | positive number

The resistance across the fundamental battery model that represents battery self-discharge.

**Dependencies**

Enabled when the **Self-discharge** parameter is set to Enabled.

**Measurement temperature — Temperature at which the block parameters are measured**

298.15 K (default) | positive number

Temperature  $T_1$ , at which the block parameters in the **Main** section are measured. For more information, see “Modeling Thermal Effects” on page 1-53.

**Dependencies**

Enabled for blocks with exposed thermal port.

**Dynamics****Charge dynamics — Battery charge dynamics model**

No dynamics (default) | One time-constant dynamics | Two time-constant dynamics | Three time-constant dynamics | Four time-constant dynamics | Five time-constant dynamics

Select how to model battery charge dynamics. This parameter determines the number of parallel RC sections in the equivalent circuit:

- **No dynamics** — The equivalent circuit contains no parallel RC sections. There is no delay between terminal voltage and internal charging voltage of the battery.
- **One time-constant dynamics** — The equivalent circuit contains one parallel RC section. Specify the time constant using the **First time constant** parameter.
- **Two time-constant dynamics** — The equivalent circuit contains two parallel RC sections. Specify the time constants using the **First time constant** and **Second time constant** parameters.
- **Three time-constant dynamics** — The equivalent circuit contains three parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, and **Third time constant** parameters.
- **Four time-constant dynamics** — The equivalent circuit contains four parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, **Third time constant**, and **Fourth time constant** parameters.
- **Five time-constant dynamics** — The equivalent circuit contains five parallel RC sections. Specify the time constants using the **First time constant**, **Second time constant**, **Third time constant**, **Fourth time constant**, and **Fifth time constant** parameters.

**First polarization resistance — First RC resistance**

0.005 Ohm (default) | positive number

The resistance of the first parallel RC section. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**First time constant — First RC time constant**

30 s (default) | positive number

The time constant of the first parallel RC section. This value is equal to  $RC$  and affects the dynamics of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Second polarization resistance — Second RC resistance**

0.005 Ohm (default) | positive number

The resistance of the second parallel RC section. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Second time constant — Second RC time constant**

30 s (default) | positive number

The time constant of the second parallel RC section. This value is equal to  $RC$  and affects the dynamics of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Third polarization resistance — Third RC resistance**

0.005 Ohm (default) | positive number

The resistance of the third parallel RC section. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Third time constant — Third RC time constant**

30 s (default) | positive number

The time constant of the third parallel RC section. This value is equal to  $RC$  and affects the dynamics of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Fourth polarization resistance — Fourth RC resistance**

0.005 Ohm (default) | positive number

The resistance of the fourth parallel RC section. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to `Four time-constant dynamics` or `Five time-constant dynamics`.

**Fourth time constant — Fourth RC time constant**

30 s (default) | positive number

The time constant of the fourth parallel RC section. This value is equal to  $RC$  and affects the dynamics of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to `Four time-constant dynamics` or `Five time-constant dynamics`.

**Fifth polarization resistance — Fifth RC resistance**

0.005 Ohm (default) | positive number

The resistance of the fifth parallel RC section. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to `Five time-constant dynamics`.

**Fifth time constant — Fifth RC time constant**

30 s (default) | positive number

The time constant of the fifth parallel RC section. This value is equal to  $RC$  and affects the dynamics of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to `Five time-constant dynamics`.

**Fade****Battery fade — Select whether to model battery performance deterioration with aging**

Disabled (default) | Enabled

Select whether to include battery fade modeling:

- **Disabled** — The battery performance is not age dependent.
- **Enabled** — The battery performance changes depending on the number of completed charge-discharge cycles. Selecting this option exposes additional parameters in this section, which define the battery performance after a certain number of discharge cycles. The block uses these parameter values to calculate the scaling coefficients  $k_1$ ,  $k_2$ , and  $k_3$ . For more information, see “Modeling Battery Fade” on page 1-52.

**Dependencies**

Enabled when the **Battery charge capacity** parameter in the **Main** section is set to `Finite`. If **Battery charge capacity** is `Infinite`, the **Fade** section is empty.

**Number of discharge cycles, N – Number of cycles that defines a second set of data points**

100 (default) | positive number

The number of charge-discharge cycles after which the other parameters in this section are measured. This second set of data points defines the scaling coefficients  $k_1$ ,  $k_2$ , and  $k_3$ , used in modeling battery fade.

**Dependencies**

Enabled when the **Battery fade** parameter is set to Enabled.

**Cell capacity after N discharge cycles – Maximum battery capacity after N discharge cycles**

45 hr\*A (default) | positive number

The maximum battery charge, in ampere-hours, after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

**Dependencies**

Enabled when the **Battery fade** parameter is set to Enabled.

**Internal resistance after N discharge cycles – Battery internal resistance after N discharge cycles**

2.02 Ohm (default) | positive number

The battery internal resistance after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

**Dependencies**

Enabled when the **Battery fade** parameter is set to Enabled and **Current Directionality** is set to Disabled.

**Average internal resistance after N discharge cycles – Battery average internal resistance after N discharge cycles**

2.02 Ohm (default) | positive number

Average of the battery charge and discharge internal resistance after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

**Dependencies**

To enable this parameter, set **Battery fade** and **Current directionality** to Enabled.

**Voltage V1 at charge AH1 after N discharge cycles – Output voltage at charge level AH1 after N discharge cycles**

10.35 V (default) | positive number

The fundamental battery model output voltage, at charge level AH1, after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

**Dependencies**

Enabled when the **Battery fade** parameter is set to Enabled.

### **Calendar Aging**

To expose these parameters, in the **Main** tab, set **Battery charge capacity** to **Finite**.

#### **Calendar aging — Calendar aging option**

Disabled (default) | Enabled

Whether to enable calendar aging of the battery.

#### **Storage condition — Storage condition**

Fixed open-circuit voltage (default) | Fixed state of charge

Whether to specify the open-circuit voltage or the state of charge during storage.

#### **Dependencies**

To enable this parameter, set **Calendar aging** to **Enabled**.

#### **Normalized open-circuit voltage during storage, $V/V_{nom}$ — Normalized open-circuit voltage during storage**

0.9 (default) | scalar

Normalized open-circuit voltage during storage.

#### **Dependencies**

To enable this parameter, set **Calendar aging** to **Enabled** and **Storage condition** to **Fixed open-circuit voltage**.

#### **Open-circuit measurement temperature — Open-circuit temperature**

298.15 K (default) | positive number

Open-circuit measurement temperature.

#### **Dependencies**

To enable this parameter, expose the block thermal port and set **Calendar aging** to **Enabled** and **Storage condition** to **Fixed open-circuit voltage**.

#### **State of charge during storage (%) — Percentage state of charge during storage**

60 (default) | positive number

State of charge during storage, in percentage.

#### **Dependencies**

To enable this parameter, set **Calendar aging** to **Enabled** and **Storage condition** to **Fixed state of charge**.

#### **Vector of time intervals — Time intervals**

[0] d (default) | scalar

Time intervals. This parameter must be equal in size to **Vector of temperatures**.

#### **Dependencies**

To enable this parameter, set **Calendar aging** to **Enabled**.

**Vector of temperatures — Storage temperatures**

[273] K (default) | scalar

Set of storage temperatures. This parameter must be equal in size to **Vector of time intervals**.

**Dependencies**

To enable this parameter, set **Calendar aging** to Enabled.

**Linear scaling for voltage, b — Voltage linear scaling**

2.2134e6 (default) | scalar

Linear scaling coefficient for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Calendar aging** to Enabled.

**Constant offset for voltage, c — Voltage constant offset**

1.632e6 (default) | scalar

Constant offset for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Calendar aging** to Enabled.

**Temperature-dependent exponential increase, d — Temperature-dependent exponential increase**

0.515833569 V (default) | scalar

Temperature-dependent exponential increase.

**Dependencies**

To enable this parameter, set **Calendar aging** to Enabled.

**Time exponent, a — Time exponent**

0.75 (default) | scalar

Time exponent.

**Dependencies**

To enable this parameter, set **Calendar aging** to Enabled.

**Temperature Dependence**

This section appears only for blocks with exposed thermal port. For more information, see “Modeling Thermal Effects” on page 1-53.

**Nominal voltage at second measurement temperature — Output voltage when battery is fully charged**

12 V (default) | positive number

The no-load voltage across the battery at the second measurement temperature when it is fully charged.

**Internal resistance at second measurement temperature – Battery internal resistance**

2.2 0hm (default) | positive number

Internal connection resistance of the battery at the second measurement temperature.

**Voltage V1 at second measurement temperature – Output voltage at charge level AH1**

11.4 V (default) | positive number

The fundamental battery model output voltage at the second measurement temperature and at charge level AH1, as specified by the **Charge AH1 when no-load voltage is V1** parameter.

**Dependencies**

Enabled when the **Battery charge capacity** parameter in the **Main** section is set to **Finite**.

**Self-discharge resistance at second measurement temperature – Resistance that represents battery self-discharge**

2200 0hm (default) | positive number

The resistance across the fundamental battery model at the second measurement temperature. This resistance represents the self-discharge.

**Dependencies**

Enabled when the **Self-discharge resistance** parameter in the **Main** section is set to **Enabled**.

**First polarization resistance at second measurement temperature – First RC resistance at second measurement temperature**

0.005 0hm (default) | positive number

The resistance of the first parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to **One time-constant dynamics**, **Two time-constant dynamics**, **Three time-constant dynamics**, **Four time-constant dynamics**, or **Five time-constant dynamics**.

**First time constant at second measurement temperature – First RC time constant at second measurement temperature**

30 s (default) | positive number

The time constant of the first parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to **One time-constant dynamics**, **Two time-constant dynamics**, **Three time-constant dynamics**, **Four time-constant dynamics**, or **Five time-constant dynamics**.

**Second polarization resistance at second measurement temperature – Second RC resistance at second measurement temperature**

0.005 0hm (default) | positive number

The resistance of the second parallel RC section at the second measurement temperature.



**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Second time constant at second measurement temperature — Second RC time constant at second measurement temperature**

30 s (default) | positive number

The time constant of the second parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Third polarization resistance at second measurement temperature — Third RC resistance at second measurement temperature**

0.005 Ohm (default) | positive number

The resistance of the third parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Third time constant at second measurement temperature — Third RC time constant at second measurement temperature**

30 s (default) | positive number

The time constant of the third parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Fourth polarization resistance at second measurement temperature — Fourth RC resistance at second measurement temperature**

0.005 Ohm (default) | positive number

The resistance of the fourth parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics.

**Fourth time constant at second measurement temperature — Fourth RC time constant at second measurement temperature**

30 s (default) | positive number

The time constant of the fourth parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics.

**Fifth polarization resistance at second measurement temperature — Fifth RC resistance at second measurement temperature**

0.005 Ohm (default) | positive number

The resistance of the fifth parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics.

**Fifth time constant at second measurement temperature — Fifth RC time constant at second measurement temperature**

30 s (default) | positive number

The time constant of the fifth parallel RC section at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics.

**Second measurement temperature — Temperature at which the block parameters in this section are measured**

273.15 K (default) | positive number

Temperature  $T_2$ , at which the block parameters in the **Temperature Dependence** section are measured. For more information, see “Modeling Thermal Effects” on page 1-53.

To specify the initial temperature at the start of simulation, use the high-priority **Temperature** variable. For more information, see “Variables” on page 1-56.

**Thermal Port**

This section appears only for blocks with exposed thermal port. For more information, see “Modeling Thermal Effects” on page 1-53.

**Thermal mass — Thermal mass associated with the thermal port**

30000 J/K (default) | positive number

Thermal mass associated with the thermal port H. It represents the energy required to raise the temperature of the thermal port by one degree.

## Version History

Introduced in R2008b

## References

- [1] Ramadass, P., B. Haran, R. E. White, and B. N. Popov. “Mathematical modeling of the capacity fade of Li-ion cells.” *Journal of Power Sources*. 123 (2003), pp. 230-240.

[2] Ning, G., B. Haran, and B. N. Popov. "Capacity fade study of lithium-ion batteries cycled at high discharge rates." *Journal of Power Sources*. 117 (2003), pp. 160-169.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Battery (Table-Based) | DC Voltage Source | Controlled Voltage Source

## Battery (Table-Based)

Tabulated battery model

**Library:** Simscape / Electrical / Sources  
Simscape / Battery / Cells



### Description

The Battery (Table-Based) block represents a high-fidelity battery model. The block calculates open-circuit voltage as a function of charge level and optional temperature using lookup tables and includes several modeling options:

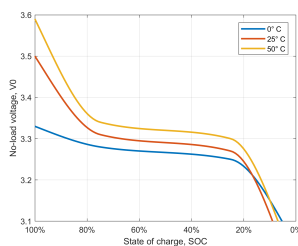
- Self-discharge
- Battery fade
- Charge dynamics
- Calendar aging

---

**Note** The block can use linear or nearest interpolation and extrapolation for all the table based parameters. For rows and columns, it follows the row-column convention, whereas rows are indexed first and, subsequently, columns.

---

The plot shows a battery whose performance varies with temperature and state of charge changes, as typically found on a datasheet.



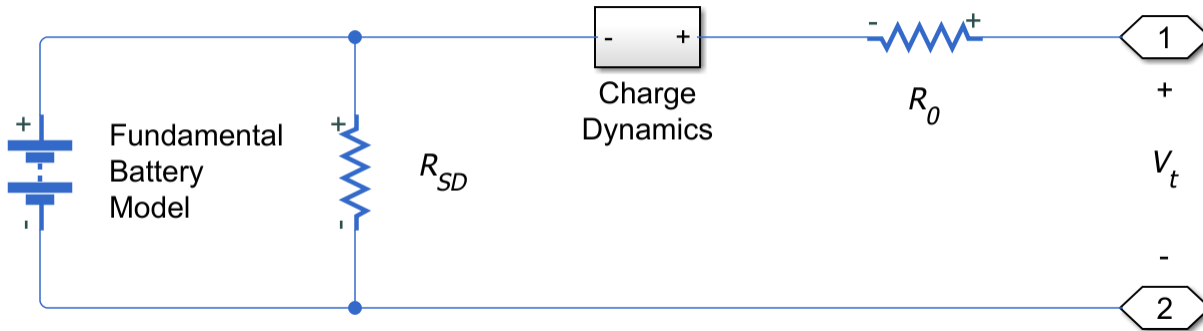
Use this block to parameterize batteries with complex open-circuit voltage behavior from datasheets or experimental results. For a simpler representation of a battery, see the Battery block.

The Battery (Table-Based) block has two optional ports that you can expose by setting the corresponding parameters. The extra physical signal port, **SOC**, outputs the internal state of charge. Use this functionality to change the load behavior as a function of state of charge, without the complexity of building a charge state estimator. To expose the **SOC** port, in the **Main** setting, set the **Expose SOC measurement port** parameter to Yes.

To expose the thermal port, in the **Thermal** setting, set the **Thermal port** parameter to Model. The thermal port represents the battery thermal mass.

You can also choose different built-in parameterizations for this block. For more information, see the “Predefined Parameterization” on page 1-77 section.

The battery equivalent circuit is made up of the fundamental battery model, the self-discharge resistance  $R_{SD}$ , the charge dynamics model, and the series resistance  $R_0$ .



### Battery Model

The block calculates the open-circuit voltage, or the voltage across the fundamental battery model by interpolation:

$$v_0 = v_0(\text{SOC}, T)$$

Where:

- $v_0$  is the open-circuit voltage of the battery. Specify the grid of lookup values using the **Open-circuit voltage, V0(SOC,T)** parameter if tabulating parameters over temperature, or **Open-circuit voltage, V0(SOC)** otherwise.
- SOC is the ratio of current charge to nominal battery capacity specified in the **Cell capacity, AH** parameter along with the effects of the temperature dependent fade percentage change in cell capacity,  $\delta_{AH}(n, T_{fade})$ , specified in the **Percentage change in cell capacity, dAH(N, Tfade)** parameter. Specify the SOC breakpoints using the **Vector of state-of-charge values, SOC** parameter. The block estimates the nominal battery capacity based on the number of cycles and the temperature of the battery by interpolating the specified temperature dependent fade characteristic and the **Cell capacity, AH** parameter.

SOC represents the normalized data with respect to  $q_{nom}$ .

For the lookup-table based fade characteristic option,

$$q_{nom}(T, n) = \left(1 + \frac{\delta_{AH}(n, T_{fade})}{100}\right) * AH \text{ Ah}.$$

For the equation-based fade characteristic option,

$$q_{nom}(T, n) = \left(1 + \frac{\delta_{AH}}{100} \sqrt{\frac{n}{N}}\right) * AH \text{ Ah}.$$

Finally, SOC is obtained from the following equation.

$$\text{SOC}(t) = \frac{1}{q_{nom}(T, n)} \int \left( i(t) - \frac{V_{open}(T, n, t)}{R_{SD}(T, n)} \right) dt$$

Where:

- $q_{nom}$  is the cell capacity of the battery. Specify this value using the **Cell capacity, AH** parameter.
- $N$  is the reference number of discharge cycles over which you specify percent change of several battery parameters. Set this value using the **Number of discharge cycles, N** parameter.
- $n$  is the present number of cycles of the battery.
- $\delta_{AH}$  is the percentage change in cell capacity of the battery after  $N$  discharge cycles.
- $T$  is the battery temperature. Specify the  $T$  breakpoints using the **Vector of temperatures, T** parameter if tabulating the parameters over temperature.

The block also models the series resistance  $R_0$  as a function of state of charge and optional temperature. Specify the grid of lookup values for the series resistance using the **Terminal resistance, R0(SOC,T)** parameter if tabulating the parameters over temperature, or **Terminal resistance, R0(SOC)** otherwise.

### Modeling Self-Discharge

When the battery terminals are open-circuit, it is still possible for internal currents to discharge the battery. This behavior is called self-discharge. To enable this effect, set the **Self-discharge** parameter to Enabled.

The block models these internal currents with a temperature-dependent resistance  $R_{SD}(T)$  across the terminals of the fundamental battery model. You can specify the lookup values for this resistance using the **Self-discharge resistance, Rleak(T)** parameter if tabulating the parameters over temperature, or **Self-discharge resistance, Rleak** otherwise.

### Modeling Charge Dynamics

Batteries are not able to respond instantaneously to load changes. They require some time to achieve a steady-state. This time-varying property is a result of battery charge dynamics and is modeled using parallel RC sections in the equivalent circuit.

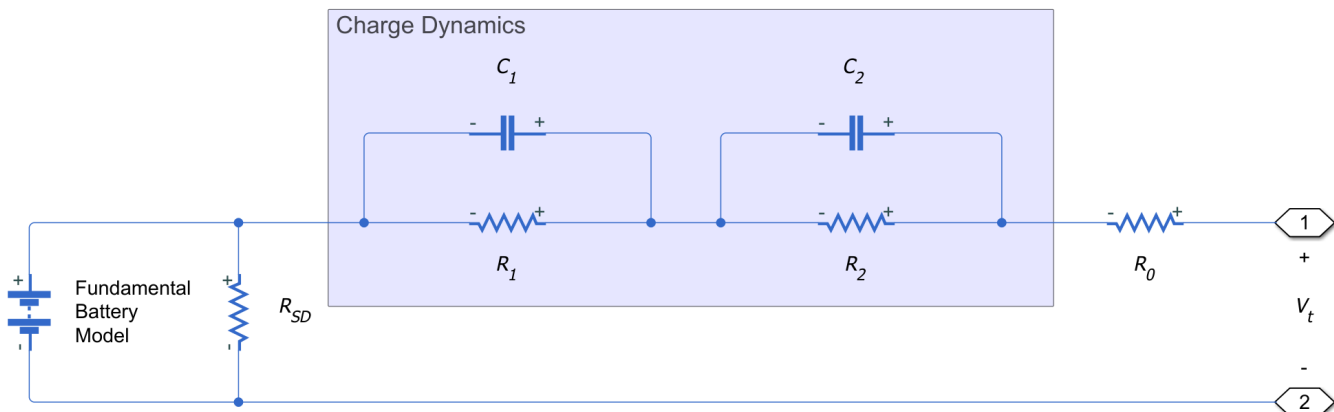
You can model battery charge dynamics using the **Charge dynamics** parameter:

- **No dynamics** — The equivalent circuit contains no parallel RC sections. There is no delay between terminal voltage and internal charging voltage of the battery.
- **One time-constant dynamics** — The equivalent circuit contains one parallel RC section. Specify the time constant using the **First time constant, tau1(SOC,T)** parameter if tabulating parameters over temperature or **First time constant, tau1(SOC)** otherwise.
- **Two time-constant dynamics** — The equivalent circuit contains two parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)** and **Second time constant, tau2(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)** and **Second time constant, tau2(SOC)** otherwise.
- **Three time-constant dynamics** — The equivalent circuit contains three parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)**, **Second time constant, tau2(SOC,T)**, and **Third time constant, tau3(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)**, **Second time constant, tau2(SOC)**, and **Third time constant, tau3(SOC)** otherwise.
- **Four time-constant dynamics** — The equivalent circuit contains four parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)**, **Second time constant,**

**tau2(SOC,T), Third time constant, tau3(SOC,T), and Fourth time constant, tau4(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC), Second time constant, tau2(SOC), Third time constant, tau3(SOC), and Fourth time constant, tau4(SOC)** otherwise.

- Five time-constant dynamics — The equivalent circuit contains five parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T), Second time constant, tau2(SOC,T), Third time constant, tau3(SOC,T), Fourth time constant, tau4(SOC,T), and Fifth time constant, tau5(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC), Second time constant, tau2(SOC), Third time constant, tau3(SOC), Fourth time constant, tau4(SOC), and Fifth time constant, tau5(SOC)** otherwise.

This diagram shows the equivalent circuit for the block configured with two time-constant dynamics.



In the diagram:

- $R_1$  and  $R_2$  are the parallel RC resistances. Specify these values with the **First polarization resistance, R1(SOC,T)** and **Second polarization resistance, R2(SOC,T)** parameters, respectively, if tabulating parameters over temperature or **First polarization resistance, R1(SOC)** and **Second polarization resistance, R2(SOC)** otherwise.
- $C_1$  and  $C_2$  are the parallel RC capacitances. The time constant  $\tau$  for each parallel section relates the  $R$  and  $C$  values using the relationship  $C = \tau/R$ . Specify  $\tau$  for each section using the **First time constant, tau1(SOC,T)** and **Second time constant, tau2(SOC,T)** parameters, respectively, if tabulating parameters over temperature or **First time constant, tau1(SOC)** and **Second time constant, tau2(SOC)** otherwise.
- $R_0$  is the series resistance. Specify this value with the **Terminal resistance, R0(SOC,T)** parameter if tabulating parameters over temperature or **Terminal resistance, R0(SOC)** otherwise.

### Modeling Battery Fade

Battery fade is the deterioration of battery performance over repeated charge and discharge cycles. When the **Fade characteristic** parameter is set to Equations, the battery fade is modeled as follows.

The open-circuit voltage across the fundamental battery model fades proportionally with the number of discharge cycles  $n$ :

$$v_{0, \text{fade}} = v_0 \left( 1 + \frac{\delta_{v_0} n}{100 N} \right)$$

Where  $\delta_{v_0}$  is the percent change in open-circuit voltage after  $N$  discharge cycles. Specify  $\delta_{v_0}$  using the **Change in open-circuit voltage after N discharge cycles (%)** parameter.

The nominal charge, from which state of charge is calculated, fades with the square root of number of discharge cycles:

$$q_{nom, \text{fade}} = q_{nom} \left( 1 + \frac{\delta_{AH} \sqrt{n}}{100 \sqrt{N}} \right)$$

All resistances in the battery model also fade with the square root of the number of discharge cycles:

$$R_{i, \text{fade}} = R_i \left( 1 + \frac{\delta_{R_i} \sqrt{n}}{100 \sqrt{N}} \right)$$

Where:

- $R_i$  is the  $i^{\text{th}}$  resistance
- $\delta_{R_i}$  is the percent change in this resistance over  $N$  cycles

Depending on how you have configured the block, the resistances might include:

- The series resistance — Specify the percent change over  $N$  cycles using the **Change in terminal resistance after N discharge cycles (%)** parameter.
- The self-discharge resistance — Specify the percent change over  $N$  cycles using the **Change in self-discharge resistance after N discharge cycles (%)** parameter.
- The first charge dynamics resistance — Specify the percent change over  $N$  cycles using the **Change in first polarization resistance after N discharge cycles (%)** parameter.
- The second charge dynamics resistance — Specify the percent change over  $N$  cycles using the **Change in second polarization resistance after N discharge cycles (%)** parameter.
- The third charge dynamics resistance — Specify the percent change over  $N$  cycles using the **Change in third polarization resistance after N discharge cycles (%)** parameter.
- The fourth charge dynamics resistance — Specify the percent change over  $N$  cycles using the **Change in fourth polarization resistance after N discharge cycles (%)** parameter.
- The fifth charge dynamics resistance — Specify the percent change over  $N$  cycles using the **Change in fifth polarization resistance after N discharge cycles (%)** parameter.

---

**Note** You can also model the battery fade characteristic by using lookup tables (temperature independent) or lookup tables (temperature dependent). Choosing any of these two options changes the blocks parameters accordingly. For more information, see the “Fade” on page 1-87 parameters tab.

---

### Modeling Thermal Effects

The battery temperature is determined from a summation of all the ohmic losses included in the model:



$$M_{th}\dot{T} = \sum_i V_{T,i}^2/R_{T,i}$$

Where:

- $M_{th}$  is the battery thermal mass.
- $i$  corresponds to the  $i^{\text{th}}$  ohmic loss contributor. Depending on how you have configured the block, the losses might include:
  - The series resistance
  - The self-discharge resistance
  - The first charge dynamics segment
  - The second charge dynamics segment
  - The third charge dynamics segment
  - The fourth charge dynamics segment
  - The fifth charge dynamics segment
- $V_{T,i}$  is the voltage drop across resistor  $i$ .
- $R_{T,i}$  is resistor  $i$ .

### Modeling Battery Aging

You can model the battery performance deterioration that occurs when the battery is not used. Calendar aging affects both the internal resistance and capacity. In particular, the resistance increase depends by various mechanisms such as the creation of Solid Electrolyte Interface (SEI) at both anode and cathode and the corrosion of the current collector. These processes mainly depends on the storage temperature, the storage state of charge, and time.

You can model the calendar aging in the Battery (Table-Based) block by setting the **Modeling option** parameter to:

- Equation-based
- Tabulated: temperature
- Tabulated: time and temperature

#### Equation-based

This equation defines the terminal resistance increase of the battery due to calendar aging:

$$\alpha_r(T, V_{oc}) = (bV_{oc} - c)e^{-\frac{qd}{kT}},$$

$$R = R_0 \left( 1 + \sum_{i=1}^{i=n} \alpha_r(T_i, V_{oc})(t_i^a - t_{i-1}^a) \right),$$

where:

- $V_{oc}$  is the **Normalized open-circuit voltage during storage, V/Vnom**.
- $R_0$  is the **Internal resistance**.
- $t_i$  is the time sample derived from the **Vector of time intervals** parameter.
- $T_i$  is derived from the **Vector of temperatures, T** parameter.

- $b$  is the **Terminal resistance linear scaling for voltage, b**.
- $c$  is the **Terminal resistance constant offset for voltage, c**.
- $d$  is the **Terminal resistance temperature-dependent exponential increase, d**.
- $a$  is the **Terminal resistance time exponent, a**.
- $q$  is the electron's elementary charge, in C.
- $k$  is the Boltzmann constant, in J/K.

This equation defines the capacity decrease of the battery due to calendar aging:

$$\alpha_c(T, V_{oc}) = (bV_{oc} - c)e^{-\frac{qd}{kT}},$$

$$C = C_0 \left( 1 - \sum_{i=1}^{i=n} \alpha_c(T_i, V_{oc})(t_i^a - t_{i-1}^a) \right),$$

where:

- $C_0$  is the **initial capacity**.
- $b$  is the **Capacity linear scaling for voltage, b**.
- $c$  is the **Capacity constant offset for voltage, c**.
- $d$  is the **Capacity temperature-dependent exponential increase, d**.
- $a$  is the **Capacity time constant, a**.

If you set the **Storage condition** parameter to Specify state-of-charge during storage, the block converts the state of charge during storage into normalized open-circuit voltage using the tabulated voltage  $V_0$  against the state of charge and the temperature during storage.

**Tabulated: temperature**

The aged terminal resistance is the product between the terminal resistance,  $R_0(SOC, T)$ , the percentage resistance increase,  $dR_0$ , and the power law that describes the time dependence of the calendar aging:

$$R_0(SOC, T; t_{st}, T_{st}) = R_0(SOC, T) \left( 1 + \sum_{i=1}^n \frac{dR_0(T_{st}, i)}{100} \left( \frac{t_{st, i}^a - t_{st, i-1}^a}{t_{st, m}^a} \right) \right),$$

where:

- $T$  is the battery temperature. Specify the  $T$  breakpoints using the **Vector of temperatures, T** parameter if tabulating the parameters over temperature.
- $T_{st}$  is the **Vector of storage temperatures**.
- $t_{st, i}$  and  $t_{st, i-1}$  are the time samples derived from the **Vector of time intervals**.
- $t_0$  is assumed to be null.
- $t_{st, m}$  is the moment in time at which the resistance increase,  $dR_0$ , is measured.

The same equation applies to the calculation of the aged battery capacity.

**Tabulated: time and temperature**

The aged terminal resistance is the product between the terminal resistance,  $R_0(SOC, T)$  and  $dR_0$ :

$$R_0(SOC, T; \Delta t_{st}, T_{st}) = R_0(SOC, T) \left( 1 + \sum_{i=1}^n \frac{dR_0(\Delta t_{st,i}, T_{st,i})}{100} \right).$$

The same equation applies to the calculation of the aged battery capacity.

### Predefined Parameterization

There are multiple available built-in parameterizations for the Battery (Table-Based) block.

This pre-parameterization data allows you to set up the block to represent components by specific suppliers. The parameterizations of these batteries match the discharge curves in the manufacturer data sheets. To load a predefined parameterization, double-click the Battery (Table-Based) block, click the **<click to select>** hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

The available pre-parameterized data model steady state electrical parameters of a lithium-ion battery. The **Open-circuit voltage, V0(SOC)**, **Terminal resistance, R0(SOC, T)**, **Cell capacity, AH**, **Percentage change in cell capacity, dAH**, and **Percentage change in Open-circuit voltage, dV0(N)** parameters are parameterized from characteristics curves in the manufacture datasheets. The change in series resistance with the battery cycle life, the thermal mass, and the dynamic RC network parameters are not parameterized. Instead, the net resistance of the RC network resistors is summed to the series resistance of the specific pre-parameterized data. If you need to fill the RC parameter data, subtract the net RC network resistance from the series resistance data.

The available data is parameterized for 1C discharge current for different temperatures up to the minimum terminal voltage in the datasheet. The data below the minimum terminal voltage is extrapolated. The **Open-circuit voltage, V0(SOC)** parameter is approximated to be temperature independent. At lower temperature, in the low SOC region, the terminal resistances are constant and equal to the value of the resistance at the minimum terminal voltage. This might lead to higher thermal losses.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

### Plotting Voltage-Charge Characteristics

A quick plot feature lets you visualize the voltage-charge characteristic for the battery model parameter values. To plot the characteristics, right-click a Battery (Table-Based) block in your model and, from the context menu, select **Electrical > 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 open-circuit voltage versus the state-of-charge (SOC) for the block. For more information, see “Plot Basic Characteristics for Battery Blocks”.

## Ports

### Output

#### SOC — Battery charge level, C

physical signal

Physical signal port that outputs the internal state of charge. Use this output port to change load behavior as a function of state of charge, without the complexity of building a charge state estimator. The state of charge is the normalized value estimated from the ratio of current charge with the nominal battery capacity,  $q_{nom}(T,n)$ . The block estimates the current battery charge by integrating the battery terminal output current. To convert the state of charge into actual charge, you must use the correct nominal battery capacity for each temperature.

#### Dependencies

To enable this port, in the **Main** setting, set the **Expose SOC measurement port** parameter to Yes.

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the battery positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the battery negative terminal.

#### H — Battery thermal mass

thermal

Thermal conserving port that represents the battery thermal mass.

#### Dependencies

To enable this port, in the **Thermal** setting, set the **Thermal port** parameter to Yes.

## Parameters

#### Selected part — Predefined parameterization option

<click to select> (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see “Load Predefined Parameterizations”.

### Main

#### Vector of state-of-charge values, SOC — SOC breakpoints

[0, .25, .75, 1] (default) | vector

Vector of state-of-charge breakpoints defining the points at which you specify lookup data. This vector must be strictly ascending. The state-of-charge value is calculated with respect to the nominal battery

capacity specified in the **Cell capacity, AH** parameter. SOC is the ratio of the available battery charge,  $q_{battery}$  and the nominal battery capacity,  $q_{nom}(T,n)$ . You must make sure that, for each temperature,  $SOC = 1$  represents the respective battery charge capacity specified in the **Cell capacity, AH** parameter, assuming a fresh battery with a number of cycles,  $N$ , equal to 1 and  $\delta_{AH}(n = 1, T_{fade}) = 0$ .

$$SOC = \frac{q_{battery}}{q_{nom}(T, n)}$$

for  $N = 1$  and  $\delta_{AH}(n, T_{fade}) = 0$ ,  $q_{nom}(T, n) = AH$ .

### Temperature dependent tables — Select whether to tabulate battery parameters over temperature

Yes - tabulate parameters over temperature (default) | No - do not tabulate parameters over temperature

Select whether to tabulate battery parameters over temperature.

### Current directionality — Enable direction of current

Disabled (default) | Enabled

Whether to enable current directionality. If you set this parameter to Enabled, the terminal resistance depends on the direction of the current.

### Terminal voltage operating range [Min Max] — Terminal voltage operating range

[0, inf] V (default) | vector of positive numbers

Operating range of the terminal voltage. This parameter must be a vector of size 1-by-2 that defines the minimum and maximum values of the terminal voltage.

### Vector of temperatures, T — T breakpoints

[273.15, 298.15, 323.15] K (default) | vector of positive numbers

Vector of temperature breakpoints defining the points at which you specify lookup data. This vector must be strictly ascending and greater than 0 K.

### Dependencies

This parameter is visible only when the **Temperature dependent tables** parameter is set to Yes - tabulate parameters over temperature.

### Open-circuit voltage, V0(SOC, T) — V0 lookup table with temperature breakpoint

[3.2, 3.1, 3.14; 3.25, 3.27, 3.3; 3.28, 3.31, 3.34; 3.33, 3.5, 3.59] V (default) | matrix of nonnegative numbers

Lookup data for open-circuit voltages across the fundamental battery model at the specified SOC and temperature breakpoints.

### Dependencies

This parameter is visible only when the **Temperature dependent tables** parameter is set to Yes - tabulate parameters over temperature.

### Open-circuit voltage, V0(SOC) — V0 lookup table

[3.1, 3.27, 3.31, 3.5] V (default) | matrix of nonnegative numbers

Lookup data for open-circuit voltages across the fundamental battery model at the specified SOC.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to No - do not tabulate parameters over temperature.

**Terminal resistance,  $R_0(SOC, T)$  —  $R_0$  lookup table with temperature breakpoint**

[.03, .015, .002; .04, .017, .008; .039, .012, .006; .027, .013, .021] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC and temperature breakpoints.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to Yes - tabulate parameters over temperature and the **Current directionality** parameter is set to Disabled.

**Terminal resistance during discharging,  $R_0(SOC, T)$  —  $R_0$  lookup table with temperature breakpoint during discharging phase**

[.03, .015, .002; .04, .017, .008; .039, .012, .006; .027, .013, .021] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC and temperature breakpoints during the discharging phase.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to Yes - tabulate parameters over temperature and the **Current directionality** parameter is set to Enabled.

**Terminal resistance during charging,  $R_0(SOC, T)$  —  $R_0$  lookup table with temperature breakpoint during charging phase**

[.03, .015, .002; .04, .017, .008; .039, .012, .006; .027, .013, .021] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC and temperature breakpoints during the charging phase.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to Yes - tabulate parameters over temperature and the **Current directionality** parameter is set to Enabled.

**Terminal resistance,  $R_0(SOC)$  —  $R_0$  lookup table**

[.015, .017, .012, .013] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to No - do not tabulate parameters over temperature and the **Current directionality** parameter is set to Disabled.

**Terminal resistance during discharging,  $R_0(\text{SOC})$  —  $R_0$  lookup table during discharging phase**

[.015, .017, .012, .013] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC during the discharging phase.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to No - do not tabulate parameters over temperature and the **Current directionality** parameter is set to Enabled.

**Terminal resistance during charging,  $R_0(\text{SOC})$  —  $R_0$  lookup table during charging phase**

[.015, .017, .012, .013] Ohm (default) | matrix of nonnegative numbers

Lookup data for series resistance of the battery at the specified SOC during the charging phase.

**Dependencies**

This parameter is visible only when the **Temperature dependent tables** parameter is set to No - do not tabulate parameters over temperature and the **Current directionality** parameter is set to Enabled.

**Cell capacity, AH — Battery capacity when fully charged**

4.1 hr\*A (default) | nonnegative scalar

Cell capacity of the battery. The block calculates the state of charge by dividing the accumulated charge by this value. The block calculates accumulated charge by integrating the battery current.

**Self-discharge — Select whether to model the self-discharge resistance of the battery**

Disabled (default) | Enabled

Select whether to model the self-discharge resistance of the battery. The block models this effect as a resistor across the terminals of the fundamental battery model.

As temperature increases, self-discharge resistance decreases, causing self-discharge to increase. If the decrease in resistance is too fast, thermal runaway of the battery and numerical instability can occur. You can resolve this instability by making any of these changes:

- Decrease the thermal resistance
- Decrease the gradient of the self-discharge resistance with respect to temperature
- Increase the self-discharge resistance

**Self-discharge resistance,  $R_{\text{leak}}(T)$  — Resistance that represents battery self-discharge at temperature breakpoints**

[8000, 7000, 6000] Ohm (default) | vector of positive numbers

Lookup data for self-discharge resistance of the battery at the specified temperature breakpoints. This resistance acts across the terminals of the fundamental battery model.

**Dependencies**

Enabled when the **Self-discharge** parameter is set to Enabled and **Temperature dependent tables** is set to Yes - tabulate parameters over temperature.

**Self-discharge resistance, R<sub>leak</sub> — Resistance that represents battery self-discharge**  
7e3 Ohm (default) | positive scalar

Lookup data for self-discharge resistance of the battery. This resistance acts across the terminals of the fundamental battery model.

#### Dependencies

This parameter is visible only when the **Self-discharge** parameter is set to Enabled and the **Temperature dependent tables** parameter is set to No - do not tabulate parameters over temperature.

.

**Extrapolation method for all tables — Method of extrapolation for tables**  
Nearest (default) | Linear | Error

Extrapolation method for all the table based parameters:

- **Linear** — Estimates values beyond the dataset by creating a tangent line at the end of the known data and extending it beyond that limit.
- **Nearest** — Extrapolates a value at query point that is the value at the nearest sample grid point.
- **Error** — Returns an error if the value goes beyond the known dataset. If you select this option, the block does not use extrapolation.

**Expose SOC measurement port — Method of extrapolation for tables**  
No (default) | Yes

Whether to expose the state-of-charge measurement port.

#### Dynamics

**Charge dynamics — Battery charge dynamics model**

No dynamics (default) | One time-constant dynamics | Two time-constant dynamics | Three time-constant dynamics | Four time-constant dynamics | Five time-constant dynamics

Select how to model battery charge dynamics. This parameter determines the number of parallel RC sections in the equivalent circuit:

- **No dynamics** — The equivalent circuit contains no parallel RC sections. There is no delay between terminal voltage and internal charging voltage of the battery.
- **One time-constant dynamics** — The equivalent circuit contains one parallel RC section. Specify the time constant using the **First time constant, tau1(SOC,T)** parameter if tabulating parameters over temperature or **First time constant, tau1(SOC)** otherwise.
- **Two time-constant dynamics** — The equivalent circuit contains two parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)** and **Second time constant, tau2(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)** and **Second time constant, tau2(SOC)** otherwise.
- **Three time-constant dynamics** — The equivalent circuit contains three parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)**, **Second time constant, tau2(SOC,T)**, and **Third time constant, tau3(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)**, **Second time constant, tau2(SOC)**, and **Third time constant, tau3(SOC)** otherwise.



- **Four time-constant dynamics** — The equivalent circuit contains four parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)**, **Second time constant, tau2(SOC,T)**, **Third time constant, tau3(SOC,T)**, and **Fourth time constant, tau4(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)**, **Second time constant, tau2(SOC)**, **Third time constant, tau3(SOC)**, and **Fourth time constant, tau4(SOC)** otherwise.
- **Five time-constant dynamics** — The equivalent circuit contains five parallel RC sections. Specify the time constants using the **First time constant, tau1(SOC,T)**, **Second time constant, tau2(SOC,T)**, **Third time constant, tau3(SOC,T)**, **Fourth time constant, tau4(SOC,T)**, and **Fifth time constant, tau5(SOC,T)** parameters if tabulating parameters over temperature or **First time constant, tau1(SOC)**, **Second time constant, tau2(SOC)**, **Third time constant, tau3(SOC)**, **Fourth time constant, tau4(SOC)**, and **Fifth time constant, tau5(SOC)** otherwise.

**First polarization resistance, R1(SOC,T) — First RC resistance at temperature breakpoints**

[.089, .076, .01; .042, .022, .099; .019, .007, .002; .051, .043, .029] Ohm (default) | matrix of positive numbers

Lookup data for the first parallel RC resistance at the specified SOC and temperature breakpoints. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**First polarization resistance, R1(SOC) — First RC resistance**

[.076, .022, .007, .043] Ohm (default) | matrix of positive numbers

Lookup data for the first parallel RC resistance at the specified SOC. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**First time constant, tau1(SOC,T) — First RC time constant at temperature breakpoints**

[44, 148, 235; 93, 110, 1000; 19, 27, 133; .5, 22, 3] s (default) | matrix of positive numbers

Lookup data for the first parallel RC time constant at the specified SOC and temperature breakpoints.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**First time constant, tau1(SOC) – First RC time constant**

[148, 110, 27, 22] s (default) | matrix of positive numbers

Lookup data for the first parallel RC time constant at the specified SOC.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Second polarization resistance, R2(SOC,T) – Second RC resistance at temperature breakpoints**

[.014, .382, .407; .028, .006, .007; .014, .007, .006; .333, .956, .912] Ohm (default) | matrix of positive numbers

Lookup data for the second parallel RC resistance at the specified SOC and temperature breakpoints. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Second polarization resistance, R2(SOC) – Second RC resistance**

[.382, .006, .007, .956] Ohm (default) | matrix of positive numbers

Lookup data for the second parallel RC resistance at the specified SOC. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Second time constant, tau2(SOC,T) – Second RC time constant at temperature breakpoints**

[1, 44, 5644; 11, 24, 506; 2, 14, 330; 3310, 13419, 30216] s (default) | matrix of positive numbers

Lookup data for the second parallel RC time constant at the specified SOC and temperature breakpoints.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Second time constant, tau2(SOC) – Second RC time constant**

[44, 24, 14, 13419] s (default) | matrix of positive numbers

Lookup data for the second parallel RC time constant at the specified SOC.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Third polarization resistance,  $R_3(SOC, T)$  – Third RC resistance at temperature breakpoints**

[.014, .382, .407; .028, .006, .007; .014, .007, .006; .333, .956, .912] Ohm (default) | matrix of positive numbers

Lookup data for the third parallel RC resistance at the specified SOC and temperature breakpoints. This parameter primarily affects the ohmic losses of the RC section and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics.

**Third polarization resistance,  $R_3(SOC)$  – Third RC resistance**

[.382, .006, .007, .956] Ohm (default) | matrix of positive numbers

Lookup data for the third parallel RC resistance at the specified SOC. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Third time constant,  $\tau_3(SOC, T)$  – Third RC time constant at temperature breakpoints**

[1, 44, 5644; 11, 24, 506; 2, 14, 330; 3310, 13419, 30216] s (default) | matrix of positive numbers

Lookup data for the third parallel RC time constant at the specified SOC and temperature breakpoints.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Third time constant,  $\tau_3(SOC)$  – Third RC time constant**

[44, 24, 14, 13419] s (default) | matrix of positive numbers

Lookup data for the third parallel RC time constant at the specified SOC.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Fourth polarization resistance, R4(SOC,T) – Fourth RC resistance at temperature breakpoints**

[.014, .382, .407; .028, .006, .007; .014, .007, .006; .333, .956, .912] Ohm (default) | matrix of positive numbers

Lookup data for the fourth parallel RC resistance at the specified SOC and temperature breakpoints. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Fourth polarization resistance, R4(SOC) – Fourth RC resistance**

[.382, .006, .007, .956] Ohm (default) | matrix of positive numbers

Lookup data for the fourth parallel RC resistance at the specified SOC. This parameter primarily affects the ohmic losses of the RC section.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Fourth time constant, tau4(SOC,T) – Fourth RC time constant at temperature breakpoints**

[1, 44, 5644; 11, 24, 506; 2, 14, 330; 3310, 13419, 30216] s (default) | matrix of positive numbers

Lookup data for the fourth parallel RC time constant at the specified SOC and temperature breakpoints.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

**Fourth time constant, tau4(SOC) – Fourth RC time constant**

[44, 24, 14, 13419] s (default) | matrix of positive numbers

Lookup data for the fourth parallel RC time constant at the specified SOC.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics or Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

**Fifth polarization resistance, R5(SOC,T) – Fifth RC resistance at temperature breakpoints**

[.014, .382, .407; .028, .006, .007; .014, .007, .006; .333, .956, .912] Ohm (default) | matrix of positive numbers

Lookup data for the fifth parallel RC resistance at the specified SOC and temperature breakpoints. This parameter primarily affects the ohmic losses of the RC section.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

#### Fifth polarization resistance, R5(SOC) – Fifth RC resistance

[.382, .006, .007, .956] Ohm (default) | matrix of positive numbers

Lookup data for the fifth parallel RC resistance at the specified SOC. This parameter primarily affects the ohmic losses of the RC section.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

#### Fifth time constant, tau5(SOC,T) – Fifth RC time constant at temperature breakpoints

[1, 44, 5644; 11, 24, 506; 2, 14, 330; 3310, 13419, 30216] s (default) | matrix of positive numbers

Lookup data for the fifth parallel RC time constant at the specified SOC and temperature breakpoints.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Temperature dependent tables** to Yes - tabulate parameters over temperature.

#### Fifth time constant, tau5(SOC) – Fifth RC time constant

[44, 24, 14, 13419] s (default) | matrix of positive numbers

Lookup data for the fifth parallel RC time constant at the specified SOC.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Temperature dependent tables** to No - do not tabulate parameters over temperature.

## Fade

### Fade characteristic – Select how to define fade characteristic

Disabled (default) | Equations | Lookup tables (temperature independent) | Lookup tables (temperature dependent)

Select how to define fade characteristic:

- **Disabled** — The block does not model fade characteristic.
- **Equations** — The cell capacity and terminal resistance will be proportional to  $\sqrt{N}$  whilst the open-circuit voltage will be proportional to  $N$ . If the self-discharge resistance or any number of the time constants are enabled, their values will be proportional to  $\sqrt{N}$ .
- **Lookup tables (temperature independent)** — Set tabulated data for the percentage change in parameters as a function of  $N$ .
- **Lookup tables (temperature dependent)** — Set tabulated data for the percentage change in parameters as a function of  $N$  and temperature.

**Number of discharge cycles, N – Reference number of cycles for percent change calculations**

100 (default) | number from 1 to Inf

The number of charge-discharge cycles over which the specified percent changes occur.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Equations.

**Change in open-circuit voltage after N discharge cycles (%) – Percent change in open-circuit voltage after N cycles**

0 (default) | scalar

Percent change in the open-circuit voltage after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Equations.

**Change in terminal resistance after N discharge cycles (%) – Percent change in series resistance after N cycles**

0 (default) | scalar

Percent change in the series resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Equations.

**Change in cell capacity after N discharge cycles (%) – Percent change in cell capacity after N cycles**

0 (default) | scalar

Percent change in the cell capacity after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Equations.

**Change in self-discharge resistance after N discharge cycles (%) – Percent change in self-discharge resistance after N cycles**

0 (default) | scalar

Percent change in the self-discharge resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Self-discharge** to Enabled and **Fade characteristic** to Equations.

**Change in first polarization resistance after N discharge cycles (%) – Percent change in first RC resistance after N cycles**

0 (default) | scalar

Percent change in the first RC resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Equations.

**Change in second polarization resistance after N discharge cycles (%) — Percent change in second RC resistance after N cycles**

0 (default) | scalar

Percent change in the second RC resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Equations.

**Change in third polarization resistance after N discharge cycles (%) — Percent change in third RC resistance after N cycles**

0 (default) | scalar

Percent change in the third RC resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Equations.

**Change in fourth polarization resistance after N discharge cycles (%) — Percent change in fourth RC resistance after N cycles**

0 (default) | scalar

Percent change in the fourth RC resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Equations.

**Change in fifth polarization resistance after N discharge cycles (%) — Percent change in fifth RC resistance after N cycles**

0 (default) | scalar

Percent change in the fifth RC resistance after the battery undergoes *N* discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Fade characteristic** to Equations.

**Vector of discharge cycle values, N — Reference vector or numbers of cycles for percent change calculations**

[100, 200, 300] (default) | number from 1 to Inf

Vector of numbers of charge-discharge cycles over which the specified percent changes occur.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature independent) or Lookuptables (temperature dependent).

**Vector of temperatures for fade data, Tfade — Vector temperatures at which fade lookup tables has been extracted**

[298.15, 323.15] K (default) | number from 1 to Inf

Vector of temperatures at which fade lookup tables has been extracted. These temperatures are completely independent of **Vectors of temperatures, T** parameter from the **Main** tab.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature dependent).

**Percentage change in open-circuit voltage, dV0(N) — Vector of percent change in open-circuit voltage after N cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent changes in the open-circuit voltage after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature independent).

**Percentage change in terminal resistance, dR0(N) — Vector of percent change in series resistance after N cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the series resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature independent).

**Percentage change in cell capacity, dAH(N) — Vector of percent change in cell capacity after N cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the cell capacity after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature independent).

**Percentage change in self-discharge resistance, dRleak(N) — Vector of percent change in self-discharge resistance after N cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the self-discharge resistance after the battery undergoes  $N$  discharge cycles.



**Dependencies**

To enable this parameter, set **Self-discharge** to Enabled and **Fade characteristic** to Lookup tables (temperature independent).

**Percentage change in first polarization resistance,  $dR1(N)$  — Vector of percent change in first RC resistance after  $N$  cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the first RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature independent).

**Percent change in second polarization resistance,  $dR2(N)$  — Vector of percent change in second RC resistance after  $N$  cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the second RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature independent).

**Percent change in third polarization resistance,  $dR3(N)$  — Vector of percent change in third RC resistance after  $N$  cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the third RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature independent).

**Percent change in fourth polarization resistance,  $dR4(N)$  — Vector of percent change in fourth RC resistance after  $N$  cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the fourth RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature independent).

**Percent change in fifth polarization resistance,  $dR5(N)$  — Vector of percent change in fifth RC resistance after  $N$  cycles**

[0, 0, 0] (default) | vector of scalars

Vector of percent change in the fifth RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature independent).

**Percentage change in open-circuit voltage,  $dV_0(N, T_{fade})$  — Matrix of percent change in open-circuit voltage after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent changes in the open-circuit voltage after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature dependent).

**Percentage change in terminal resistance,  $dR_0(N, T_{fade})$  — Matrix of percent change in series resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the series resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature dependent).

**Percentage change in cell capacity,  $dAH(N, T_{fade})$  — Matrix of percent change in cell capacity after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the cell capacity after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Fade characteristic** to Lookup tables (temperature dependent).

**Percentage change in self-discharge resistance,  $dR_{leak}(N, T_{fade})$  — Matrix of percent change in self-discharge resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the self-discharge resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Self-discharge** to Enabled and **Fade characteristic** to Lookup tables (temperature dependent).

**Percentage change in first polarization resistance,  $dR_1(N, T_{fade})$  — Matrix of percent change in first RC resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the first RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to One time-constant dynamics, Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature dependent).

**Percent change in second polarization resistance,  $dR2(N, T_{fade})$  — Matrix of percent change in second RC resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the second RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Two time-constant dynamics, Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature dependent).

**Percent change in third polarization resistance,  $dR3(N, T_{fade})$  — Matrix of percent change in third RC resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the third RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Three time-constant dynamics, Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature dependent).

**Percent change in fourth polarization resistance,  $dR4(N, T_{fade})$  — Matrix of percent change in fourth RC resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the fourth RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Four time-constant dynamics, or Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature dependent).

**Percent change in fifth polarization resistance,  $dR5(N, T_{fade})$  — Matrix of percent change in fifth RC resistance after  $N$  cycles**

[0, 0; 0, 0; 0, 0] (default) | matrix of scalars

Matrix of percent change in the fifth RC resistance after the battery undergoes  $N$  discharge cycles.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Five time-constant dynamics and **Fade characteristic** to Lookup tables (temperature dependent).

## **Calendar Aging**

### **Internal resistance calendar aging – Calendar aging for internal resistance**

Disabled (default) | Enabled

Whether to enable calendar aging of the internal resistance of the battery.

### **Capacity calendar aging – Calendar aging for capacity**

Disabled (default) | Enabled

Whether to enable calendar aging of the capacity of the battery.

### **Modeling option – Calendar aging option**

Equation-based (default) | Tabulated: temperature | Tabulated: time and temperature

Modeling option for internal resistance and capacity calendar aging.

#### **Dependencies**

To enable this parameter, set either **Internal resistance calendar aging** or **Capacity calendar aging** to Enabled.

### **Vector of time intervals – Time intervals**

[0] d (default) | scalar

Time intervals. This parameter must be equal in size to **Vector of storage temperatures**.

#### **Dependencies**

To enable this parameter, set **Modeling option** to either Equation-based, Tabulated: temperature, or Tabulated: time and temperature.

### **Vector of storage temperatures – Storage temperatures**

[273] K (default) | scalar

Set of storage temperatures. This parameter must be equal in size to **Vector of time intervals**.

#### **Dependencies**

To enable this parameter, set **Modeling option** to either Equation-based, Tabulated: temperature, or Tabulated: time and temperature.

### **Storage condition – Storage condition**

Specify open-circuit voltage during storage (default) | Specify state-of-charge during storage

Whether to specify the open-circuit voltage or the state of charge during storage.

#### **Dependencies**

To enable this parameter, set **Modeling option** to Equation-based.

### **Normalized open-circuit voltage during storage, V/Vnom – Normalized open-circuit voltage during storage**

0.9 (default) | scalar

Normalized open-circuit voltage during storage.

**Dependencies**

To enable this parameter, set **Modeling option** to Equation-based and **Storage condition** to Specify open-circuit voltage during storage.

**State of charge during storage — Percentage state of charge during storage**

60 (default) | positive number

State of charge during storage, in percentage.

**Dependencies**

To enable this parameter, set **Modeling option** to Equation-based and **Storage condition** to Specify state-of-charge during storage.

**Terminal resistance linear scaling for voltage, b — Voltage linear scaling in terminal resistance**

2.2134e6 (default) | scalar

Linear scaling coefficient in terminal resistance for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Equation-based.

**Terminal resistance constant offset for voltage, c — Voltage constant offset in terminal resistance**

1.632e6 (default) | scalar

Constant offset in terminal resistance for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Equation-based.

**Terminal resistance temperature-dependent exponential increase, d — Temperature-dependent exponential increase in terminal resistance**

0.515833569 V (default) | scalar

Temperature-dependent exponential increase in terminal resistance.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Equation-based.

**Terminal resistance time exponent, a — Time exponent in terminal resistance**

0.75 (default) | scalar

Time exponent in terminal resistance.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to either Equation-based or Tabulated: temperature.

**Capacity linear scaling for voltage, b — Voltage linear scaling in capacity**

1.5097e07 (default) | scalar

Linear scaling coefficient in capacity for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Equation-based.

**Capacity constant offset for voltage, c — Voltage constant offset in capacity**

8.3625e06 (default) | scalar

Constant offset in capacity for open-circuit voltage.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Equation-based.

**Capacity temperature-dependent exponential increase, d — Temperature-dependent exponential increase in capacity**

0.6011 V (default) | scalar

Temperature-dependent exponential increase in capacity.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Equation-based.

**Capacity time exponent, a — Time exponent in capacity**

0.75 (default) | scalar

Time exponent in capacity.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to either Equation-based or Tabulated: temperature.

**Vector of sampled temperatures for terminal resistance calendar aging, T\_ar — Sampled temperatures for terminal resistance calendar aging**

[273.15, 298.15, 323.15] K (default) | vector of scalars

Vector of sampled temperatures for terminal resistance calendar aging.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to either Tabulated: temperature or Tabulated: time and temperature.

**Percentage change in terminal resistance due to calendar aging, dR0(T\_ar) — Percentage change in terminal resistance due to calendar aging**

[0, 0, 0] (default) | vector of scalars

Percentage change in terminal resistance due to calendar aging.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Tabulated: time.

**Time between terminal resistance beginning of life and  $dR(T_{ar})$  measurement — Time between terminal resistance beginning of life and  $dR(T_{ar})$  measurement**

100 d (default) | scalar

Time between the beginning of life of the terminal resistance and the  $dR(T_{ar})$  measurement.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Tabulated: time.

**Vector of sampled temperatures for capacity calendar aging,  $T_{ac}$  — Sampled temperatures for capacity calendar aging**

[273.15, 298.15, 323.15] K (default) | vector of scalars

Vector of sampled temperatures for capacity calendar aging.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to either Tabulated: temperature or Tabulated: time and temperature.

**Percentage change in capacity due to calendar aging,  $dAH(T_{ac})$  — Percentage change in capacity due to calendar aging**

[0, 0, 0] (default) | vector of scalars

Percentage change in capacity due to calendar aging.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Tabulated: time.

**Time between capacity beginning of life and  $dAH(T_{ac})$  measurement — Time between capacity beginning of life and  $dAH(T_{ac})$  measurement**

100 d (default) | scalar

Time between the beginning of life of the capacity and the  $dR(T_{ac})$  measurement.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Tabulated: time.

**Vector of sampled storage time intervals for terminal resistance calendar aging,  $t_{ar}$  — Sampled storage time intervals for terminal resistance calendar aging**

[90, 180, 270, 360] d (default) | vector of scalars

Vector of sampled storage time intervals for terminal resistance calendar aging.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Tabulated: time and temperature.

**Percentage change in terminal resistance due to calendar aging,  $dR0(t_{ar}, T_{ar})$  – Percentage change in terminal resistance due to calendar aging**

[0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0] (default) | matrix of scalars

Percentage change in terminal resistance due to calendar aging.

**Dependencies**

To enable this parameter, set **Internal resistance calendar aging** to Enabled and **Modeling option** to Tabulated: time and temperature.

**Vector of sampled storage time intervals for capacity calendar aging,  $t_{ac}$  – Sampled storage time intervals for capacity calendar aging**

[90, 180, 270, 360] d (default) | vector of scalars

Vector of sampled storage time intervals for capacity calendar aging.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Tabulated: time and temperature.

**Percentage change in capacity due to calendar aging,  $dAH(t_{ac}, T_{ac})$  – Percentage change in capacity due to calendar aging**

[0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0] (default) | matrix of scalars

Percentage change in capacity due to calendar aging.

**Dependencies**

To enable this parameter, set **Capacity calendar aging** to Enabled and **Modeling option** to Tabulated: time and temperature.

**Thermal**

**Thermal port – Thermal port visibility**

Omit (default) | Model

Whether to expose the thermal port.

**Simulation temperature – Battery temperature**

298.15 K (default) | positive number

Battery temperature used in lookup tables throughout simulation.

**Dependencies**

To enable this parameter, set **Thermal port** to Omit and **Temperature dependent tables** to Yes - tabulate parameters over temperature or **Fade characteristic** to Lookup tables (temperature dependent). For more information, see “Modeling Thermal Effects” on page 1-53.

**Thermal mass – Thermal mass associated with the thermal port**

30000 J/K (default)



Thermal mass associated with the thermal port H. It represents the energy required to raise the temperature of the thermal port by one degree.

**Dependencies**

To enable this parameter, set **Thermal port** to Model. For more information, see “Modeling Thermal Effects” on page 1-53.

**Version History**

**Introduced in R2018a**

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Battery

**Topics**

“List of Pre-Parameterized Components”

# Bidirectional DC-DC Converter

Controller-driven bidirectional DC-DC step-up and step-down voltage regulator

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Bidirectional DC-DC Converter block represents a converter that steps up or steps down DC voltage from either side of the converter to the other as driven by an attached controller and gate-signal generator. Bidirectional DC-DC converters are useful for switching between energy storage and use, for example, in electric vehicles.

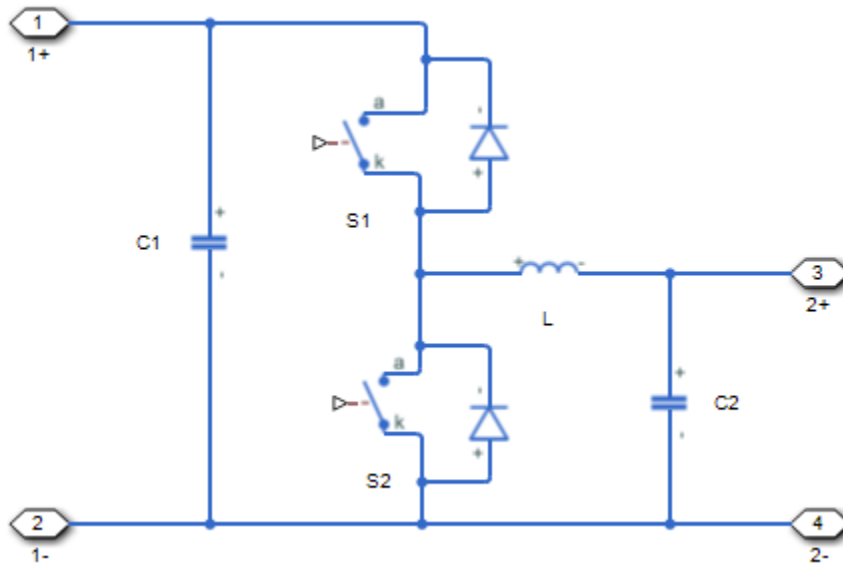
The Bidirectional DC-DC Converter block allows you to model a nonisolated converter with two switching devices, an isolated converter with six switching devices, or a dual active bridge converter with eight switching devices. Options for the type of switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

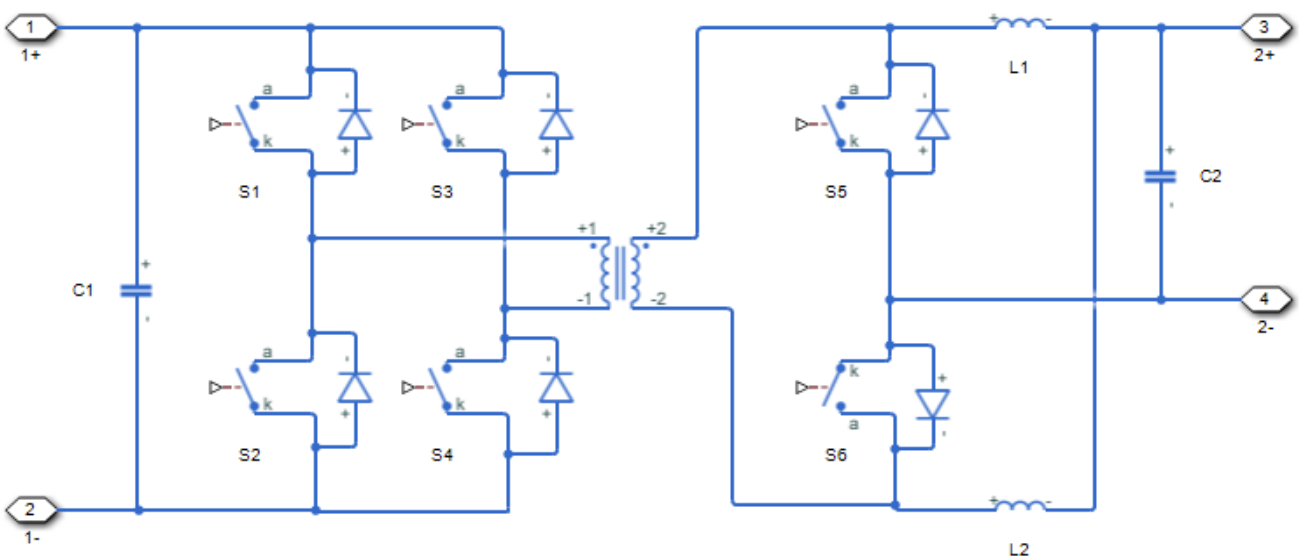
## Model

You can model three different types of bidirectional DC-DC converter. To access the different options, double-click the block and set the **Modeling option** parameter to either:

- **Nonisolated converter** — Bidirectional DC-DC converter without an electrical barrier. This converter contains an inductor, two capacitors, and two switches that are of the same device type.

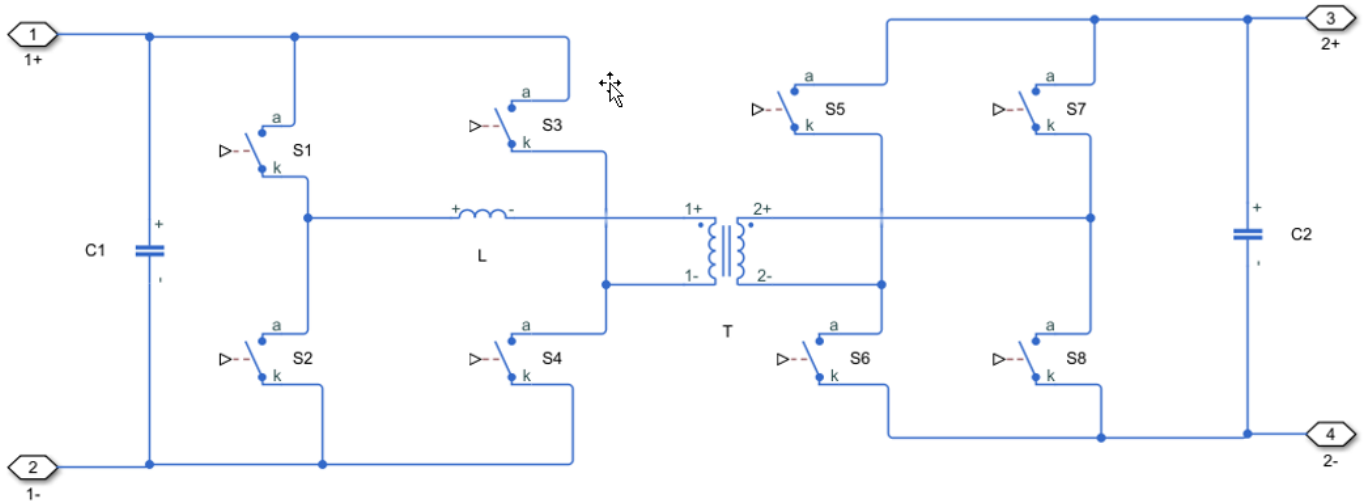


- **Isolated converter** — Bidirectional DC-DC converter with an electrical barrier. This converter contains four additional switches that form a full bridge. The full bridge is on the input or high-voltage (HV) side of the converter. The other two switches are on the output or low-voltage (LV) side of the converter. You can select different semiconductor types for the HV and LV switching devices. For example, you can use a GTO for the HV switching devices and an IGBT for the LV switching devices. To provide separation between the input and output voltages, the model uses a high-frequency transformer.



- **Dual Active Bridge converter** — This bidirectional DC-DC converter contains two full-bridges. The left bridge is the input or high-voltage (HV) side of the converter. The right bridge is the output or low-voltage (LV) side of the converter. You can select different semiconductor types for the HV and LV switching devices. For example, you can use a GTO for the HV switching

devices and an IGBT for the LV switching devices. To provide separation between the input and output voltages, the model uses a high-frequency transformer.



### Protection

The block contains an integral protection diode for each switching device. The integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

To configure the internal protection diode block, use the **Protection Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.

Goals	Value to Select	Integral Protection Diode
Prioritize simulation speed.	Diode with no dynamics	The Diode block
Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

### Gate Control

To connect Simulink® gate-control voltage signals to the gate ports of the switching devices:

- 1 Convert each voltage signal using a Simulink-PS Converter block.
- 2 Mux the converted gate signals into a single vector.
  - For a nonisolated converter model, use a Two-Pulse Gate Multiplexer block.
  - For an isolated converter model, use a Six-Pulse Gate Multiplexer block.

- For a dual active bridge converter model, use a Eight-Pulse Gate Multiplexer block.

3 Connect the vector signal to the **G** port.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to **Averaged switch** and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

### Assumptions

A source impedance or a nonzero equivalent-series resistance (ESR) is connected to the left side of the Bidirectional DC-DC Converter block.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **G** — Switching device gate control

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

Data Types: double

#### **1+** — Positive DC voltage 1

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

**1- – Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

**2+ – Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

**2- – Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

## **Parameters**

**Modeling option – Converter model**

Nonisolated converter (default) | Isolated converter | Dual Active Bridge converter

Whether to model a nonisolated converter with two switching devices, an isolated converter with six switching devices, or a dual active bridge converter with eight switching devices.

### **Switching Devices**

These tables show how the visibility of **Switching Devices** parameters depends on the converter model and switching devices that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Nonisolated Converter Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

**Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies**

<b>Parameters and Options</b>					
<b>Switching device HV</b>					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance HV	Forward voltage HV	Forward voltage HV	Drain-source on resistance HV	Forward voltage HV	On-state resistance HV
Off-state conductance HV	On-state resistance HV	On-state resistance HV	Off-state conductance HV	On-state resistance HV	
Threshold voltage HV	Off-state conductance HV	Off-state conductance HV	Threshold voltage HV	Off-state conductance HV	
	Gate trigger voltage HV, Vgt_hv	Threshold voltage HV		Gate trigger voltage HV, Vgt_hv	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage HV, Vgt_off_hv			Gate turn-off voltage HV, Vgt_off_hv	
	Holding current HV			Holding current HV	
<b>Switching device LV</b>					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance LV	Forward voltage LV	Forward voltage LV	Drain-source on resistance LV	Forward voltage LV	On-state resistance LV
Off-state conductance LV	On-state resistance LV	On-state resistance LV	Off-state conductance LV	On-state resistance LV	
Threshold voltage LV	Off-state conductance LV	Off-state conductance LV	Threshold voltage LV	Off-state conductance LV	
	Gate trigger voltage LV, Vgt	Threshold voltage LV		Gate trigger voltage LV, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage LV, Vgt_off_lv			Gate turn-off voltage LV, Vgt_off_lv	
	Holding current LV			Holding current LV	

**Switching device – Switch type**

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the nonisolated converter model.



**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Forward voltage — Voltage**

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**On-state resistance — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance

- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Gate trigger voltage, Vgt — Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off — Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Holding current — Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Nonisolated Converter Switching Devices Parameter Dependencies table.

**Switching device HV — Switch**

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the high-voltage side of the isolated converter model.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Forward voltage HV — Voltage**

0.8 Ohm (default) | scalar

For the different switching device types, the **Forward voltage HV** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Drain-source on resistance HV — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**On-state resistance HV — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance HV** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Off-state conductance HV — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance HV**.

For the different switching device types, the **On-state resistance HV** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Threshold voltage HV — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Gate trigger voltage HV, Vgt\_hv — Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Gate turn-off voltage HV, Vgt\_off\_hv — Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Holding current HV — Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Switching device LV – Switch**

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the low-voltage side of the isolated converter model.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Forward voltage LV – Voltage**

0.8 Ohm (default) | scalar

For the different switching device types, the **Forward voltage LV** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Drain-source on resistance LV – Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**On-state resistance LV – Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance LV** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Off-state conductance LV – Conductance**

1e-5 1/0hm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance LV**.

For the different switching device types, the **On-state resistance LV** is taken as:

- GTO – Anode-cathode conductance
- Ideal semiconductor switch – Anode-cathode conductance
- IGBT – Collector-emitter conductance
- MOSFET – Drain-source conductance
- Thyristor – Anode-cathode conductance

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Threshold voltage LV – Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch – Gate-emitter voltage
- IGBT – Gate-cathode voltage
- MOSFET – Gate-source voltage

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Gate trigger voltage LV, Vgt\_lv – Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Gate turn-off voltage LV, Vgt\_off\_lv – Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Holding current LV – Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Isolated Converter and Dual Active Bridge Converter Switching Devices Parameter Dependencies table.

**Integer for piecewise constant approximation of gate input (0 for disabled) – Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Protection Diode**

The visibility of **Protection Diode** parameters depends on how you configure the protection diode **Model dynamics** and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Protection Diode Parameter Dependencies**

Parameters and Options			
<b>Model dynamics</b>			
<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>		
Forward voltage	Forward voltage		
On resistance	On resistance		
Off conductance	Off conductance		
	Junction capacitance		
	Peak reverse current, iRM		
	Initial forward current when measuring iRM		
	Rate of change of current when measuring iRM		
	<b>Reverse recovery time parameterization</b>		
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>
Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr	

**Model dynamics – Diode model**

Diode with no dynamics (default) | Diode with charge dynamics

Diode type. The options are:

- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

---

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Forward voltage – Voltage**

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.**On resistance – Resistance**

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.**Off conductance – Conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Junction capacitance – Capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Peak reverse current,  $i_{RM}$  – Current**

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Initial forward current when measuring  $i_{RM}$  – Current**

300 A (default) | scalar greater than 0



Initial forward current when measuring peak reverse current. This value must be greater than zero.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Rate of change of current when measuring $i_{RM}$ – Current change rate

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery time parameterization – Recovery-time model

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select `Specify stretch factor` or `Specify reverse recovery charge`, you can specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery time stretch factor – Stretch factor

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery time, $t_{rr}$ – Time

15 us (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery charge, $Q_{rr}$ – Charge

1500 s\*uA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Transformer

To enable these parameters, set **Modeling option** to Isolated converter.

#### Transformer inductance L1 — Inductance

10 H (default) | positive scalar

Self-inductance of the first winding of the transformer.

### Dependencies

To enable this parameter, set **Modeling option** to Isolated converter.

#### Transformer inductance L2 — Inductance

0.1 H (default) | positive scalar

Self-inductance of the second winding of the transformer.

### Dependencies

To enable this parameter, set **Modeling option** to Isolated converter.

#### Transformer coefficient of coupling — Coupling coefficient

0.9 (default) | positive scalar greater than zero and less than 1

Defines the mutual inductance of the transformer.

### Dependencies

To enable this parameter, set **Modeling option** to Isolated converter.

### LC Parameters

#### Inductance, L — Inductance

1e-6 H (default) | positive scalar

Converter inductance. If you set the **Modeling option** parameter to Isolated converter, the two inductors are identical.

**Inductor series resistance – Inductor series resistance**

0 Ohm (default) | zero or positive scalar

Series resistance of the inductor.

**Capacitance, C1 – Capacitance**

1e-7 F (default) | positive scalar

Capacitance of the first DC terminal.

**Capacitance, C2 – Capacitance**

1e-7 F (default) | positive scalar

Capacitance of the second DC terminal.

**C1 effective series resistance – Resistance**

1e-6 Ohm (default) | zero or positive scalar

Series resistance of capacitor *C1*.**C2 effective series resistance – Resistance**

1e-6 Ohm (default) | zero or positive scalar

Series resistance of capacitor *C2*.**Snubbers**The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.**Snubbers Parameter Dependencies**

<b>Snubbers Parameter Dependencies</b>			
<b>Modeling option</b>			
<i>Nonisolated Converter</i>		<i>Isolated Converter</i>	
<b>Snubber</b>		<b>Snubber HV</b>	
<i>None</i>	<i>RC Snubber</i>	<b>None</b>	<i>RC Snubber</i>
	Snubber resistance		Snubber resistance HV
	Snubber capacitance		Snubber capacitance HV
		<b>Snubber LV</b>	
		<i>None</i>	<i>RC Snubber</i>
			Snubber resistance LV
			Snubber capacitance LV

**Snubber – Snubber model**

None (default) | RC snubber

Snubber for each switching device.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber resistance — Resistance**

0.1 Ohm (default) | scalar

Resistance of the snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber capacitance — Capacitance**

1e-7 F (default) | scalar

Capacitance of the snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber HV — Snubber model**

None (default) | RC snubber

HV snubber for each switching device.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber resistance HV — Resistance**

0.1 Ohm (default) | scalar

Resistance of the high-voltage snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber capacitance HV — Capacitance**

1e-7 F (default) | scalar

Capacitance of the high-voltage snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber LV — Snubber model**

None (default) | RC snubber

LV snubber for each switching device.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber resistance LV — Resistance**

0.1 Ohm (default) | scalar

Resistance of the low-voltage snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber capacitance LV — Capacitance**

1e-7 F (default) | scalar

Capacitance of the low-voltage snubbers.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Version History**

Introduced in R2018a

**References**

- [1] Saleh, M., Y. Esa, Y. Mhandi, W. Brandauer, and A. Mohamed. *Design and implementation of CCNY DC microgrid testbed*. Industry Applications Society Annual Meeting. Portland, OR: 2016, pp 1-7.
- [2] Kutkut, N. H., and G. Luckjiff. *Current mode control of a full bridge DC-to-DC converter with a two inductor rectifier*. Power Electronics Specialists Conference. Saint Louis, MO: 1997, pp 203-209.
- [3] Nene, H. *Digital control of a bi-directional DC-DC converter for automotive applications*. Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC). Long Beach, CA: 2013, pp 1360-1365.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

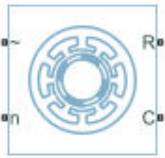
**See Also**

Average-Value DC-DC Converter | Buck Converter | Buck-Boost Converter | Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear)

## BLDC

Three-winding brushless DC motor with trapezoidal flux distribution

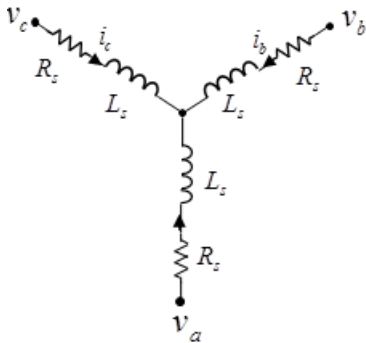
**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



### Description

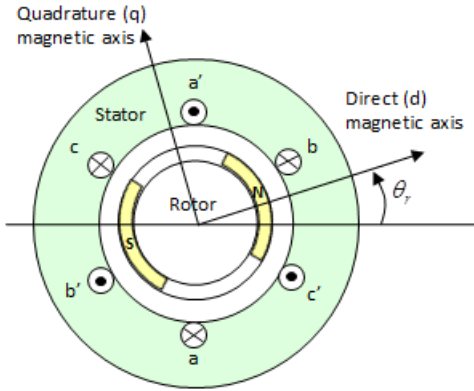
The BLDC block models a permanent magnet synchronous machine with a three-phase wye-wound stator. The block has four options for defining the permanent magnet flux distribution as a function of rotor angle. Two options allow for simple parameterization by assuming a perfect trapezoid for the back emf. For simple parameterization, you specify either the flux linkage or the rotor-induced back emf. The other two options give more accurate results using tabulated data that you specify. For more accurate results, you specify either the flux linkage partial derivative or the measured back emf constant for a given rotor speed.

The figure shows the equivalent electrical circuit for the stator windings.



### Motor Construction

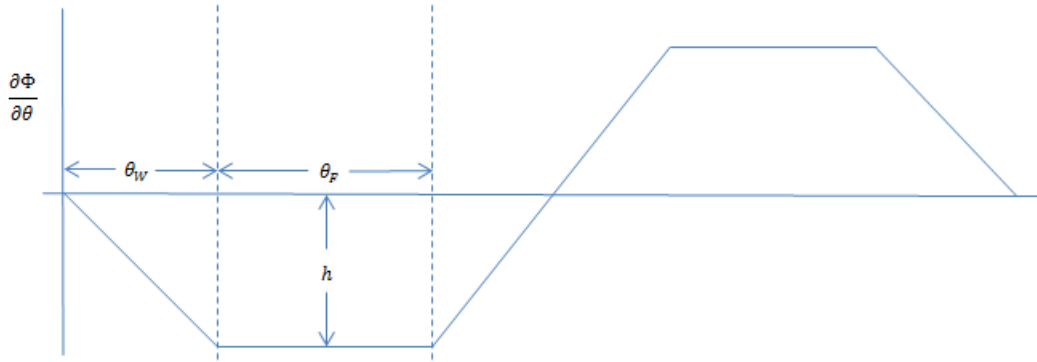
This figure shows the motor construction with a single pole-pair on the rotor.



For the axes convention in the preceding figure, the *a*-phase and permanent magnet fluxes are aligned when rotor angle  $\theta_r$  is zero. The block supports a second rotor-axis definition. For the second definition, the rotor angle is the angle between the *a*-phase magnetic axis and the rotor *q*-axis.

**Trapezoidal Rate of Change of Flux**

The rotor magnetic field due to the permanent magnets create a trapezoidal rate of change of flux with rotor angle. The figure shows this rate of change of flux.



Back emf is the rate of change of flux, defined by

$$\frac{d\Phi}{dt} = \frac{\partial\Phi}{\partial\theta} \frac{d\theta}{dt} = \frac{\partial\Phi}{\partial\theta} \omega,$$

where:

- $\Phi$  is the permanent magnet flux linkage.
- $\theta$  is the rotor angle.
- $\omega$  is the mechanical rotational speed.

The height *h* of the trapezoidal rate of change of flux profile is derived from the permanent magnet peak flux.

Integrating  $\frac{\partial\Phi}{\partial\theta}$  over the range 0 to  $\pi/2$ ,

$$\Phi_{max} = \frac{h}{2}(\theta_F + \theta_W),$$

where:

- $\Phi_{max}$  is the permanent magnet flux linkage.
- $h$  is the rate of change of flux profile height.
- $\theta_F$  is the rotor angle range over which the back emf that the permanent magnet flux induces in the stator is constant.
- $\theta_W$  is the rotor angle range over which back emf increases or decreases linearly when the rotor moves at constant speed.

Rearranging the preceding equation,

$$h = 2\Phi_{max}/(\theta_F + \theta_W).$$

### Electrical Defining Equations

Voltages across the stator windings are defined by

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the external voltages applied to the three motor electrical connections.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ , and  $\frac{d\psi_c}{dt}$

are the rates of change of magnetic flux in each stator winding.

The permanent magnet and the three windings contribute to the total flux linking each winding. The total flux is defined by

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix},$$

where:

- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the total fluxes linking each stator winding.
- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ , etc. are the mutual inductances of the stator windings.
- $\psi_{am}$ ,  $\psi_{bm}$ , and  $\psi_{cm}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of rotor angle, defined by

$$L_{aa} = L_s + L_m \cos(2\theta_r),$$



$$L_{bb} = L_s + L_m \cos(2(\theta_r - 2\pi/3)),$$

$$L_{cc} = L_s + L_m \cos(2(\theta_r + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_r + \pi/6)),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_r + \pi/6 - 2\pi/3)),$$

and

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_r + \pi/6 + 2\pi/3)),$$

where:

- $L_s$  is the stator self-inductance per phase — The average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation — The fluctuation in self-inductance and mutual inductance with changing rotor angle.
- $M_s$  is the stator mutual inductance — The average mutual inductance between the stator windings.

The permanent magnet flux linking each stator winding follows the trapezoidal profile shown in the figure. The block implements the trapezoidal profile using lookup tables to calculate permanent magnet flux values.

### Simplified Equations

The defining voltage and torque equations for the block are

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \left( \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - N\omega \begin{bmatrix} \frac{\partial \psi_{am}}{\partial \theta_r} \\ \frac{\partial \psi_{bm}}{\partial \theta_r} \\ \frac{\partial \psi_{cm}}{\partial \theta_r} \end{bmatrix} \right),$$

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N\omega i_q L_q$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega i_d L_d,$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

and

$$T = \frac{3}{2} N (i_q i_d L_d - i_d i_q L_q) + [i_a \ i_b \ i_c] \begin{bmatrix} \frac{\partial \psi_{am}}{\partial \theta_r} \\ \frac{\partial \psi_{bm}}{\partial \theta_r} \\ \frac{\partial \psi_{cm}}{\partial \theta_r} \end{bmatrix},$$

where:

- $v_d$ ,  $v_q$ , and  $v_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence voltages.
- $P$  is Park's Transformation, defined by

$$P = 2/3 \begin{bmatrix} \cos\theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin\theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

- $N$  is the number of rotor permanent magnet pole pairs.
- $\omega$  is the rotor mechanical rotational speed.
- $\frac{\partial \psi_{am}}{\partial \theta_r}$ ,  $\frac{\partial \psi_{bm}}{\partial \theta_r}$ , and  $\frac{\partial \psi_{cm}}{\partial \theta_r}$

are the partial derivatives of instantaneous permanent magnet flux linking each phase winding.

- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

- $L_d = L_s + M_s + 3/2 L_m$ .  $L_d$  is the stator  $d$ -axis inductance.
- $L_q = L_s + M_s - 3/2 L_m$ .  $L_q$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 2M_s$ .  $L_0$  is the stator zero-sequence inductance.
- $T$  is the rotor torque. Torque flows from the motor case (block physical port C) to the motor rotor (block physical port R).

### Calculating Iron Losses

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. The iron losses model, which is based on the work of Mellor [3].

The term representing the main magnetizing path depends on the induced RMS stator voltage,  $V_{m_{rms}}$ :

$$P_{OC}(V_{m_{rms}}) = \frac{a_h}{k} V_{m_{rms}} + \frac{a_j}{k^2} V_{m_{rms}}^2 + \frac{a_{ex}}{k^{1.5}} V_{m_{rms}}^{1.5}$$

This is the dominant term during no-load operation.  $k$  is the back emf constant relating RMS volts per Hz. It is defined as  $k = V_{m_{rms}}/f$ , where  $f$  is the electrical frequency. The first term on the right-hand side is the magnetic hysteresis loss, the second is the eddy current loss and the third is the excess loss. The three coefficients appearing on the numerators are derived from the values that you provide for the open-circuit hysteresis, eddy, and excess losses.

The term representing the cross-tooth tip path becomes important when a demagnetizing field is set up and can be determined from a finite element analysis short-circuit test. It depends on the RMS emf associated with the cross-tooth tip flux,  $V_{d_{rms}}^*$ :

$$P_{SC}(V_{d_{rms}}^*) = \frac{b_h}{k} V_{d_{rms}}^* + \frac{b_j}{k^2} V_{d_{rms}}^{*2} + \frac{b_{ex}}{k^{1.5}} V_{d_{rms}}^{*1.5}$$

The three numerator terms are derived from the values you provide for the short-circuit hysteresis, eddy, and excess losses.

### Predefined Parameterization

There are multiple available built-in parameterizations for the BLDC block.

This pre-parameterization data allows you to set up the block to represent components by specific suppliers. The parameterizations of these brushless DC motors match the manufacturer data sheets. To load a predefined parameterization, double-click the BLDC block, click the **<click to select>** hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

~ — **Three-phase port**  
electrical

Expandable three-phase port.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**n – Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

**Dependencies**

To enable this port, set **Winding Type** to Wye-wound and **Zero sequence** to Include.

**R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the motor rotor.

**C – Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

**HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Selected part — Predefined parameterization option**

<click to select> (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see "Load Predefined Parameterizations".

**Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Rotor****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Winding type — Windings configuration**

Wye-wound (default) | Delta-wound

Select the configuration for the windings:

- Wye-wound — The windings are wye-wound.
- Delta-wound — The windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

**Back EMF profile — Back EMF profile**

Perfect trapezoid - specify maximum flux linkage (default) | Perfect trapezoid - specify maximum rotor-induced back emf | Tabulated - specify flux partial derivative with respect to rotor angle | Tabulated - specify rotor-induced back emf as a function of rotor angle

Parameterization for defining the permanent magnet flux distribution as a function of rotor angle. Choose:

- Perfect trapezoid - specify maximum flux linkage to specify the maximum flux linkage for the permanent magnet and the rotor angle where the back emf is constant. The block assumes a perfect trapezoid for the back emf. This is the default value.
- Perfect trapezoid - specify maximum rotor-induced back emf to specify the maximum rotor-induced back emf and the corresponding rotor speed. The block assumes a perfect trapezoid for the back emf.
- Tabulated - specify flux partial derivative with respect to rotor angle to specify values for the partial derivative of flux linkage and the corresponding rotor angles.
- Tabulated - specify rotor-induced back emf as a function of rotor angle to specify the measured back emf constant and the corresponding rotor speed and angles.

**Maximum permanent magnet flux linkage — Maximum permanent magnet flux linkage**  
0.03 Wb (default)

Peak permanent magnet flux linkage with any of the stator windings.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Perfect trapezoid - specify maximum flux linkage.

**Rotor angle over which back emf is constant — Rotor angle over which back emf is constant**

$\pi / 12$  rad (default)

Rotor angle range over which the permanent magnet flux linking the stator winding is constant. This angle is  $\theta_F$  in the figure that shows the “Trapezoidal Rate of Change of Flux” on page 1-121.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Perfect trapezoid - specify maximum flux linkage or Perfect trapezoid - specify maximum rotor-induced back emf.

**Maximum rotor-induced back emf — Maximum rotor-induced back emf**  
9.6 V (default)

Peak rotor-induced back emf into the stator windings.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Perfect trapezoid - specify maximum rotor-induced back emf.

**Rotor-induced back emf — Rotor-induced back emf**

[0, -9.6, -9.6, 9.6, 9.6, 0] V (default)

Vector of values for the rotor-induced back emf as a function of rotor angle. The first and last values must be the same, and are normally both zero. For more information, see the **Corresponding rotor angles** parameter. First and last values are the same because flux is cyclic with period  $2\pi/N$ , where  $N$  is the number of permanent magnet pole pairs.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Tabulated - specify rotor-induced back emf as a function of rotor angle.

**Flux linkage partial derivative with respect to rotor angle — Flux linkage partial derivative with respect to rotor angle**

[0, -.1528, -.1528, .1528, .1528, 0] Wb/rad (default)

Vector of values for the partial derivative of flux linkage (where flux linkage is flux times number of winding turns) with respect to rotor angle. The first and last values must be the same, and are normally both zero. For more information, see the **Corresponding rotor angles** parameter. First and last values are the same because flux is cyclic with period  $2\pi/N$ , where  $N$  is the number of permanent magnet pole pairs.

**Corresponding rotor angles — Corresponding rotor angles**

[0, 7.5, 22.5, 37.5, 52.5, 60] deg (default)

Vector of rotor angles where the flux linkage partial derivative or rotor-induced back emf is defined. Rotor angle is defined as the angle between the  $a$ -phase magnetic axis and the  $d$ -axis. That is, when the angle is zero, the magnetic fields due to the rotor and the  $a$ -phase winding align. This definition is used regardless of your block setting for rotor angle definition. The first value is zero, and the last value is  $2\pi/N$ , where  $N$  is the number of permanent magnet pole pairs.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Tabulated - specify flux partial derivative with respect to rotor angle or Tabulated - specify rotor-induced back emf as a function of rotor angle.

**Rotor speed used for back emf measurement — Rotor speed used for back emf measurement**

600 rpm (default)

Specify the rotor speed corresponding to the maximum rotor-induced back emf.

**Dependencies**

To enable this parameter, set **Back EMF profile** to Perfect trapezoid - specify maximum rotor-induced back emf or Tabulated - specify rotor-induced back emf as a function of rotor angle.

**Number of pole pairs — Number of pole pairs**

6 (default)

Number of permanent magnet pole pairs on the rotor.

**Rotor angle definition — Reference point for the rotor angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. The default value is Angle between the a-phase magnetic axis and the d-axis. This definition is shown in the “Motor Construction” on page 1-120 figure. When you select this value, the rotor and a-phase fluxes are aligned when the rotor angle is zero.

The other value you can choose for this parameter is Angle between the a-phase magnetic axis and the q-axis. When you select this value, the a-phase current generates maximum torque when the rotor angle is zero.

**Stator****Modeling fidelity — Modeling fidelity**

Constant  $L_d$  and  $L_q$  (default) | Tabulated  $L_d$  and  $L_q$

Select the modeling fidelity:

- Constant  $L_d$  and  $L_q$  —  $L_d$  and  $L_q$  values are constant and defined by their respective parameters.
- Tabulated  $L_d$  and  $L_q$  —  $L_d$  and  $L_q$  values are computed online from DQ currents look-up tables as follows:

$$L_d = f_1(i_d, i_q)$$

$$L_q = f_2(i_d, i_q)$$

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .

**Stator parameterization — Stator parameterization**

Specify  $L_d$ ,  $L_q$ , and  $L_0$  (default) | Specify  $L_s$ ,  $L_m$ , and  $M_s$

Choose Specify  $L_d$ ,  $L_q$ , and  $L_0$  or Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator d-axis inductance,  $L_d$  — Stator d-axis inductance**

0.00022 H (default)

D-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and **Modeling fidelity** to Constant  $L_d$  and  $L_q$ .

**Stator q-axis inductance,  $L_q$  — Stator q-axis inductance**

0.00022 H (default)

Q-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and **Modeling fidelity** to Constant  $L_d$  and  $L_q$ .



**Direct-axis current vector,  $i_D$  — Direct-axis current vector**

[-200, 0, 200] A (default)

Direct-axis current vector,  $i_D$ .**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld and Lq.

**Quadrature-axis current vector,  $i_Q$  — Quadrature-axis current vector**

[-200, 0, 200] A (default)

Quadrature-axis current vector,  $i_Q$ .**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld and Lq.

**Ld matrix,  $L_d(i_d, i_q)$  — Ld matrix**

0.00022 \* ones(3, 3) H (default)

Ld matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld and Lq.

**Lq matrix,  $L_q(i_d, i_q)$  — Lq matrix**

0.00022 \* ones(3, 3) H (default)

Lq matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld and Lq.

**Stator zero-sequence inductance,  $L_0$  — Stator zero-sequence inductance**

0.00016 H (default)

Zero-sequence inductance.

**Dependencies**

To enable this parameter either:

- Set **Winding Type** to Wye-wound, **Zero sequence** to Include, and **Stator parameterization** to Specify Ld, Lq, and L0.
- Set **Winding Type** to Delta-wound and **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator self-inductance per phase,  $L_s$  — Stator self-inductance per phase**

0.0002 H (default)

Average self-inductance of each of the three stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator inductance fluctuation, Lm — Stator inductance fluctuation**

0 H (default)

Fluctuation in self-inductance and mutual inductance of the stator windings with rotor angle.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator mutual inductance, Ms — Stator mutual inductance**

0.00002 H (default)

Average mutual inductance between the stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator resistance per phase, Rs — Stator resistance per phase**

0.013 Ohm (default)

Resistance of each of the stator windings.

**Zero sequence — Zero sequence option**

Include (default) | Exclude

Option to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. To prioritize model fidelity, use this default setting. Using this option:
  - Results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
  - Exposes a zero-sequence parameter in the **Impedances** settings.
- **Exclude** — Exclude zero-sequence terms. To prioritize simulation speed for desktop simulation or real-time deployment, select this option.

**Dependencies**

This parameter is visible only when you set the **Winding Type** parameter to Wye-wound.

**Iron Losses****Model — Enable iron losses computation**

None (default) | Empirical

Specify the iron losses computational model.

**Open-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] — Open-circuit iron losses**

[0, 0, 0] W (default)

Row vector, of length 3, of the open-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

#### Dependencies

To enable this parameter, set **Model** to Empirical.

#### Short-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] — Short-circuit iron losses

[0, 0, 0] W (default)

Row vector, of length 3, of the short-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

#### Dependencies

To enable this parameter, set **Model** to Empirical.

#### Electrical frequency at which losses determined — Electrical frequency at which losses determined

60 Hz (default)

Electrical frequency at which the open-circuit and short-circuit iron losses were measured.

#### Dependencies

To enable this parameter, set **Model** to Empirical.

#### Short-circuit RMS current for short-circuit iron losses — Short-circuit RMS current for short-circuit iron losses

95 A (default)

The resulting short-circuit RMS phase current when measuring the short-circuit losses.

#### Dependencies

To enable this parameter, set **Model** to Empirical.

### Mechanical

#### Rotor inertia — Rotor inertia

0.01 kg\*m<sup>2</sup> (default)

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

#### Rotor damping — Rotor damping

0 N\*m/(rad/s) (default)

Rotary damping.

### Temperature Dependence

These parameters appear only for blocks with exposed thermal ports.

#### Measurement temperature — Measurement temperature

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 1/K (default)

The fractional rate of change of permanent magnet flux density with temperature. It is used to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal Port**

These parameters appear only for blocks with exposed thermal ports.

**Thermal mass for each stator winding – Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Percentage of main flux path iron losses associated with the rotor – Percentage of main flux path iron losses associated with the rotor**

90 (default)

The percentage of the main flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

**Dependencies**

To enable this parameter, set **Model** to **Empirical**.

**Percentage of cross-tooth flux path iron losses associated with the rotor – Percentage of cross-tooth flux path iron losses associated with the rotor**

30 (default)

The percentage of the cross-tooth flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

**Dependencies**

To enable this parameter, set **Model** to **Empirical**.

## Version History

Introduced in R2013b

## Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.
- [3] Mellor, P.H., R. Wrobel, and D. Holliday. "A computationally efficient iron loss model for brushless AC machines that caters for rated flux and field weakened operation." *IEEE Electric Machines and Drives Conference*. May 2009.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Permanent Magnet Synchronous Motor | Hybrid Excitation Synchronous Machine

### Blocks

BLDC Commutation Logic | BLDC Current Controller with PWM Generation | BLDC Current Controller

### Topics

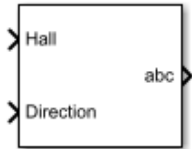
"Expand and Collapse Three-Phase Ports on a Block"

"List of Pre-Parameterized Components"

# BLDC Commutation Logic

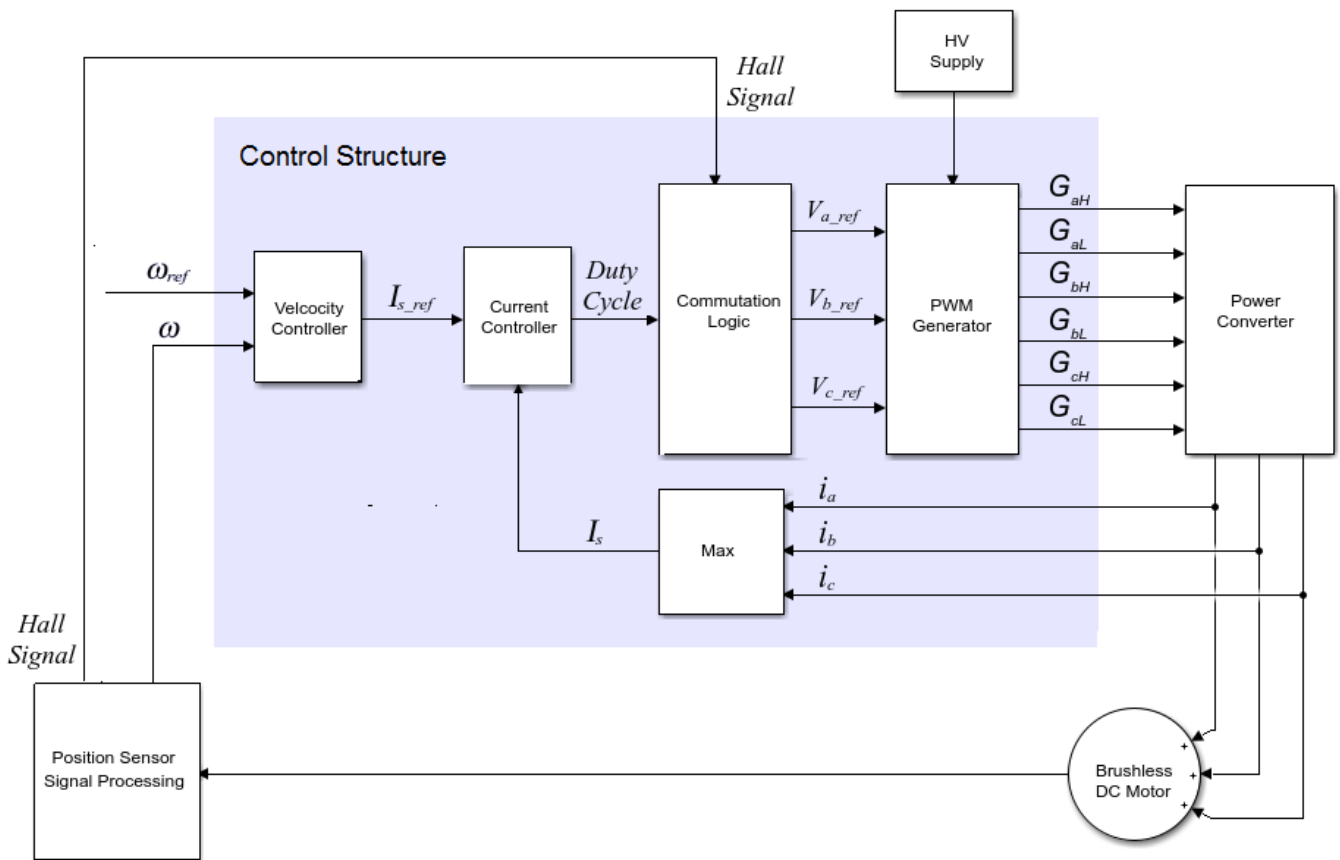
Switch-commutation logic for brushless DC motors

**Library:** Simscape / Electrical / Control / BLDC Control



## Description

The BLDC Commutation Logic block implements a commutation logic for brushless DC motors as part of this control algorithm.



The commutation logic is based on the Hall signals as summarized in this table.

Hall Sensors			Motor Phases		
Hall a	Hall b	Hall c	Phase a	Phase b	Phase c

Hall Sensors			Motor Phases		
0	0	0	0	0	0
1	1	0	0	1	-1
0	1	0	-1	1	0
0	1	1	-1	0	1
0	0	1	0	-1	1
1	0	1	1	-1	0
1	0	0	1	0	-1
1	1	1	0	0	0

## Ports

### Input

#### Hall — Hall sensor

vector

Hall sensor data.

Data Types: single | double

#### Direction — Motor direction

scalar

Direction of motor rotation.

Data Types: single | double

### Output

#### abc — Motor phase

vector

Motor phase indicated by the commutation logic.

Data Types: single | double

## Parameters

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## **Version History**

**Introduced in R2018a**

## **References**

- [1] Stirban, A., I. Boldea, and G. D. Andreescu. "Motion-Sensorless Control of BLDC-PM Motor With Offline FEM-Information-Assisted Position and Speed Observer." *IEEE Transactions on Industry Applications*. 48, no. 6 (2012): 1950-1958.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

BLDC Current Controller | BLDC Current Controller with PWM Generation

### **Simscape Blocks**

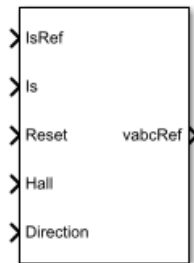
BLDC



# BLDC Current Controller

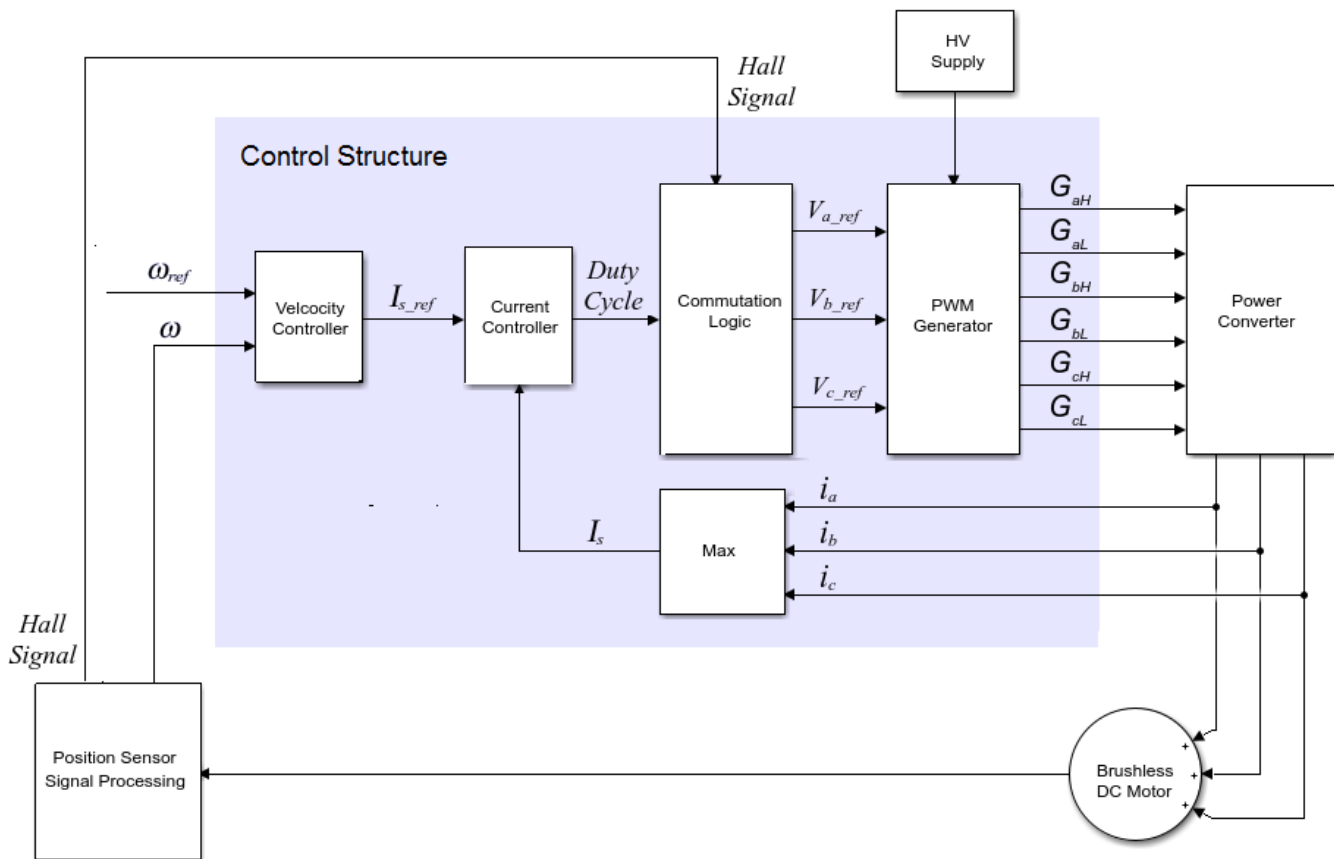
Discrete-time Brushless DC Motor current PI controller

**Library:** Simscape / Electrical / Control / BLDC Control



## Description

The BLDC Current Controller block uses this algorithm to control current in a DC brushless motor.



## Equations

The BLDC Current Controller produces the duty cycle for a BLDC block by implementing proportional-integral (PI) current control using this equation.

$$D = \left( K_p + K_i \frac{T_s z}{z-1} \right) (I_{s\_ref} - I_s)$$

Where:

- $D$  is the duty cycle.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the time period.
- $I_{s\_ref}$  is the reference current.
- $I_s$  is the measured current.
- $G_{zc}$  is the zero cancellation polynomial.

The closed-loop transfer function for the PI control algorithm yields a zero that can be cancelled by using zero-cancellation in the feedforward path. The zero-cancellation transfer function in discrete-time is:

$$G_{ZC}(z) = \frac{\frac{T_s K_i}{K_p}}{z + \left( \frac{T_s - \frac{K_p}{K_i}}{\frac{K_p}{K_i}} \right)}$$

The block obtains control signals for the three phases by multiplying the duty cycle by the commutation signals. The resulting three control signals are normalized over the interval [-1, 1].

## Ports

### Input

#### IsRef — Reference current

scalar

Reference current for control.

Data Types: `single` | `double`

#### Is — Measured current

scalar

Actual current.

Data Types: `single` | `double`

#### Reset — External reset

scalar

External reset signal (rising edge) for the integrator.

Data Types: single | double

### **Hall — Hall sensor**

vector

Hall sensor data.

Data Types: single | double

### **Direction — Motor direction**

scalar

Direction of motor rotation.

Data Types: single | double

### **Output**

#### **vabcRef — Reference voltage**

vector

Reference voltage for the *a*-, *b*-, and *c*-phases.

Data Types: single | double

### **Parameters**

#### **Proportional gain — Controller proportional gain, $K_p$**

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

#### **Integral gain — Integral gain, $K_i$**

5 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.

#### **Anti-windup gain — Anti-windup gain, $K_{aw}$**

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

### **Dependencies**

If you set **Sample time (-1 for inherited)** to -1 and select the **Enable zero cancellation** option, the **Discretization sample time** parameter becomes visible.

**Discretization sample time — Sample time for discretization**

0.001 (default) | positive scalar

Time, in s, between consecutive discretizations. Discretization is required for zero cancellation.

**Dependencies**

This parameter is only visible when both these conditions are met:

- **Sample time** is set to -1.
- **Enable zero cancellation** is selected.

**Enable zero cancellation — Feedforward zero cancellation**

off (default) | on

Option to use zero cancellation on the feedforward path.

**Dependencies**

If you select the **Enable zero cancellation** option and set **Sample time (-1 for inherited)** to -1, the **Discretization sample time** parameter becomes visible.

## Version History

Introduced in R2018a

## References

- [1] Stirban, A., I. Boldea, and G. D. Andreescu. "Motion-Sensorless Control of BLDC-PM Motor With Offline FEM-Information-Assisted Position and Speed Observer." *IEEE Transactions on Industry Applications*. 48, no. 6 (2012): 1950-1958.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

BLDC Commutation Logic | BLDC Current Controller with PWM Generation

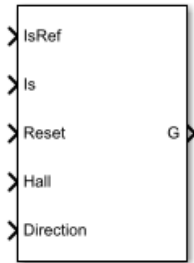
**Simscape Blocks**

BLDC

# BLDC Current Controller with PWM Generation

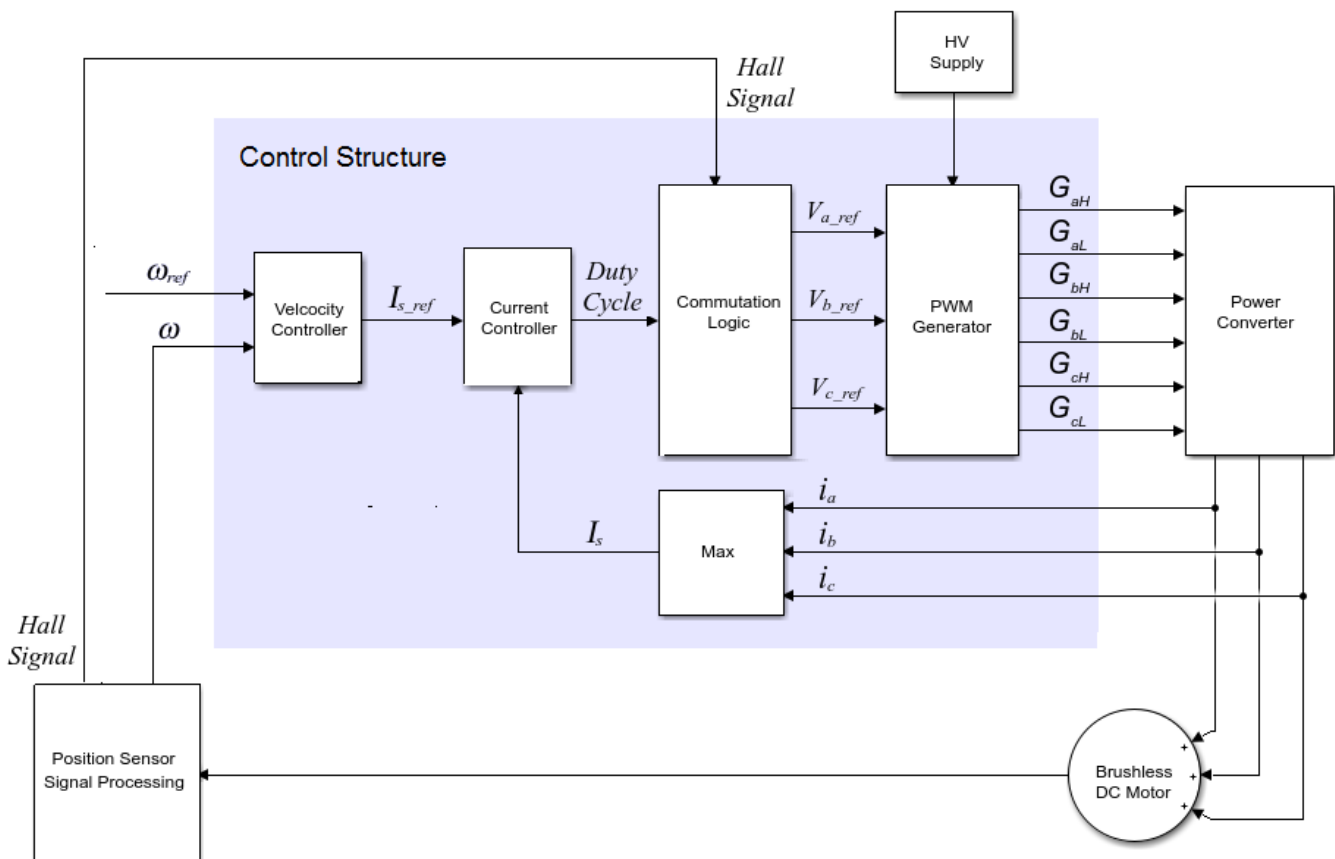
Discrete-time brushless DC Motor current PI controller with pulse width modulation generation

**Library:** Simscape / Electrical / Control / BLDC Control



## Description

The BLDC Current Controller with PWM Generation block generates a pulse width modulation (PWM) signal and controls current in a brushless DC motor. The controller uses this algorithm.



## Equations

The BLDC Current Controller with PWM Generation produces the duty cycle for a BLDC block by implementing proportional-integral (PI) current control using this equation

$$D = \left( K_p + K_i \frac{T_s z}{z-1} \right) (I_{s\_ref} - I_s)$$

where:

- $D$  is the duty cycle.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the time period.
- $I_{s\_ref}$  is the reference current.
- $I_s$  is the measured current.
- $G_{zc}$  is the zero cancellation polynomial.

The closed-loop transfer function for the PI control algorithm yields a zero that can be cancelled by using zero-cancellation block in the feedforward path. The zero-cancellation transfer function in discrete-time is:

$$G_{ZC}(z) = \frac{\frac{T_s K_i}{K_p}}{z + \left( \frac{T_s - \frac{K_p}{K_i}}{\frac{K_p}{K_i}} \right)}$$

The block obtains control signals for the three phases by multiplying the duty cycle by the commutation signals. The resulting three control signals are normalized over the interval  $[-1, 1]$ .

The PWM generator outputs a 1 when the value of the control signal is greater than the carrier counter value. Otherwise, the PWM generator outputs a 0.

## Ports

### Input

#### IsRef — Reference current

scalar

Reference current for control.

Data Types: `single` | `double`

#### Is — Measured current

scalar

Actual current.

Data Types: `single` | `double`

**Reset — External reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: single | double

**Hall — Hall sensor**

vector

Hall sensor data.

Data Types: single | double

**Direction — Motor direction**

scalar

Direction of motor rotation.

Data Types: single | double

**Output****G — Gate pulses**

vector

Pulse waveforms that determine switching behavior in the attached block.

Data Types: single | double

**Parameters****Control Parameters****Proportional gain — Controller proportional gain,  $K_p$** 

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.**Integral gain — Integral gain,  $K_i$** 

5 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.**Anti-windup gain — Anti-windup gain,  $K_{aw}$** 

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.**Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

#### Dependencies

If you set **Sample time (-1 for inherited)** to -1 and select the **Enable zero cancellation** option, the **Discretization sample time** parameter becomes visible.

#### Discretization sample time — Sample time for discretization

0.001 (default) | positive scalar

Time, in s, between consecutive discretizations. Discretization is required for zero cancellation.

#### Dependencies

This parameter is only visible when both these conditions are met:

- **Sample time** is set to -1.
- **Enable zero cancellation** is selected.

#### Enable zero cancellation — Feedforward zero cancellation

off (default) | on

Option to use zero cancellation on the feedforward path.

#### Dependencies

If you select the **Enable zero cancellation** option and set **Sample time (-1 for inherited)** to -1, the **Discretization sample time** parameter becomes visible.

### PWM Generator

#### Carrier counter — Carrier counter model

Up (default) | Down | Up-Down

Use the carrier counter strategy to change the initial behavior of the PWM output:

- Up counter — PWM output begins at the start of the on state.
- Down counter — PWM output begins at the start of the off state.
- Up-down counter — PWM output begins in the middle of the on state.

#### Timer period (s) — PWM timer period

0.001 (default) | positive scalar

Pulse width modulation timer period,  $T_{per}$ , in seconds.

#### Fundamental sample time (s) — Sample time for PWM generation

0.0001 (default) | positive scalar

Time, in s, between consecutive PWM generator executions. During execution, the block produces PWM output and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

To ensure adequate resolution in the generated PWM signal, set the fundamental sample time so that  $0 < T_{s\_pwm} \leq 10T_{per}$ , where:



- $T_{s\_pwm}$  is the **Fundamental sample time (s)**.
- $T_{per}$  is the **Timer period (s)**.

## Version History

Introduced in R2018a

## References

- [1] Stirban, A., I. Boldea, and G. D. Andreescu. "Motion-Sensorless Control of BLDC-PM Motor With Offline FEM-Information-Assisted Position and Speed Observer." *IEEE Transactions on Industry Applications*. 48, no. 6 (2012): 1950-1958.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

BLDC Commutation Logic | BLDC Current Controller

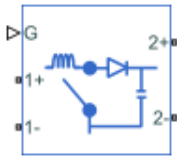
### Simscape Blocks

BLDC

# Boost Converter

Controller-driven DC-DC step-up voltage regulator

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Boost Converter block represents a converter that steps up DC voltage as driven by an attached controller and gate-signal generator. Boost converters are also known as step-up voltage regulators because they increase voltage magnitude.

The Boost Converter block allows you to model a nonsynchronous converter with one switching device or a synchronous converter with two switching devices. Options for the type of switching devices are:

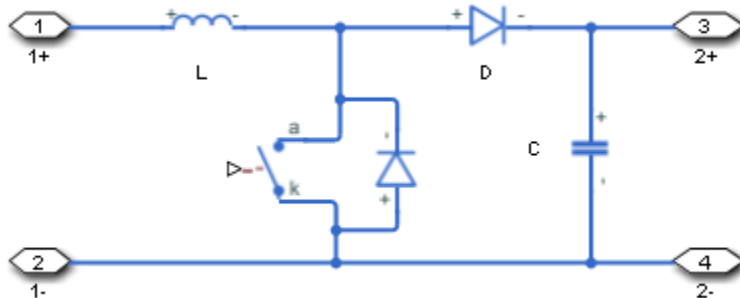
- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

## Converter Topology

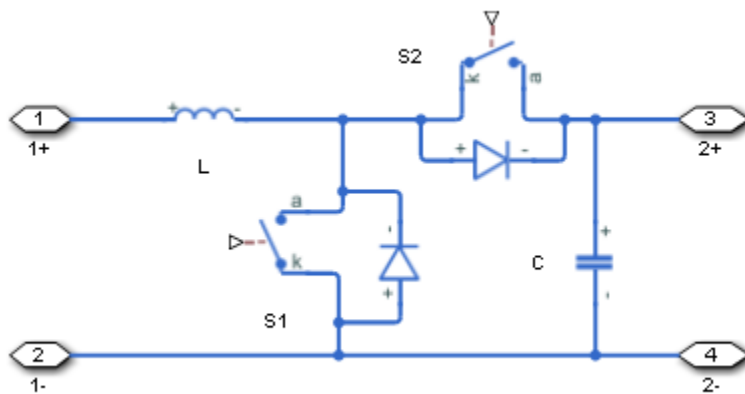
You can model this converter as a nonsynchronous converter with a physical signal gate control port or with two electrical control ports, or as a synchronous converter with an electrical control port. To select the converter topology, set the **Modeling option** parameter to either:

- Nonsynchronous converter — Nonsynchronous converter with optional physical or electrical gate control ports.
- Synchronous converter — Synchronous converter with multiplexed gate signals.

The nonsynchronous boost converter models contain an inductor, a switching device, a diode, and an output capacitor.



The synchronous boost converter model contains an inductor, two switching devices, and an output capacitor.



In each case, the capacitor smoothes the output voltage.

### Protection

For the synchronous converter model, you can include an integral protection diodes. Integral diodes protect the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

To include and configure the internal protection diodes, use the **Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.

Goals		Value to Select	Integral Protection Diode
Do not include protection.		None	None
Include protection.	Prioritize simulation speed.	Diode with no dynamics	The Diode block

Goals		Value to Select	Integral Protection Diode
	Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

### Gate Control

To connect gate-control voltage signals to the gate ports of the switching devices, for the:

- Nonsynchronous converter model:
  - PS control port model:
    - 1 Convert a Simulink gate-control voltage signal to a physical signal using a Simulink-PS Converter block.
    - 2 Connect the Simulink-PS Converter block to the **G** port.
  - Electrical control ports model:
    - 1 Connect a Simscape electrical-domain positive DC voltage signal to the **G+** port.
    - 2 Connect the Simscape electrical-domain negative DC voltage signal to the **G-** port.
- Synchronous converter model:
  - 1 Convert each Simulink gate-control voltage signal to a physical signal using Simulink-PS Converter blocks.
  - 2 Multiplex the converted gate-control signals into a single vector using a Two-Pulse Gate Multiplexer.
  - 3 Connect the vector signal to the **G** port.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to **Averaged switch** and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the

range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

Only a PWM-driven averaged switch converter captures both continuous conduction mode (CCM) and discontinuous conduction mode (DCM). A duty cycle-driven averaged switch converter captures CCM only.

## Ports

### Input

#### G – Switching device gate control

physical signal | scalar

Physical signal port associated with the gate terminals of the switching device.

#### Dependencies

To enable this port, set **Modeling option** to Nonsynchronous converter and **Gate-control port** to PS.

Data Types: double

### Conserving

#### G – Switching device gate control

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

#### Dependencies

To enable this port, set **Modeling option** to Synchronous converter.

Data Types: double

#### G+ – Switching device gate control positive terminal

electrical | scalar

Positive electrical conserving port associated with the positive gate terminal of the switching device.

**Dependencies**

To enable this port, set **Modeling option** to `Nonsynchronous converter` and **Gate-control port** to `Electrical`.

Data Types: `double`

**G- — Switching device gate control negative terminal**

`electrical` | `scalar`

Negative electrical conserving port associated with the negative gate terminal of the switching device.

**Dependencies**

To enable this port, set **Modeling option** to `Nonsynchronous converter` and **Gate-control port** to `Electrical`.

Data Types: `double`

**1+ — Positive DC voltage 1**

`electrical` | `scalar`

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: `double`

**1- — Negative DC voltage 1**

`electrical` | `scalar`

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: `double`

**2+ — Positive DC voltage 2**

`electrical` | `scalar`

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: `double`

**2- — Negative DC voltage 2**

`electrical` | `scalar`

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: `double`

**Parameters****Modeling option — Whether to model nonsynchronous or synchronous converter**

`Nonsynchronous converter` (default) | `Synchronous converter`

Whether to model nonsynchronous or synchronous converter.

## Switching Devices

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

### Gate-control port — Whether to specify physical or electrical control port for the switching gate

PS (default) | Electrical

Whether to specify physical or electrical control port for the switching gate.

#### Dependencies

To enable this port, set **Modeling option** to Nonsynchronous converter.

### Switching device — Switch type

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the converter. For the synchronous model, the switches are identical.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**

- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**On-state resistance — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:



- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate trigger voltage, $V_{gt}$ — Voltage threshold

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate turn-off voltage, $V_{gt\_off}$ — Voltage threshold

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Holding current — Current threshold

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

#### Dependencies

To enable this parameter, set **Switching device** to Averaged Switch.

#### Diode

This table shows how the visibility of **Diode** parameters depends on how you configure the **Modeling option**, **Model dynamics**, and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Diode Parameter Dependencies**

Parameters and Options						
<b>Modeling option</b>						
<i>PS control port or Electrical control ports</i>				<i>Synchronous converter</i>		
<b>Model dynamics</b>				<b>Model dynamics</b>		
<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>			<i>None</i>	<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>
Forward voltage	Forward voltage			Forward voltage	Forward voltage	
On resistance	On resistance			On resistance	On resistance	
Off conductance	Off conductance			Off conductance	Off conductance	
	Junction capacitance				Junction capacitance	
	Peak reverse current, iRM				Peak reverse current, iRM	
	Initial forward current when measuring iRM				Initial forward current when measuring iRM	
	Rate of change of current when measuring iRM				Rate of change of current when measuring iRM	
	<b>Reverse recovery time parameterization</b>				<b>Reverse recovery time parameterization</b>	
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>		<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>
Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr	Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr	

**Model dynamics – Diode model**

Diode with no dynamics (default) | Diode with charge dynamics | None

Diode type. The options are:

- **None** — This option is not available for the nonsynchronous converter.
- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block. This option is the default for the nonsynchronous converter.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you select **Averaged Switch** for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

---

### Dependencies

See the Diode Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### On resistance — Resistance

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

### Off conductance — Conductance

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

### Junction capacitance — Capacitance

50 nF (default) | scalar

Diode junction capacitance.

### Dependencies

See the Diode Parameter Dependencies table.

### Peak reverse current, $i_{RM}$ — Current

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

### Dependencies

See the Diode Parameter Dependencies table.

### Initial forward current when measuring $i_{RM}$ — Current

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

### Dependencies

See the Diode Parameter Dependencies table.

### Rate of change of current when measuring $i_{RM}$ — Current change rate

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time parameterization – Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select `Specify stretch factor` or `Specify reverse recovery charge`, you can specify a value that the block uses to derive the reverse recovery time.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time, trr**. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time, trr – Time**

15 us (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, i<sub>RM</sub>** parameter divided by the value of the **Rate of change of current when measuring i<sub>RM</sub>** parameter.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery charge, Q<sub>rr</sub> – Charge**

1500 s\*uA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, i<sub>RM</sub>**.

- $a$  is the value specified for **Rate of change of current when measuring iRM**.

### Dependencies

See the Diode Parameter Dependencies table.

### LC Parameters

#### Inductance – Inductance

1e-6 H (default) | positive scalar

Inductance.

#### Inductor series resistance – Inductor series resistance

0 Ohm (default) | zero or positive scalar

Series resistance of the inductor.

#### Capacitance – Capacitance

1e-7 F (default) | positive scalar

Capacitance.

#### Capacitor effective series resistance – Capacitor resistance

1e-6 Ohm (default) | zero or positive scalar

Series resistance of the capacitor.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.

#### Snubbers Parameter Dependencies

Snubbers Parameter Dependencies	
Snubber	
None	RC Snubber
	Snubber resistance
	Snubber capacitance

#### Snubber – Snubber model

None (default) | RC snubber

Switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

#### Snubber resistance – Resistance

0.1 (default) | Ohm | scalar

Resistance of the switching device snubber.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber capacitance — Capacitance**

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

**Dependencies**

See the Snubbers Parameter Dependencies table.

## Version History

Introduced in R2018a

## References

- [1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics, 2nd Edition*. Hoboken, NJ: John Wiley & Sons Inc., 2010.
- [2] Han, D. and B. Sarlioglu, "Deadtime Effect on GaN-Based Synchronous Boost Converter and Analytical Model for Optimal Deadtime Selection." *IEEE Transactions on Power Electronics*. Vol. 31, Number 1, 2016, pp 601-612.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear)

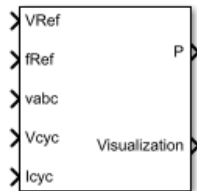
**Topics**

"How the Block Calculates TM and Tau" on page 1-452

## Bridge Cycloconverter Voltage Controller (Three-Phase)

RMS Voltage PI control for three-phase bridge cycloconverters

**Library:** Simscape / Electrical / Control / Converter Control



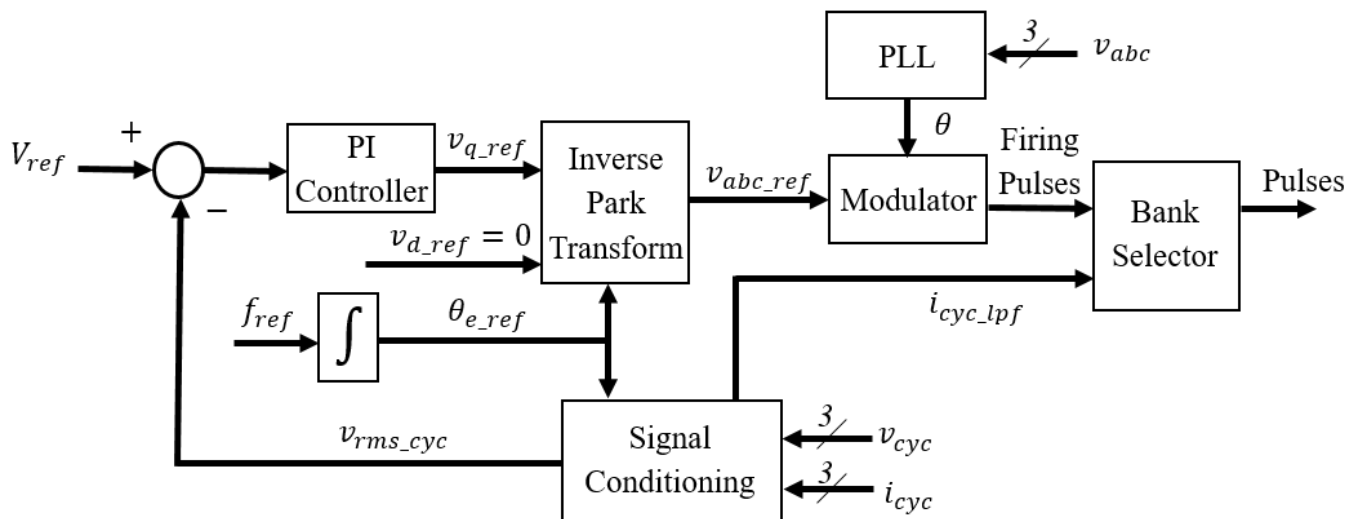
### Description

The Bridge Cycloconverter Voltage Controller (Three-Phase) block implements a PI-based root-mean-square (RMS) voltage controller for three-phase bridge cycloconverters.

To convert a three-phase signal directly from a higher frequency to a lower frequency, use this block with a three-phase bridge cycloconverter. Refer to "Three-Phase Bridge Cycloconverter" for an example of such a conversion.

### Operating Principle

The controller regulates the cycloconverter line-to-neutral RMS voltage to a given value and a given electrical frequency. The structure of the cycloconverter controller is illustrated in this diagram.



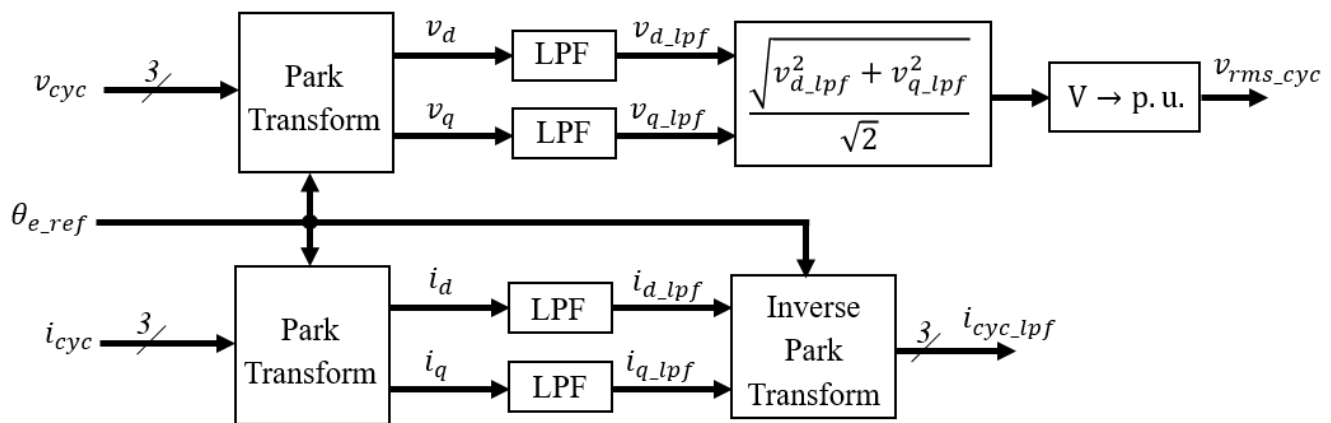
In the diagram:

- The controller integrates the desired output frequency  $f_{ref}$  to produce the reference electrical angle  $\theta_{e\_ref}$ .

- The Signal Conditioning block filters the cycloconverter line-to-neutral voltage  $v_{cyc}$  and current  $i_{cyc}$  to produce the per-unit RMS voltage  $v_{rms\_cyc}$  and smoothed current signal  $i_{cyc\_lpf}$ .
- The PI Controller generates a reference phase voltage in the  $q$ -axis from the error between the desired output RMS voltage  $V_{ref}$  and  $v_{rms\_cyc}$ .
- The Inverse Park Transform block converts the reference phase voltage in  $dq0$ -coordinates to a phase voltage  $v_{abc\_ref}$  in  $abc$ -coordinates.
- The Sinusoidal Power Measurement (PLL, Three-Phase) block estimates the phase angle  $\theta$  of the input voltage signal  $v_{abc}$ .

The Modulator and Bank Selector blocks create the 36 pulses to drive the cycloconverter using the reference phase voltage  $v_{abc\_ref}$ , estimated phase angle  $\theta$ , and filtered cycloconverter current  $i_{cyc\_lpf}$ . To generate the firing angles, the controller uses the cosine wave crossing method.

This diagram shows the signal conditioning logic.



In the diagram:

- The Park Transform blocks convert the measured cycloconverter voltage  $v_{cyc}$  and current  $i_{cyc}$  into  $d$ - and  $q$ -axis components ( $v_d, v_q, i_d, i_q$ ) using the reference electrical angle  $\theta_{e\_ref}$ .
- The Low-Pass Filter (LPF) blocks remove the high-frequency noise from each of the  $d$ - and  $q$ -axis voltage and currents to produce the filtered components ( $v_{d\_lpf}, v_{q\_lpf}, i_{d\_lpf}, i_{q\_lpf}$ ).
- The block calculates the cycloconverter per-unit RMS voltage  $v_{rms\_cyc}$  by taking the squared sum of the  $dq$  components, dividing by  $\sqrt{2}$ , and finally converting from SI to per-unit representation.
- The Inverse Park Transform converts the  $dq$  filtered current back to the  $abc$ -axis and outputs it as  $i_{cyc\_lpf}$ .

The cycloconverter reference line-to-neutral rms voltage output is given in per-unit representation.

### Visualization

The block outputs a bus containing six signals for visualization:

- The estimated phase angle  $\theta$  of the input voltage signal  $v_{abc}$
- The desired RMS voltage  $V_{ref}$  of the output signal



- The reference phase voltages  $v_{abc\_ref}$  of the desired output signal
- The filtered line-to-neutral cycloconverter RMS voltage  $v_{rms\_cyc}$
- The filtered cycloconverter phase currents  $i_{cyc\_lpf}$
- The filtered cycloconverter phase voltages  $v_{cyc\_lpf}$

## Ports

### Input

#### **VRef — Reference voltage**

scalar

Reference line-to-neutral RMS voltage, expressed in per-unit representation.

Data Types: `single` | `double`

#### **fRef — Reference frequency**

scalar

Reference electrical frequency, in Hz.

Data Types: `single` | `double`

#### **vabc — Phase voltages**

vector

Measured phase voltages of the source, in V.

Data Types: `single` | `double`

#### **Vcyc — Cycloconverter voltages**

vector

Measured cycloconverter phase voltages, in V.

Data Types: `single` | `double`

#### **Icyc — Cycloconverter currents**

vector

Measured cycloconverter phase currents, in A.

Data Types: `single` | `double`

### Output

#### **P — Pulses**

vector

Thyristor pulse vector to control a three-phase bridge cycloconverter.

Data Types: `single` | `double`

#### **Visualization — Visualization bus**

bus

Bus containing internal signals for visualization. For a full list of signals, refer to the “Visualization” on page 1-162 section.

Data Types: single | double

## Parameters

### **Rated voltage (phase-to-phase RMS) — Rated RMS voltage**

6000 (default) | positive number

Rated RMS voltage for per-unit conversion calculations, in V.

### **Loop filter proportional gain — LF proportional gain**

2 (default) | positive number

Loop filter proportional gain for the phase-locked loop (PLL) estimating the phase of the input signal. This value determines the aggressiveness of the PLL in tracking and locking to the phase angle. Increase this value to improve reaction time of the tracking to step changes in the phase angle.

### **Loop filter integral gain — LF integral gain**

20 (default) | positive number

Loop filter integral gain for the phase-locked loop (PLL) estimating the phase of the input signal. Increase this value to increase the rate at which steady-state error is eliminated in the phase angle. This value also determines the aggressiveness of the PLL in tracking and locking to the phase.

### **Filters time constant (s) — Time constant**

1e-2 (default) | positive number

Time constant of the low-pass filters in the Signal Conditioning block of the controller. These filters reduce undesired high-frequency noise in the cycloconverter phase voltage and current measurements.

### **Controller proportional gain — Proportional gain**

1 (default) | positive number

Proportional gain for the PI-controller that generates the reference phase voltage for the cycloconverter. Increase this value to increase the aggressiveness of the controller.

### **Controller integral gain — Integral gain**

12 (default) | positive number

Integral gain of the PI-controller that generates the reference phase voltage for the cycloconverter. Increase this value to increase the rate at which steady-state error is eliminated in the phase voltage signal.

### **Controller anti-windup gain — Anti-windup gain**

10 (default) | positive number

Anti-windup gain of the PI-controller that generates the reference phase voltage for the cycloconverter.

### **Thyristor pulse width (rad) — Pulse width**

5\*pi/6 (default) | positive number

Angular width of pulses sent to the cycloconverter.

**Bank selector current threshold (A) – Current threshold**

5 (default) | positive number

Current threshold for switching between positive and negative converters.

**Pulse ordering – Pulse ordering rule**

Sequential device order (default) | Natural order of commutation

Strategy used for the ordering of generated pulses.

**Sample time (-1 for inherited) – Sample time**

-1 (default) | positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, set the sample time explicitly.

## Version History

Introduced in R2017b

## References

[1] Chen, H., M. H. Johnson, and D. C. Aliprantis. "Low-frequency AC transmission for offshore wind power." *IEEE Transactions on Power Delivery*. Vol. 28, Number 4, 2013, pp. 2236-2244.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

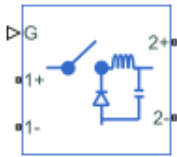
**Blocks**

Sinusoidal Power Measurement (PLL, Three-Phase) | Park Transform | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

# Buck Converter

Controller-driven DC-DC step-down voltage regulator

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Buck Converter block represents a converter that steps down DC voltage as driven by an attached controller and gate-signal generator. Buck converters are also known as step-down voltage regulators because they decrease voltage magnitude.

The Buck Converter block allows you to model a nonsynchronous converter with one switching device or a synchronous converter with two switching devices. Options for the type of switching devices are:

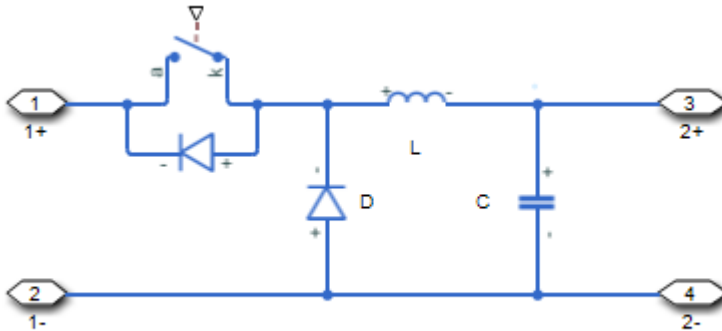
- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

## Converter Topology

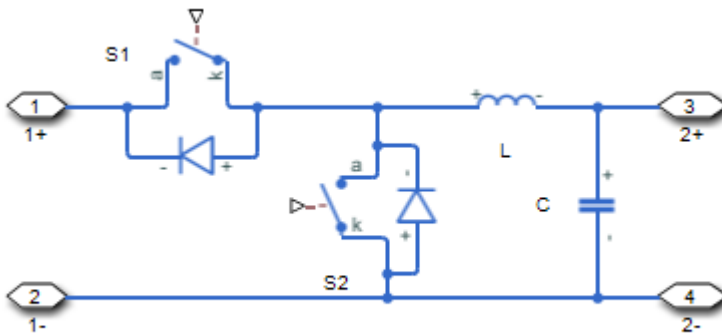
You can model this converter as a nonsynchronous converter with a physical signal gate control port or with two electrical control ports, or as a synchronous converter with an electrical control port. To select the converter topology, set the **Modeling option** parameter to either:

- Nonsynchronous converter — Nonsynchronous converter with optional physical or electrical gate control ports.
- Synchronous converter — Synchronous converter with multiplexed gate signals.

The nonsynchronous buck converter models contain a switching device, a diode, an inductor, and an output capacitor.



The synchronous buck converter model contains two switching devices, an inductor, and an output capacitor.



In each case, the capacitor smoothes the output voltage.

### Protection

For the synchronous converter model, you can include an integral protection diode for the S2 switching device. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

To include and configure the internal protection diode block, use the **Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.

Goals		Value to Select	Integral Protection Diode
Do not include protection.		None	None
Include protection.	Prioritize simulation speed.	Diode with no dynamics	The Diode block

Goals		Value to Select	Integral Protection Diode
	Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

### Gate Control

To connect gate-control voltage signals to the gate ports of the switching devices, for the:

- Nonsynchronous converter model:
  - PS control port model:
    - 1 Convert a Simulink gate-control voltage signal to a physical signal using a Simulink-PS Converter block.
    - 2 Connect the Simulink-PS Converter block to the **G** port.
  - Electrical control ports model:
    - 1 Connect a Simscape electrical-domain positive DC voltage signal to the **G+** port.
    - 2 Connect the Simscape electrical-domain negative DC voltage signal to the **G-** port.
- Synchronous converter model:
  - 1 Convert each Simulink gate-control voltage signal to a physical signal using Simulink-PS Converter blocks.
  - 2 Multiplex the converted gate-control signals into a single vector using a Two-Pulse Gate Multiplexer.
  - 3 Connect the vector signal to the **G** port.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the

range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

Only a PWM-driven averaged switch converter captures both continuous conduction mode (CCM) and discontinuous conduction mode (DCM). A duty cycle-driven averaged switch converter captures CCM only.

## Ports

### Input

#### G – Switching device gate control

physical signal | scalar

Physical signal port associated with the gate terminals of the switching device.

#### Dependencies

To enable this port, set **Modeling option** to Nonsynchronous converter and **Gate-control port** to PS.

Data Types: double

### Conserving

#### G – Switching device gate control

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

#### Dependencies

To enable this port, set **Modeling option** to Synchronous converter.

Data Types: double

#### G+ – Switching device gate control positive terminal

electrical | scalar

Positive electrical conserving port associated with the positive gate terminal of the switching device.

**Dependencies**

To enable this port, set **Modeling option** to Nonsynchronous converter and **Gate-control port** to Electrical.

Data Types: double

**G- — Switching device gate control negative terminal**

electrical | scalar

Negative electrical conserving port associated with the negative gate terminal of the switching device.

**Dependencies**

To enable this port, set **Modeling option** to Nonsynchronous converter and **Gate-control port** to Electrical.

Data Types: double

**1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

**1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

**2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

**2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

**Parameters****Modeling option — Whether to model nonsynchronous or synchronous converter**

Nonsynchronous converter (default) | Synchronous converter

Whether to model nonsynchronous or synchronous converter.



## Switching Devices

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

### Gate-control port — Whether to specify physical or electrical control port for the switching gate

PS (default) | Electrical

Whether to specify physical or electrical control port for the switching gate.

#### Dependencies

To enable this port, set **Modeling option** to Nonsynchronous converter.

### Switching device — Switch type

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the converter. For the synchronous model, the switches are identical.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**

- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**On-state resistance — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged Switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate trigger voltage, $V_{gt}$ — Voltage threshold

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate turn-off voltage, $V_{gt\_off}$ — Voltage threshold

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Holding current — Current threshold

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

#### Dependencies

To enable this parameter, set **Switching device** to Averaged Switch.

#### Diode

This table shows how the visibility of **Diode** parameters depends on how you configure the **Modeling option**, **Model dynamics**, and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Diode Parameter Dependencies**

Parameters and Options							
<b>Modeling option</b>							
<i>PS control port or Electrical control ports</i>				<i>Synchronous converter</i>			
<b>Model dynamics</b>				<b>Model dynamics</b>			
<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>			<i>None</i>	<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>	
Forward voltage	Forward voltage			Forward voltage	Forward voltage		
On resistance	On resistance			On resistance	On resistance		
Off conductance	Off conductance			Off conductance	Off conductance		
	Junction capacitance				Junction capacitance		
	Peak reverse current, iRM				Peak reverse current, iRM		
	Initial forward current when measuring iRM				Initial forward current when measuring iRM		
	Rate of change of current when measuring iRM				Rate of change of current when measuring iRM		
	<b>Reverse recovery time parameterization</b>				<b>Reverse recovery time parameterization</b>		
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>		<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>
	Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr		Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr

**Model dynamics – Diode model**

Protection diode with no dynamics (default) | Protection diode with charge dynamics | None

Diode type. The options are:

- None — This option is not available for the nonsynchronous converter.
- Diode with no dynamics — Select this option to prioritize simulation speed using the Diode block. This option is the default for the nonsynchronous converter.

- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

---

### Dependencies

See the Diode Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### On resistance — Resistance

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

### Off conductance — Conductance

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

### Junction capacitance — Capacitance

50 nF (default) | scalar

Diode junction capacitance.

### Dependencies

See the Diode Parameter Dependencies table.

### Peak reverse current, $i_{RM}$ — Current

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

### Dependencies

See the Diode Parameter Dependencies table.

### Initial forward current when measuring $i_{RM}$ — Current

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

### Dependencies

See the Diode Parameter Dependencies table.

### Rate of change of current when measuring $i_{RM}$ — Current change rate

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time parameterization – Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select **Specify stretch factor** or **Specify reverse recovery charge**, you can specify a value that the block uses to derive the reverse recovery time.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time, trr**. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time, trr – Time**

15 us (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, iRM** parameter divided by the value of the **Rate of change of current when measuring iRM** parameter.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery charge, Qrr – Charge**

1500 s\*uA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

### Dependencies

See the Diode Parameter Dependencies table.

### LC Filter

#### Inductance – Filter inductance

1e-6 H (default) | positive scalar

Inductance of the LC filter.

#### Inductor series resistance – Inductor series resistance

0 Ohm (default) | zero or positive scalar

Series resistance of the inductor.

#### Capacitance – Filter capacitance

1e-7 F (default) | positive scalar

Capacitance of the LC filter.

#### Capacitor effective series resistance – Capacitor resistance

1e-6 Ohm (default) | zero or positive scalar

Series resistance of the capacitor.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.

#### Snubbers Parameter Dependencies

Snubbers Parameter Dependencies	
Snubber	
None	RC Snubber
	Snubber resistance
	Snubber capacitance

#### Snubber – Snubber model

None (default) | RC snubber

Switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

#### Snubber resistance – Resistance

0.1 (default) | Ohm | scalar

Resistance of the switching device snubber.

**Dependencies**

See the Snubbers Parameter Dependencies table.

**Snubber capacitance — Capacitance**

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

**Dependencies**

See the Snubbers Parameter Dependencies table.

## Version History

Introduced in R2018a

## References

- [1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics, 2nd Edition*. Hoboken, NJ: John Wiley & Sons Inc., 2010.
- [2] Hedayati, M. H., P. Bharadwaj, and V. John. "Hybrid synchronous DC-DC buck power converter using Si and GaN transistors." *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. Trivandrum, India: 2016, pp 1-6.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck-Boost Converter | Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear)

**Topics**

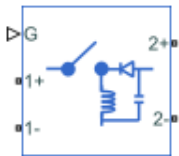
"How the Block Calculates TM and Tau" on page 1-452



# Buck-Boost Converter

Controller-driven DC-DC inverting or four-switch step-up or step-down voltage regulator

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Buck-Boost Converter block represents a DC-DC converter that can either step up or step down DC voltage from one side of the converter to the other as driven by an attached controller and gate-signal generator. Buck-boost converters are also known as step-up/step-down voltage regulators because they can increase or decrease voltage magnitude.

The block can also invert voltage so that the polarity of the output voltage is the opposite of the polarity of the input voltage. The magnitude of the output voltage depends on the duty cycle.

The Buck-Boost Converter block allows you to model an inverting buck-boost converter with one switching device or a buck-boost converter with four switching devices. Options for the type of switching devices are:

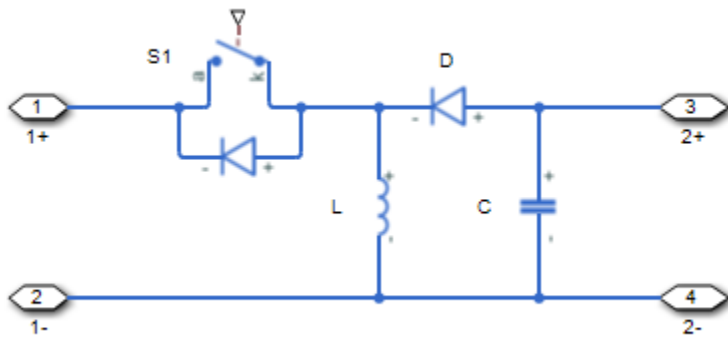
- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

## Converter Topology

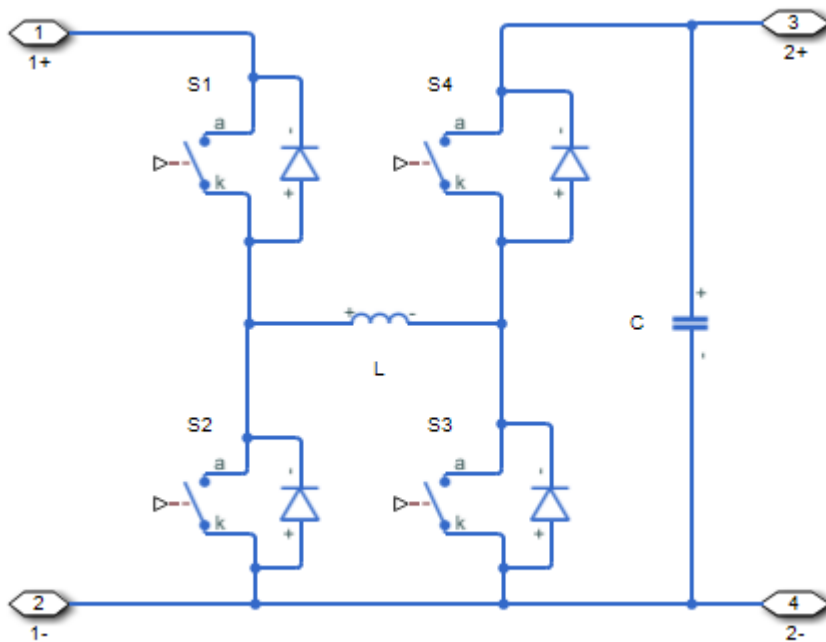
You can model this converter as an inverting buck-boost converter with a physical signal gate control port or with two electrical control ports, or as a four-switch buck-boost converter with an electrical control port. To select the converter topology, set the **Modeling option** parameter to either:

- Inverting converter — Inverting buck-boost converter with physical or electrical gate control port.
- Four-switch converter — Four-switch buck-boost converter with multiplexed gate signals.

The inverting converter models contain a switching device, a diode, an inductor, and an output capacitor.



The four-switch converter model contains four switching devices, an inductor, and an output capacitor.



In each case, the capacitor smoothes the output voltage.

## Protection

You can include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

## Gate Control

To connect gate-control voltage signals to the gate ports of the switching devices, for the:

- Inverting converter model:
  - PS control port model:
    - 1 Convert a Simulink gate-control voltage signal to a physical signal using a Simulink-PS Converter block.
    - 2 Connect the Simulink-PS Converter block to the **G** port.
  - Electrical control ports model:
    - 1 Connect a Simscape electrical-domain positive DC voltage signal to the **G+** port.
    - 2 Connect the Simscape electrical-domain negative DC voltage signal to the **G-** port.
- Synchronous converter model:
  - 1 Convert each Simulink gate-control voltage signal to a physical signal using Simulink-PS Converter blocks.
  - 2 Multiplex the converted gate-control signals into a single vector using a Four-Pulse Gate Multiplexer
  - 3 Connect the vector signal to the **G** port.

## Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

Only a PWM-driven averaged switch converter captures both continuous conduction mode (CCM) and discontinuous conduction mode (DCM). A duty cycle-driven averaged switch converter captures CCM only.

## Ports

### Input

#### **G – Switching device gate control**

physical signal | scalar

Physical signal port associated with the gate terminals of the switching device.

#### **Dependencies**

To enable this port, set **Modeling option** to Inverting converter and **Gate-control port** to PS.

Data Types: double

### Conserving

#### **G – Switching device gate control**

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

#### **Dependencies**

To enable this port, set **Modeling option** to Four-switch converter.

Data Types: double

#### **G+ – Switching device gate control positive terminal**

electrical | scalar

Positive electrical conserving port associated with the positive gate terminal of the switching device.

#### **Dependencies**

To enable this port, set **Modeling option** to Inverting converter and **Gate-control port** to Electrical.

Data Types: double

**G- — Switching device gate control negative terminal**

electrical | scalar

Negative electrical conserving port associated with the negative gate terminal of the switching device.

**Dependencies**

To enable this port, set **Modeling option** to Inverting converter and **Gate-control port** to Electrical.

Data Types: double

**1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

**1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

**2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

**2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

**Parameters****Modeling option — Whether to model inverting or four-switch buck-boost converter**

Inverting converter (default) | Four-switch converter

Whether to model an inverting or a four-switch buck-boost converter.

**Switching Devices**

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

## Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	Integer for piecewise constant approximation of gate input (0 for disabled)
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

### Gate-control port — Whether to specify physical or electrical control port for the switching gate

PS (default) | Electrical

Whether to specify physical or electrical control port for the switching gate.

#### Dependencies

To enable this port, set **Modeling option** to Inverting converter.

### Switching device — Switch type

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the converter. For the four-switch model, the switches are identical.

model.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**On-state resistance — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage

- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate trigger voltage, Vgt — Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off — Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Holding current — Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Diode**

This table shows how the visibility of **Diode** parameters depends on how you configure the **Model dynamics** and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.



## Diode Parameter Dependencies

Parameters and Options			
<b>Model dynamics</b>			
<i>Diode with no dynamics</i>	<i>Diode with charge dynamics</i>		
Forward voltage	Forward voltage		
On resistance	On resistance		
Off conductance	Off conductance		
	Junction capacitance		
	Peak reverse current, iRM		
	Initial forward current when measuring iRM		
	Rate of change of current when measuring iRM		
	<b>Reverse recovery time parameterization</b>		
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>
Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr	

### Model dynamics — Diode model

*Diode with no dynamics (default) | Diode with charge dynamics*

Diode type. The options are:

- *Diode with no dynamics* — Select this option to prioritize simulation speed using the Diode block.
- *Diode with charge dynamics* — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and *Diode with no dynamics* is automatically selected.

---

### Dependencies

See the Diode Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### On resistance — Resistance

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

**Off conductance – Conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Junction capacitance – Capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

See the Diode Parameter Dependencies table.

**Peak reverse current,  $i_{RM}$  – Current**

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

**Dependencies**

See the Diode Parameter Dependencies table.

**Initial forward current when measuring  $i_{RM}$  – Current**

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

See the Diode Parameter Dependencies table.

**Rate of change of current when measuring  $i_{RM}$  – Current change rate**

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time parameterization – Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select `Specify stretch factor` or `Specify reverse recovery charge`, you can specify a value that the block uses to derive the reverse recovery time.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge.

The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

See the Diode Parameter Dependencies table.

#### Reverse recovery time, $t_{rr}$ – Time

15  $\mu$ s (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

See the Diode Parameter Dependencies table.

#### Reverse recovery charge, $Q_{rr}$ – Charge

1500  $\text{s} \cdot \mu\text{A}$  (default) | scalar

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

#### Dependencies

See the Diode Parameter Dependencies table.

### LC Parameters

#### Inductance – Inductance

1e-6 H (default) | positive scalar

Inductance.

#### Inductor series resistance – Inductor series resistance

0  $\Omega$  (default) | zero or positive scalar

Series resistance of the inductor.

#### Capacitance – Capacitance

1e-7 F (default) | positive scalar

Capacitance.

### Capacitor effective series resistance – Capacitor resistance

1e-6 Ohm (default) | zero or positive scalar

Series resistance of the capacitor.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Snubbers Parameter Dependencies

Snubbers Parameter Dependencies	
Snubber	
None	RC Snubber
	Snubber resistance
	Snubber capacitance

### Snubber – Snubber model

None (default) | RC snubber

Switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber resistance – Resistance

0.1 (default) | Ohm | scalar

Resistance of the switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber capacitance – Capacitance

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

## Version History

Introduced in R2018a

## References

- [1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics, 2nd Edition*. Hoboken, NJ: John Wiley & Sons Inc., 2010.
- [2] Xiaoyong, R., Z. Tang, X. Ruan, J. Wei and G. Hua. *Four Switch Buck-Boost Converter for Telecom DC-DC power supply applications*. Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition. Austin, TX: 2008, pp 1527-1530.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Boost Converter | Converter | GTO | IGBT (Ideal Switching) | MOSFET (Ideal Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear)

### Topics

“How the Block Calculates TM and Tau” on page 1-452

# Busbar

Load flow analysis busbar connector

**Library:** Simscape / Electrical / Connectors & References



## Description

The Busbar block is a connector for load-flow analysis. Use this block with three-phase Simscape Electrical networks that are compatible with frequency-time simulation mode. After completing simulation, the block adds an annotation showing the voltage, phase, active power, and reactive power at each of its three-phase electrical conserving ports at the time the simulation started. For the annotation to be added, Simscape logging must be enabled.

The block will simulate in time-based simulation mode, however it will not add the annotation with initial load flow results unless the network is compatible with frequency-time simulation mode and, on the Solver Configuration block, you check `Start simulation from steady state`. However, when using time-based simulation mode, the block calculates approximate values for voltage, phase, active and reactive powers. You can view these values from Simscape logging or by exposing the optional measurement ports on the block. The level of approximation depends on how balanced the system is and on the level of harmonics present.

## Ports

### Output

#### Vt — Voltage

physical signal

Physical signal port associated with the voltage measurement. The value is reported in per-unit, i.e. phase-to-phase root mean square voltage divided by the **Rated busbar voltage (phase-to-phase RMS)**.

#### Dependencies

This port is visible only when, in the **Parameters** settings, the **Measurements ports** parameter is set to Yes.

#### ph — Phase

physical signal

Physical signal port associated with the voltage phase measurement.

#### Dependencies

This port is visible only when, in the **Parameters** settings, the **Measurements ports** parameter is set to Yes.

#### P — Active power

physical signal

Physical signal port associated with the active (real) power measurement.

**Dependencies**

This port is visible only when, in the **Parameters** settings, the **Number of connections** parameter is set to 2, 3, or 4 and the **Measurements ports** parameter is set to Yes.

**Q — Reactive power**

physical signal

Physical signal port associated with the reactive (imaginary) power measurement.

**Dependencies**

This port is visible only when, in the **Parameters** settings, the **Number of connections** parameter is set to 2, 3, or 4 and the **Measurements ports** parameter is set to Yes.

**Conserving**

**~1 — Three-phase connection 1**

electrical

Composite three-phase electrical conserving port associated with the first three-phase connection.

**~2 — Three-phase connection 2**

electrical

Composite three-phase electrical conserving port associated with the second three-phase connection.

**Dependencies**

This port is visible only when, in the **Parameters** settings, the **Number of connections** parameter is set to 2, 3, or 4.

**~3 — Three-phase connection 3**

electrical

Composite three-phase electrical conserving port associated with the third three-phase connection..

**Dependencies**

This port is visible only when, in the **Parameters** settings, the **Number of connections** parameter is set to 3 or 4.

**~4 — Three-phase connection 4**

electrical

Composite three-phase electrical conserving port associated with the fourth three-phase connection.

**Dependencies**

This port is visible only when, in the **Parameters** settings, the **Number of connections** parameter is set to 4.

## Parameters

### Number of connections — Number of port connections

2 (default) | 1 | 3 | 4

Number of port connections that are exposed on the block.

The number of exposed ports on the block correlates to this parameter.

### Rated busbar voltage (phase-to-phase RMS) — Busbar rated voltage

$\sqrt{3} * 100 / \sqrt{2}$  V (default)

Rated phase-to-phase root mean square voltage of the busbar.

### Network frequency — Network frequency

60 Hz (default)

Frequency of the connected network.

### Measurement ports — Measurement port exposure

No (default) | Yes

Visibility for the voltage, phase, active power, and reactive power physical signal output measurement ports.

### Dependencies

Ports **Vt** and **ph** are visible only when this parameter is set to Yes.

Ports **P** and **Q** are visible only when this parameter is set to Yes and, in the **Parameters** settings, the **Number of connections** parameter is set to 2, 3, or 4.

## Version History

Introduced in R2019b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Load Flow Source | **Load-Flow Analysis**

## Topics

“Perform a Load-Flow Analysis Using Simscape Electrical”



# Busbar (DC)

Load flow analysis DC busbar connector

**Library:** Simscape / Electrical / Connectors & References



## Description

The Busbar (DC) block is a DC connector for load-flow analysis. You can use this block with Simscape Electrical DC networks that are configured to run in time or frequency-time simulation mode. You can view the voltage and power values by enabling simulation data logging or by exposing the optional measurement ports on the block. For more information about data logging, see “About Simulation Data Logging”.

## Ports

### Output

#### Vt — Voltage, V

physical signal

Physical signal port associated with the per-unit voltage measurement, in Volts.

#### Dependencies

To enable this port, in the **Parameters** settings, set the **Measurements ports** parameter to Yes.

#### P — Power, W

physical signal

Physical signal port associated with the power measurement, in watts.

#### Dependencies

To enable this port, in the **Parameters** settings, set the **Measurements ports** parameter to Yes and the **Number of connections** parameter to 2, 3, or 4.

### Conserving

#### 1+ — Positive connection 1

electrical

Electrical conserving port associated with the first positive connection.

#### 2+ — Positive connection 2

electrical

Electrical conserving port associated with the second positive connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 2, 3, or 4.

#### **3+ — Positive connection 3**

electrical

Electrical conserving port associated with the third positive connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 3 or 4.

#### **4+ — Positive connection 4**

electrical

Electrical conserving port associated with the fourth positive connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 4.

#### **1- — Negative connection 1**

electrical

Electrical conserving port associated with the first negative connection.

#### **2- — Negative connection 2**

electrical

Electrical conserving port associated with the second negative connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 2, 3, or 4.

#### **3- — Negative connection 3**

electrical

Electrical conserving port associated with the third negative connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 3 or 4.

#### **4- — Negative connection 4**

electrical

Electrical conserving port associated with the fourth negative connection.

### **Dependencies**

To enable this port, in the **Parameters** settings, set the **Number of connections** parameter to 4.

## Parameters

### Number of connections — Number of port connections

2 (default) | 1 | 3 | 4

Number of port connections to expose on the block.

### Rated DC voltage — Busbar rated DC voltage

1e3 V (default) | positive scalar

Rated DC voltage of the busbar. This block uses this value to scale the per-unit voltage value.

### Measurement type — Measurement type

Instantaneous (default) | Average over a specified period

Type of measurement. In time simulation mode, you can set this parameter to either **Instantaneous** or **Average over a specified period**. In frequency-time simulation mode, you must set this parameter to **Instantaneous**.

### Averaging period — Averaging period of measurement

1e-2 s (default) | positive scalar

Period over which the voltage and power values are averaged.

### Dependencies

To enable this parameter, set **Measurement type** to **Average over a specified period**.

### Measurement ports — Measurement port exposure

No (default) | Yes

Whether to expose the **Vt** and **P** physical signal output measurement ports.

## Version History

Introduced in R2022a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Busbar | Load Flow Source | **Load-Flow Analysis**

### Topics

“Mixed AC/DC System Loadflow”

# Capacitor

Linear or nonlinear capacitor with optional tolerance, operational limits and fault behavior

**Library:** Simscape / Electrical / Passive



## Description

The Capacitor block lets you model linear, nonlinear (table-based), and frequency-dependent capacitors, including polar capacitors. Optionally, you can also model the following effects:

- “Tolerances” on page 1-199
- “Operating Limits” on page 1-200
- “Faults” on page 1-200

You can turn these modeling options on and off independently of each other. When all the additional options are turned off, the component behavior is identical to the Simscape Foundation library Capacitor block.

In its simplest form, the Capacitor block models a linear capacitor, described with the following equation:

$$I = C \frac{dV}{dt}$$

where:

- $I$  is the current.
- $C$  is the capacitance.
- $V$  is the voltage.
- $t$  is the time.

To model a nonlinear or polar capacitor, set the **Capacitance model** parameter to `Lookup table` and provide a lookup table of capacitance-voltage values:

- For polar capacitors, where this lookup table is asymmetric with respect to the applied terminal voltage, set the **Symmetric C-V table** parameter to `No - use C-V data as-is`.
- For other types of nonlinear capacitor, ensure symmetry of the capacitance with regards to the applied terminal voltage by setting the **Symmetric C-V table** parameter to `Yes - use voltage magnitude when computing C`.

To model a frequency-dependent capacitor with resistive and dielectric losses, set the **Capacitance model** parameter to `Dielectric relaxation (Debye)`. The Debye relaxation model considers a collection of noninteracting dipoles in the frequency domain. The result is in terms of a complex permittivity. The real ( $\epsilon'$ ) and imaginary ( $\epsilon''$ ) parts of the complex permittivity are given by the equations:

$$\varepsilon' = \varepsilon_{\infty} + \frac{\varepsilon_s - \varepsilon_{\infty}}{1 + \omega^2\tau^2}$$

$$\varepsilon'' = \frac{(\varepsilon_s - \varepsilon_{\infty})\omega\tau}{1 + \omega^2\tau^2},$$

where  $\omega$  is the radial frequency,  $\varepsilon_{\infty}$  is the real permittivity at very high frequency,  $\varepsilon_s$  is the real permittivity at low frequency and  $\tau$  is the relaxation time constant.

In the time-domain, the characteristic equation for a capacitor in the Debye model is

$$\tau i = -q + C_s(\alpha\tau\dot{v} + v),$$

where  $C_s$  is the low-frequency capacitance,  $q = Q/C_s$  is the charge,  $i$  is the current and  $v$  is the voltage across the capacitor.

Most datasheets do not provide explicit values for the complex permittivity nor for the relaxation time; however, often the dissipation factor is provided at two frequencies. The parameters  $\tau$  and  $\alpha$  may be derived from these values as described by the equations:

$$\tau = \frac{DF_1DF_2(\omega_2^2 - \omega_1^2) + \sqrt{DF_1^2DF_2^2(\omega_2^2 - \omega_1^2)^2 + 4\omega_1\omega_2(DF_2\omega_2 - DF_1\omega_1)(DF_1\omega_2 - DF_2\omega_1)}}{2\omega_1\omega_2(DF_2\omega_2 - DF_1\omega_1)}$$

$$\alpha = \frac{\omega_1\tau - DF_1}{\omega_1\tau(1 + DF_1\omega_1\tau)},$$

where  $\omega_1$  and  $\omega_2$  are two different frequencies, and  $DF_1$  and  $DF_2$  are the dissipation factors calculated at said frequencies, respectively.

The argument of the square root must be positive for a valid parameterization in the Debye model.

## Tolerances

You can apply tolerances to the nominal value you provide for the **Capacitance** parameter. Datasheets typically provide a tolerance percentage for a given capacitor type. The table shows how the block applies tolerances and calculates capacitance based on the selected **Tolerance application** option.

Option	Capacitance Value
None – use nominal value	$C$
Random tolerance	Uniform distribution: $C \cdot (1 - tol + 2 \cdot tol \cdot rand)$ Gaussian distribution: $C \cdot (1 + tol \cdot randn / nSigma)$
Apply maximum tolerance value	$C \cdot (1 + tol)$
Apply minimum tolerance value	$C \cdot (1 - tol)$

In the table,

- $C$  is the **Capacitance** parameter value, nominal capacitance.
- $tol$  is the fractional tolerance, **Capacitance tolerance (%)** /100.

- $nSigma$  is the value you provide for the **Number of standard deviations for quoted tolerance** parameter.
- `rand` and `randn` are standard MATLAB® functions for generating uniform and normal distribution random numbers.

---

**Note** If you choose the `Random tolerance` option and you are in "Fast Restart" mode, the random tolerance value is updated on every simulation if at least one between the fractional tolerance,  $tol$ , or the **Number of standard deviations for quoted tolerance**,  $nSigma$ , is set to Run-time and is defined with a variable (even if you do not modify that variable).

---

### Operating Limits

You can specify operating limits in terms of maximum working voltage and the maximum (instantaneous) power dissipation in the series resistance and in the parallel conductance of the capacitor.

For polar capacitors, you can define the working voltage range in such a way that the block provides a warning, or an error, if the polarity of the applied voltage becomes incorrect.

When an operating limit is exceeded, the block can either generate a warning or stop the simulation with an error. For more information, see "Operating Limits" on page 1-204.

### Faults

Instantaneous changes in capacitor parameters are unphysical. Therefore, when the Capacitor block enters the faulted state, the capacitance, resistance, and conductance transition to their faulted values over a period of time, according to the following formula:

$$CurrentValue = FaultedValue - (FaultedValue - UnfaultedValue) \cdot \operatorname{sech}(\Delta t / \tau) \quad (1-1)$$

where:

- $\Delta t$  is the time since the onset of the fault condition.
- $\tau$  is the user-defined time constant associated with the fault transition.

The block can trigger the start of fault transition:

- At a specific time
- When terminal voltage is outside the permissible voltage range for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault transition takes precedence. In other words, component fails no more than once per simulation.

You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Variables

Use the **Variables** section of the block interface to set the priority and initial target values for the block variables prior to simulation. For more information, see “Set Priority and Initial Target for Block Variables”.

The **Capacitor voltage** variable lets you specify a high-priority target for the initial capacitor voltage at the start of simulation.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the capacitor positive voltage.

#### - – Negative terminal

electrical

Electrical conserving port associated with the capacitor negative voltage.

## Parameters

### Main

#### Capacitance model – Capacitor type

Constant (default) | Lookup table | Dielectric relaxation (Debye)

Select the type of capacitor:

- **Constant** — Model a linear capacitor, with nominal capacitance defined by the **Capacitance** parameter value. This is the default.
- **Lookup table** — Model a nonlinear capacitor, where the nominal capacitance value changes based on the value of applied terminal voltage.
- **Dielectric relaxation (Debye)** — Model a frequency-dependent capacitor with resistive and dielectric losses.

#### Capacitance — Nominal capacitance

1e-6 F (default)

The nominal capacitance value for linear capacitor, or the low-frequency capacitance in the Debye parameterization.

### Dependencies

This parameter is visible only when you select **Constant** or **Dielectric relaxation (Debye)** for the **Capacitance model** parameter.

#### Capacitance values — Vector of capacitance values

[1e-05, 1e-06] F (default) | positive vector

The vector of capacitance values, for table lookup based on the corresponding voltage value. Capacitance values must be greater than 0. The vector length must be the same as the voltage vector length.

**Dependencies**

This parameter is visible only when you select `Lookup table` for the **Capacitance model** parameter.

**Corresponding voltage values – Input vector of voltage values**

[0, 10] V (default)

The input vector of voltage values for table-based capacitance calculation. The vector length must be greater than or equal to 2, and the values must be strictly monotonic, either increasing or decreasing.

**Dependencies**

This parameter is visible only when you select `Lookup table` for the **Capacitance model** parameter.

**Symmetric C-V table – Table data**

Yes - use voltage magnitude when computing C (default) | No - use C-V data as-is

Specify how to use the table data:

- **Yes - use voltage magnitude when computing C** — Use this option to ensure symmetry of the capacitance with regards to the applied terminal voltage. This is the default.
- **No - use C-V data as-is** — Use this option to model polar capacitors. For example, with default parameter values for table-based capacitance, applied voltage of -10 V would produce nominal capacitance of 1e-6 F. However, if you select **No - use C-V data as-is** for the **Symmetric C-V table** parameter, the resulting capacitance value is 1e-5 F, because the block uses the nearest input value for extrapolation.

**Dependencies**

This parameter is visible only when you select `Lookup table` for the **Capacitance model** parameter.

**Frequencies for specifying dissipation factors [f1 f2] – Frequencies for dissipation factors calculation**

[1, 10] kHz (default)

Frequencies at which the dissipation factors **[DF1 DF2]** are calculated, in kHz.

**Dependencies**

This parameter is visible only when you select `Dielectric relaxation (Debye)` for the **Capacitance model** parameter.

**Dissipation factors (%) at f1 and f2 [DF1 DF2] – Dissipation factors at f1 and f2**

[.8, 1.2] (default)

Ratio between the equivalent series resistance and the capacitive reactance, or the tangent of the loss angle. Dissipation factors are a common metric for capacitors.



**Dependencies**

This parameter is visible only when you select `Dielectric relaxation (Debye)` for the **Capacitance model** parameter.

**Capacitance tolerance (%) – Capacitor tolerance**

5 (default)

The capacitor tolerance as defined on the manufacturer datasheet. For table-based capacitors, this tolerance is applied to the entire table at once.

**Tolerance application – Tolerance application**

None – use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation:

- `None – use nominal value` — The block does not apply tolerance, uses the nominal capacitance value. This is the default.
- `Random tolerance` — The block applies random offset to the capacitance value, within the tolerance value limit. You can choose `Uniform` or `Gaussian` distribution for calculating the random number by using the **Tolerance distribution** parameter.
- `Apply maximum tolerance value` — The capacitance is increased by the specified tolerance percent value.
- `Apply minimum tolerance value` — The capacitance is decreased by the specified tolerance percent value.

**Tolerance distribution – Tolerance distribution type**

Uniform (default) | Gaussian

Select the distribution type:

- `Uniform` — Uniform distribution. This is the default.
- `Gaussian` — Gaussian distribution.

**Dependencies**

This parameter is visible only when you select `Random tolerance` for the **Tolerance application** parameter.

**Number of standard deviations for quoted tolerance – Number of standard deviations for quoted tolerance**

4 (default)

Number of standard deviations for calculating the Gaussian random number.

**Dependencies**

This parameter is visible only when you select `Gaussian` for the **Tolerance distribution** parameter.

**Series resistance – Series resistance**

1e-6 Ohm (default)

Simulation of some circuits may require the presence of the small series resistance. Equivalent series resistance (ESR) is sometimes specified on manufacturer datasheets. If not, you can define this

resistance for a linear capacitor via the dissipation factor (DF), which is also shown on many datasheets. The relationship is  $DF = 2\pi \cdot f \cdot C \cdot ESR$ , where  $f$  is signal frequency. For a Debye capacitor, the **Dissipation factors (%) at f1 and f2 [DF1 DF2]** are corrected for this additional series resistance prior to computing  $\tan \delta$  and  $\cos \phi$ .

### **Parallel conductance — Parallel leakage path**

0 1/Ohm (default)

Parallel leakage path associated with the capacitor. For capacitors connected in series, the presence of a small parallel conductance can help with convergence.

### **Averaging period for power logging — Averaging period for power logging**

0 s (default)

Averaging period for power logging, in s.

---

**Note** If this parameter is set to 0, the output is the instantaneous power.

---

### **Dependencies**

This parameter is visible only when you select **Dielectric relaxation (Debye)** for the **Capacitance model** parameter.

### **Operating Limits**

#### **Enable operating limits — Enable operating limits**

No (default) | Yes

Select **Yes** to enable reporting when the operational limits are exceeded. The associated parameters become visible on the **Operating Limits** tab to let you select the reporting method and specify the operating limits in terms of power and working voltage.

#### **Reporting if operating limits exceeded — Report behavior**

Warn (default) | Error

Select what happens when an operating limit is exceeded:

- **Warn** — The block issues a warning. This is the default.
- **Error** — Simulation stops with an error.

#### **Working voltage range — Working voltage range**

[-25, 25] V (default)

Range of voltage values allowed for normal block operation, specified as a vector of size 2.

#### **Power rating — Maximum instantaneous power dissipation**

1 W (default)

Maximum instantaneous power dissipation in the resistance and conductance elements associated with the capacitor.

## Faults

### Enable faults — Enable faults

No (default) | Yes

Select **Yes** to enable faults modeling. The associated **Faults** parameters become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together. The default value is **No**.

### Reporting when a fault occurs — Report behavior

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- **None** — The block does not issue an assertion. This is the default.
- **Warn** — The block issues a warning.
- **Error** — Simulation stops with an error.

### Faulted capacitance as % of unfaulted — Faulted capacitance as % of unfaulted

100 (default)

Relative change in the capacitance when the block is in the faulted state, as compared to the unfaulted state. For table-based capacitances, the relative change is applied to all elements of the vector. The default value is 100, which means that the faulted capacitance is equal to the unfaulted capacitance.

### Faulted series resistance — Faulted series resistance

1e-3 Ohm (default)

Equivalent series resistance of the capacitor when the block is in the faulted state.

### Faulted parallel conductance — Faulted parallel conductance

0 (default) | 1/0hm

Parallel leakage conductance of the capacitor when the block is in the faulted state.

### Fault transition time constant — Fault transition time constant

1e-3 s (default)

Time constant associated with the transition to the faulted state, as described in “Faults” on page 1-200.

### Enable temporal fault trigger — Time-based fault triggering

No (default) | Yes

Select **Yes** to enable time-based fault triggering.

### Simulation time for fault event — Simulation time for fault event

1 s (default)

Set the simulation time at which you want the block to start entering the fault state.

### Dependencies

This parameter is visible only when you select **Yes** for the **Enable temporal fault trigger** parameter.

**Enable behavioral fault trigger – Behavioral fault triggering**

No (default) | Yes

Select Yes to enable behavioral fault triggering.

**Permissible voltage range – Minimum and maximum permissible voltage**

[-100, 100] V (default)

Specify the minimum and maximum permissible voltage. If the voltage value is outside this range for longer than the **Time to fail when exceeding voltage range** parameter value, then the block starts entering the fault state.

**Dependencies**

This parameter is visible only when you select Yes for the **Enable behavioral fault trigger** parameter.

**Time to fail when exceeding voltage range – Time to fail when exceeding voltage range**

1 s (default)

Set the maximum length of time that the voltage can be outside the permissible voltage range without triggering the fault.

**Dependencies**

This parameter is visible only when you select Yes for the **Enable behavioral fault trigger** parameter.

## Version History

**Introduced in R2016a**

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Fault | Inductor | Resistor

# Cauer Thermal Model

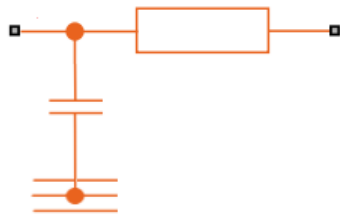
Heat transfer through multiple layers of a semiconductor module

**Library:** Simscape / Electrical / Passive / Thermal



## Description

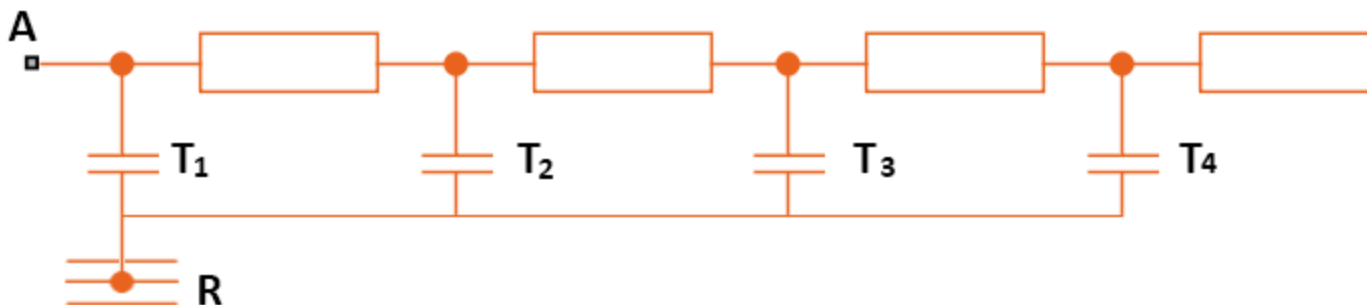
The Cauer Thermal Model block represents heat transfer through multiple layers of a semiconductor module. A Cauer Thermal Model contains multiple Cauer Thermal Model Element components. The figure shows an equivalent circuit for a Cauer Thermal Model Element.



The multiple layers of a Cauer thermal model include chip, solder, substrate, solder, and base. Other terms that describe a Cauer thermal model are:

- Continued fraction circuit
- T model
- Ladder network

This figure shows an equivalent circuit for a Cauer Thermal Model.



The temperature at **R** is equal to the absolute zero temperature.

## Equations

The defining equations for each Cauer thermal model element are

$$C_{thermal} = \frac{\tau}{R_{thermal}},$$

$$Q_{AB} = \frac{T_{AB}}{R_{thermal}},$$

and

$$Q_{AR} = C_{thermal} \frac{dT_{AR}}{dt},$$

where:

- $C_{thermal}$  is the thermal capacity.
- $\tau$  is the thermal time constant.
- $R_{thermal}$  is the thermal resistance.
- $Q_{AB}$  is the heat flow through the material.
- $T_{AB}$  is the temperature difference between the material layers.
- $Q_{AR}$  is the heat flow through the thermal capacity.
- $T_{AR}$  is the temperature drop across the thermal capacity.

### Parameterize Cauer Model from Foster Coefficients

The datasheets of semiconductor devices often use Foster coefficients to specify the thermal model. However, Cauer thermal models are more useful as you can extend them with additional thermal components such as heatsinks and radiation or convection elements.

To parameterize the Cauer Thermal Model block from Foster thermal model data, select the **Parameterize Cauer model using Foster coefficient data** parameter.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

To specify the initial temperatures of each thermal capacity, set the value of the **Vector of thermal mass temperatures** variable. The length of this vector must be equal to the number of Cauer thermal model elements. It corresponds to the temperature drop across each thermal capacity in the model.

### Ports

#### Conserving

**A — First surface of the semiconductor individual layer**  
thermal

Thermal conserving port associated with the first surface of the individual layer of the semiconductor.

### **B – Second surface of the semiconductor last layer**

thermal

Thermal conserving port associated with the second surface of the last layer of the semiconductor.

## **Parameters**

### **Parameterize Cauer model using Foster coefficient data – Parameterize Cauer model using Foster coefficient data**

Off (default) | On

Whether to parameterize the Cauer model by using coefficient data from a Foster model.

### **Thermal resistance data – Thermal resistances**

$1e-3 * [2.6468, 3.8701, 1.2007, .8824]$  K/W (default)

Vector of thermal resistances.

#### **Dependencies**

To enable this parameter, clear the **Parameterize Cauer model using Foster coefficient data** parameter.

### **Thermal time constant data – Thermal time constants**

$[.0086, .054, .3624, 1.651]$  s (default)

Vector of thermal time constants.

#### **Dependencies**

To enable this parameter, clear the **Parameterize Cauer model using Foster coefficient data** parameter.

### **Thermal resistance data (Foster) – Thermal resistances parameterized from Foster data**

$[.0016, .0043, .0013, .0014]$  K/W (default)

Vector of thermal resistances parameterized by using data from a Foster model.

#### **Dependencies**

To enable this parameter, select the **Parameterize Cauer model using Foster coefficient data** parameter.

### **Thermal time constant data (Foster) – Thermal time constants parameterized from Foster data**

$[.0068, .064, .32, 2]$  s (default)

Vector of thermal time constants parameterized by using data from a Foster model.

#### **Dependencies**

To enable this parameter, select the **Parameterize Cauer model using Foster coefficient data** parameter.

## Version History

Introduced in R2016a

### Removal of R port and Show absolute reference port parameter

*Behavior changed in R2022b*

The Cauer Thermal Model block no longer supports the external **R** port. The reference temperature is set to 0 Kelvin. In your models, delete the blocks previously connected to the **R** port of the Cauer Thermal Model block.

## References

- [1] Schütze, T. *AN2008-03: Thermal equivalent circuit models*. Application Note. V1.0. Germany: Infineon Technologies AG, 2008.
- [2] T. G. Subhash Joshi and V. John, *Combined transient thermal impedance estimation for pulse-power applications*. 2017 National Power Electronics Conference (NPEC), 2017, pp. 42-47, doi: 10.1109/NPEC.2017.8310432.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Foster Thermal Model | Thermal Resistor

### Functions

ee\_getCauerFromFoster

### Topics

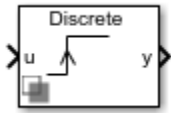
“Simulating Thermal Effects in Semiconductors”



# Change Detector

Boolean signal change detector

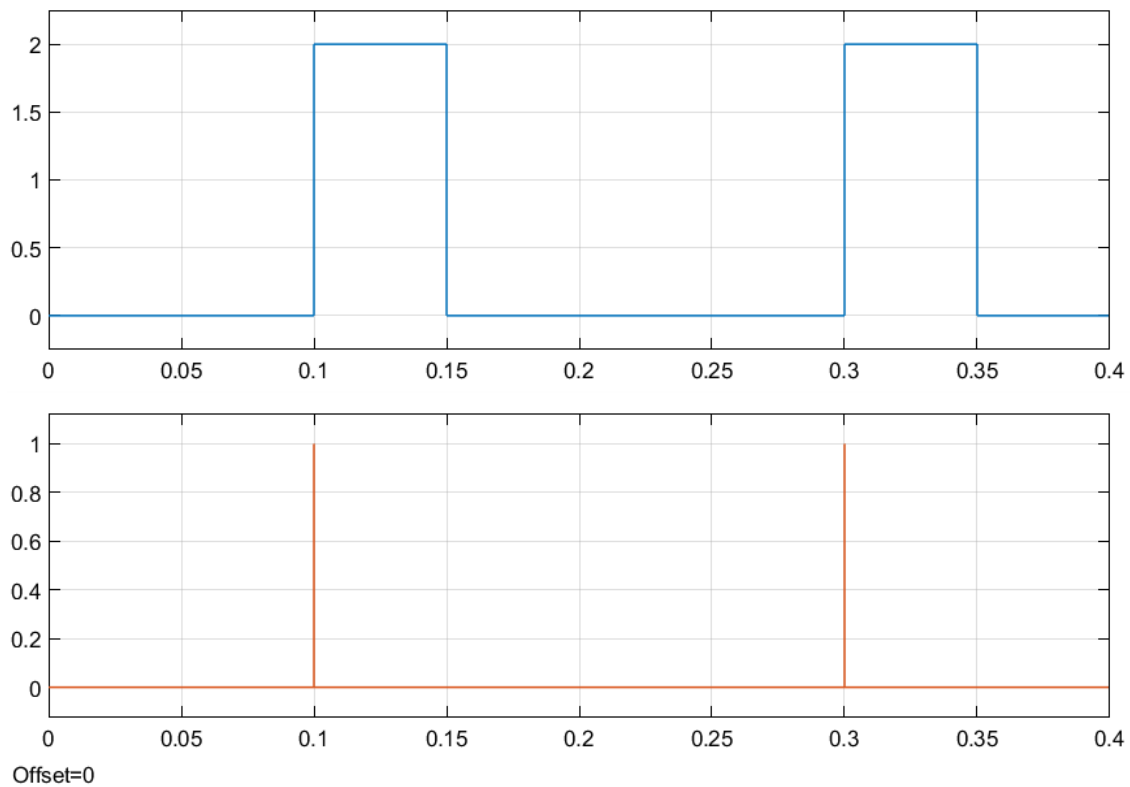
**Library:** Simscape / Electrical / Control / General Control



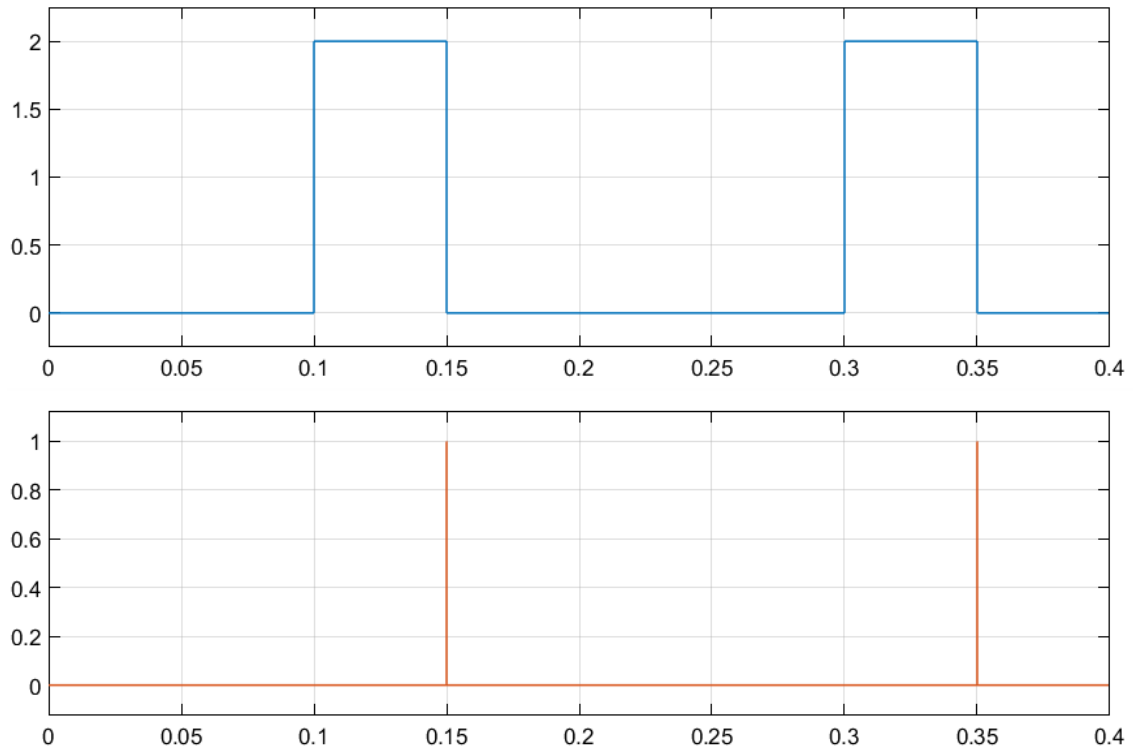
## Description

The Change Detector block outputs a Boolean response of `true` when it detects a change in the Boolean input signal that meets one of these change criteria:

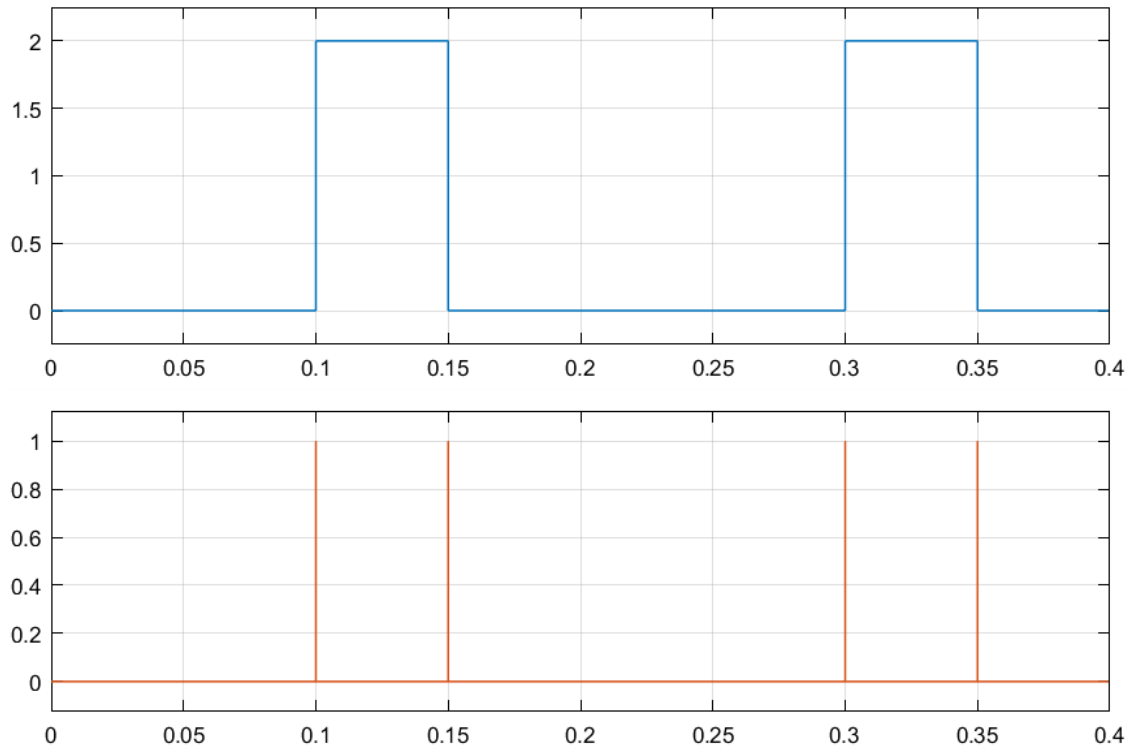
- Rising edge — The input goes from false to true.



- Falling edge — The input goes from true to false.



- Either edge — The input goes from true to false or from false to true.



## Ports

### Input

#### **u** — Boolean input

0 or 1

Input Boolean signal. If false, 0. If true, 1.

Data Types: Boolean

### Output

#### **y** — Change report

0 or 1

Output is true, 1, when the block detects a change that corresponds to the specified criteria (rising, falling, or either edge). Otherwise, output is false, 0.

Data Types: Boolean

## Parameters

#### **Change detection** — Change criteria

Rising edge (default) | Falling edge | Either edge

Criteria for change detection.

**Initial condition — Initial Boolean value**

0 (default) | 1

Initial value of the previous input. If the input at the start of simulation is different from the initial condition value, the block detects an edge.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2018b

## Extended Capabilities

**C/C++ Code Generation**

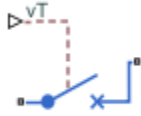
Generate C and C++ code using Simulink® Coder™.

## See Also

# Circuit Breaker

Single-pole single-throw circuit breaker

**Library:** Simscape / Electrical / Switches & Breakers



## Description

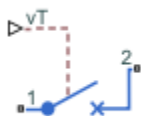
The Circuit Breaker block models a single-phase circuit breaker that uses an external signal and phase current information to break an electrical circuit.

The table shows how the external signal **vT** and the **Breaker Behavior** parameter control the block behavior.

Condition		Block Behavior	Resistance Parameter Used
<b>vT &lt; Threshold</b>		The breaker is closed. Port <b>1</b> connects to port <b>2</b> .	<b>Closed Resistance</b>
<b>vT ≥ Threshold</b>	Open when current crosses zero after vT ≥ Threshold	When the current in port <b>1</b> goes through zero, the phase disconnects from port <b>2</b> . The breaker is open.	<b>Open Conductance</b>
	Open when vT ≥ Threshold	The breaker is open..	

## Ports

Refer to the figure for port locations.



## Input

**vT — Scalar control port**

physical signal

Scalar physical signal control port.

**Dependencies**

To enable this port, set **Modeling option** to `PS control port`.

**Conserving****1 – Port 1**

electrical

Electrical conserving port.

**2 – Port 2**

electrical

Electrical conserving port.

**+ – Positive port**

electrical

Positive electrical conserving port.

**Dependencies**

To enable this port, set **Modeling option** to `Electrical control port`.

**- – Negative port**

electrical

Negative electrical conserving port.

**Dependencies**

To enable this port, set **Modeling option** to `Electrical control port`.

**Parameters****Modeling option – Whether to specify physical or electrical control port**

`PS control port` (default) | `Electrical control port`

Whether to specify physical or electrical control port for the switch gate.

**Closed resistance – Closed resistance**

`0.001 Ohm` (default)

Resistance between ports **1** and **2** when the breaker is closed.

**Open conductance – Open conductance**

`1e-6 1/Ohm` (default)

Conductance between ports **1** and **2** when the breaker is open.

**Threshold – Threshold voltage**

`0.5 V` (default)

Threshold voltage for the control port **vT**. The block uses the threshold voltage and the value of **vT** at the start of the simulation to determine whether the breaker is initially open or closed. If you set

**Breaker Behavior** to Open when current crosses zero after  $vT \geq \text{Threshold}$ , when the voltage rises above the threshold, the breaker opens as its current crosses zero. If you set **Breaker Behavior** to Open when  $vT \geq \text{Threshold}$ , the breaker opens as soon as the voltage rises above the threshold. When the control port voltage falls below the threshold, the breaker closes.

#### **Breaker Behavior — Breaker Behavior when $vT$ rises above threshold**

Open when current crosses zero after  $vT \geq \text{Threshold}$  (default) | Open when  $vT \geq \text{Threshold}$

Breaker Behavior when the value at the control port  **$vT$**  is greater than or equal to the value of the **Threshold** parameter. The breaker can open either as soon as the voltage rises above the threshold, or when its current also crosses zero.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Circuit Breaker (Three-Phase) | Circuit Breaker (with arc)

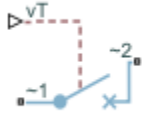
## **Topics**

“Switch Between Physical Signal and Electrical Ports”

## Circuit Breaker (Three-Phase)

Three-phase circuit breaker controlled by external signal

**Library:** Simscape / Electrical / Switches & Breakers



### Description

The Circuit Breaker (Three-Phase) block models a three-phase circuit breaker that uses an external signal and phase current information to break an electrical circuit.

The table shows how the external signal **vT** and the **Breaker Behavior** parameter control the block behavior.

Condition		Block Behavior	Resistance Parameter Used
<b>vT &lt; Threshold</b>		The breaker is closed. Each phase in the composite three-phase port <b>~1</b> connects to the corresponding phase in the port <b>~2</b> .	<b>Closed Resistance</b>
<b>vT ≥ Threshold</b>	Open when current crosses zero after <b>vT ≥ Threshold</b>	When the current in any phase of the composite port <b>~1</b> crosses zero, the phase disconnects from the corresponding phase at port <b>~2</b> . The breaker is open.	<b>Open Conductance</b>
	Open when <b>vT ≥ Threshold</b>	The breaker is open.	

### Ports

#### Input

##### **vT — Scalar control port**

physical signal

Scalar physical signal control port.

#### Dependencies

To enable this port, set **Modeling option** to Expanded three-phase ports | PS control port or Composite three-phase ports | PS control port.



**Conserving****~1 — Expandable three-phase port**

electrical

Expandable three-phase electrical conserving port.

**~2 — Expandable three-phase port**

electrical

Expandable three-phase electrical conserving port.

**+ — Positive port**

electrical

Positive electrical conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports | Electrical control port or Composite three-phase ports | Electrical control port.

**- — Negative port**

electrical

Negative electrical conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports | Electrical control port or Composite three-phase ports | Electrical control port.

**Parameters****Modeling option — Connection and gate control ports**

Composite three-phase ports | PS control port (default) | Composite three-phase ports | Electrical control port | Expanded three-phase ports | PS control port | Expanded three-phase ports | Electrical control port

Whether to model composite or expanded three-phase ports and physical signal or electrical control ports.

**Closed resistance — Closed resistance**

0.001 Ohm (default)

Resistance between ports ~1 and ~2 when the breaker is closed.

**Open conductance — Open conductance**

1e-6 1/Ohm (default)

Conductance between ports ~1 and ~2 when the breaker is open.

**Threshold — Threshold voltage**

0.5 V (default)

Threshold voltage for the control port **vT**. The block uses the threshold voltage and the value of **vT** at the start of the simulation to determine whether the breaker is initially open or closed. If you set

**Breaker Behavior** to Open when current crosses zero after  $vT \geq \text{Threshold}$ , when the voltage rises above the threshold, the breaker opens each phase as its current crosses zero. If you set **Breaker Behavior** to Open when  $vT \geq \text{Threshold}$ , the breaker opens each phase as soon as the voltage rises above the threshold. When the control port voltage falls below the threshold, the breaker closes.

**Breaker Behavior — Breaker Behavior when  $vT$  rises above threshold**

Open when current crosses zero after  $vT \geq \text{Threshold}$  (default) | Open when  $vT \geq \text{Threshold}$

Breaker Behavior when the value at the control port  **$vT$**  is greater than or equal to the value of the **Threshold** parameter. The breaker can open either as soon as the voltage rises above the threshold, or when its current also crosses zero.

## Version History

Introduced in R2013b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Circuit Breaker | Circuit Breaker (with arc)

## Topics

“Expand and Collapse Three-Phase Ports on a Block”

“Switch Between Physical Signal and Electrical Ports”

## Circuit Breaker (with arc)

Single-pole single-throw circuit breaker with Mayr arc representation

**Library:** Simscape / Electrical / Switches & Breakers



### Description

The Circuit Breaker (with arc) block represents a single-phase circuit breaker with Mayr arc representation controlled by an external control signal  $vT$ . If  $vT$  is less than the threshold, then the breaker is closed. If  $vT$  is greater than or equal to the threshold, then the breaker opens with an arc during the current interruption. The external signal can open and close the breaker repeatedly.

The table shows how the external signal  $vT$  controls the block behavior.

Condition	Block Behavior
$vT < \text{Threshold}$	The circuit breaker is closed. Port <b>1</b> is connected to port <b>2</b> .
$vT \geq \text{Threshold}$	The circuit breaker is either opening or open. Port <b>1</b> is connected to port <b>2</b> via a nonlinear conductance.

The Circuit Breaker (with arc) block has a higher computational overhead than the Circuit Breaker block. If the fidelity of the representation of arc current or voltage is your overriding requirement, use the Circuit Breaker (with arc) block and use a global Simulink variable-step solver. Otherwise, use the Circuit Breaker block.

### Mayr Arc Model Equations

The defining equations for the breaker are

$$x = \ln(g)$$

and

$$i = gv,$$

where:

- $g$  is the arc conductance.
- $x$  is an internal state variable.
- $v$  is the voltage across the breaker.
- $i$  is the current through the breaker.

When the breaker is closed,

$$\frac{dx}{dt} = 0.$$

When the breaker is opening or open,

$$\frac{dx}{dt} = \frac{1}{\tau} \left( \frac{gv^2}{P} - 1 \right),$$

where:

- $\tau$  is the arc time constant.
- $P$  is the cooling power.

## Ports

Refer to the figure for port locations.



### Input

#### vT — Scalar control port

physical signal

Scalar physical signal control port.

#### Dependencies

To enable this port, set **Modeling option** to PS control port.

### Conserving

#### 1 — Port 1

electrical

Electrical conserving port.

#### 2 — Port 2

electrical

Electrical conserving port.

#### + — Positive port

electrical

Positive electrical conserving port.

#### Dependencies

To enable this port, set **Modeling option** to Electrical control port.

#### - — Negative port

electrical

Negative electrical conserving port.

### Dependencies

To enable this port, set **Modeling option** to Electrical control port.

## Parameters

### Modeling option — Whether to specify physical or electrical control port

PS control port (default) | Electrical control port

Whether to specify physical or electrical control port for the switch gate.

### Arc time constant, tau — Arc time constant

0.3e-6 s (default)

Time constant for the Mayr arc..

### Cooling power, P — Cooling power

30900 W (default)

Cooling power.

### Initial arc conductance, g0 — Initial arc conductance

1e4 s (default)

Conductance between ports **1** and **2** when the breaker is closed.

### Threshold — Threshold voltage

0.5 V (default)

Threshold voltage for the control port  $vT$ . The block uses the threshold voltage and the value of  $vT$  at the start of the simulation to determine whether the breaker is initially open or closed. When the voltage rises above the threshold, the breaker opens. When the control port voltage falls below the threshold, the breaker closes.

## Version History

Introduced in R2015b

## References

- [1] Schavemaker, P. H., and L. Van der Sluis. "The Arc Model Blockset." *Proceedings of the Second IASTED International Conference POWER AND ENERGY SYSTEMS (EuroPES)*. Crete, Greece, June 25-28, 2002, pp. 644-648.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Circuit Breaker | Circuit Breaker (Three-Phase)

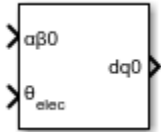
**Topics**

“Switch Between Physical Signal and Electrical Ports”

# Clarke to Park Angle Transform

Implement  $\alpha\beta 0$  to  $dq0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

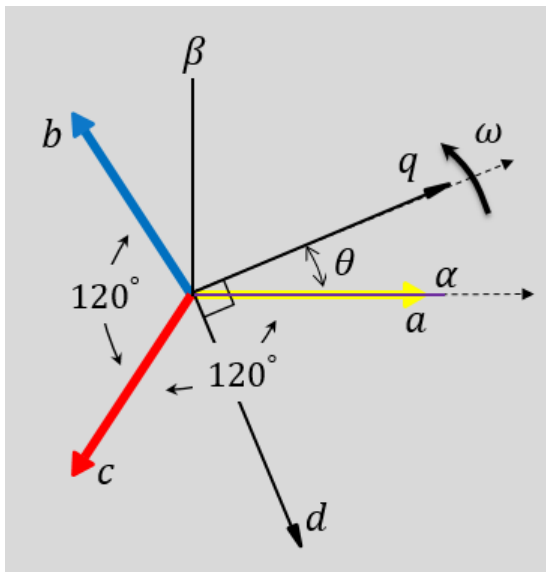


## Description

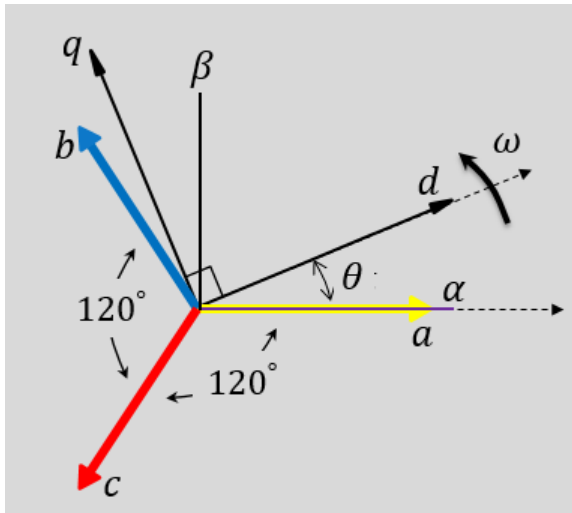
The Clarke to Park Angle Transform block converts the alpha, beta, and zero components in a stationary reference frame to direct, quadrature, and zero components in a rotating reference frame. For balanced three-phase systems, the zero components are equal to zero.

You can configure the block to align the phase  $a$ -axis of the three-phase system to either the  $q$ - or  $d$ -axis of the rotating reference frame at time,  $t = 0$ . The figures show the direction of the magnetic axes of the stator windings in the three-phase system, a stationary  $\alpha\beta 0$  reference frame, and a rotating  $dq0$  reference frame where:

- The  $a$ -axis and the  $q$ -axis are initially aligned.



- The  $a$ -axis and the  $d$ -axis are initially aligned.



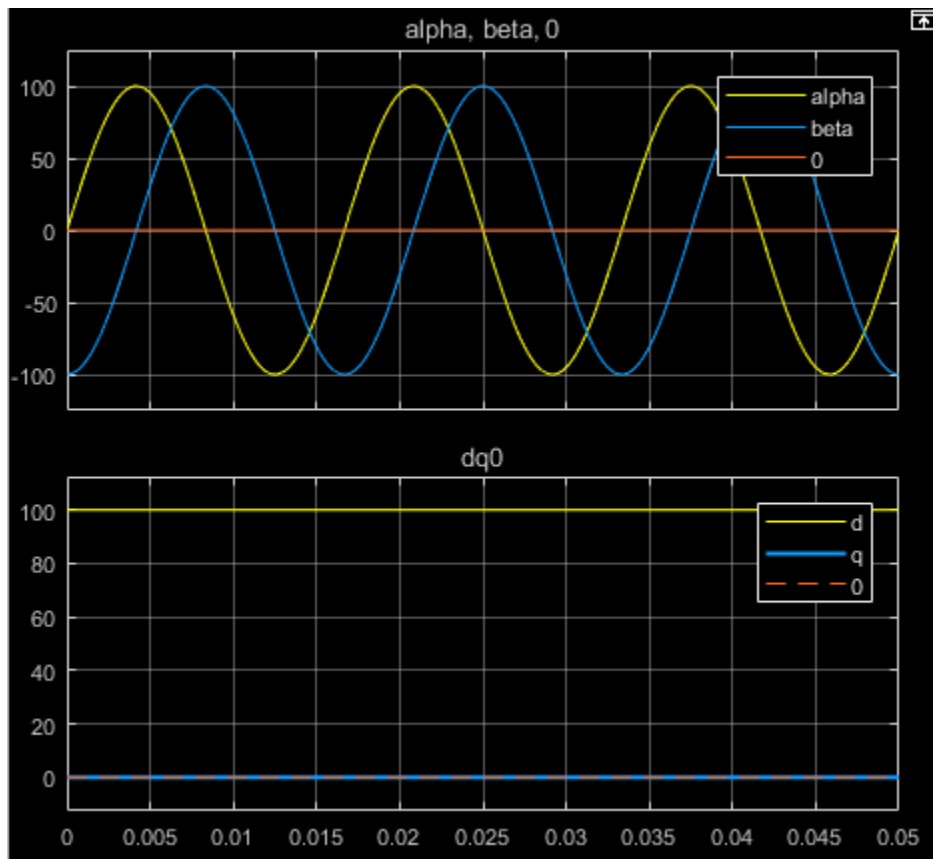
In both cases, the angle  $\theta = \omega t$ , where

- $\theta$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment.
- $\omega$  is the rotational speed of the  $d$ - $q$  reference frame.
- $t$  is the time, in s, from the initial alignment.

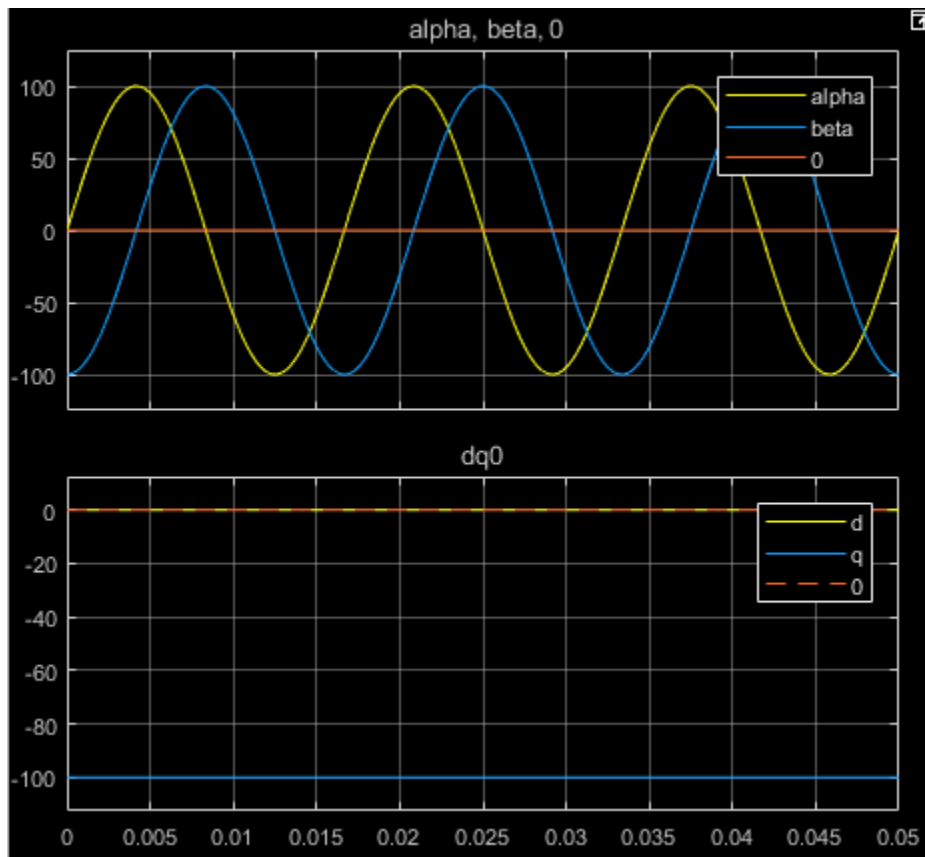
The figures show the time-response of the individual components of equivalent balanced  $\alpha\beta 0$  and  $dq 0$  for an:

- Alignment of the  $a$ -phase vector to the  $q$ -axis





- Alignment of the  $a$ -phase vector to the  $d$ -axis



### Equations

The Clarke to Park Angle Transform block implements the transform for an  $\alpha$ -phase to  $q$ -axis alignment as

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & 0 \\ \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}$$

where:

- $\alpha$  and  $\beta$  are the alpha-axis and beta-axis components of the two-phase system in the stationary reference frame.
- $0$  is the zero component.
- $d$  and  $q$  are the direct-axis and quadrature-axis components of the two-axis system in the rotating reference frame.

For an  $\alpha$ -phase to  $d$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}$$

## Ports

### Input

#### $\alpha\beta 0$ — $\alpha$ - $\beta$ axis and zero components

vector

Alpha-axis,  $\alpha$ , beta-axis,  $\beta$ , and zero components of the two-phase system in the stationary reference frame.

Data Types: single | double

#### $\theta_{abc}$ — Rotational angle

scalar | in radians

Angular position of the rotating reference frame. The value of this parameter is equal to the polar distance from the vector of the  $a$ -phase in the  $abc$  reference frame to the initially aligned axis of the  $dq0$  reference frame.

Data Types: single | double

### Output

#### $dq0$ — $d$ - $q$ axis and zero components

vector

Direct-axis and quadrature-axis components and the zero component of the system in the rotating reference frame.

Data Types: single | double

## Parameters

#### Phase-a axis alignment — $dq0$ reference frame alignment

Q-axis (default) | D-axis

Align the  $a$ -phase vector of the  $abc$  reference frame to the  $d$ - or  $q$ -axis of the rotating reference frame.

## Version History

Introduced in R2017b

## References

- [1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscatawy, NJ: Wiley-IEEE Press, 2013.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Clarke Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform

# Clarke Transform

Implement  $abc$  to  $\alpha\beta 0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

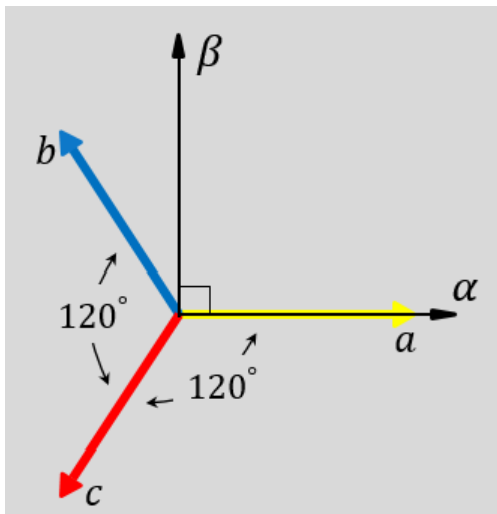


## Description

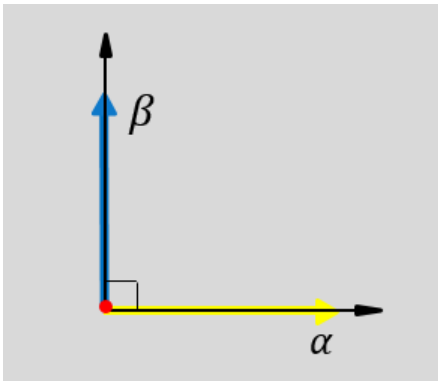
The Clarke Transform block converts the time-domain components of a three-phase system in an  $abc$  reference frame to components in a stationary  $\alpha\beta 0$  reference frame. The block can preserve the active and reactive powers with the powers of the system in the  $abc$  reference frame by implementing a power invariant version of the Clarke transform. For a balanced system, the zero component is equal to zero.

The figures show:

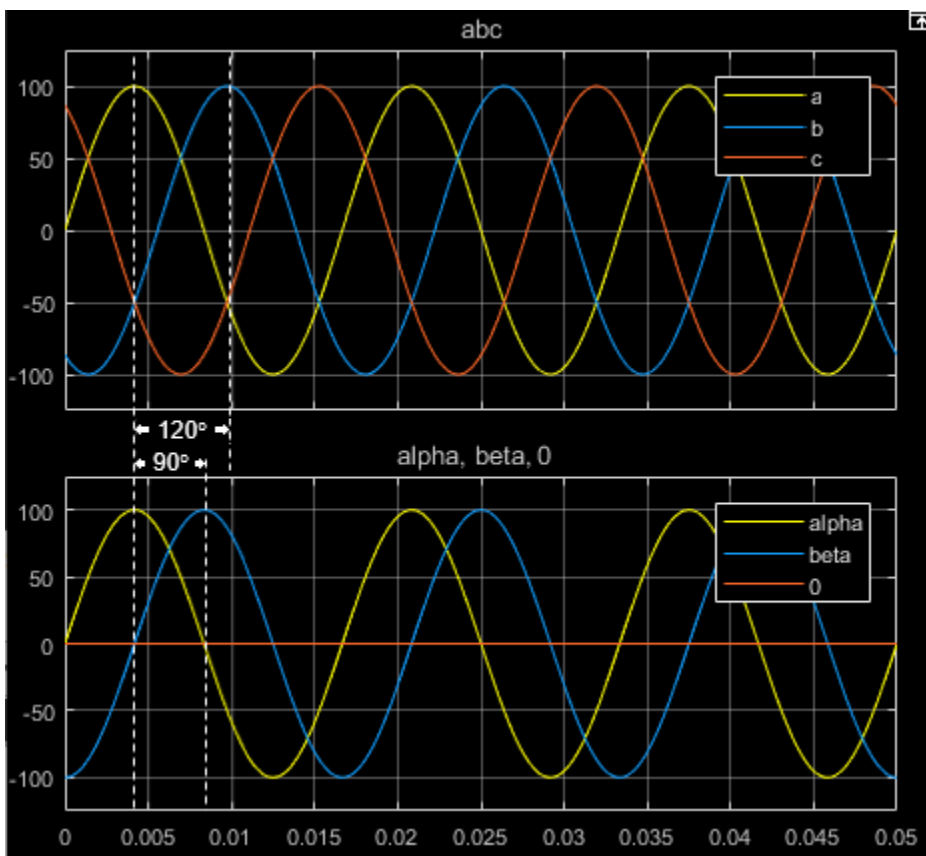
- The direction of the magnetic axes of the stator windings in the  $abc$  reference frame and the stationary  $\alpha\beta 0$  reference frame



- Equivalent  $\alpha$ ,  $\beta$ , and zero components in the stationary reference frame



- The time-response of the individual components of equivalent balanced  $abc$  and  $\alpha\beta 0$  systems



### Equations

The block implements the Clarke transform as

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

where:

- $a$ ,  $b$ , and  $c$  are the components of the three-phase system in the  $abc$  reference frame.
- $\alpha$  and  $\beta$  are the components of the two-axis system in the stationary reference frame.
- $0$  is the zero component of the two-axis system in the stationary reference frame.

The block implements the power invariant version of the Clarke transform as

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{1}}{\sqrt{2}} & \frac{\sqrt{1}}{\sqrt{2}} & \frac{\sqrt{1}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

## Ports

### Input

#### **abc** — **a-, b-, and c-phase components**

vector

Components of the three-phase system in the  $abc$  reference frame.

Data Types: single | double

### Output

#### **$\alpha\beta 0$** — **$\alpha$ - $\beta$ axis and zero components**

vector

Alpha-axis component  $\alpha$ , beta-axis component  $\beta$ , and zero component in the stationary reference frame.

Data Types: single | double

## Parameters

#### **Power Invariant** — **Power invariant transform**

off (default) | on

Preserve the active and reactive power of the system in the  $abc$  reference frame.

## Version History

Introduced in R2017b

## References

- [1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscatawy, NJ: Wiley-IEEE Press, 2013.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform



# CMOS AND

Behavioral model of CMOS AND gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



## Description

The CMOS AND block represents a CMOS AND logic gate behaviorally:

- The block output logic level is **HIGH** if the logic levels of both of the gate inputs are 1.
- The block output logic level is **LOW** otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-236 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### **Voltage Plot**

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## **Assumptions and Limitations**

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## **Ports**

### **Conserving**

#### **A — Input port**

electrical

Electrical conserving port associated with the CMOS AND input.

#### **B — Input port**

electrical

Electrical conserving port associated with the CMOS AND input.

#### **J — Output port**

electrical

Electrical conserving port associated with the CMOS AND output.

## **Parameters**

### **Inputs**

#### **Low level input voltage — Low level input voltage**

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### **High level input voltage — High level input voltage**

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

### **Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

### **Outputs**

#### **Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

#### **Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

#### **Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

#### **High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

#### **Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

#### **Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

#### **Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

#### **Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at  $I_{OH}$  — Logic HIGH output resistance at zero current and at  $I_{OH}$** 

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current  $I_{OH}$  when shorted to ground — Logic HIGH output current  $I_{OH}$  when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at  $I_{OL}$  — Logic LOW output resistance at zero current and at  $I_{OL}$** 

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> – Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay – Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance – Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage – Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state – Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

**Version History****Introduced in R2008b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS NOT | CMOS OR | CMOS XOR | Schmitt Trigger | S-R Latch

## CMOS Buffer

Behavioral model of CMOS Buffer gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS Buffer block represents a CMOS Buffer logic gate behaviorally:

- The block output logic level is HIGH if the logic level of the gate input is 1.
- The block output logic level is LOW otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-242 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.
- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### **A — Input port**

electrical

Electrical conserving port associated with the CMOS Buffer input.

#### **J — Output port**

electrical

Electrical conserving port associated with the CMOS Buffer output.

## Parameters

### Inputs

#### **Low level input voltage — Low level input voltage**

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### **High level input voltage — High level input voltage**

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

#### **Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the



**Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

## Outputs

### Output current-voltage relationship – Output current-voltage relationship

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

### Low level output voltage – Low level output voltage

0 V (default)

Voltage value at the output when the output logic level is LOW.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### High level output voltage – High level output voltage

5 V (default)

Voltage value at the output when the output logic level is HIGH.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### Output resistance – Output resistance

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### Supply voltage – Supply voltage

5 V (default)

Supply voltage value applied to the gate in your circuit.

#### Dependencies

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I<sub>OH</sub> — Logic HIGH output resistance at zero current and at I<sub>OH</sub>**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I<sub>OH</sub> when shorted to ground — Logic HIGH output current I<sub>OH</sub> when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I<sub>OL</sub> — Logic LOW output resistance at zero current and at I<sub>OL</sub>**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage  $V_{cc}$ .

### Dependencies

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

### Propagation delay — Propagation delay

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

### Protection diode on resistance — Protection diode on resistance

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

### Dependencies

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

### Protection diode forward voltage — Protection diode forward voltage

0.6 V (default)

The voltage above which the protection diode is turned on.

### Dependencies

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

### Initial Conditions

#### Output initial state — Output initial state

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

CMOS AND | CMOS NAND | CMOS NOR | CMOS NOT | CMOS OR | CMOS XOR | Schmitt Trigger | S-R Latch

## CMOS NAND

Behavioral model of CMOS NAND gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS NAND block represents a CMOS NAND logic gate behaviorally:

- The block output logic level is HIGH if the logic levels of both of the gate inputs are 0.
- The block output logic level is LOW otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-247 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For Quadratic model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### Voltage Plot

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the Quadratic option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### A — Input port

electrical

Electrical conserving port associated with the CMOS NAND input.

#### B — Input port

electrical

Electrical conserving port associated with the CMOS NAND input.

#### J — Output port

electrical

Electrical conserving port associated with the CMOS NAND output.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### High level input voltage — High level input voltage

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I\_OH — Logic HIGH output resistance at zero current and at I\_OH**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I\_OH when shorted to ground — Logic HIGH output current I\_OH when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I\_OL — Logic LOW output resistance at zero current and at I\_OL**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b



## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NOR | CMOS NOT | CMOS OR | CMOS XOR | Schmitt Trigger | S-R Latch

## CMOS NOR

Behavioral model of CMOS NOR gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS NOR block represents a CMOS NOR logic gate behaviorally:

- The block output logic level is **LOW** if the logic levels of any of the gate inputs are 1.
- The block output logic level is **HIGH** otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-253 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### Voltage Plot

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### A — Input port

electrical

Electrical conserving port associated with the CMOS NOR input.

#### B — Input port

electrical

Electrical conserving port associated with the CMOS NOR input.

#### J — Output port

electrical

Electrical conserving port associated with the CMOS NOR output.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### High level input voltage — High level input voltage

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I\_OH — Logic HIGH output resistance at zero current and at I\_OH**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I\_OH when shorted to ground — Logic HIGH output current I\_OH when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I\_OL — Logic LOW output resistance at zero current and at I\_OL**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOT | CMOS OR | CMOS XOR | Schmitt Trigger | S-R Latch

## CMOS NOT

Behavioral model of CMOS NOT gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS NOT block represents a CMOS NOT logic gate behaviorally:

- The block output logic level is HIGH if the logic level of the gate input is 0.
- The block output logic level is LOW otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-259 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:



- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.
- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### Voltage Plot

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### A — Input port

electrical

Electrical conserving port associated with the CMOS NOT input.

#### J — Output port

electrical

Electrical conserving port associated with the CMOS NOT output.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### High level input voltage — High level input voltage

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I\_OH — Logic HIGH output resistance at zero current and at I\_OH**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I\_OH when shorted to ground — Logic HIGH output current I\_OH when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I\_OL — Logic LOW output resistance at zero current and at I\_OL**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS OR | CMOS XOR | Schmitt Trigger | S-R Latch

## CMOS OR

Behavioral model of CMOS OR gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS OR block represents a CMOS OR logic gate behaviorally:

- The block output logic level is HIGH if the logic levels of any of the gate inputs are 1.
- The block output logic level is LOW otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-265 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### Voltage Plot

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### A — Input port

electrical

Electrical conserving port associated with the CMOS OR input.

#### B — Input port

electrical

Electrical conserving port associated with the CMOS OR input.

#### J — Output port

electrical

Electrical conserving port associated with the CMOS OR output.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### High level input voltage — High level input voltage

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.



**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I\_OH — Logic HIGH output resistance at zero current and at I\_OH**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I\_OH when shorted to ground — Logic HIGH output current I\_OH when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I\_OL — Logic LOW output resistance at zero current and at I\_OL**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS NOT | CMOS XOR | Schmitt Trigger  
| S-R Latch

## CMOS XOR

Behavioral model of CMOS XOR gate

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The CMOS XOR block represents a CMOS XOR logic gate behaviorally:

- The block output logic level is **HIGH** if the logic level of exactly one of the gate inputs is 1.
- The block output logic level is **LOW** otherwise.

The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-271 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

### Voltage Plot

To access a voltage plot for each electrical port, in your model, right-click the block and, from the context menu, select **Electrical > Basic Characteristics**.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

Circuits that involve a feedback path around a set of logic gates may require a nonzero propagation delay to be set on one or more gates.

## Ports

### Conserving

#### A — Input port

electrical

Electrical conserving port associated with the CMOS XOR input.

#### B — Input port

electrical

Electrical conserving port associated with the CMOS XOR input.

#### J — Output port

electrical

Electrical conserving port associated with the CMOS XOR output.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### High level input voltage — High level input voltage

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I\_OH — Logic HIGH output resistance at zero current and at I\_OH**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I\_OH when shorted to ground — Logic HIGH output current I\_OH when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I\_OL — Logic LOW output resistance at zero current and at I\_OL**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2008b



## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS NOT | CMOS OR | Schmitt Trigger | S-R Latch

# Comparator

Behavioral model of a comparator integrated circuit

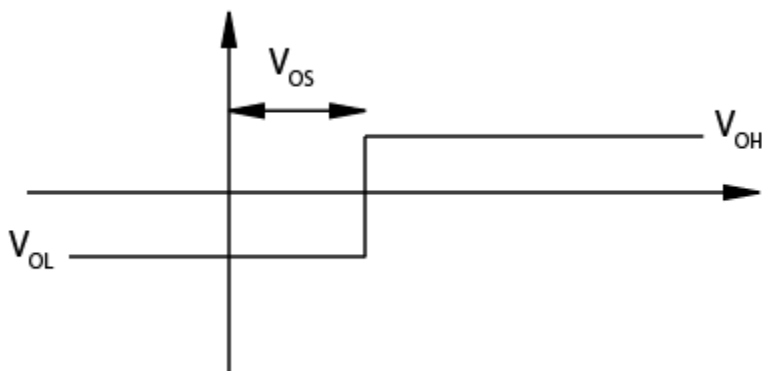
**Library:** Simscape / Electrical / Integrated Circuits



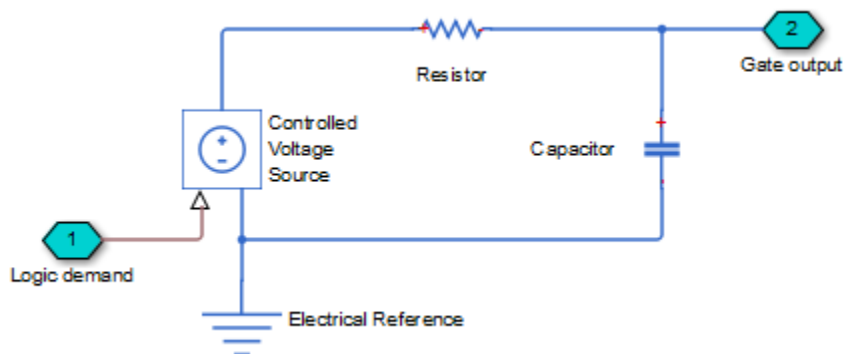
## Description

The Comparator block is an abstracted behavioral model of a comparator integrated circuit. It does not model an internal transistor-level implementation. Therefore, the block runs quickly during simulation but retains the correct I/O behavior. The block models differential inputs electrically as having infinite resistance and a finite or zero capacitance.

The block models the gate output as a voltage source driving a series resistor and a capacitor that connects to ground. The output pin connects to the resistor-capacitor connection node. If the difference in the inputs is greater than the input threshold voltage, then the output is equal to the **High level output voltage** ( $V_{OH}$ ). Otherwise, the output is equal to the **Low level output voltage** ( $V_{OL}$ ).



The output model is shown in the following illustration.



## Assumptions and Limitations

Modeling of the output as a controlled voltage source is representative of a totem-pole or push-pull output stage. To model a device with an open-collector:

- 1 Connect the output pin to the base of an NPN Bipolar Transistor or PNP Bipolar Transistor block.
- 2 Set the **Output resistance** parameter to a suitable value.

## Ports

### Conserving

#### + – Non-inverting input

electrical

Electrical conserving port associated with the comparator non-inverting input.

#### - – Inverting input

electrical

Electrical conserving port associated with the comparator inverting input.

#### OUT – Output port

electrical

Electrical conserving port associated with the comparator output

## Parameters

### Input

#### Input offset voltage – Input offset voltage

5 mV (default)

The voltage which the difference in the input voltages must be greater than so that the comparator gives a logic output 1.

#### Average input capacitance – Average input capacitance

0 pF (default)

You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero can result in faster simulation times.

### Output

#### Low level output voltage – Low level output voltage

0 V (default)

The steady-state output voltage,  $V_{OL}$ , when the voltage difference across the inputs is less than or equal to the threshold voltage, and the output current is zero.

#### High level output voltage – High level output voltage

5 V (default)

The steady-state output voltage,  $V_{OH}$ , when the voltage difference across the inputs is greater than the threshold voltage, and the output current is zero.

**Output resistance – Output resistance**

50 Ohm (default)

This parameter is the ratio of output voltage drop to output current. Set this parameter to  $(V_{OH} - V_{OH1})/I_{OH1}$ , where  $V_{OH1}$  is the reduced output high voltage when the output current is  $I_{OH1}$ .

**Propagation delay – Propagation delay**

0 s (default)

Set this value based on the high-to-low and low-to-high propagation delays.

## Version History

Introduced in R2009b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

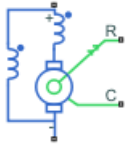
**See Also**

CMOS Buffer

# Compound Motor

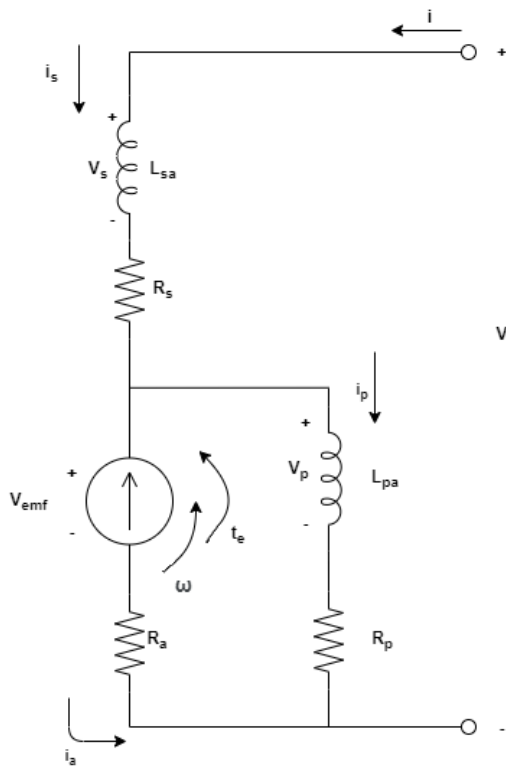
Compound motor model with electrical and torque characteristics and fault modeling

**Library:** Simscape / Electrical / Electromechanical / Brushed Motors

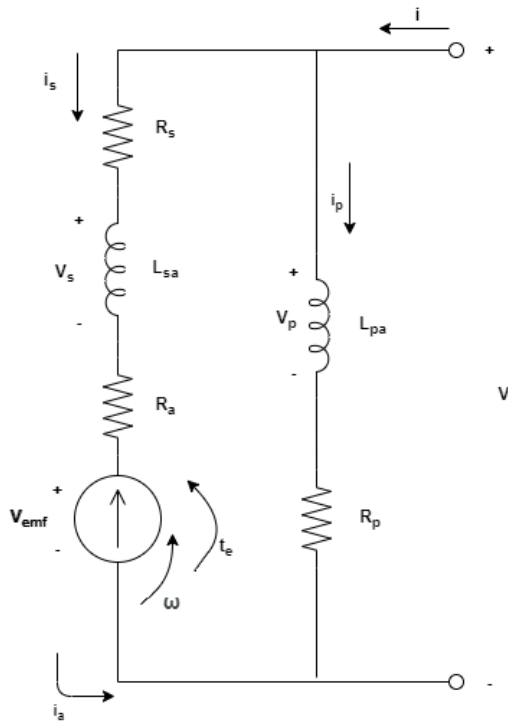


## Description

The Compound Motor block represents the electrical and torque characteristics of a compound motor. This figure shows the equivalent circuit for a short-shunt compound motor:



This figure shows the equivalent circuit for a long-shunt compound motor:



where:

- $i$  is the total current.
- $i_s$  is the series field winding current.
- $i_p$  is the parallel field winding current.
- $i_a$  is the armature current.
- $V$  is the total voltage.
- $V_s$  is the series field winding voltage.
- $V_p$  is the parallel field winding voltage.
- $V_a$  is the armature voltage.
- $\omega$  is the angular velocity.
- $t_e$  is the torque.

If you set the **Steady-state parameterization** parameter to **By equivalent circuit parameters**, you can specify the equivalent circuit parameters for this model:

- $R_a$  — **Armature resistance, Ra**
- $R_s$  — **Series field winding resistance, Rs**
- $R_p$  — **Shunt field winding resistance, Rp**
- $L_{sa}$  — **Series field winding to armature back EMF constant, Lsa**
- $L_{pa}$  — **Shunt field winding to armature back EMF constant, Lpa**

### Short-Shunt Equations

When **Electrical circuit topology** is set to **Short - shunt**, the electrical dynamic equations are:

$$\begin{aligned}
V_s &= R_s i_s + L_s \frac{di_s}{dt} + L_{sp} \frac{di_p}{dt} \\
V_p &= R_p i_p + L_p \frac{di_p}{dt} + L_{sp} \frac{di_s}{dt} \\
V_{emf} &= k_v \omega = (L_{sa} i_s + L_{pa} i_p) \omega \\
V &= V_s + V_p \\
V_p &= k_v \omega + R_a i_a \\
i &= i_s \\
i_a &= i_s - i_p
\end{aligned}$$

These are the mechanical dynamic equations for the short-shunt compound motor:

$$\begin{aligned}
J \dot{\omega} + D \omega &= t_e + t_{load} \\
t_e &= k_v i_a = (L_{sa} i_s + L_{pa} i_p) i_a
\end{aligned}$$

From these dynamic equations, the block obtains the steady-state equations by making the derivatives equal to zero:

$$\begin{aligned}
V_{emf} &= L_{sa} i_s \omega + L_{pa} i_p \omega \\
t_{elec} &= k_v (i_s - i_p) = V_{emf} \frac{i_s - i_p}{\omega} \\
V &= R_s i_s + R_p i_p \\
R_p i_p &= V_{emf} + R_a (i_s - i_p)
\end{aligned}$$

Then, it computes the steady-state currents and torque as follows:

$$\begin{aligned}
t_{elec}(\omega, V) &= \frac{-V^2 (L_{pa} \omega + L_{sa} \omega - R_p) (R_a L_{pa} + R_a L_{sa} + R_p L_{sa})}{(R_a R_p + R_a R_s + R_p R_s + L_{sa} R_p \omega - L_{pa} R_s \omega)^2} \\
i &= i_s(\omega, V) = \frac{V (R_a + R_p - L_{pa} \omega)}{R_a R_p + R_a R_s + R_p R_s + L_{sa} R_p \omega - L_{pa} R_s \omega}
\end{aligned}$$

### Long-Shunt Equations

When **Electrical circuit topology** is set to Long - shunt, the electrical dynamic equations are:

$$\begin{aligned}
V_s &= R_s i_s + L_s \frac{di_s}{dt} + L_{sp} \frac{di_p}{dt} \\
V_p &= R_p i_p + L_p \frac{di_p}{dt} + L_{sp} \frac{di_s}{dt} \\
V_{emf} &= k_v \omega = (L_{sa} i_s + L_{pa} i_p) \omega \\
V &= V_p \\
V_p &= k_v \omega + R_a i_a + V_s \\
i &= i_s + i_p \\
i_a &= i_s
\end{aligned}$$

These are the mechanical dynamic equations for the long-shunt compound motor:

$$J\dot{\omega} + D\omega = t_e + t_{load}$$

$$t_e = k_v i_a = (L_{sa} i_s + L_{pa} i_p) \dot{i}_a$$

From these dynamic equations, the block obtains the steady-state equations by making the derivatives equal to zero:

$$V_{emf} = L_{sa} i_s \omega + L_{pa} i_p \omega$$

$$t_{elec} = k_v i_s = V_{emf} \frac{i_s}{\omega}$$

$$V = R_p i_p$$

$$R_p i_p = V_{emf} + (R_a + R_s) i_s$$

Then, it computes the steady-state currents and torque as follows:

$$t_{elec}(\omega, V) = \frac{V^2 (R_a - L_{pa} \omega) (R_a L_{pa} + R_s L_{pa} + R_p L_{sa})}{R_p^2 (R_a + R_s + L_{sa} \omega)^2}$$

$$i(\omega, V) = i_s + i_p = \frac{V}{R_p} + \frac{R_p V - L_{pa} V \omega}{R_p (R_a + R_s + L_{sa} \omega)}$$

## Faults

The Compound Motor block allows you to model three types of faults:

- Armature winding fault — The armature winding fails and becomes open circuit.
- Series field winding fault — The series field winding fails and becomes open circuit.
- Shunt field winding fault — The shunt field winding fails and becomes open circuit.

The block can trigger fault events:

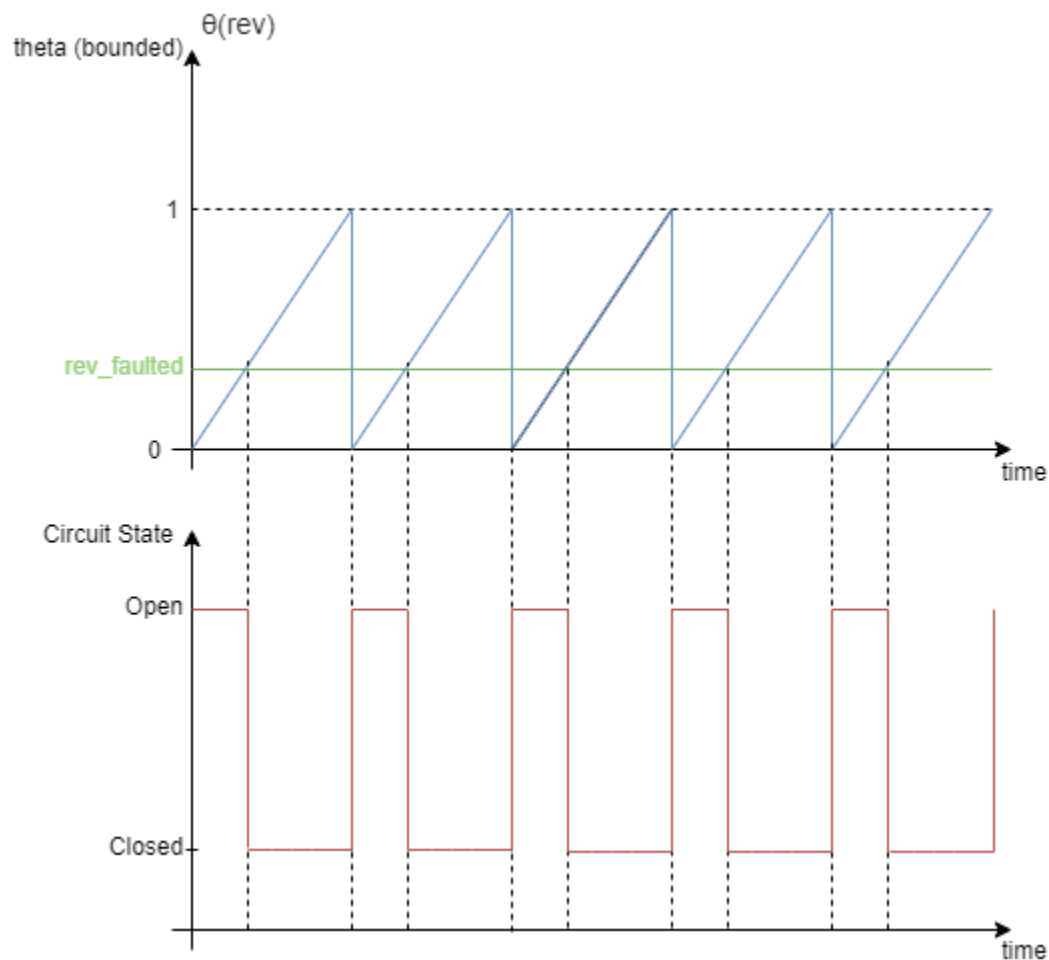
- At a specific time (temporal fault)
- When a current limit is exceeded for longer than a specific time interval (behavioral fault)

You can enable or disable these trigger mechanisms separately.

You can choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

If you set the **Enable armature winding open-circuit fault** parameter to On, the armature fails at the time specified by the **Simulation time for armature winding fault event** parameter for a temporal fault, or when the winding currents exceeds the value of the **Maximum permissible armature winding current** parameter for a behavioral fault. When the armature fails, the voltage source connected to this block observes an open circuit for a fraction of the total motor revolution, specified by the **Fraction of revolution during which armature is open-circuit** parameter. This figure illustrates the circuit state behavior and the open-circuit state (`rev_faulted`) for a revolution period:





### Model Thermal Effects

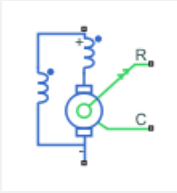
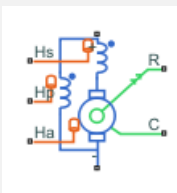
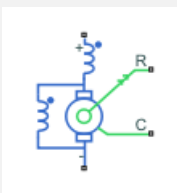
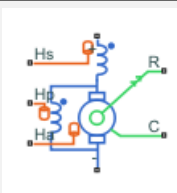
You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain thermal ports.
- `Show thermal port` — The block contains thermal conserving ports for the series field winding, the shunt field winding, and the armature.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Ports

The type, visibility, and location of the block ports depend on how you configure the **Electrical circuit topology** parameter in the **Configuration** tab, and if you expose the thermal ports:

Electrical circuit topology	Thermal ports	Block
Long-shunt	Hidden	
	Visible	
Short-shunt	Hidden	
	Visible	

**Conserving**

**+ – Positive terminal**  
electrical

Electrical conserving port associated with the compound motor positive terminal.

**- – Negative terminal**  
electrical

Electrical conserving port associated with the compound motor negative terminal.

**C – Motor case**  
mechanical

Mechanical rotational conserving port associated with the compound motor case.

**R – Motor rotor**  
mechanical

Mechanical rotational conserving port associated with the compound motor rotor.

**Hs – Series field winding thermal port**  
thermal

Thermal conserving port associated with the series field winding. For more information, see “Model Thermal Effects” on page 1-1735.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### Hp — Shunt field winding thermal port

thermal

Thermal conserving port associated with the shunt field winding.. For more information, see “Model Thermal Effects” on page 1-1735.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### Ha — Armature thermal port

thermal

Thermal conserving port associated with the armature. For more information, see “Model Thermal Effects” on page 1-1735.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

#### Modeling option — Whether to enable thermal ports

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

#### Configuration

#### Electrical circuit topology — Topology of electrical circuit

Long-shunt (default) | Short-shunt

Topology of the electrical circuit.

#### Shunt winding polarity — Shunt winding polarity

Cumulative (default) | Differential

Magnetic orientation of the shunt winding.

#### Steady-State

#### Steady-state parameterization — Block parameterization

By equivalent circuit parameters (default) | By rated, stall, and no-load datasheet parameters

Select one of the following methods for block parameterization:

- By equivalent circuit parameters — Provide electrical parameters for an equivalent circuit model of the motor.

- By rated, stall, and no-load datasheet parameters — Provide current and speed parameters that the block converts to an equivalent circuit model of the motor.

**Armature resistance,  $R_a$  — Armature resistance**

1.68 Ohm (default) | positive scalar

Resistance of the conducting portion of the motor.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By equivalent circuit parameters.

**Series field winding resistance,  $R_s$  — Series field winding resistance**

1.68 Ohm (default) | positive scalar

Resistance of the series field winding.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By equivalent circuit parameters.

**Shunt field winding resistance,  $R_p$  — Shunt field winding resistance**

303 Ohm (default) | positive scalar

Resistance of the shunt field winding.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By equivalent circuit parameters.

**Series field winding to armature back EMF constant,  $L_{sa}$  — Series field winding to armature back EMF constant**

0.69 H (default) | positive scalar

Series field winding to armature back EMF constant.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By equivalent circuit parameters.

**Shunt field winding to armature back EMF constant,  $L_{pa}$  — Shunt field winding to armature back EMF constant**

0.64 H (default) | positive scalar

Shunt field winding to armature back EMF constant.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By equivalent circuit parameters.

**Rated voltage — Rated voltage**

100 V (default) | positive scalar

Voltage at which the motor is rated to operate.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Rated speed — Rated speed**

1000 rpm (default) | positive scalar

Speed at which the motor is rated to operate.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Rated electrical power — Rated electrical power**

136 W (default) | positive scalar

Power at which the motor is rated to operate.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Rated efficiency (percent) — Rated efficiency**

85 (default) | positive scalar

Efficiency at which the motor is rated to operate, as a percentage.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Stall current — Zero speed current**

30 A (default) | positive scalar

Amount of current generated by the motor when the speed is approximately zero. This parameter must be greater than the value of the **No-load current** parameter.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**No-load speed — No-load speed**

4500 rpm (default) | positive scalar

Speed of the motor when no load is applied. This parameter must be greater than the value of the **Rated speed** parameter.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**No-load current — No-load current**

0.33 A (default) | positive scalar

Current of the motor when no load is applied.

**Dependencies**

To enable this parameter, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Ratio of shunt field winding resistance divided by series field winding resistance — Ratio of shunt field winding resistance divided by series field winding resistance**

190 (default) | nonnegative scalar

Ratio of the shunt field winding resistance divided by the series field winding resistance.

**Dependencies**

To enable this parameter, in the **Configuration** tab, set **Electrical circuit topology** to Short-shunt and, in the **Steady-state** tab, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Ratio of armature resistance divided by series field winding resistance — Ratio of armature resistance divided by series field winding resistance**

1.0 (default) | nonnegative scalar

Ratio of armature resistance divided by series field winding resistance.

**Dependencies**

To enable this parameter, in the **Configuration** tab, set **Electrical circuit topology** to Long-shunt and, in the **Steady-state** tab, set **Steady-state parameterization** to By rated, stall, and no-load datasheet parameters.

**Mechanical****Rotor inertia — Rotor inertia**

1e-4 kg\*m<sup>2</sup> (default) | scalar

Resistance of the rotor to change in motor motion. The value can be zero.

**Rotor damping — Rotor dissipated energy**

1e-6 N\*m/(rad/s) (default) | scalar

Energy dissipated by the rotor. The value can be zero.

**Dependencies**

To enable this parameter, in the **Steady-State** tab, set **Steady-state parameterization** to By equivalent circuit parameters.

**Electrical Dynamics****Series field winding inductance, L<sub>s</sub> — Series field winding inductance**

0.9 H (default) | positive scalar

Inductance of the series field winding.

**Shunt field winding inductance,  $L_p$  — Shunt field winding inductance**

0.85 H (default) | positive scalar

Inductance of the shunt field winding.

**Coefficient of mutual inductance between shunt-series windings,  $k_{sp}$  — Coefficient of mutual inductance between shunt-series windings**

0.8 (default) | nonnegative scalar between 0 and 1

Coefficient of mutual inductance between shunt-series windings.

**Thermal**

**Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default) | scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature. The default value is for copper.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Measurement temperature — Measurement temperature**

25 degC (default) | scalar

Temperature for which motor parameters are quoted.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Thermal masses, [ $M_s$   $M_p$   $M_a$ ] — Thermal masses**

[100, 100, 100] J/K (default) | vector

1-by-3 row vector that defines the thermal mass for the series and shunt field windings and armature winding. The thermal mass is the energy required to raise the temperature by one degree.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Faults**

**Enable armature winding open-circuit fault — Whether to enable armature winding faults**

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

**Fraction of revolution during which armature is open-circuit — Fraction of revolution during which armature is open-circuit**

1/3 (default) | scalar

Fraction of the revolution of the compound motor during which the armature winding is faulted and the voltage source observes an open circuit. For more information, see the “Faults” on page 1-388 section.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Faulted armature winding open-circuit conductance — Open-circuit conductance of faulted armature winding**

1e-5 S (default) | scalar

Open-circuit conductance when the armature winding is faulted.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Armature winding fault trigger — Fault trigger for armature winding**

Temporal (default) | Behavioral

Whether the trigger for the armature winding fault is temporal or behavioral.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Simulation time for armature winding fault event — Simulation time for armature winding fault event**

1 s (default) | scalar

Simulation time for the armature winding becomes faulted.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Temporal.

**Maximum permissible armature winding current — Maximum permissible armature winding current**

inf A (default)

Upper load-current threshold for open-circuit armature winding faults.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Behavioral.

**Time to fail when exceeding armature winding current — Time to fail when exceeding armature winding current**

1 s (default) | scalar

Amount of time that the winding current must continuously exceed the maximum permissible armature winding current before a behavioral fault is triggered.



**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Behavioral.

**Enable series field winding open-circuit fault — Whether to enable series field winding faults**

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

**Faulted series field winding open-circuit conductance — Open-circuit conductance of faulted field winding**

1e-5 S (default) | scalar

Open-circuit conductance when the series field winding is faulted.

**Dependencies**

To enable this parameter, set **Enable series field winding open-circuit fault** to On.

**Series field winding fault trigger — Fault trigger for series field winding**

Temporal (default) | Behavioral

Whether the trigger for the series field winding fault is temporal or behavioral.

**Dependencies**

To enable this parameter, set **Enable series field winding open-circuit fault** to On.

**Simulation time for series field winding fault event — Simulation time for series field winding fault event**

1 s (default) | scalar

Simulation time for the series field winding becomes faulted.

**Dependencies**

To enable this parameter, set **Enable series field winding open-circuit fault** to On and **Series field winding fault trigger** to Temporal.

**Maximum permissible series field winding current — Maximum permissible series field winding current**

inf A (default)

Upper load-current threshold for open-circuit series field winding faults.

**Dependencies**

To enable this parameter, set **Enable series field winding open-circuit fault** to On and **Series field winding fault trigger** to Behavioral.

**Time to fail when exceeding series field winding current — Time to fail when exceeding series field winding current**

1 s (default) | scalar

Amount of time that the winding current must continuously exceed the maximum permissible series field winding current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable series field winding open-circuit fault** to On and **Series field winding fault trigger** to Behavioral.

**Enable shunt field winding open-circuit fault – Whether to enable shunt field winding faults**

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

**Faulted shunt field winding open-circuit conductance – Open-circuit conductance of faulted shunt winding**

1e-5 S (default) | scalar

Open-circuit conductance when the shunt field winding is faulted.

**Dependencies**

To enable this parameter, set **Enable shunt field winding open-circuit fault** to On.

**Shunt field winding fault trigger – Fault trigger for shunt field winding**

Temporal (default) | Behavioral

Whether the trigger for the shunt field winding fault is temporal or behavioral.

**Dependencies**

To enable this parameter, set **Enable shunt field winding open-circuit fault** to On.

**Simulation time for shunt field winding fault event – Simulation time for shunt field winding fault event**

1 s (default) | scalar

Simulation time for the shunt field winding becomes faulted.

**Dependencies**

To enable this parameter, set **Enable shunt field winding open-circuit fault** to On and **Shunt field winding fault trigger** to Temporal.

**Maximum permissible shunt field winding current – Maximum permissible shunt field winding current**

inf A (default)

Upper load-current threshold for open-circuit shunt field winding faults.

**Dependencies**

To enable this parameter, set **Enable shunt field winding open-circuit fault** to On and **Shunt field winding fault trigger** to Behavioral.

**Time to fail when exceeding shunt field winding current — Time to fail when exceeding shunt field winding current**

1 s (default) | scalar

Amount of time that the winding current must continuously exceed the maximum permissible shunt field winding current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable shunt field winding open-circuit fault** to On and **Shunt field winding fault trigger** to Behavioral.

**Reporting when a fault occurs — Reporting when a fault occurs**

None (default) | Warn | Error

Simulation reporting when a fault occurs:

- None — Does not generate a warning or an error.
- Warn — Generates a warning.
- Error — The simulation stops and generates an error.

**Dependencies**

To enable this parameter, set at least one of these parameters to On:

- **Enable armature winding open-circuit fault**
- **Enable series field winding open-circuit fault**
- **Enable shunt field winding open-circuit fault**

## Version History

Introduced in R2021a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

DC Motor

## Constant Current Load

Constant current load for DC or AC supply

**Library:** Simscape / Electrical / Passive



### Description

The Constant Current Load block implements a constant current load for a DC or AC supply.

If you set the **Load type** parameter to DC:

- The consumed current of this block is equal to the value of the **Consumed current** parameter as long as the voltage from the DC supply is equal to or greater than the value specified for the **Minimum supply voltage** parameter.
- When the voltage from the DC supply drops below **Minimum supply voltage**, the load behaviour changes and the block models a resistive load. If the supply voltage becomes negative, the block models a small open-circuit conductance.

To ensure smooth transitions between these behaviours, the block uses a third-order polynomial spline with continuous derivatives. You can specify the width of this transition using the **Transition voltage width** parameter.

If you set the **Load type** parameter to AC:

- The root mean square consumed current of this block is equal to the value of the **Consumed current (RMS)** parameter as long as the voltage from the AC supply is equal to or greater than the value specified for the **Minimum supply voltage (RMS)** parameter.
- When the voltage from the AC supply drops below the **Minimum supply voltage (RMS)** parameter, the load behaviour changes and the block models a load with constant resistance.

### Equations

If you set the **Load type** parameter to AC, the block calculates the peak voltage,  $V_{pk}$ , through harmonic approximation of the instantaneous voltage by using a one-period-integrated Fourier transform:

$$\text{Re} = f_0 \int v \sin(2\pi f_0 t)$$

$$\text{Im} = f_0 \int v \sin\left(2\pi f_0 t + \frac{\pi}{2}\right)$$

$$V_{pk} = 2\sqrt{\text{Re}^2 + \text{Im}^2}$$

During the first period, the peak voltage is equal to  $\theta$  V. The current is defined by this equation:

$$i(t) = \frac{V(t)}{R_{equiv}}$$

where  $R_{equiv}$  is the equivalent resistance and depends on the value of the **Minimum supply voltage (RMS)** parameter.

If the voltage from the three-phase supply is greater than the value specified for the **Minimum supply voltage (RMS)** parameter, the equivalent resistance is defined by:

$$R_{equiv} = \frac{\frac{V_{pk}}{\sqrt{2}}}{I_{RMSconsumed}},$$

where  $V_{pk}$  is the voltage peak magnitude and  $I_{RMSconsumed}$  is the value of the **Consumed current (RMS)** parameter.

If the voltage from the three-phase supply is less than the value specified for the **Minimum supply voltage (RMS)** parameter, the equivalent resistance is defined by:

$$R_{equiv} = \frac{V_{RMSmin}}{I_{RMSconsumed}},$$

where  $V_{RMSmin}$  is the value of the **Minimum supply voltage (RMS)** parameter.

## Faults

The Constant Current Load block allows you to model an electrical fault as an open circuit. The block can trigger fault events either:

- At a specific time
- When a power limit is exceeded for longer than a specific time interval

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Load-Flow Analysis

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the constant current load positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the constant current load negative terminal.

## Parameters

### Main

#### **Load type — Load type**

DC (default) | AC

Type of load. Select DC to use this load with a DC supply or AC to use this load with an AC supply.

#### **Consumed current — Current consumed**

1 A (default) | scalar

Current consumed by the block.

#### **Dependencies**

To enable this parameter, set **Load Type** to DC.

#### **Minimum supply voltage — Minimum DC supply voltage**

5 V (default) | scalar

Minimum voltage the DC supply needs to give the block to act as a constant current load.

#### **Dependencies**

To enable this parameter, set **Load Type** to DC.

#### **Reverse conductance — Reverse conductance**

1e-6 S (default) | scalar

Value of the reverse conductance when the supply voltage becomes negative.

#### **Dependencies**

To enable this parameter, set **Load Type** to DC.

#### **Transition voltage width — Transition voltage width**

1e-3 V (default) | scalar

Transition voltage width used to ensure smooth transitions between block behaviours when the supply voltage drops below the value of the **Minimum supply voltage** parameter or becomes negative.

#### **Dependencies**

To enable this parameter, set **Load Type** to DC.

#### **Consumed current (RMS) — Root mean square consumed current**

100 A (default) | scalar

Root mean square current consumed by the block.

#### **Dependencies**

To enable this parameter, set **Load Type** to AC.

#### **Rated frequency — Rated frequency**

60 Hz (default) | scalar

Rated frequency.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Minimum supply voltage (RMS) — Minimum AC supply voltage

200 V (default) | scalar

Minimum voltage, in RMS, that the AC supply must provide for the block to act as a constant current load.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Faults

#### Enable open-circuit fault — Whether to enable faults modeling

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

#### Open-circuit conductance — Open-circuit conductance

1e-4 S (default) | scalar

Conductance during the open-circuit state.

#### Dependencies

To enable this parameter, set **Enable open-circuit fault** to On.

#### Reporting when a fault occurs — Whether to issue an assertion when a fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

#### Dependencies

To enable this parameter, set **Enable open-circuit fault** to On.

#### Fault trigger — Trigger mechanism

Temporal (default) | Behavioral

Whether to trigger the fault based on temporal or behavioral conditions.

#### Dependencies

To enable this parameter, set **Enable open-circuit fault** to On.

#### Simulation time for fault event — Time before entering faulted state

1 s (default) | scalar

Simulation time when the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Temporal.

**Maximum permissible power — Power threshold to fault transition**

inf W (default)

Maximum permissible power value. If the power exceeds this value for longer than the **Time to fail when exceeding power** parameter value, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Behavioral.

**Time to fail when exceeding power — Maximum length of time the power exceeds the threshold**

1 s (default)

Maximum length of time that the power can exceed the maximum permissible value without triggering the fault.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Behavioral.

## Version History

Introduced in R2021a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Constant Current Load (Three-Phase)

**Topics**

“Perform a Load-Flow Analysis Using Simscape Electrical”

“Frequency and Time Simulation Mode”



## Constant Current Load (Three-Phase)

Three-phase constant current load

**Library:** Simscape / Electrical / Passive



### Description

The Constant Current Load (Three-Phase) block implements a constant current load for a three-phase supply.

This block consumes a current equal to the value of the **Consumed current (RMS)** parameter as long as the voltage from the three-phase supply is equal to or greater than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter.

When the voltage from the three-phase supply drops below the value of **Minimum supply voltage (phase-to-phase RMS)**, the load behaviour changes and the block models a load with constant resistance.

### Equations

If the three-phase voltages are balanced, the phase-to-phase RMS voltage is defined by this equation:

$$V_{lineRMS} = \sqrt{(v_a - v_{dc})^2 + (v_b - v_{dc})^2 + (v_c - v_{dc})^2},$$

$$\text{where } v_{dc} = \frac{v_a + v_b + v_c}{3}.$$

The currents for each of the three phases are defined by this equation:

$$i_k(t) = \frac{v_k(t)}{R_{equiv}},$$

where  $R_{equiv}$  is the equivalent resistance and depends on value of the **Minimum supply voltage (phase-to-phase RMS)** parameter.

If the voltage from the three-phase supply is greater than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter, the equivalent resistance is defined by:

$$R_{equiv} = \frac{\frac{v_{pk}}{\sqrt{2}}}{I_{RMSconsumed}},$$

where  $v_{pk}$  is the voltage peak magnitude and  $I_{RMSconsumed}$  is the value of the **Consumed current (RMS)** parameter.

If the voltage from the three-phase supply is less than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter, the equivalent resistance is defined by:

$$R_{equiv} = \frac{v_{phRMSmin}}{I_{RMSconsumed}},$$

where  $v_{phRMSmin} = \frac{V_{lineRMSmin}}{\sqrt{3}}$  is the phase-to-ground RMS minimum voltage.

## Faults

The Constant Current Load (Three-Phase) block allows you to model an electrical fault as an open circuit. The block can trigger fault events either:

- At a specific time
- When a power limit is exceeded for longer than a specific time interval

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Load-Flow Analysis

If the block is in a network that is compatible with frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Ports

### Conserving

~ — **Three-phase voltage**  
electrical

Expandable three-phase electrical port associated with the three-phase voltage.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Main****Consumed current (RMS) — Root mean square consumed current**

1 A (default) | scalar

Root mean square current consumed by the block.

**Minimum supply voltage (phase-to-phase RMS) — Minimum supply voltage**

10 V (default) | scalar

Minimum voltage, in phase-to-phase RMS, that the supply must provide for the block to act as a current load.

**Faults****Enable open-circuit fault — Whether to enable faults modeling**

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

**Open-circuit conductance — Open-circuit conductance**

1e-4 S (default) | scalar

Conductance during the open-circuit state.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On.

**Reporting when a fault occurs — Whether to issue an assertion when a fault occurs**

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On.

**Fault trigger — Trigger mechanism**

Temporal (default) | Behavioral

Whether to trigger the fault based on temporal or behavioral conditions.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On.

**Simulation time for fault event — Time before entering faulted state**

1 s (default) | scalar

Simulation time when the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Temporal.

**Maximum permissible power — Power threshold to fault transition**

inf W (default)

Maximum permissible power value. If the power exceeds this value for longer than the **Time to fail when exceeding power** parameter value, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Behavioral.

**Time to fail when exceeding power — Maximum length of time the power exceeds the threshold**

1 s (default)

Maximum length of time that the power can exceed the maximum permissible value without triggering the fault.

**Dependencies**

To enable this parameter, set **Enable open-circuit fault** to On and **Fault trigger** to Behavioral.

## Version History

Introduced in R2021a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Constant Current Load

**Topics**

“Perform a Load-Flow Analysis Using Simscape Electrical”

“Frequency and Time Simulation Mode”

# Constant Power Load

Constant power load for DC or AC supply

**Library:** Simscape / Electrical / Passive



## Description

The Constant Power Load block implements a constant power load for a DC or AC supply.

If you set the **Load type** parameter to DC:

- The block outputs a nominal rated power as long as the voltage from the DC supply is equal to or greater than the value specified for the **Minimum supply voltage** parameter.
- When the voltage from the DC supply drops below the **Minimum supply voltage**, the load behaviour changes and the block acts as a constant resistance. If the supply voltage becomes negative, the block acts as a constant reverse conductance.

To ensure smooth transitions between these behaviours, the block uses a third-order polynomial spline with continuous derivatives. You can specify the width of this transition using the **Transition voltage width** parameter.

If you set the **Load type** parameter to AC:

- The block outputs a nominal rated power as long as the voltage from the AC supply is equal to or greater than the value specified for the **Minimum supply voltage (RMS)** parameter.
- When the voltage from the AC supply drops below the **Minimum supply voltage (RMS)**, the load behaviour changes and the block models a load with constant impedance.

For more information about the equations for the AC load type in a three-phase system, see the “Equations” on page 1-307 section of the Constant Power Load (Three-Phase) block.

## Faults

The Constant Power Load block allows you to model an electrical fault as an open circuit. The block can trigger fault events at a specific time.

You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Load-Flow Analysis

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the constant power load positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the constant power load negative terminal.

## Parameters

### Main

#### Load type – Load type

DC (default) | AC

Type of load. Select DC to use this load with a DC supply or AC to use this load with an AC supply.

#### Power consumed – Power consumed

100 W (default) | scalar

Nominal power consumed by the block.

#### Dependencies

To enable this parameter, set **Load Type** to DC.

#### Minimum supply voltage – Minimum DC supply voltage

20 V (default) | scalar

Minimum voltage the DC supply needs to give the block to act as a constant power load.

#### Dependencies

To enable this parameter, set **Load Type** to DC.

#### Reverse conductance – Reverse conductance

1e-6 S (default) | scalar

Value of the reverse conductance when the supply voltage becomes negative.

#### Dependencies

To enable this parameter, set **Load Type** to DC.

#### Transition voltage width – Transition voltage width

1e-3 V (default) | scalar

Transition voltage width used to ensure smooth transitions between the block's behaviours when the supply voltage drops below the **Minimum supply voltage** or becomes negative.

#### Dependencies

To enable this parameter, set **Load Type** to DC.

#### Active power consumed — Active power consumed

500 W (default) | scalar

Active power consumed by the block.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Reactive power consumed — Reactive power consumed

0 V\*A (default) | scalar

Reactive power consumed by the block.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Rated frequency — Rated frequency

60 Hz (default) | scalar

Rated frequency.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Minimum supply voltage (RMS) — Minimum AC supply voltage

200 V (default) | scalar

Minimum voltage, in RMS, the AC supply needs to give the block to act as a constant power load.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Faults

#### Enable open circuit fault — Whether to enable faults modeling

Off (default) | On

Select On to enable faults modeling and expose the associated parameters in the **Faults** section to select the reporting method and specify the time at which the fault triggers.

#### Reporting when a fault occurs — Whether to issue an assertion when a fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.

- Warn — The block issues a warning.
- Error — Simulation stops with an error.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Simulation time for fault event — Time before entering faulted state**

1 s (default) | scalar

Simulation time when the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Open-circuit conductance — Open-circuit conductance**

1e-4 S (default) | scalar

Conductance during the open-circuit state.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

## Version History

Introduced in R2020a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Constant Power Load (Three-Phase)

**Topics**

“Perform a Load-Flow Analysis Using Simscape Electrical”

“Frequency and Time Simulation Mode”



# Constant Power Load (Three-Phase)

Three-phase constant power load

**Library:** Simscape / Electrical / Passive



## Description

The Constant Power Load (Three-Phase) block implements a constant power load for a three-phase supply.

The block outputs a nominal rated power as long as the voltage from the three-phase supply is equal to or greater than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter.

When the voltage from the three-phase supply drops below the value of **Minimum supply voltage (phase-to-phase RMS)**, the load behaviour changes and the block models a load with constant impedance.

## Equations

When the voltage from the three-phase supply is greater than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter, the block acts in constant PQ mode.

The active and reactive power are defined by:

$$P = \frac{3}{2} V_{pk} I_{pk} \cos(\varphi_V - \varphi_I)$$

$$Q = \frac{3}{2} V_{pk} I_{pk} \sin(\varphi_V - \varphi_I)$$

where:

- $V_{pk}$  is the voltage peak magnitude.
- $\varphi_I = \varphi_V - \text{atan}\left(\frac{Q}{P}\right)$  is the phase of the current.
- $I_{pk} = \frac{2}{3} \frac{P}{V_{pk} \cos(\varphi_V - \varphi_I)}$  is the current peak magnitude.

When the voltage from the three-phase supply is less than or equal to the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter, the block acts in constant Z mode.

The active and reactive power are defined by:

$$P = \frac{3}{2} V_{pk}^2 \frac{R}{R^2 + X^2}$$

$$Q = \frac{3}{2} V_{pk}^2 \frac{X}{R^2 + X^2}$$

where:

- $R = V_{line\ RMS_{min}} \frac{P}{P^2 + Q^2}$  is the constant resistance.
- $X = V_{line\ RMS_{min}} \frac{Q}{P^2 + Q^2}$  is the constant reactance.
- $V_{line\ RMS_{min}}$  is the **Minimum supply voltage (phase-to-phase RMS)**.

### Load-Flow Analysis

If the block is in a network that is compatible with frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

### Limitations

If you connect this block to an inductive node or a current source, you must provide a large parasitic.

### Ports

#### Conserving

##### ~ — Three-phase voltage

electrical

Expandable three-phase electrical port associated with the three-phase voltage.

### Parameters

#### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Main****Active power consumed — Active power consumed**

1 kW (default) | scalar

Active power consumed by the block.

**Reactive power consumed — Reactive power consumed**

0 A\*kV (default) | scalar

Reactive power consumed by the block.

**Minimum supply voltage (phase-to-phase RMS) — Minimum supply voltage**

100 V (default) | scalar

Minimum voltage, in phase-to-phase RMS, that the supply needs to give the block to act as a constant power load.

**Faults****Enable open circuit fault — Whether to enable faults modeling**

Off (default) | On

Select On to enable faults modeling and expose the associated parameters in the **Faults** section, which enables you to select the reporting method and specify the time at which the fault triggers.**Reporting when a fault occurs — Whether to issue an assertion when a fault occurs**

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — The simulation stops with an error.

**Dependencies**To enable this parameter, set **Enable open circuit fault** to On.**Simulation time for fault event — Time at which fault is triggered**

1 s (default) | scalar

Simulation time when the block enters the faulted state.

**Dependencies**To enable this parameter, set **Enable open circuit fault** to On.**Open-circuit conductance — Open-circuit conductance**

1e-4 S (default) | scalar

Conductance during the open-circuit state.

**Dependencies**To enable this parameter, set **Enable open circuit fault** to On.

## **Version History**

**Introduced in R2020b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Power Sensor (Three-Phase) | Constant Power Load

## **Topics**

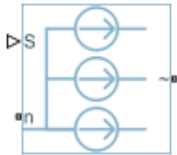
“Perform a Load-Flow Analysis Using Simscape Electrical”

“Frequency and Time Simulation Mode”

## Controlled Current Source (Three-Phase)

Ideal three-phase controlled current source

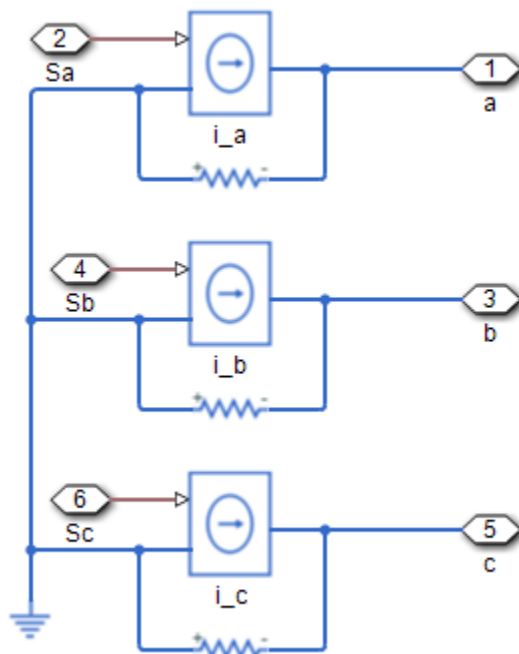
**Library:** Simscape / Electrical / Sources



### Description

The Controlled Current Source (Three-Phase) block represents an ideal three-phase current source that is powerful enough to maintain the specified current through it regardless of the voltage across it. The output currents are  $[i_a \ i_b \ i_c] = S$ , where  $S$  is a vector containing the numerical values presented at the physical signal port.

The figure shows the equivalent circuit for the expanded implementation of the Controlled Current Source (Three-Phase) block.



To use the Controlled Current Source (Three-Phase) block as an abstracted current controller in an electrical drive, connect the conserving ports for the output current directly to the machine.

To choose the input type of this block, set the **Input type** parameter to Instantaneous or Sinusoidal magnitude and phase shift. If you select Instantaneous, this block works only

in time simulation mode. If you select `Sinusoidal magnitude and phase shift`, this block works in both time and frequency-time simulation modes. For more information, see “Frequency and Time Simulation Mode”.

To access the implementation with an expanded, three-phase port, double-click the block and set the **Modeling option** parameter to `Expanded three-phase ports`.

## Ports

### Input

#### **S — Control**

physical signal

Physical signal input port associated with the control signal.

#### **Dependencies**

To enable this port, set **Input type** to `Instantaneous`.

#### **I — RMS current**

physical signal

Physical signal input port associated with the RMS current.

#### **Dependencies**

To enable this port, set **Input type** to `Sinusoidal magnitude and phase shift`.

#### **ph — Phase shift**

physical signal

Physical signal input port associated with the phase shift, in radians.

#### **Dependencies**

To enable this port, set **Input type** to `Sinusoidal magnitude and phase shift`.

### Conserving

#### **n — Neutral**

electrical

Electrical conserving port associated with the neutral phase.

#### **~ — Three-phase current**

electrical

Expandable three-phase port.

#### **Dependencies**

To enable this port, set **Modeling option** to `Composite three-phase ports`.

#### **a — a-phase current**

electrical

Electrical conserving port associated with the *a*-phase output current.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports.

**b – b-phase current**

electrical

Electrical conserving port associated with the *b*-phase output current.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports.

**c – c-phase current**

electrical

Electrical conserving port associated with the *c*-phase output current.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports.

## Parameters

**Modeling option – Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Input type – Type of input**

Instantaneous (default) | Sinusoidal magnitude and phase shift

Type of physical signal input. Choose between:

- **Instantaneous** — The output currents,  $[i_a i_b i_c]$ , are equal to the values of the input port **S**.
- **Sinusoidal magnitude and phase shift** — The output currents are equal to:

$$i_a = (\sqrt{2} I) \sin(2\pi F_{rated} + ph)$$

$$i_b = (\sqrt{2} I) \sin\left(2\pi F_{rated} + ph - 2\frac{\pi}{3}\right)$$

$$i_c = (\sqrt{2} I) \sin\left(2\pi F_{rated} + ph + 2\frac{\pi}{3}\right)$$

**Rated electrical frequency – Rated electrical frequency**

60 Hz | scalar

Rated electrical frequency. The value of this parameter must be equal to the value of the rated electrical frequencies of other sinusoidal sources in your model, if present.

**Dependencies**

To enable this parameter, set **Input type** to Sinusoidal magnitude and phase shift.

**Version History**

**Introduced in R2018b**

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Battery | Current Source (Three-Phase) | DC Current Source | Voltage Source (Three-Phase)

**Topics**

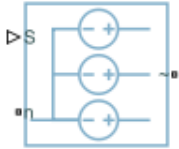
“Expand and Collapse Three-Phase Ports on a Block”



# Controlled Voltage Source (Three-Phase)

Ideal three-phase controlled voltage source

**Library:** Simscape / Electrical / Sources



## Description

The Controlled Voltage Source (Three-Phase) block represents an ideal three-phase voltage source that maintains the specified voltage regardless of the current through it.

To choose the input type of this block, set the **Input type** parameter to `Instantaneous` or `Sinusoidal magnitude and phase shift`. If you select `Instantaneous`, this block works only in time simulation mode. If you select `Sinusoidal magnitude and phase shift`, this block works in both time and frequency-time simulation modes. For more information, see “Frequency and Time Simulation Mode”.

To access the implementation with an expanded, three-phase port, double-click the block and set the **Modeling option** parameter to `Expanded three-phase ports`.

## Ports

### Input

#### S — Control

physical signal | vector

Physical signal input port associated with the control signal. The output voltages are  $[v_a \ v_b \ v_c] = S$ . For an ideal sinusoidal source, input three sine waves that have the requisite amplitude and phase delays of  $\left[0 \ -\frac{2\pi}{3} \ \frac{2\pi}{3}\right]$ .

#### Dependencies

To enable this port, set **Input type** to `Instantaneous`.

#### Vt — Phase-to-phase RMS voltage

physical signal

Physical signal input port associated with the phase-to-phase RMS voltage.

#### Dependencies

To enable this port, set **Input type** to `Sinusoidal magnitude and phase shift`.

#### ph — Phase shift

physical signal

Physical signal input port associated with the phase shift, in radians.

**Dependencies**

To enable this port, set **Input type** to `Sinusoidal magnitude and phase shift`.

**Conserving****n — Neutral**

electrical

Electrical conserving port associated with the neutral phase.

**~ — Three-phase voltage**

electrical

Expandable three-phase port associated with the output voltage.

**Dependencies**

To enable this port, set **Modeling option** to `Composite three-phase ports`.

**a — a-phase voltage**

electrical

Electrical conserving port associated with the *a*-phase output voltage.

**Dependencies**

To enable this port, set **Modeling option** to `Expanded three-phase ports`.

**b — b-phase voltage**

electrical

Electrical conserving port associated with the *b*-phase output voltage.

**Dependencies**

To enable this port, set **Modeling option** to `Expanded three-phase ports`.

**c — c-phase voltage**

electrical

Electrical conserving port associated with the *c*-phase output voltage.

**Dependencies**

To enable this port, set **Modeling option** to `Expanded three-phase ports`.

**Parameters****Modeling option — Whether to model composite or expanded three-phase ports**

`Composite three-phase ports (default)` | `Expanded three-phase ports`

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Input type — Type of input

Instantaneous (default) | Sinusoidal magnitude and phase shift

Type of physical signal input. Choose between:

- **Instantaneous** — The output voltages,  $[v_a v_b v_c]$ , are equal to the values of the input port **S**.
- **Sinusoidal magnitude and phase shift** — The output voltages are equal to:

$$v_a = \left(\sqrt{2} \frac{V_t}{\sqrt{3}}\right) \sin(2\pi F_{rated} + ph)$$

$$v_b = \left(\sqrt{2} \frac{V_t}{\sqrt{3}}\right) \sin\left(2\pi F_{rated} + ph - 2\frac{\pi}{3}\right)$$

$$v_c = \left(\sqrt{2} \frac{V_t}{\sqrt{3}}\right) \sin\left(2\pi F_{rated} + ph + 2\frac{\pi}{3}\right)$$

### Rated electrical frequency — Rated electrical frequency

60 Hz | scalar

Rated electrical frequency. The value of this parameter must be equal to the value of the rated electrical frequencies of other sinusoidal sources in your model, if present.

#### Dependencies

To enable this parameter, set **Input type** to Sinusoidal magnitude and phase shift.

## Version History

Introduced in R2019a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Controlled Current Source (Three-Phase) | Voltage Source | Exponential Voltage Source

## Controlled PWM Voltage

Pulse-width modulated voltage source

**Library:** Simscape / Electrical / Integrated Circuits



### Description

The Controlled PWM Voltage block represents a pulse-width modulated (PWM) voltage source. You can model electrical or physical signal input ports by setting the **Modeling option** parameter to either:

- **Electrical input ports** — The block calculates the duty cycle based on the reference voltage across its **ref+** and **ref-** ports. This option is the default.
- **PS input** — Specify the duty cycle value directly by using an input physical signal port.

If you set **Modeling option** to **Electrical input ports**, the demanded duty cycle is

$$100 * \frac{V_{ref} - V_{min}}{V_{max} - V_{min}} \text{ percent}$$

where:

- $V_{ref}$  is the reference voltage across the **ref+** and **ref-** ports.
- $V_{min}$  is the minimum reference voltage.
- $V_{max}$  is the maximum reference voltage.

The value of the **Output voltage amplitude** parameter determines amplitude of the output voltage.

At time zero, the pulse is initialized as high, unless the **Pulse delay time** parameter is greater than zero, or the demanded duty cycle is zero.

You can use parameters **Pulse delay time** and **Pulse width offset** to add a small turn-on delay and a small turn-off advance. This can be useful when fine-tuning switching times so as to minimize switching losses.

In PWM mode, the block has two options for the type of switching event when moving between output high and output low states:

- **Asynchronous** — **Best for variable-step solvers** — Asynchronous events are better suited to variable step solvers, because they require fewer simulation steps for the same level of accuracy. In asynchronous mode the PWM switching events generate zero crossings, and therefore switching times are always determined accurately, regardless of the simulation maximum step size.
- **Discrete-time** — **Best for fixed-step solvers** — Discrete-time events are better suited to fixed-step operation, because then the switching events are always synchronized with the simulation step. Using an asynchronous implementation with fixed-step solvers may sometimes

result in events being up to one simulation step late. For more information, see “Simulating with Fixed Time Step — Local and Global Fixed-Step Solvers”.

If you use a fixed-step or local solver and the discrete-time switching event type, the following restrictions apply to the **Sample time** parameter value:

- The sample time must be a multiple of the simulation step size.
- The sample time must be small compared to the PWM period, to ensure sufficient resolution.

### Assumptions and Limitations

The model is based on the following assumptions:

- The REF output of this block is floating, it is not tied to the Electrical Reference. One consequence of this is that if you connect the PWM and REF electrical ports directly to the PWM and REF electrical ports of an H-bridge or a gate driver, you must attach an Electrical Reference block to the REF connection line.
- Do not connect the Controlled PWM block directly to a semiconductor gate, because this omits the gate driver output impedance that determines switching dynamics. Use a Gate Driver or a Half-Bridge Driver block to set the gate-source or the gate-emitter voltage.
- Do not use the Controlled PWM block to drive a motor block directly. A PWM motor driver goes open circuit in between pulses. Use the H-Bridge block to drive a motor block.
- When driving a motor via the H-Bridge block, set the **Simulation mode** parameter to **Averaged** to speed up simulations. You must also set the **Simulation mode** parameter of the H-Bridge block to **Averaged** mode. This applies the average of the demanded PWM voltage to the motor. The **Averaged** mode assumes that the impedance of the motor inductive term is small at the **PWM frequency**. To verify this assumption, run the simulation using the **PWM** mode and compare the results to those obtained from using the **Averaged** mode.
- If you are linearizing your model, set the **Simulation mode** parameter to **Averaged** and ensure that you have specified the operating point of the block correctly. You can only linearize the block for inputs corresponding to a duty cycle greater than zero and less than 100 percent.
- When you use this block in **PWM** mode with the **Use local solver option** selected in the Solver Configuration block, set the **Switching event type** parameter to **Discrete-time – Best for fixed-step solvers**. Using the **Asynchronous – Best for variable-step solvers** option in this situation may produce inaccuracies, because simulation with the local solver implies fixed step, and the PWM events will not always coincide precisely with the simulation steps. This results in PWM events sometimes occurring one simulation step late.

## Ports

### Input

#### **u** — Control signal, unitless

physical signal

Input physical signal that specifies the duty cycle.

### Dependencies

To enable this port, set **Modeling option** to **PS** input.

## Conserving

### **ref+ — Positive terminal**

electrical

Positive electrical reference voltage.

#### **Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

### **ref- — Negative terminal**

electrical

Negative electrical reference voltage

#### **Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

### **PWM — Pulse-width modulated signal**

electrical

Electrical conserving port associated with the pulse-width modulated signal.

### **REF — Floating zero volt reference**

electrical

Electrical conserving port associated with the floating zero volt reference.

## Parameters

### **Modeling option — Whether to model electrical or physical signal input ports**

Electrical input ports (default) | PS input

Whether to model electrical or physical signal input ports.

#### **PWM**

### **PWM frequency — PWM output signal frequency**

1000 Hz (default)

Frequency of the PWM output signal.

### **Pulse delay time — Turn-on delay**

0 s (default)

The pulse train does not start until the simulation time is equal to the **Pulse delay time**. You can specify a small value for **Pulse delay time** to fine-tune switching times and ensure that an off-going device is fully off before the on-going device starts to turn on. You can also use larger delay times, for example, if you need the pulse train to start only after a number of cycles. The value you provide must be greater than or equal to zero.

#### **Dependencies**

Enabled when the **Simulation mode** parameter is set to PWM.

**Pulse width offset — Lengthens or shortens the pulse**

0 s (default)

The demanded pulse width as defined by the product of the demanded duty cycle and one over the pulse frequency can be offset by the value you provide for **Pulse width offset**. A positive value acts to lengthen the pulse by a fixed amount. A negative value acts to shorten the pulse. You can use this parameter, along with the **Pulse delay time**, to fine-tune switching times so as to minimize switching losses in some circuits.

**Dependencies**

Enabled when the **Simulation mode** parameter is set to PWM.

**Minimum pulse width — Minimum length of pulse**

0 s (default)

The minimum pulse length, based on the internal clock or defined programmatically, to protect the device being driven. The value you provide must be greater than or equal to zero.

**Dependencies**

Enabled when the **Simulation mode** parameter is set to PWM.

**Simulation mode — Select type of output voltage**

PWM (default) | Averaged

Select the type of output voltage:

- **PWM** — Produces a pulse-width modulated signal.
- **Averaged** — The output is a constant whose value is equal to the average value of the PWM signal.

**Switching event type — Select type of switching event**

Asynchronous — Best for variable-step solvers (default) | Discrete-time — Best for fixed-step solvers

Select the switching event type when moving between output high and output low states:

- **Asynchronous — Best for variable-step solvers** — This option is more efficient for desktop simulation with variable-step solvers, because it requires fewer simulation steps for the same level of accuracy.
- **Discrete-time — Best for fixed-step solvers** — Use with fixed-step solvers, including the local solver. For more information, see “Simulating with Fixed Time Step — Local and Global Fixed-Step Solvers”.

**Dependencies**

Enabled when the **Simulation mode** parameter is set to PWM.

**Sample time — Discrete sampling time**

1e-6 s (default)

The time between updates of the block output state. The sample time must be a multiple of the simulation step size. In order for the PWM control to have sufficient resolution, set the sample time to less than one hundredth of the PWM period. (The PWM period is one over the PWM frequency.)

**Dependencies**

Enabled when the **Switching event type** parameter is set to Discrete-time – Best for fixed-step solvers.

**Input Scaling****Input voltage for 0% duty cycle – Vmin**

0 V (default)

Value of the input voltage at which the PWM signal has a 0% duty cycle.

**Dependencies**

To enable this parameter, set **Modeling option** to Electrical input ports.

**Input voltage for 100% duty cycle – Vmax**

5 V (default)

Value of the input voltage at which the PWM signal has a 100% duty cycle.

**Dependencies**

To enable this parameter, set **Modeling option** to Electrical input ports.

**Input value for 0% duty cycle – Minimum signal value**

0 (default)

Value of the input signal at which the PWM signal has a 0% duty cycle.

**Dependencies**

To enable this parameter, set **Modeling option** to PS input.

**Input value for 100% duty cycle – Maximum signal value**

1 (default)

Value of the input signal at which the PWM signal has a 100% duty cycle

**Dependencies**

To enable this parameter, set **Modeling option** to PS input.

**Output Voltage****Output voltage amplitude – Signal amplitude for high output**

5 V (default)

Amplitude of the PWM signal when the output is high.

**Version History**

**Introduced in R2008b**



## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Gate Driver | H-Bridge | Half-Bridge Driver

# Controller LCFB1

Load controller for turbine-governor model

**Library:** Simscape / Electrical / Control / Turbine-Governors

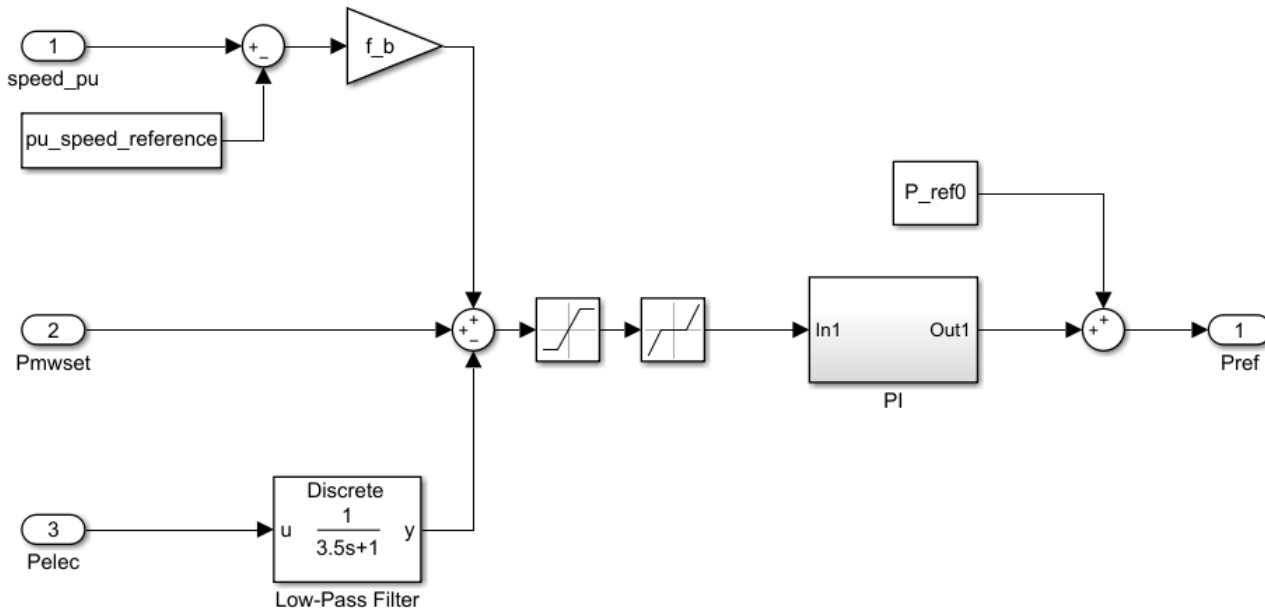


## Description

The Controller LCFB1 block models a model LCFB1 turbine load controller. You can use this load controller with any turbine-governor model, such as the Governor Type 1 or Governor Type 3 blocks.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the controller for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the controller for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the block:



## Ports

### Input

**speed\_pu** — Speed load  
scalar

Speed load, specified as a scalar.

Data Types: `single` | `double`

### **Pmwset — Power setpoint**

scalar

Power setpoint value, specified as a scalar.

Data Types: `single` | `double`

### **Pelec — Electrical power input**

scalar

Electrical power input, specified as a scalar.

Data Types: `single` | `double`

## **Output**

### **Pref — Bias**

scalar

Bias to apply to a turbine-governor speed load reference, returned as a scalar.

Data Types: `single` | `double`

## **Parameters**

### **Speed reference (pu) — Speed reference**

1 (default) | scalar

Per-unit speed reference.

### **Initial power reference, P\_ref0 (pu) — Initial power reference**

0.45 (default) | scalar

Per-unit power reference at the start of the simulation.

### **Controller deadband (pu) — Deadband of controller**

0 (default) | scalar

Per-unit deadband of the controller.

### **Maximum error, e\_max (pu) — Maximum error**

0.1 (default) | scalar

Maximum possible error.

### **Frequency bias gain, f\_b (pu) — Gain of frequency bias**

0 (default) | scalar

Per-unit gain of the frequency bias.

### **Proportional gain, K\_p (pu) — Proportional gain for PI control**

0 (default) | scalar

Per-unit proportional gain for PI control.

**Integral gain,  $K_i$  (pu/s) – Integral gain for PI control**

0.1 (default) | scalar

Per-unit integral gain for PI control.

**Power transducer time constant,  $T_{pelec}$  (s) – Time constant of power transducer**

3.5 (default) | scalar

Time constant of the power transducer, in seconds. The power transducer is represented by the block labeled "Low-Pass Filter" in the diagram.

**Maximum output,  $I_{rmax}$  (pu) – Upper limit of the output**

0.035 (default) | scalar

Per-unit upper saturation limit for the output.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see "What Is Sample Time?" and "Specify Sample Time".

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see "Promote Block Parameters on a Mask".

## Version History

Introduced in R2020a

## References

[1] Task Force on Turbine-Governor Modeling, *Dynamic models for turbine-governors in power system studies*, IEEE Power Energy Society, January 2013.

## See Also

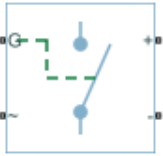
### Blocks

Governor Type 1 | Governor Type 3

# Converter (Three-Phase)

Controller-driven bidirectional AC/DC three-arm converter

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



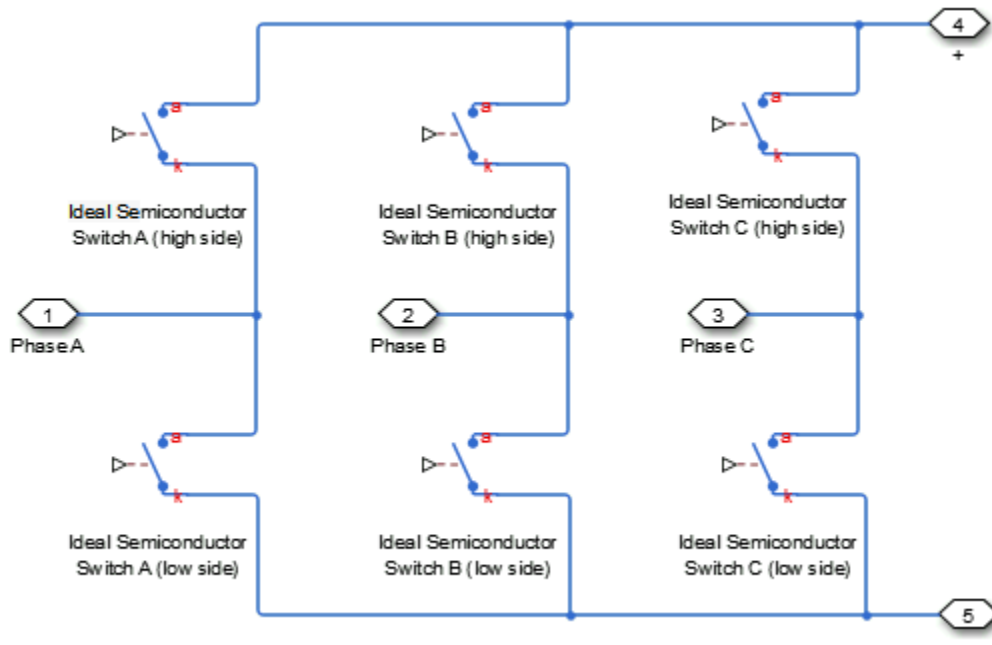
## Description

The Converter (Three-Phase) block models a three-arm converter circuit that connects a three-phase AC network to a DC network.

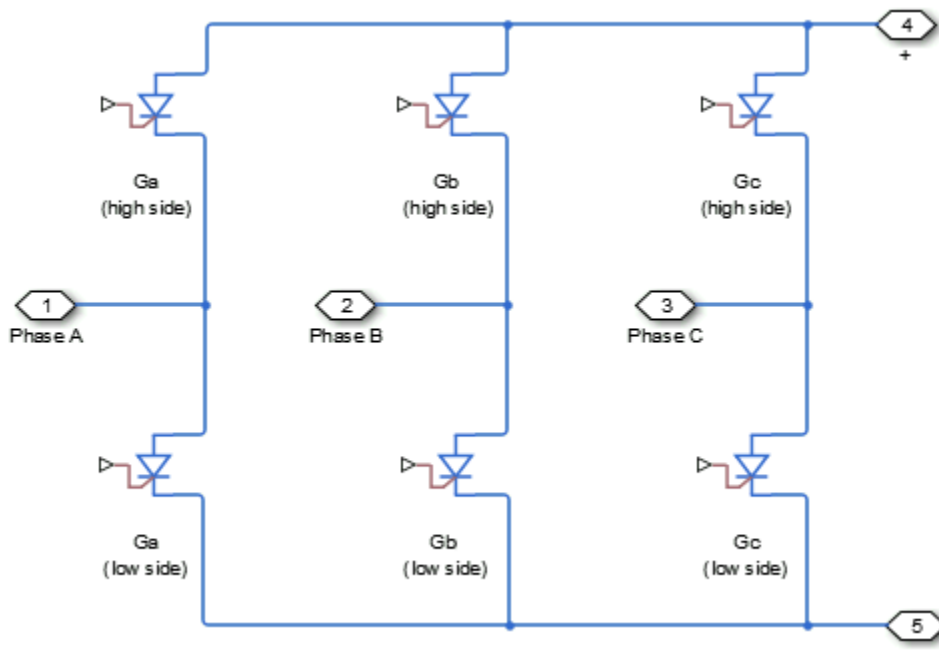
Each component in the three-arm circuit is the same switching device, which you specify using an option on the Converter (Three-Phase) block dialog box. The switching devices that you can specify are implementations of blocks in the **Simscape > Electrical > Semiconductors & Converters > Semiconductors** library:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

The figure shows the equivalent circuit for a converter with fully controlled switching devices (e.g. IGBTs, GTOs).



The figure shows the equivalent circuit for a converter with partially controlled switching devices (e.g. thyristors).



Control the gate ports of the six switching devices via an input to port **G** on the Converter (Three-Phase) block:

- 1 Multiplex all six gate signals into a single vector with a Six-Pulse Gate Multiplexer block.

- 2 Connect the output of the Six-Pulse Gate Multiplexer block to the Converter (Three-Phase) block **G** port.

You can specify an integral protection diode for each switching device. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

The table shows you how to set the **Integral protection diode** parameter based on your goals.

Goals	Value to Select	Integral Protection Diode
Prioritize simulation speed.	Diode with no dynamics	The Diode block
Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can include a snubber circuit, consisting of a resistor and capacitor connected in series, for each switching device. Snubber circuits protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of change of current when a switching device turns on.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

### Assumptions and Limitations

If, in the Solver Configuration block inside your model, you set the **Solver type** parameter to *Partitioning*, the averaged switches introduce instability during dead time, when all gate inputs are set to 0. Where possible, use the open-zero state by setting all high-side switches to 0 and all low-side switches to 1.

## Ports

### Conserving

#### **G — Gate terminals**

electrical

Vector input port associated with the gate terminals of the switching devices. Connect this port to a Six-Pulse Gate Multiplexer block.

#### **~ — Three-phase port**

electrical

Expandable three-phase port

#### **Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the DC positive terminal

#### **- — Negative terminal**

electrical

Electrical conserving port associated with the DC negative terminal



## Parameters

### Electrical connection — Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

### Switching device — Switching device

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Converter switching device. The default value is Ideal Semiconductor Switch.

The switching devices you can select are:

- “Switching Devices: GTO” on page 1-331
- “Switching Devices: Ideal Semiconductor Switch” on page 1-332
- “Switching Devices: IGBT” on page 1-332
- “Switching Devices: MOSFET” on page 1-333
- “Switching Devices: Thyristor” on page 1-333
- “Switching Devices: Averaged Switch” on page 1-334

### Dependencies

Multiple additional parameters will become visible depending on the choice of the specific switching device.

### Switching Devices: GTO

For more information, see GTO.

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

### Forward voltage, $V_f$ — Forward voltage

0.8 V (default)

Minimum voltage required across the anode and cathode block ports for the gradient of the device i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

### On-state resistance — On-state resistance

0.001 Ohm (default)

Rate of change of voltage versus current above the forward voltage.

### Off-state Conductance — Off-state conductance

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

### Gate trigger voltage, $V_{gt}$ — Gate trigger voltage

1 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Gate turn-off voltage, Vgt\_off – Gate turn-off voltage**

-1 V (default)

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Holding current – Holding current**

1 A (default)

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Switching Devices: Ideal Semiconductor Switch**

For more information, see Ideal Semiconductor Switch.

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**On-state resistance – On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Off-state Conductance – Off-state conductance**

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Threshold voltage, Vth – Threshold voltage**

6 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Switching Devices: IGBT**

For more information, see IGBT (Ideal, Switching).

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**Forward voltage, Vf – Forward voltage**

0.8 V (default)

Minimum voltage required across the collector and emitter block ports for the gradient of the diode  $i$ - $v$  characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

**On-state resistance – On-state resistance**

0.001 Ohm (default)

Collector-emitter resistance when the device is on.

**Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Collector-emitter conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Threshold voltage,  $V_{th}$  — Threshold voltage**

6 V (default)

Gate-emitter voltage at which the device turns on.

**Switching Devices: MOSFET**

For more information, see MOSFET (Ideal, Switching).

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**Drain-source on resistance,  $R_{DS(on)}$  — Drain-source on resistance**

0.001 Ohm (default)

Drain-source resistance when the device is on.

**Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Drain-source conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Threshold voltage,  $V_{th}$  — Threshold voltage**

6 V (default)

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value.

**Switching Devices: Thyristor**

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

For more information, see Thyristor (Piecewise Linear).

**Forward voltage,  $V_f$  — Forward voltage**

0.8 V (default)

Forward voltage at which the device turns on.

**On-state resistance — On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Gate trigger voltage, Vgt – Gate trigger voltage**

1 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Holding current – Holding current**

1 A (default)

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Switching Devices: Averaged Switch**

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

---

**Note** If you select this mode, the value of the gate signals must be between 0 and 1.

---

**On-state resistance – On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Integer for piecewise constant approximation of gate input (0 for disabled) – Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Integral Diodes****Integral protection diode – Integral protection diode**

None (default) | Diode with no dynamics | Diode with charge dynamics

Integral protection diode for each switching device.

The diodes you can select are:

- Diode with no dynamics
- Diode with charge dynamics

---

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

---

**Forward voltage – Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

#### On resistance — On resistance

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

#### Off conductance — Off conductance

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

#### Junction capacitance — Junction capacitance

50e-9 F (default)

Diode junction capacitance.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

#### Peak reverse current, $i_{RM}$ — Peak reverse current

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

#### Initial forward current when measuring $i_{RM}$ — Initial forward current when measuring $i_{RM}$

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** 

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Reverse recovery time,  $t_{rr}$  — Reverse recovery time**

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery time directly.

**Reverse recovery time stretch factor — Reverse recovery time stretch factor**

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify stretch factor.

**Reverse recovery charge,  $Q_{rr}$  — Reverse recovery charge**

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery charge.

For more information on these parameters, see Diode.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

#### Snubber — Snubber

None (default) | RC snubber

Snubber for each switching device:

- None - This is the default value.
- RC snubber

#### Snubber resistance — Snubber resistance

0.1 Ohm (default)

Snubber resistance.

### Dependencies

This parameter is visible only when the **Snubber** parameter is set to RC snubber.

#### Snubber capacitance — Snubber capacitance

1e-7 F (default)

Snubber capacitance.

### Dependencies

This parameter is visible only when the **Snubber** parameter is set to RC snubber.

## Version History

### Introduced in R2013b

#### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Boost Converter | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear)

### **Topics**

“Expand and Collapse Three-Phase Ports on a Block”



## Coordinate Transform (Five-Phase)

Implement  $abcde$  to  $dqxy0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



### Description

The Coordinate Transform (Five-Phase) block converts the time-domain components of a five-phase system in an  $abcde$  reference frame to direct, quadrature,  $x$ ,  $y$ , and zero components in a rotating reference frame.

You can configure the block to align the phase  $a$ -axis of the five-phase system to either the  $d$ -axis or  $q$ -axis of the rotating reference frame at time  $t = 0$ .

$\theta = \omega t + \text{Offset}$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment, where:

- $\omega$  is the rotational speed of the rotational reference frame.
- $t$  is the time, in s, from the initial alignment.
- $\text{Offset} = \frac{\pi}{2}$ ,  $d$  - axis alignment  
0,  $q$  - axis alignment

### Equations

The Coordinate Transform (Five-Phase) block implements the transform as

$$\begin{bmatrix} d \\ q \\ x \\ y \\ 0 \end{bmatrix} = \frac{2}{5} \begin{bmatrix} \sin\theta & \sin\left(\theta - \frac{2\pi}{5}\right) & \sin\left(\theta - \frac{4\pi}{5}\right) & \sin\left(\theta + \frac{4\pi}{5}\right) & \sin\left(\theta + \frac{2\pi}{5}\right) \\ \cos\theta & \cos\left(\theta - \frac{2\pi}{5}\right) & \cos\left(\theta - \frac{4\pi}{5}\right) & \cos\left(\theta + \frac{4\pi}{5}\right) & \cos\left(\theta + \frac{2\pi}{5}\right) \\ \sin\theta & \sin\left(\theta + \frac{4\pi}{5}\right) & \sin\left(\theta - \frac{2\pi}{5}\right) & \sin\left(\theta + \frac{2\pi}{5}\right) & \sin\left(\theta - \frac{4\pi}{5}\right) \\ \cos\theta & \cos\left(\theta + \frac{4\pi}{5}\right) & \cos\left(\theta - \frac{2\pi}{5}\right) & \cos\left(\theta + \frac{2\pi}{5}\right) & \cos\left(\theta - \frac{4\pi}{5}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}$$

### Ports

#### Input

**abcde** —  $a$ -,  $b$ -,  $c$ -,  $d$ -, and  $e$ -phase components

vector

Components of the five-phase system in an *abcde* reference frame, specified as a vector.

Data Types: `single` | `double`

### **$\theta_{elec}$ — Rotational angle**

scalar

Angular position of the rotating reference frame, specified as a scalar.

Data Types: `single` | `double`

### **Output**

#### **$dqxy0$ — Direct, quadrature, x, y, and zero components**

vector

Direct, quadrature, x-axis, y-axis, and zero components of the system in the rotating reference frame, returned as a vector.

Data Types: `single` | `double`

### **Parameters**

#### **Phase-a axis alignment — $dqxy0$ reference frame alignment**

Q-axis (default) | D-axis

Whether to align the *a*-phase vector of the *abcde* reference frame to the *d*-axis or *q*-axis of the rotating reference frame.

## **Version History**

Introduced in R2021a

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

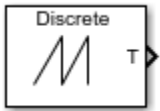
### **See Also**

Inverse Coordinate Transform (Five-Phase) | Inverse Decoupled Transform (Six-Phase) | Decoupled Transform (Six-Phase)

# Counter

Discrete- or continuous-time counter

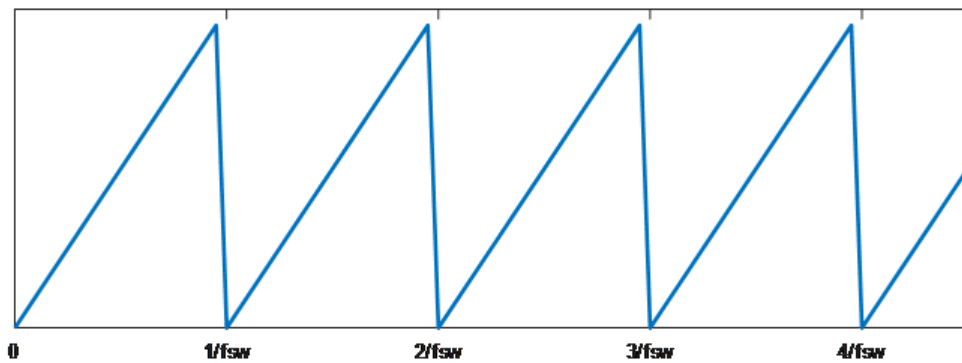
**Library:** Simscape / Electrical / Control / General Control



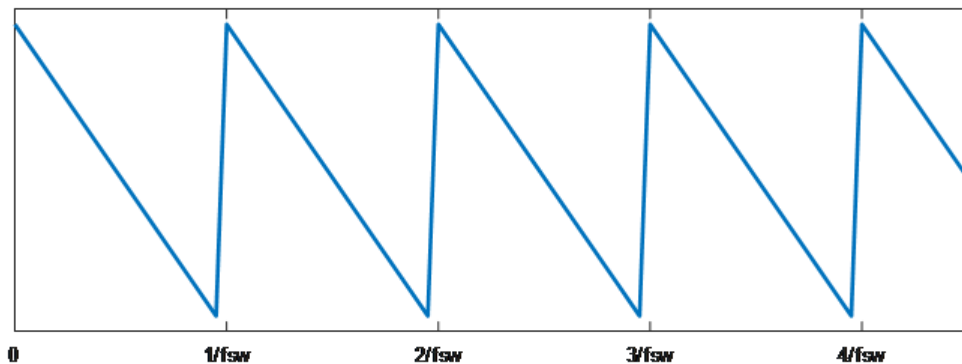
## Description

The Counter block implements an incremental counter in either discrete or continuous time. The counter increments in response to one of these criteria:

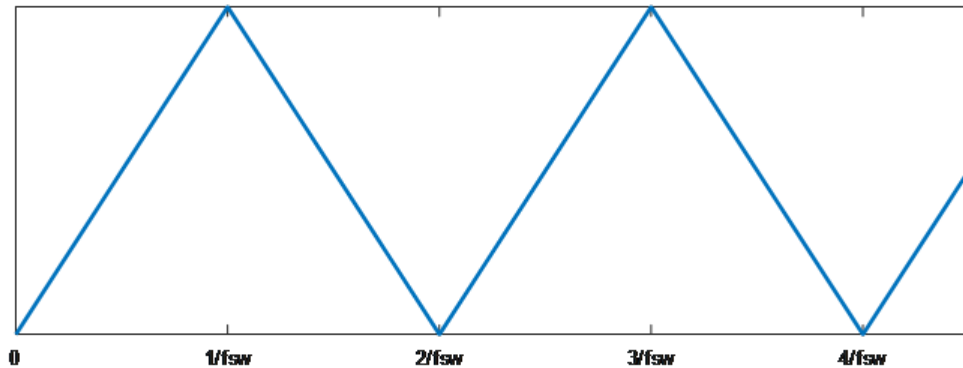
- Up — The input goes up.



- Down — The input goes down.



- Up-Down — The input goes up or down.



## Ports

### Output

#### T – Counter value

scalar

Counter value.

Data Types: single | double

## Parameters

### Counter type – Counter strategy

Up (default) | Down | Up-Down

Counter type.

### Output range – Range

[0 1] (default) | [-1 1]

Range for the output.

### Timer period (s) – Period

0.001 (default) | positive scalar

Timer period, in s.

### Phase delay (s) – Delay

0 (default) | 0 or positive scalar

Phase delay, in s. Add a phase delay to change the initial state of the counter.

### Sample time – Block sample time

0.0001 (default) | 0 or positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For discrete-time simulation, set the sample time to a positive scalar. For continuous-time simulation, set the sample time to 0.

## **Version History**

**Introduced in R2018b**

## **Extended Capabilities**

### **C/C++ Code Generation**

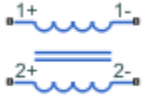
Generate C and C++ code using Simulink® Coder™.

## **See Also**

## Coupled Lines (Pair)

Magnetically couple two lines

**Library:** Simscape / Electrical / Passive / Lines



### Description

The Coupled Lines (Pair) block models two magnetically coupled lines. Each line has a self-inductance, series resistance, and parallel conductance. In addition, there is a mutual inductance and mutual resistance between the two lines.

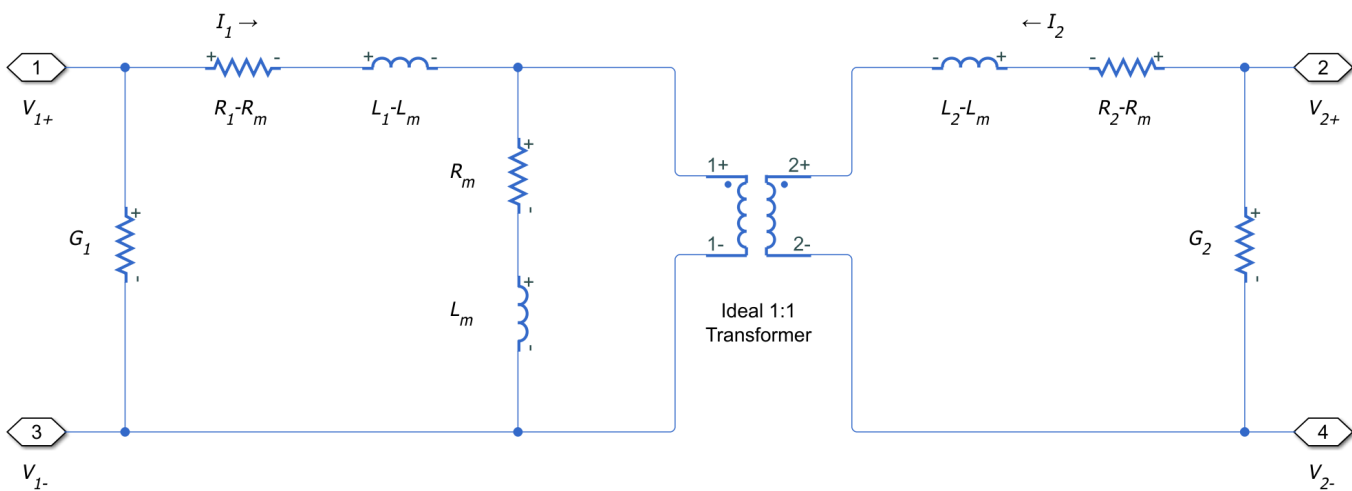
Use this block when the magnetic coupling between the two lines is nonnegligible. These effects are most prominent when:

- The lines are parallel and close together.
- The self-inductances of the lines are high.
- The AC frequency of the network is high.

To model magnetic coupling of a three-phase line, use the Coupled Lines block.

### Equivalent Circuit

The figure shows the equivalent circuit for a pair of coupled lines.



Here:

- $R_1$  and  $R_2$  are the series resistances of lines 1 and 2, respectively.

- $L_1$  and  $L_2$  are the self-inductances of lines 1 and 2, respectively.
- $R_m$  is the mutual resistance between the two lines. You can use this parameter to account for losses in a common return path.
- $L_m$  is the mutual inductance between the two lines.
- $G_1$  and  $G_2$  are the leakage conductances of lines 1 and 2, respectively.
- $V_1$  and  $V_2$  are voltage drops across lines 1 and 2, respectively.
- $I_1$  and  $I_2$  are the currents through the resistors  $R_1 - R_m$  and  $R_2 - R_m$ , respectively.

### Equations

The defining equation for this block is:

$$V = \begin{bmatrix} R_1 & R_m \\ R_m & R_2 \end{bmatrix} I + \begin{bmatrix} L_1 & L_m \\ L_m & L_2 \end{bmatrix} \frac{dI}{dt},$$

where:

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

$$I = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}.$$

$I_1$  and  $I_2$  are, in general, not equal to the currents in line 1 and line 2. These terminal currents make up the vector:

$$I_{total} = I + \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} V.$$

### Inductive Coupling

To quantify the strength of the coupling between the two lines, you can use a coupling factor or coefficient of coupling  $k$ . The coupling factor relates the mutual inductance to the line self-inductances:

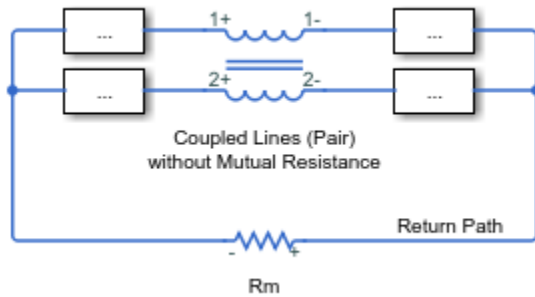
$$L_m = k\sqrt{L_1 L_2}.$$

This coupling factor must fall in the range  $-1 < k < 1$ , where a negative coupling factor indicates a reversal in orientation of one of the coils. The magnitude of  $k$  indicates:

- $|k| = 0$  — There is no magnetic coupling between the two lines.
- $0 < |k| < 0.5$  — The two lines are loosely coupled and mutual magnetic effects are small.
- $0.5 \leq |k| < 1$  — The two lines are strongly coupled and mutual magnetic effects are large.

### Mutual Resistance

If the two lines share a common return path, you can model the resistance of this return path using the **Mutual resistance** parameter. This workflow is equivalent to setting the **Mutual resistance** to zero and explicitly modeling the return path resistance  $R_m$ , as shown in this diagram.



If the two lines do not share a common return path, set the mutual resistance parameter to zero and model each of the return resistances explicitly.

## Ports

### Conserving

#### 1+ – Line 1 positive terminal

electrical

Electrical conserving port associated with the positive terminal of line 1.

#### 1- – Line 1 negative terminal

electrical

Electrical conserving port associated with the negative terminal of line 1.

#### 2+ – Line 2 positive terminal

electrical

Electrical conserving port associated with the positive terminal of line 2.

#### 2- – Line 2 negative terminal

electrical

Electrical conserving port associated with the negative terminal of line 2.

## Parameters

### Parameters

#### Parameterization – Line impedance parameterization

Balanced impedance (default) | General impedance

Specify how to parameterize the impedance of the two lines:

#### Balanced impedance

Specify the same series resistance, series inductance, and parallel leakage conductance for both lines.



**General impedance**

Specify the series resistance, series inductance, and parallel leakage conductance separately for each line.

**Line 1 inductance — Line 1 self-inductance**

1e-3 H (default)

Self-inductance of line 1. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line 2 inductance — Line 2 self-inductance**

1e-3 H (default)

Self-inductance of line 2. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line inductance — Self-inductance**

1e-3 H (default)

Self-inductance of line 1 and line 2. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

**Mutual inductance — Mutual inductance between lines**

3e-4 H (default)

Mutual inductance between line 1 and line 2. If you know the coupling factor, set this value to  $k\sqrt{L_1L_2}$ . To have a physically realizable mutual inductance, this value must satisfy:

$$-\sqrt{L_1L_2} < L_m < \sqrt{L_1L_2}.$$

**Line 1 resistance — Line 1 series resistance**

0.001 Ohm (default)

Series resistance of line 1. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line 2 resistance — Line 2 series resistance**

0.001 Ohm (default)

Series resistance of line 2. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line resistance — Series resistance**

0.001 Ohm (default)

Series resistance of line 1 and line 2. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

**Mutual resistance — Mutual resistance between lines**

0 Ohm (default)

Mutual resistance between line 1 and line 2. This value must be greater than or equal to zero. Use this value to account for losses in a common return path.

**Line 1 leakage conductance — Line 1 parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line 1. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line 2 leakage conductance — Line 2 parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line 2. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line leakage conductance — Parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line 1 and line 2. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

## Version History

Introduced in R2018a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

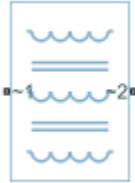
**See Also**

Coupled Lines

## Coupled Lines (Three-Phase)

Magnetically couple three-phase lines

**Library:** Simscape / Electrical / Passive / Lines



### Description

The Coupled Lines (Three-Phase) block models three magnetically coupled lines. Each line has a self-inductance, series resistance, and parallel conductance. In addition, there is a mutual inductance and mutual resistance between each pair of lines.

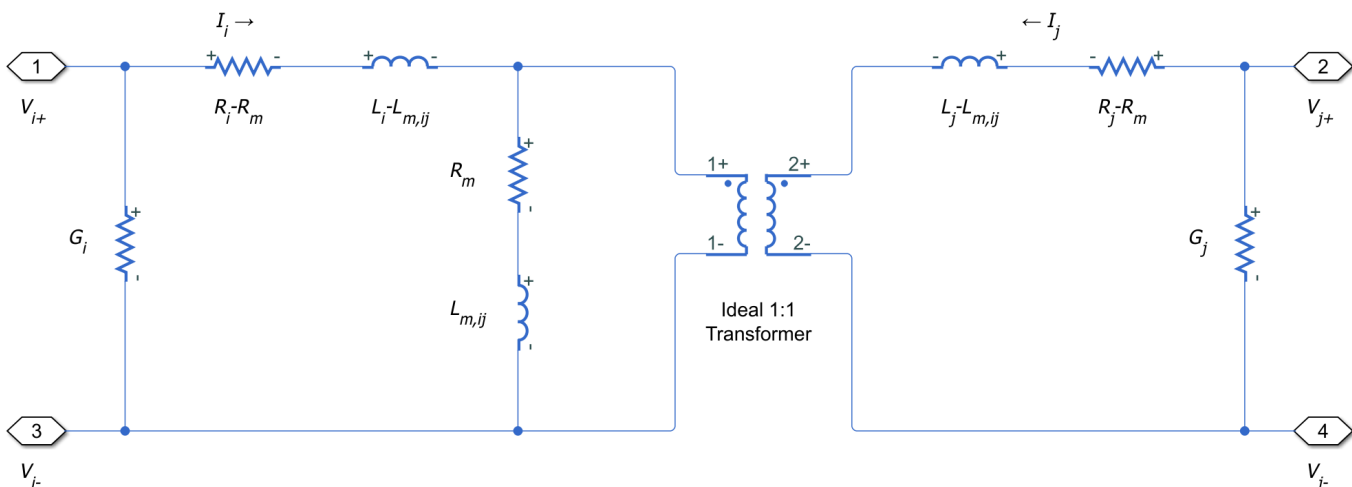
Use this block when the magnetic coupling in a three-phase network is nonnegligible. These effects are most prominent when:

- The lines are parallel and close together.
- The self-inductances of the lines are high.
- The AC frequency of the network is high.

To model magnetic coupling of a single pair of lines, use the Coupled Lines (Pair) block. To model capacitive coupling between the lines, use the Transmission Line block.

### Equivalent Circuit

The equivalent circuit shows the coupling between two arbitrary phases  $i$ , and  $j$ . The block models the magnetic coupling using such an equivalent circuit between each of the three phases  $a$ ,  $b$ , and  $c$ .



Here:

- $R_i$  and  $R_j$  are the series resistances of lines  $i$  and  $j$ , respectively.
- $L_i$  and  $L_j$  are the self-inductances of lines  $i$  and  $j$ , respectively.
- $R_m$  is the mutual resistance between the two lines. You can use this parameter to account for losses in a common return path.
- $L_{m,ij}$  is the mutual inductance between lines  $i$  and  $j$ , respectively.
- $G_i$  and  $G_j$  are the leakage conductances of lines  $i$  and  $j$ , respectively.
- $V_i$  and  $V_j$  are voltage drops across lines  $i$  and  $j$ , respectively.
- $I_i$  and  $I_j$  are the currents through the resistors  $R_i-R_m$  and  $R_j-R_m$ , respectively.

### Equations

The defining equation for this block is:

$$V = \begin{bmatrix} R_a & R_m & R_m \\ R_m & R_b & R_m \\ R_m & R_m & R_c \end{bmatrix} I + \begin{bmatrix} L_a & L_{m,ab} & L_{m,ac} \\ L_{m,ab} & L_b & L_{m,bc} \\ L_{m,ac} & L_{m,bc} & L_c \end{bmatrix} \frac{dI}{dt},$$

where:

$$V = \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix},$$

$$I = \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}.$$

$I_a$ ,  $I_b$ , and  $I_c$  are, in general, not equal to the currents in lines  $a$ ,  $b$ , and  $c$ . These terminal currents make up the vector:

$$I_{total} = I + \begin{bmatrix} G_a & 0 & 0 \\ 0 & G_b & 0 \\ 0 & 0 & G_c \end{bmatrix} V.$$

### Inductive Coupling

To quantify the strength of the coupling between the two lines, you can use a coupling factor or coefficient of coupling  $k$ . The coupling factor relates the mutual inductance to the line self-inductances:

$$L_{m,ij} = k\sqrt{L_i L_j}.$$

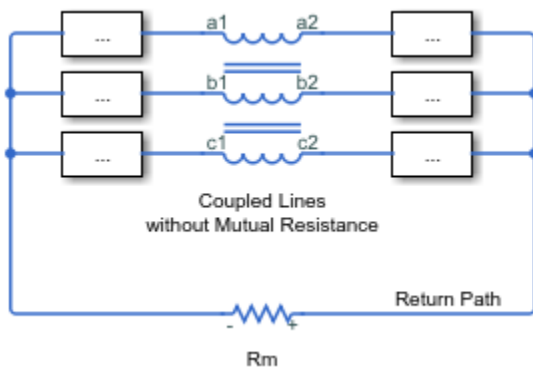
This coupling factor must fall in the range  $-1 < k < 1$ , where a negative coupling factor indicates a reversal in orientation of one of the coils. The magnitude of  $k$  indicates:

- $|k| = 0$  — There is no magnetic coupling between the two lines.
- $0 < |k| < 0.5$  — The two lines are loosely coupled and mutual magnetic effects are small.

- $0.5 \leq |k| < 1$  — The two lines are strongly coupled and mutual magnetic effects are large.

### Mutual Resistance

If the three lines share a common return path, you can model the resistance of this return path using the **Mutual resistance** parameter  $R_m$ . This workflow is equivalent to setting the **Mutual resistance** to zero and explicitly modeling the return path resistance  $R_m$ , as shown in this diagram.



If the three lines do not share a common return path, set the mutual resistance parameter to zero and model each of the return resistances explicitly.

## Ports

### Conserving

#### ~1 — Line 1 positive terminal

electrical

Expandable three-phase electrical conserving port associated with the positive terminals of lines *a*, *b*, and *c*.

#### Dependencies

To enable this port, set **Modeling option** to Composite three-phase ports

#### ~2 — Line 1 negative terminal

electrical

Expandable three-phase electrical conserving port associated with the negative terminals of lines *a*, *b*, and *c*.

#### Dependencies

To enable this port, set **Modeling option** to Composite three-phase ports

#### a1 — Line a positive terminal

electrical

Electrical conserving port associated with the positive terminal of line *a*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**a2 — Line a negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line *a*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**b1 — Line b positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line *b*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**b2 — Line b negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line *b*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**c1 — Line c positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line *c*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**c2 — Line c negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line *c*.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports

**Parameters****Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Main

### Parameterization — Line impedance parameterization

Balanced impedance (default) | General impedance

Specify how to parameterize the impedance of the three lines:

#### Balanced impedance

Specify the same series resistance, series inductance, and parallel leakage conductance for all lines.

#### General impedance

Specify the series resistance, series inductance, and parallel leakage conductance separately for each line.

### Line a inductance — Line a self-inductance

1e-3 H (default)

Self-inductance of line *a*. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Parameterization** to General impedance.

### Line b inductance — Line b self-inductance

1e-3 H (default)

Self-inductance of line *b*. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Parameterization** to General impedance.

### Line c inductance — Line c self-inductance

1e-3 H (default)

Self-inductance of line *c*. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Parameterization** to General impedance.

### Line inductance — Self-inductance

1e-3 H (default)

Self-inductance of lines *a*, *b*, and *c*. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Parameterization** to Balanced impedance.

**Line a-b mutual inductance – Mutual inductance between a-b**

3e-4 H (default)

Mutual inductance between lines  $a$  and  $b$ . If you know the coupling factor, set this value to  $k\sqrt{L_a L_b}$ . To have a physically realizable mutual inductance, this value must satisfy:

$$-\sqrt{L_a L_b} < L_{m,ab} < \sqrt{L_a L_b}.$$

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line b-c mutual inductance – Mutual inductance between b-c**

3e-4 H (default)

Mutual inductance between lines  $b$  and  $c$ . If you know the coupling factor, set this value to  $k\sqrt{L_b L_c}$ . To have a physically realizable mutual inductance, this value must satisfy:

$$-\sqrt{L_b L_c} < L_{m,bc} < \sqrt{L_b L_c}.$$

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line a-c mutual inductance – Mutual inductance between a-c**

3e-4 H (default)

Mutual inductance between lines  $a$  and  $c$ . If you know the coupling factor, set this value to  $k\sqrt{L_a L_c}$ . To have a physically realizable mutual inductance, this value must satisfy:

$$-\sqrt{L_a L_c} < L_{m,ac} < \sqrt{L_a L_c}.$$

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Mutual inductance – Mutual inductance**

3e-4 H (default)

Mutual inductance between each pair of lines. If you know the coupling factor, set this value to  $kL$ , where  $L$  is the series inductance of each of the lines. To have a physically realizable mutual inductance, this value must satisfy:

$$-L < L_m < L.$$

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

**Resistance****Line a resistance – Line a series resistance**

0.001 Ohm (default)

Series resistance of line  $a$ . This value must be greater than or equal to zero.



**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line b resistance — Line b series resistance**

0.001 Ohm (default)

Series resistance of line *b*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line c resistance — Line c series resistance**

0.001 Ohm (default)

Series resistance of line *c*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line resistance — Series resistance**

0.001 Ohm (default)

Series resistance of lines *a*, *b*, and *c*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

**Mutual resistance — Mutual resistance**

0 Ohm (default)

Mutual resistance between each pair of lines. This value must be greater than or equal to zero. Use this value to account for losses in a common return path.

**Line a leakage conductance — Line a parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line *a*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line b leakage conductance — Line b parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line *b*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line c leakage conductance — Line c parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of line *a*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to General impedance.

**Line leakage conductance — Parallel leakage conductance**

1e-9 1/Ohm (default)

Parallel leakage conductance of lines *a*, *b*, and *c*. This value must be greater than or equal to zero.

**Dependencies**

To enable this parameter, set **Parameterization** to Balanced impedance.

## Version History

Introduced in R2018a

### Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Coupled Lines (Pair)

# Crystal

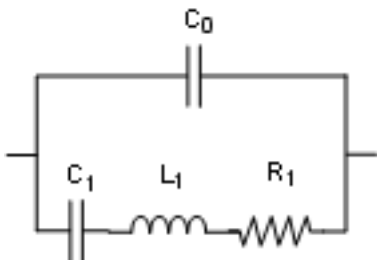
Stable resonator

**Library:** Simscape / Electrical / Passive



## Description

The Crystal block represents the electrical characteristics of a crystal. The following figure shows the equivalent circuit model of the Crystal block.



You specify the equivalent circuit parameters for this model when you set the **Parameterization** parameter to `Equivalent circuit parameters`.

- The capacitor  $C0$  corresponds to the capacitance you specify in the **Shunt capacitance, C0** parameter.
- The capacitor  $C1$  corresponds to the capacitance you specify in the **Motional capacitance, C1** parameter.
- The inductor  $L1$  corresponds to the inductance you specify in the **Motional inductance, L1** parameter.
- The resistor  $R1$  corresponds to the resistance you specify in the **Equivalent series resistance, R1** parameter.

Most datasheets specify crystal frequency rather than inductance, so the block optionally accepts frequency data.

- When you set the **Parameterization** parameter to `Series resonance data`, the block uses the following relationship to calculate  $L1$  from the series resonant frequency:

$$f_s = \frac{1}{2\pi\sqrt{L_1 C_1}}$$

Where  $f_s$  is the **Series resonance, fs** parameter value.

- When you set the **Parameterization** parameter to `Parallel resonance data`, the block uses the following relationship to calculate  $L1$  from the parallel resonant frequency:

$$f_a = \frac{1}{2\pi\sqrt{L_1 C_1 (C_0 + C_L) / (C_1 + C_0 + C_L)}}$$

Where:

- $f_a$  is the **Parallel resonance, fa** parameter value.
- $C_L$  is the **Load capacitance, CL** parameter value.

Some datasheets specify quality factor rather than equivalent series resistance, so the block optionally accepts quality factor data. When you set the **R1 parameterization** parameter to **Quality factor Q**, the block uses the following relationship to calculate  $R1$  from the quality factor:

$$Q = \frac{2\pi f L_1}{R_1}$$

Where  $Q$  is the **Quality factor, Q** parameter value.

---

**Note** The **R1 parameterization** parameter is only visible when you select **Series resonance data** or **Parallel resonance data** for the **Parameterization** parameter.

---

## Assumptions and Limitations

- The Crystal block models only the fundamental crystal vibration mode.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the crystal positive voltage.

#### - – Negative terminal

electrical

Electrical conserving port associated with the crystal negative voltage.

## Parameters

### Parameterization – Block parameterization

Series resonance data (default) | Parallel resonance data | Equivalent circuit parameters

Select one of the following methods for block parameterization:

- **Series resonance data** — Provide series resonant frequency and capacitance data for the crystal. This method is the default.
- **Parallel resonance data** — Provide parallel resonant frequency and capacitance data for the crystal.
- **Equivalent circuit parameters** — Provide electrical parameters for an equivalent circuit model of the crystal.

**Series resonance,  $f_s$  — Crystal series resonant frequency**

32.764 kHz (default)

Crystal series resonant frequency.

**Dependencies**

This parameter is visible only when you select **Series resonance** data for the **Parameterization** parameter.

**Parallel resonance,  $f_a$  — Crystal parallel resonant frequency**

32.768 kHz (default)

Crystal parallel resonant frequency that corresponds to operating with a parallel load capacitance specified by the **Load capacitance,  $C_L$**  parameter.

**Dependencies**

This parameter is visible only when you select **Parallel resonance** data for the **Parameterization** parameter.

**Motional inductance,  $L_1$  — Motional inductance**

6742 H (default)

Inductance that represents the mechanical mass of the crystal.

**Dependencies**

This parameter is visible only when you select **Equivalent circuit** parameters for the **Parameterization** parameter.

**R1 parameterization — Series resistance parameterization**Equivalent series resistance  $R_1$  (default) | Quality factor  $Q$ 

Select one of the following methods for series resistance parameterization:

- **Equivalent series resistance  $R_1$**  — Provide the resistance value directly. This is the default method.
- **Quality factor  $Q$**  — Provide the quality factor that the block uses to calculate the resistance value.

**Dependencies**

This parameter is visible only when you select **Series resonance** data or **Parallel resonance** data for the **Parameterization** parameter.

**Quality factor,  $Q$  — Crystal quality factor**

90000 (default)

Crystal quality factor. This parameter is only visible when you make one of the following selections:

- **Series resonance** data for the **Parameterization** parameter and **Quality factor  $Q$**  for the **R1 parameterization** parameter
- **Parallel resonance** data for the **Parameterization** parameter and **Quality factor  $Q$**  for the **R1 parameterization** parameter

**Dependencies**

This parameter is visible only when you select:

- Series resonance data for the **Parameterization** parameter and Quality factor Q for the **R1 parameterization** parameter
- Parallel resonance data for the **Parameterization** parameter and Quality factor Q for the **R1 parameterization** parameter

**Equivalent series resistance, R1 — Motional damping**

15 kOhm (default)

Motional damping term. This parameter is only visible when you make one of the following selections:

- Series resonance data for the **Parameterization** parameter and Equivalent series resistance R1 for the **R1 parameterization** parameter
- Parallel resonance data for the **Parameterization** parameter and Equivalent series resistance R1 for the **R1 parameterization** parameter
- Equivalent circuit parameters for the **Parameterization** parameter

**Dependencies**

This parameter is visible only when you select:

- Series resonance data for the **Parameterization** parameter and Equivalent series resistance R1 for the **R1 parameterization** parameter
- Parallel resonance data for the **Parameterization** parameter and Equivalent series resistance R1 for the **R1 parameterization** parameter
- Equivalent circuit parameters for the **Parameterization** parameter

**Motional capacitance, C1 — Motional capacitance**

0.0035 pF (default)

Capacitance that represents crystal mechanical stiffness under load.

**Load capacitance, CL — Load capacitance**

12.5 pF (default)

Load capacitance that corresponds to the **Parallel resonance, fa** parameter value.

**Dependencies**

This parameter is visible only when you select Parallel resonance data for the **Parameterization** parameter.

**Shunt capacitance, C0 — Shunt capacitance**

1.6 pF (default) | positive scalar

Electrical capacitance between the two crystal electrical connections.

**Initial voltage — Initial output voltage**

0 V (default)

The output voltage at the start of the simulation when the output current is zero.

## **Version History**

Introduced in R2009a

## **Extended Capabilities**

### **C/C++ Code Generation**

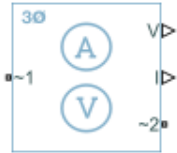
Generate C and C++ code using Simulink® Coder™.

## **See Also**

## Current and Voltage Sensor (Three-Phase)

Ideal three-phase current and voltage sensor

**Library:** Simscape / Electrical / Sensors & Transducers



### Description

The Current and Voltage Sensor (Three-Phase) block measures the voltage and current of a three-phase electrical node.

---

**Note** You must connect the Current and Voltage Sensor (Three-Phase) block in series with a three-phase element. For more information on how to use this block, see “Build and Simulate Composite and Expanded Three-Phase Models”.

---

### Ports

#### Output

##### V — Voltage

physical signal

Three-element physical signal vector output port associated with the *a*-, *b*-, and *c*-phase voltages.

##### I — Current

physical signal

Three-element physical signal vector output port associated with the *a*-, *b*-, and *c*-phase currents.

#### Conserving

##### ~1 — Three-phase voltage and current

electrical

Expandable three-phase electrical conserving port 1 associated with three-phase voltage and current.

##### ~2 — Three-phase voltage and current

electrical

Expandable three-phase electrical conserving port 2 associated with three-phase voltage and current.

### Parameters

#### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports



Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Voltage measurement type — Voltage measurement type**  
 phase-to-phase voltage (default) | phase-to-ground voltage

Type of voltage measurement.

**Measurement output unit — Measurement system**  
 Per unit (default) | SI

System of units for output current and voltage measurements.

**Nominal power — Nominal power**  
 100e6 V\*A (default)

Nominal power.

#### **Dependencies**

To enable this parameter, set **Measurement output unit** to Per unit.

**Nominal voltage (phase-to-phase RMS) — Nominal voltage**  
 24e3 V (default)

Nominal phase-to-phase root-mean-square (RMS) voltage.

#### **Dependencies**

To enable this parameter, set **Measurement output unit** to Per unit.

## **Version History**

**Introduced in R2019a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Current Sensor (Three-Phase) | Line Voltage Sensor (Three-Phase) | Phase Voltage Sensor (Three-Phase) | Power Sensor

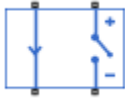
### **Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Current-Controlled Switch

Current-controlled switch with hysteresis

**Library:** Simscape / Electrical / Additional Components / SPICE Passives



## Description

The Current-Controlled Switch block represents the electrical characteristics of a switch whose state is controlled by the current through the input ports (the controlling current).

This block models either a variable-resistance or a short-transition switch. For a variable-resistance switch, set the **Switch model** parameter to Smooth transition between  $I_{on}$  and  $I_{off}$ . For a short-transition switch, set **Switch model** to Abrupt transition after delay.

### Short-transition Switch

In a short-transition switch, the transition between the off and on states is instantaneous:

- When the controlling current is greater than or equal to the sum of the **Threshold current,  $I_T$**  and **Hysteresis current,  $I_H$**  parameter values, the switch is closed and has a resistance equal to the **On resistance,  $R_{ON}$**  parameter value.
- When the controlling current is less than the **Threshold current,  $I_T$**  parameter value minus the **Hysteresis current,  $I_H$**  parameter value, the switch is open and has a resistance equal to the **Off resistance,  $R_{OFF}$**  parameter value.
- When the controlling current is greater than or less than the **Threshold current,  $I_T$**  parameter value by an amount less than or equal to the **Hysteresis current,  $I_H$**  parameter value, the current is in the crossover region and the state of the switch remains unchanged.

---

**Note** If the **Hysteresis current,  $I_H$**  parameter value is less than 0, the block models a variable-resistance switch independently of the value you set for the **Switch model** parameter.

---

### Variable-resistance Switch

In a variable-resistance switch, the transition between the off and on states is exponential:

- When the **Control current for on state,  $I_{ON}$**  is greater than the **Control current for off state,  $I_{OFF}$** :
  - If the controlling current is greater than the **Control current for on state,  $I_{ON}$**  parameter value, the switch is closed and has a resistance equal to the **On resistance,  $R_{ON}$**  parameter value.

- If the controlling current is less than the **Control current for off state, IOFF** parameter value, the switch is open and has a resistance equal to the **Off resistance, ROFF** parameter value.
- If the controlling current is greater than the **Control current for off state, IOFF** parameter value or less than the **Control current for on state, ION** parameter value, the resistance is defined by:

$$R_s = \exp \left[ L_m + 3L_r \left( \frac{I_c - I_m}{2I_d} \right) - 2L_r \left( \frac{(I_c - I_m)^3}{I_d^3} \right) \right]$$

where:

- $R_s$  is the switch resistance.
  - $L_m = \ln \left[ (R_{ON} * R_{OFF})^{\frac{1}{2}} \right]$  is the log-mean of resistor values.
  - $L_r = \ln \left( \frac{R_{ON}}{R_{OFF}} \right)$  is the log-ratio of resistor values.
  - $I_c$  is the controlling current.
  - $I_m = \frac{I_{ON} + I_{OFF}}{2}$  is the mean of the control currents.
  - $I_d = I_{ON} - I_{OFF}$  is the difference between the control currents.
- When the **Control current for on state, ION** is less than the **Control current for off state, IOFF**:
    - If the controlling current is less than the **Control current for on state, ION** parameter value, the switch is closed and has a resistance equal to the **On resistance, RON** parameter value.
    - If the controlling current is greater than the **Control current for off state, IOFF** parameter value, the switch is open and has a resistance equal to the **Off resistance, ROFF** parameter value.
    - If the controlling current is less than the **Control current for off state, IOFF** parameter value or greater than the **Control current for on state, ION** parameter value, the resistance is defined by:

$$R_s = \exp \left[ L_m + 3L_r \left( \frac{I_c - I_m}{2I_d} \right) - 2L_r \left( \frac{(I_c - I_m)^3}{I_d^3} \right) \right]$$

## Assumptions and Limitations

The block output resistance model is discontinuous during switching. The discontinuity might cause numerical issues. Try the following actions to resolve the issues:

- Set the **On resistance, RON** and **Off resistance, ROFF** parameter values to keep  $R_{ON}$  divided by  $R_{OFF}$  as large as possible, and more than  $1e+12$ .
- Increase the **Hysteresis current, IH** parameter value to reduce switch chatter.
- If a variable-step solver is employed, decrease the **Max step size** parameter value (in the Configuration Parameters block dialog box).

---

**Note** This increases the simulation time.

---

## Ports

### Conserving

#### **1+ – Positive input port**

electrical

Electrical conserving port associated with the current controlled switch positive input.

#### **1- – Negative input port**

electrical

Electrical conserving port associated with the current controlled switch negative input.

#### **2+ – Positive output port**

electrical

Electrical conserving port associated with the current controlled switch positive output.

#### **2- – Negative output port**

electrical

Electrical conserving port associated with the current controlled switch negative output.

## Parameters

### **Switch model – Switch model**

Abrupt transition after delay (default) | Smooth transition between  $I_{on}$  and  $I_{off}$

Whether the transition between on state and off state current is instantaneous or not. To model this block as a variable-resistance switch, select Smooth transition between  $I_{on}$  and  $I_{off}$ . To model this block as a short-transition switch, select Abrupt transition after delay.

### **Threshold current, IT – Threshold current**

0 A (default)

Current above which the block interprets the controlling current as HIGH.

---

**Note** The controlling current must differ from the threshold current by at least the **Hysteresis current, IH** parameter value to change the state of the switch.

---

### Dependencies

To enable this parameter, set **Switch model** to Abrupt transition after delay.

### **Hysteresis current, IH – Hysteresis current**

0 A (default)

Amount by which the controlling current must exceed or fall below the **Threshold current, IT** parameter value to change the state of the switch.

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

**Control current for on state, ION – Control current for on state**

1e-3 A (default)

Switch control current for the on state.

**Dependencies**

To enable this parameter, set **Switch model** to Smooth transition between Ion and Ioff.

**Control current for off state, IOFF – Control current for off state**

0 A (default)

Switch control current for the off state.

**Dependencies**

To enable this parameter, set **Switch model** to Smooth transition between Ion and Ioff.

**On resistance, RON – Closed switch resistance**

1 Ohm (default)

Resistance of the switch when it is closed.

**Use 1/GMIN as Off resistance – Whether to use 1/GMIN as off resistance**

No (default) | Yes

Whether to automatically calculate the off resistance of this switch as 1/GMIN.

---

**Note** The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

By default, *GMIN* matches the **GMIN** parameter of the Environment Parameters block, whose default value is 1e-12. To change *GMIN*, add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

---

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

**Off resistance, ROFF – Open switch resistance**

1e12 Ohm (default) | 1e6

Resistance of the switch when it is open.

**Dependencies**

To enable this parameter either:

- Set **Switch model** to Abrupt transition after delay and **Use 1/GMIN as Off resistance** to No. The default value is 1e12.
- Set **Switch model** to Smooth transition between Ion and Ioff. The default value is 1e6.

, .

**Time delay, TD — Time delay**

0 s (default)

Time delay after which the switch transition takes place.

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

## Version History

Introduced in R2009a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | Voltage-Controlled Switch

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Current Limiter

Behavioral model of current limiter

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

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

The current limiting acts for both positive and negative currents. For applications where limiting is required in only one direction, you can augment the Current Limiter block with a series diode (blocks any reverse current) or parallel diode (no limiting in the reverse direction).

The block implements current limiting by using a hyperbolic tangent function:

$$i = i_{LIM} \tanh\left(\frac{4v}{v_{LIM}}\right) + g_{LIM}v$$

where:

- $i$  is the current through the component.
- $v$  is the voltage drop across the component.
- $i_{LIM}$  is the current limit.
- $v_{LIM}$  is the approximate voltage drop across the component when the current limit becomes active.
- $g_{LIM}$  is the rate of change of current with voltage drop when on the current limit (limit-state conductance).

When  $v = v_{LIM}$ , then

$$i = i_{LIM} \tanh(4) + g_{LIM}v = 0.9993i_{LIM} + g_{LIM}v$$

Therefore the current is approximately equal to the limit. Choose the value for  $g_{LIM}$  such that  $g_{LIM}v$  is small compared to  $i_{LIM}$  for the maximum expected voltage drop. This term is included in the block equation to improve numerical properties during simulation.

When choosing the value of  $v_{LIM}$ , consider that making it too small will require tight solver tolerances and small step sizes. In practice, current limiters can be implemented using a MOSFET and series source resistor, the gate-source voltage being driven by the series resistor. This implementation does not produce a sharp limit, similar to the tanh curve used in this block. You can use a datasheet plot of current against voltage to pick a suitable value for  $v_{LIM}$ .

## Thermal Port

This block has one optional thermal port. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

The thermal port model contains a thermal mass. The power dissipated by the current limiter, plus the heat flow into the thermal port, drives the thermal mass differential equation:

$$m \frac{dT}{dt} = P_{loss} + Q_H$$

where:

- $m$  is the thermal mass.
- $T$  is the thermal port temperature.
- $P_{loss}$  is the electrical loss,  $v \cdot i$ .
- $Q_H$  is the heat flow from the external network into the thermal port.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the current limiter positive terminal

#### - — Negative terminal

electrical

Electrical conserving port associated with the current limiter negative terminal

#### H — Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to **Show thermal port**.

## Parameters

### Modeling option — Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model thermal parameters.

### Parameters

#### Current limit — Current limit

1 A (default)

The maximum current magnitude.



**Voltage drop when current starts to limit – Voltage drop when current starts to limit**

0.1 V (default)

When the voltage drop is equal to this value, then the current is limited at 0.9993 times the current limit value.

**Limit-state conductance – Limit-state conductance**

1e-3 1/Ohm (default)

When the current is limited, this parameter defines the rate of change of current with voltage drop if the current is driven harder onto the limit.

**Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

**Thermal mass – Thermal mass**

100 J/K (default)

The heat energy required to raise the temperature by one degree.

**Initial temperature – Initial temperature**

25 C (default)

The temperature at the start of the simulation.

## Version History

Introduced in R2015a

## Extended Capabilities

**C/C++ Code Generation**

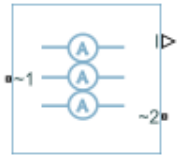
Generate C and C++ code using Simulink® Coder™.

## See Also

## Current Sensor (Three-Phase)

Ideal three-phase current measurement

**Library:** Simscape / Electrical / Sensors & Transducers



### Description

The Current Sensor (Three-Phase) block represents an ideal three-phase current sensor. The block measures each of the three currents flowing from port ~1 to port ~2 and outputs a single three-element, physical signal vector. Each element of the physical signal output vector is proportional to the current in its respective phase.

---

**Note** You must connect the Current Sensor (Three-Phase) block in series with a three-phase element. For more information on how to use this block, see “Build and Simulate Composite and Expanded Three-Phase Models”.

---

### Ports

#### Output

##### **I — Three-elements vector**

physical signal

Three-element physical signal vector output port associated with the phase currents.

#### Conserving

##### **~1 — Three-phase port**

electrical

Expandable three-phase electrical conserving port.

##### **~2 — Three-phase port**

electrical

Expandable three-phase electrical conserving port.

### Parameters

#### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

[Current and Voltage Sensor \(Three-Phase\)](#) | [Line Voltage Sensor \(Three-Phase\)](#) | [Phase Voltage Sensor \(Three-Phase\)](#) | [Power Sensor](#)

## **Topics**

[“Build and Simulate Composite and Expanded Three-Phase Models”](#)

[“Expand and Collapse Three-Phase Ports on a Block”](#)

# Current Source

Current source with optional DC, AC, and noise components

**Library:** Simscape / Electrical / Sources



## Description

The Current Source block implements a current source with DC, AC, and noise components. The current flowing through the source from the - terminal to the + terminal is given by:

$$i = i_{DC} + i_{AC}\sin(2\pi ft + \phi) + i_N$$

where:

- $i_{DC}$  is the steady-state DC current component.
- $i_{AC}$  is the amplitude of the AC current component.
- $f$  is the frequency of the AC component.
- $\phi$  is the phase offset of the AC component.
- $i_N$  is the noise current.

You can configure your source as DC-only, AC-only, or a combination of both. By default, both AC and DC components are set to 0. Define the AC/DC current by specifying nonzero parameter values after placing the block in your model.

The noise component is also optional. If you set the **Noise mode** parameter to **Enabled**, then the added noise current is given by:

$$i_N = \sqrt{P_i/2} \frac{N(0, 1)}{\sqrt{h}}$$

where:

- $P_i$  is the single-sided noise power spectral density for a 1 ohm load, in  $A^2/Hz$ .
- $N$  is a Gaussian random number with zero mean and standard deviation of one.
- $h$  is the sampling interval.

By default, the **Noise mode** parameter is set to **Disabled**, and the current source generates no thermal noise.

## Noise Options

The block generates Gaussian noise by using the PS Random Number source in the Simscape Foundation library. You can control the random number seed by setting the **Repeatability** parameter:

- **Not repeatable** — Every time you simulate your model, the block resets the random seed using the MATLAB random number generator:

```
seed = randi(2^32-1);
```

- **Repeatable** — The block automatically generates a seed value and stores it inside the block, to always start the simulation with the same random number. This auto-generated seed value is set when you add a Current Source block from the block library to the model. When you make a new copy of the Current Source block from an existing one in a model, a new seed value is generated. The block sets the value using the MATLAB random number generator command shown above.
- **Specify seed** — If you select this option, the additional **Seed** parameter lets you directly specify the random number seed value.

## Assumptions and Limitations

- Simulating with noise enabled slows down simulation. Choose the sample time ( $h$ ) so that noise is generated only at frequencies of interest, and not higher.

## Ports

### Conserving

#### + — Positive port

electrical

Electrical conserving port associated with the current source positive terminal

#### - — Negative port

electrical

Electrical conserving port associated with the current source negative terminal

## Parameters

### DC & AC Components

#### DC current — DC current

0 A (default)

DC component of the output current. Specify a nonzero value to add a DC component to the current source.

#### AC current peak amplitude — AC current amplitude

0 A (default)

Amplitude of the AC component of the output current. Specify a nonzero value to add an AC component to the current source.

#### AC current phase shift — AC current phase offset

0 rad (default)

Phase offset of the AC component of the output current.

#### AC current frequency — AC current frequency

60 Hz (default)

Frequency of the AC component of the output current.

## Noise

### Noise mode — Noise option

Disabled (default) | Enabled

Select the noise option:

- **Disabled** — No noise is produced by the current source. This is the default.
- **Enabled** — The current source generates thermal noise, and the associated parameters become visible on the **Noise** tab.

### Power spectral density — Power spectral density

0 A<sup>2</sup>/Hz (default)

The single-sided spectrum noise power. Strictly-speaking, this is a density function for the square of the current, commonly thought of as a power into a 1 ohm load, and therefore units are A<sup>2</sup>/Hz. To avoid this unit ambiguity, some datasheets quote noise current as a noise density with units of A/√Hz. In this case, you should enter the square of the noise density quoted in the datasheet as the parameter value.

### Repeatability — Noise control option

Not repeatable (default) | Repeatable | Specify seed

Select the noise control option:

- **Not repeatable** — The random sequence used for noise generation is not repeatable. This is the default.
- **Repeatable** — The random sequence used for noise generation is repeatable, with a system-generated seed.
- **Specify seed** — The random sequence used for noise generation is repeatable, and you control the seed by using the **Seed** parameter.

### Auto-generated seed used for repeatable option — Auto-generated random number seed

3241354618 (default)

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Seed** parameter.

### Dependencies

This parameter is visible only when you select **Repeatable** for the **Repeatability** parameter.

### Seed — Random number seed

0 (default)

Random number seed used by the noise random number generator.

### Dependencies

This parameter is visible only when you select **Specify seed** for the **Repeatability** parameter.

**Sample time – Sample time**

1e-3 s (default)

Defines the rate at which the noise source is sampled. Choose it to reflect the frequencies of interest in your model. Making the sample time too small will unnecessarily slow down your simulation.

**Version History****Introduced in R2013a****Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

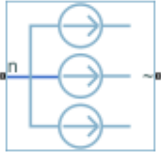
**See Also**

Resistor | Voltage Source

## Current Source (Three-Phase)

Ideal three-phase current source

**Library:** Simscape / Electrical / Sources



### Description

The Current Source (Three-Phase) block models an ideal three-phase current source that maintains sinusoidal currents of the specified magnitude through its terminals, independent of the voltage across the source.

The output current is defined by the following equations:

$$I_0 = \sqrt{2}i_{\text{phase\_rms}}$$

$$i_a = I_0 \sin(2\pi ft + \varphi)$$

$$i_b = I_0 \sin(2\pi ft + \varphi - 120^\circ)$$

$$i_c = I_0 \sin(2\pi ft + \varphi + 120^\circ),$$

where:

- $I_0$  is the peak phase current.
- $i_{\text{phase\_rms}}$  is the RMS phase current.
- $i_a, i_b, i_c$  are the respective phase currents.
- $f$  is the frequency.
- $\varphi$  is the phase shift.
- $t$  is the time.

The arrow indicates the positive direction of the current flow. The source has a wye configuration, and port **n** provides a connection to the center of the wye.

### Ports

#### Conserving

~ — **Three-phase port**

electrical

Expandable three-phase port associated with the three phases,  $a$ ,  $b$ , and  $c$ .



**n – Wye center**

electrical

Electrical conserving port associated with the center of the wye.

**Parameters****Modeling option – Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Current (phase RMS) – RMS phase current**

100 / sqrt(2) A (default)

RMS phase current.

**Phase shift – Phase shift**

0 deg (default)

Phase shift, in angular units.

**Frequency – Current frequency**

60 Hz (default)

Current frequency, specified in Hz or units directly convertible to Hz (where Hz is defined as 1/s). For example, kHz and MHz are valid units, but rad/s is not.

**Version History**

Introduced in R2013b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Battery | Controlled Current Source (Three-Phase) | Voltage Source (Three-Phase)

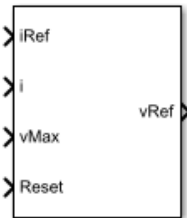
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

## DC Current Controller

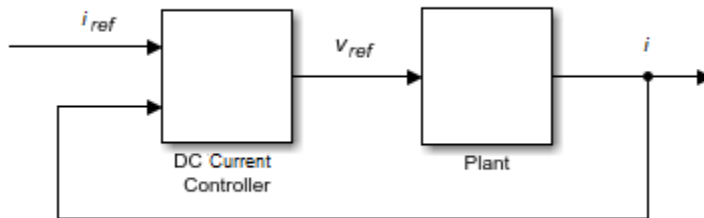
Discrete-time DC current PI control with integral anti-windup

**Library:** Simscape / Electrical / Control / General Machine Control



### Description

The DC Current Controller block implements a discrete-time proportional-integral (PI) DC voltage controller. The block can implement zero cancellation in the feedforward path. To avoid saturation of the integral gain, the block can implement anti-windup gain.



### Equations

The equation that the DC Current Controller block uses to calculate the reference voltage is

$$v_{ref} = \left( K_p + K_i \frac{T_s z}{z-1} \right) (i_{ref} - i),$$

where:

- $v_{ref}$  is the reference voltage.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the sample time.
- $i_{ref}$  is the reference current.
- $i$  is the measured current.

The PI control calculation yields a zero in the closed-loop transfer function. To cancel the zero, the block uses this discrete-time zero-cancellation transfer function:

$$G_{ZC}(z) = \frac{\frac{T_s K_i}{K_p}}{z + \frac{T_s - \frac{K_p}{K_i}}{\frac{K_p}{K_i}}}$$

To avoid saturation of the integrator output, the block uses an anti-windup mechanism. The integrator gain is then equal to

$$K_i + K_{aw}(v_{ref\_sat} - v_{ref\_unsat}),$$

where:

- $K_{aw}$  is the anti-windup gain.
- $v_{ref\_sat}$  is the saturated reference voltage signal, which the block calculates as  $v_{ref\_sat} = \min(\max(v_{ref\_unsat}, v_{min}), v_{max})$ ,

where:

- $v_{ref\_unsat}$  is the unsaturated reference voltage signal.
- $v_{min}$  is the lower limit for the output voltage. For positive voltage only,  $v_{min} = 0$ . For positive and negative voltage,  $v_{min} = -v_{max}$
- $v_{max}$  is the upper limit for the output voltage.

## Ports

### Input

#### **iRef — Reference current**

scalar

Desired output current for the plant.

Data Types: single | double

#### **i — Measured current**

scalar

Measured current for the plant.

Data Types: single | double

#### **v<sub>max</sub> — Maximum DC voltage**

scalar

Maximum DC output voltage for the plant.

Data Types: single | double

#### **Reset — External reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: single | double

### Output

#### **vRef — Reference DC voltage**

scalar

Desired DC output voltage for the plant.

Data Types: single | double

### Parameters

#### **Proportional gain — Controller proportional gain, $K_p$**

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

#### **Integral gain — Integral gain, $K_i$**

5 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.

#### **Anti-windup gain — Anti-windup gain, $K_{aw}$**

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.

#### **Voltage limitation — Negative voltage restriction**

Positive and negative voltage (default) | Positive voltage only

Allow for both positive and negative DC voltage output or limit the output to positive DC voltage.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

### Dependencies

If you set **Sample time (-1 for inherited)** to -1 and select the **Enable zero cancellation** option, the **Discretization sample time** parameter becomes visible.

#### **Discretization sample time — Sample time for discretization**

0.001 (default) | positive scalar

Time, in s, between consecutive discretizations. Discretization is required for zero cancellation.

### Dependencies

This parameter is only visible when both these conditions are met:

- **Sample time** is set to -1.
- **Enable zero cancellation** is selected.

**Enable zero cancellation — Feedforward zero cancellation**

off (default) | on

Option to use zero cancellation on the feedforward path.

**Dependencies**

If you select the **Enable zero cancellation** option and set **Sample time (-1 for inherited)** to -1, the **Discretization sample time** parameter becomes visible.

## Version History

Introduced in R2018a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

DC-DC Voltage Controller | DC Voltage Controller | PWM Generator

# DC Current Source

Constant current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The DC Current Source block represents a constant current source whose output current value is independent of the voltage across its terminals.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance *GMIN*:

- By default, *GMIN* matches the **GMIN** parameter of the Environment Parameters block, whose default value is 1e-12.
- To change *GMIN*, add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

#### + — Positive voltage

electrical

Electrical conserving port associated with the DC current source positive voltage

#### - — Negative voltage

electrical

Electrical conserving port associated with the DC current source negative voltage

## Parameters

### Constant value, DC — DC output current

0 A (default)

Value of the DC output current, in A.

### Include parallel conductance, GMIN — Whether to include parallel conductance

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

### Values of parallel conductance, *GMIN*

Include parallel conductance, <i>GMIN</i>		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b><i>GMIN</i></b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

DC Current Source | DC Voltage Source | Environment Parameters | Exponential Current Source | Pulse Current Source | Piecewise Linear Voltage Source | SFFM Current Source | Sinusoidal Current Source

### Functions

subcircuit2ssc

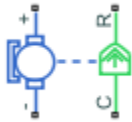
### Topics

“Additional Parameterization Workflows”  
 “Converting a SPICE Netlist to Simscape Blocks”  
 “Parameterize an Exponential Diode from SPICE Netlist”

## DC Motor

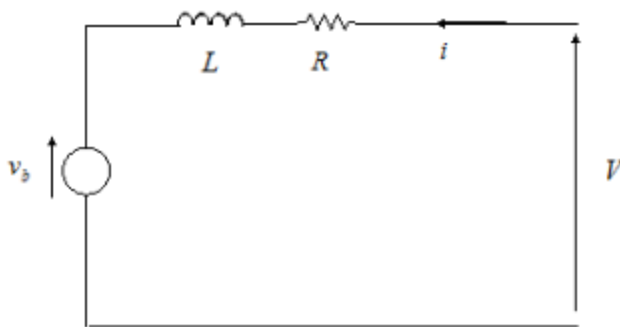
DC motor model with electrical and torque characteristics and fault modeling

**Library:** Simscape / Electrical / Electromechanical / Brushed Motors



### Description

The DC Motor block represents the electrical and torque characteristics of a DC motor using the following equivalent circuit model:



You specify the equivalent circuit parameters for this model when you set the **Model parameterization** parameter to *By equivalent circuit parameters*. The resistor  $R$  corresponds to the resistance you specify in the **Armature resistance** parameter. The inductor  $L$  corresponds to the inductance you specify in the **Armature inductance** parameter.

You can specify how to generate the magnetic field of the DC motor by setting the **Field type** parameter to the desired option. The permanent magnets in the motor induce the following back emf  $v_b$  in the armature:

$$v_b = k_v \omega$$

where  $k_v$  is the **Back-emf constant** and  $\omega$  is the angular velocity. The motor produces the following torque, which is proportional to the motor current  $i$ :

$$T_E = k_t i$$

where  $k_t$  is the **Torque constant**. The DC Motor block assumes that there are no electromagnetic losses. This means that mechanical power is equal to the electrical power dissipated by the back emf in the armature. Equating these two terms gives:

$$T_E \omega = v_b i$$

$$k_t i \omega = k_v \omega i$$

$$k_v = k_t$$



As a result, you specify either  $k_v$  or  $k_t$  in the block parameters.

If the magnetic field is generated from the current flowing through the windings, the **Back-emf constant** depends on the field current  $I_f$ :

$$k_v = L_{af}I_f$$

where  $L_{af}$  is the **Field-armature mutual inductance**.

The torque-speed characteristic for the DC Motor block is related to the parameters in the preceding figure. When you set the **Model parameterization** parameter to **By stall torque & no-load speed** or **By rated power, rated speed & no-load speed**, the block solves for the equivalent circuit parameters as follows:

- 1 For the steady-state torque-speed relationship,  $L$  has no effect.
- 2 Sum the voltages around the loop and rearrange for  $i$ :

$$i = \frac{V - v_b}{R} = \frac{V - k_v\omega}{R}$$

- 3 Substitute this value of  $i$  into the equation for torque:

$$T_E = \frac{k_t}{R}(V - k_v\omega)$$

When you set the **Model parameterization** parameter to **By stall torque & no-load speed**, the block uses the stall torque and no-load speed to determine values for  $R$  and  $k_t$  (and equivalently  $k_v$ ).

When you set the **Model parameterization** parameter to **By rated power, rated speed & no-load speed**, the block uses the rated speed and power to calculate the rated torque. The block uses the rated torque and no-load speed values to determine values for  $R$  and  $k_t$ .

The block models motor inertia  $J$  and damping  $\lambda$  for all values of the **Model parameterization** parameter. The resulting torque across the block is:

$$T = \frac{k_t}{R}(V - k_v\omega) - J\dot{\omega} - \lambda\omega$$

It is not always possible to measure rotor damping, and rotor damping is not always provided on a manufacturer datasheet. An alternative is to use the no-load current to infer a value for rotor damping.

For no-load, the electrically-generated mechanical torque must equal the rotor damping torque:

$$k_t i_{noload} = \lambda \omega_{noload}$$

where  $i_{noload}$  is the no-load current. If you select **By no-load current** for the **Rotor damping parameterization** parameter, then this equation is used in addition to the torque-speed equation to determine values for  $\lambda$  and the other equation coefficients.

The value for rotor damping, whether specified directly or in terms of no-load current, is taken into account when determining equivalent circuit parameters for **Model parameterization** options **By stall torque and no-load speed** and **By rated power, rated speed and no-load speed**.

When a positive current flows from the electrical **+** to **-** ports, a positive torque acts from the mechanical **C** to **R** ports.

### Faults

The DC Motor block allows you to model two types of faults:

- Armature winding fault — The armature winding fails and goes open circuit.
- Field winding fault — The field winding that creates the magnetic field fails and goes open circuit.

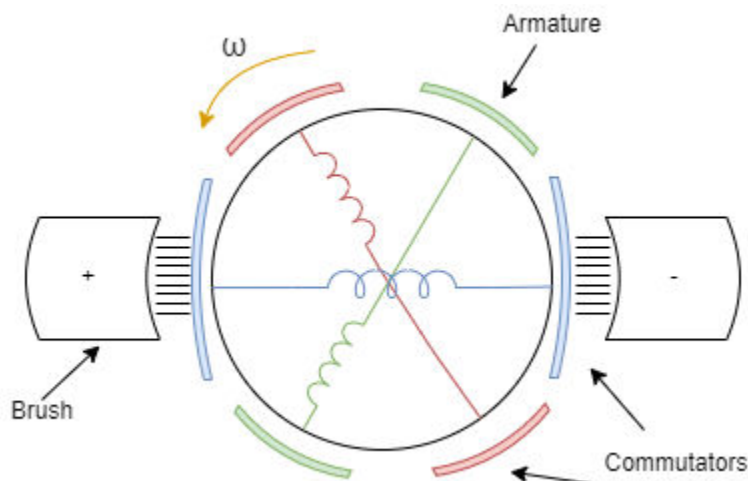
The block can trigger fault events:

- At a specific time (temporal fault).
- When a current limit is exceeded for longer than a specific time interval (behavioral fault).

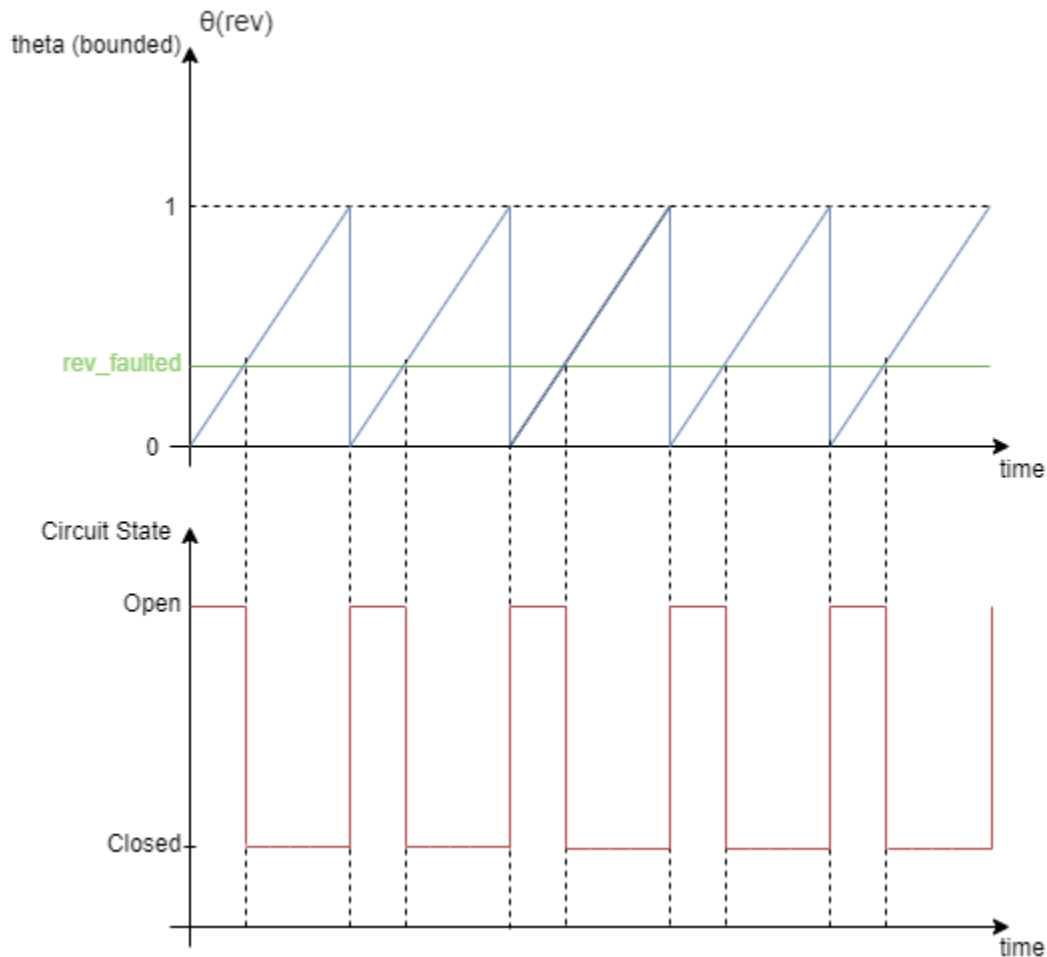
You can enable or disable these trigger mechanisms separately.

You can choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

This figure shows a generic representation of a DC motor, with brushes, armatures, commutators, and their windings:



If you set the **Enable armature winding open-circuit fault** parameter to On, the armature fails at the time specified by the **Simulation time for armature winding fault event** parameter for a temporal fault, or when the winding currents exceeds the value of the **Maximum permissible armature winding current** parameter for a behavioral fault. When the armature fails, the voltage source connected to this block observes an open circuit for a fraction of the total motor revolution, specified by the **Fraction of revolution during which armature is open-circuit** parameter, `rev_faulted`. This figure illustrates the circuit state behaviour for a certain `rev_faulted` during the entire revolution period:



### Predefined Parameterization

There are multiple available built-in parameterizations for the permanent magnet DC Motor block.

This pre-parameterization data allows you to set up the block to represent components by specific suppliers. The parameterizations of these DC motors match the manufacturer data sheets. To load a predefined parameterization, double-click the DC Motor block, click the **<click to select>** hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

## Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the DC motor positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the DC motor negative terminal.

#### C — Motor case

mechanical

Mechanical rotational conserving port associated with the DC motor case.

#### R — Motor rotor

mechanical

Mechanical rotational conserving port associated with the DC motor rotor.

#### F+ — Positive field

electrical

Electrical conserving port associated with the positive field winding.

### Dependencies

To enable this port, set **Field type** to Wound.

#### F- — Negative field

electrical

Electrical conserving port associated with the negative field winding.

### Dependencies

To enable this port, set **Field type** to Wound.

#### H — Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Whether to enable thermal ports

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

### Selected part — Predefined parameterization option

<click to select> (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see “Load Predefined Parameterizations”.

### Electrical Torque

#### Field type — Type of field

Permanent Magnet (default) | Wound

Select one of the following methods for field type:

- Permanent Magnet — Generate the DC Motor magnetic field using a permanent magnet.
- Wound — Generate the DC Motor magnetic field using the current flowing through the windings.

#### Model parameterization — Block parameterization

By equivalent circuit parameters (default) | By stall torque & no-load speed | By rated power, rated speed & no-load speed

Select one of the following methods for block parameterization:

- By equivalent circuit parameters — Provide electrical parameters for an equivalent circuit model of the motor.
- By stall torque & no-load speed — Provide torque and speed parameters that the block converts to an equivalent circuit model of the motor.
- By rated power, rated speed & no-load speed — Provide power and speed parameters that the block converts to an equivalent circuit model of the motor.

### Dependencies

To enable this parameter, set **Field type** to Permanent magnet.

#### Field resistance — Field resistance

100 Ohm (default)

Resistance of the magnetic field.

**Dependencies**

To enable this parameter, set **Field type** to Wound.

**Field inductance – Field inductance**

1 H (default)

Inductance of the magnetic field.

**Dependencies**

To enable this parameter, set **Field type** to Wound.

**Field-armature mutual inductance – Field-armature mutual inductance**

0.15 H (default)

Mutual inductance between the magnetic field and the conducting portion of the motor.

**Dependencies**

To enable this parameter, set **Field type** to Wound.

**Initial field current – Initial field current**

0 A (default)

Initial field current.

**Dependencies**

To enable this parameter, set **Field type** to Wound.

**Armature resistance – Armature resistance**

3.9 Ohm (default)

Resistance of the conducting portion of the motor.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Armature inductance – Armature inductance**

12e-6 H (default)

Inductance of the conducting portion of the motor. If you do not have information about this inductance, set the value of this parameter to a small, nonzero number.

**Define back-emf or torque constant – Constants specification**

Specify back-emf constant (default) | Specify torque constant

Indicate whether you will specify the motor's back-emf constant or torque constant. When you specify them in SI units, these constants have the same value, so you only specify one or the other in the block dialog box.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Back-emf constant — Back-emf constant**

0.072e-3 (default) | V/rpm

The ratio of the voltage generated by the motor to the speed at which the motor is spinning.

**Dependencies**

This parameter is visible only when you set the **Define back-emf or torque constant** parameter to Specify back-emf constant.

**Torque constant — Torque constant**

6.876e-4 N\*m/A (default)

The ratio of the torque generated by the motor to the current delivered to it.

**Dependencies**

This parameter is visible only when you set the **Define back-emf or torque constant** parameter to Specify torque constant.

**Stall torque — Stall torque**

0.24e-3 N\*m. (default)

The amount of torque generated by the motor when the speed is approximately zero.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By stall torque & no-load speed.

**No-load speed — No-load speed**

19100 rpm (default)

Speed of the motor when not driving a load.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By stall torque & no-load speed or By rated power, rated speed & no-load speed.

**Rated speed (at rated load) — Rated speed**

15000 rpm (default)

Motor speed at the rated mechanical power level.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Rated load (mechanical power) — Rated load**

0.08 W (default)

The mechanical power the motor is designed to deliver at the rated speed.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Rated DC supply voltage — Rated DC supply voltage**

1.5 V (default)

The voltage at which the motor is rated to operate.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to **By stall torque & no-load speed** or **By rated power, rated speed & no-load speed**.

**Rotor damping parameterization — Rotor damping parameterization**

By damping value (default) | By no-load current

Select one of the following methods to specify rotor damping:

- **By damping value** — Specify a value for rotor damping directly, by using the **Rotor damping** parameter in the **Mechanical** parameters.
- **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 for the **Mechanical** parameters.

**No-load current — No-load current**

0 A (default)

Specify the no-load current value, to be used for calculating the rotor damping. This parameter is only visible when you select **By no-load current** for the **Rotor damping parameterization** parameter.

**Dependencies**

This parameter is visible only when you set the **Rotor damping parameterization** parameter to **By no-load current**.

**DC supply voltage when measuring no-load current — DC supply voltage when measuring no-load current**

1.5 V (default)

Specify the DC supply voltage corresponding to the no-load current value, to be used for calculating the rotor damping.

**Dependencies**

This parameter is visible only when you set the **Rotor damping parameterization** parameter to **By no-load current**.

**Mechanical****Rotor inertia — Rotor inertia**0.01 g\*cm<sup>2</sup> (default)

Resistance of the rotor to change in motor motion. The value can be zero.

**Rotor damping — Rotor dissipated energy**

0 N\*m/(rad/s) (default)

Energy dissipated by the rotor. The value can be zero.



## Dependencies

This parameter is visible only when you set the **Rotor damping parameterization** parameter in the **Electrical Torque** setting to By damping value.

## Initial rotor speed — Initial rotor speed

0 rpm (default)

Speed of the rotor at the start of the simulation.

## Temperature Dependence

This tab appears only for blocks with exposed thermal port.

## Resistance temperature coefficient — Resistance temperature coefficient

$3.93e-3$  1/K (default)

Resistance temperature coefficient.

## Measurement temperature — Measurement temperature

25 degC (default)

The temperature for which motor parameters are defined.

## Thermal Port

This tab appears only for blocks with exposed thermal port.

## Thermal mass — Thermal mass

100 J/K (default)

The thermal mass is the energy required to raise the temperature by one degree.

## Initial temperature — Initial temperature

25 degC (default)

Temperature of the thermal port at the start of simulation.

## Faults

### Enable armature winding open-circuit fault — Whether to enable armature winding faults

Off (default) | On

Select **On** to enable faults modeling and expose the associated parameters in the **Faults** section, which enables you to select the reporting method and specify the time at which the armature winding fault triggers.

### Fraction of revolution during which armature is open-circuit — Fraction of revolution during which armature is open-circuit

1/3 (default)

Fraction of the revolution of the DC motor during which the armature winding is faulted and the voltage source observes an open circuit. For more information, see the “Faults” on page 1-388 section.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Faulted armature winding open-circuit conductance — Open-circuit conductance of faulted armature winding**

1e-5 S (default)

Open-circuit conductance when the armature winding is faulted.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Armature winding fault trigger — Fault trigger for armature winding**

Temporal (default) | Behavioral

Whether the trigger for the armature winding fault is temporal or behavioral.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On.

**Simulation time for armature winding fault event — Simulation time for armature winding fault event**

1 s (default)

Time at which the armature winding becomes faulted.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Temporal.

**Maximum permissible armature winding current — Maximum permissible armature winding current**

inf A (default)

Upper load-current threshold for open-circuit armature winding faults.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Behavioral.

**Time to fail when exceeding armature winding current — Time to fail when exceeding armature winding current**

1 s (default)

Amount of time that the winding current must continuously exceed the maximum permissible armature winding current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable armature winding open-circuit fault** to On and **Armature winding fault trigger** to Behavioral.

**Enable field winding open-circuit fault — Whether to enable field winding faults**

Off (default) | On

Select On to enable field winding faults modeling and expose the associated parameters in the **Faults** section, which enables you to select the reporting method and specify the time at which the field winding fault triggers.

**Dependencies**

To enable this parameter, set **Field type** to Wound.

**Faulted field winding open-circuit conductance — Open-circuit conductance of faulted field winding**

1e-5 S (default)

Open-circuit conductance when the field winding is faulted.

**Dependencies**

To enable this parameter, set **Enable field winding open-circuit fault** to On.

**Field winding fault trigger — Fault trigger for field winding**

Temporal (default) | Behavioral

Whether the trigger for the field winding fault is temporal or behavioral.

**Dependencies**

To enable this parameter, set **Enable field winding open-circuit fault** to On.

**Simulation time for field winding fault event — Simulation time for field winding fault event**

1 s (default)

Time at which the field winding becomes faulted.

**Dependencies**

To enable this parameter, set **Enable field winding open-circuit fault** to On and **Field winding fault trigger** to Temporal.

**Maximum permissible field winding current — Maximum permissible field winding current**

inf A (default)

Upper load-current threshold for open-circuit field winding faults.

**Dependencies**

To enable this parameter, set **Enable field winding open-circuit fault** to On and **Field winding fault trigger** to Behavioral.

**Time to fail when exceeding field winding current — Time to fail when exceeding field winding current**

1 s (default)

Amount of time that the winding current must continuously exceed the maximum permissible field winding current before a behavioral fault is triggered.

### Dependencies

To enable this parameter, set **Enable field winding open-circuit fault** to On and **Field winding fault trigger** to Behavioral.

### Reporting when a fault occurs — Reporting when a fault occurs

None (default) | Warn | Error

Simulation reporting when a fault occurs:

- None — Does not generate a warning or an error.
- Warn — Generates a warning.
- Error — The simulation stops and generates an error.

### Dependencies

To enable this parameter, set **Enable armature winding open-circuit fault** or **Enable field winding open-circuit fault** to On.

## Version History

Introduced in R2008a

### References

[1] Bolton, W. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 3rd edition Pearson Education, 2004..

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

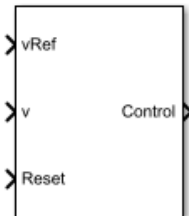
### See Also

Induction Machine (Single-Phase) | Motor & Drive (System Level) | Shunt Motor | Universal Motor

# DC Voltage Controller

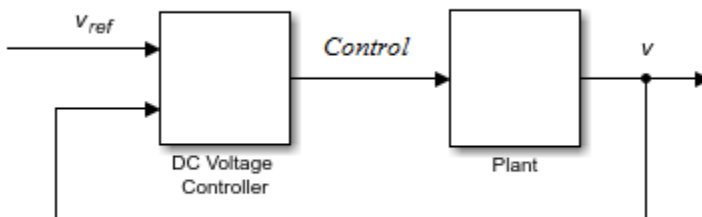
Discrete-time DC voltage PI control with feedforward zero cancellation and integral anti-windup

**Library:** Simscape / Electrical / Control / General Machine Control



## Description

The DC Voltage Controller block implements discrete-time PI-based DC voltage control. The block can implement zero cancellation in the feedforward path. To avoid saturation of the integral gain, the block can implement anti-windup gain.



## Equations

The equation that the DC Voltage Controller block uses to calculate the control signal is

$$control = \left( K_p + K_i \frac{T_s z}{z-1} \right) (v_{ref} - v),$$

where:

- $control$  is the control signal, which is expressed as a duty cycle or a current.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the sample time.
- $v_{ref}$  is the reference voltage.
- $v$  is the measured voltage.

The PI control calculation yields a zero in the closed-loop transfer function. To cancel the zero, the block uses this discrete-time zero-cancellation transfer function:

$$G_{ZC}(z) = \frac{\frac{T_s K_i}{K_p}}{z + \left( \frac{T_s - \frac{K_p}{K_i}}{\frac{K_p}{K_i}} \right)}.$$

To avoid saturation of the integrator output, the block uses an anti-windup mechanism. The integrator gain is then equal to

$$K_i + K_{aw}(\text{control}_{sat} - \text{control}_{unsat}),$$

where:

- $K_{aw}$  is the anti-windup gain.
- $\text{control}_{sat}$  is the saturated control signal, which the block calculates as  $\text{control}_{sat} = \min(\max(\text{control}_{unsat}, \text{control}_{min}), \text{control}_{max})$ ,

where:

- $\text{control}_{unsat}$  is the unsaturated control signal.
- $\text{control}_{min}$  is the lower limit for the control signal.
- $v_{max}$  is the upper limit for the control signal.

## Ports

### Input

#### **vRef — Reference DC voltage**

scalar

Desired DC output voltage for the plant.

Data Types: `single` | `double`

#### **v — Measured DC voltage**

scalar

Measured DC output voltage for the plant.

Data Types: `single` | `double`

#### **Reset — External reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `single` | `double`

### Output

#### **Control — Control signal**

scalar

Control signal, *control*, expressed as a duty cycle or a current.

Data Types: single | double

## Parameters

### Proportional gain — Proportional gain, $K_p$

0.1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

### Integral gain — Integral gain, $K_i$

50 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.

### Anti-windup gain — Anti-windup gain, $K_{aw}$

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.

### Control action upper limit — Upper limit for the control signal, $control_{max}$

1 (default) | positive scalar

Upper limit for the **Control** output signal. The value must be greater than the value of the **Control action lower limit** parameter.

### Control action lower limit — Lower limit for the control signal, $control_{min}$

0 (default) | non-negative scalar

Lower limit for the **Control** output signal.

### Sample time (-1 for inherited) — Block sample time

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

### Discretization sample time — Sample time for discretization

0.001 (default) | positive scalar

Time, in s, between consecutive discretizations.

### Time constant voltage filter — DC voltage filter time constant, $\tau$

0.001 (default) | positive scalar

Time constant,  $\tau$ , for the DC voltage filter.

## Dependencies

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** checkbox is selected.

**Enable zero cancellation — Feedforward zero cancellation**

off (default) | on

Option to use zero cancellation on the feedforward path.

**Dependencies**

If you select the **Enable zero cancellation** option and set **Sample time (-1 for inherited)** to -1, the **Discretization sample time** parameter becomes visible.

**Filter DC voltage — DC voltage filter option**

on (default) | off

To enable the filter on voltage measurement path, select the checkbox. To disable the filter, clear the check box.

**Dependencies**

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** check box is selected.

## Version History

Introduced in R2018a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

DC-DC Voltage Controller | DC Current Controller



# DC Voltage Source

Constant voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The DC Voltage Source block represents a constant voltage source whose output voltage value is independent of the current through the source.

## Ports

### Conserving

#### + — Positive voltage

electrical

Electrical conserving port associated with the DC current source positive voltage

#### - — Negative voltage

electrical

Electrical conserving port associated with the DC current source negative voltage

## Parameters

### Constant value, DC — DC output voltage

0 V (default)

Value of the DC output voltage, in V.

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

DC Current Source | Environment Parameters | Exponential Voltage Source | Pulse Voltage Source | Piecewise Linear Voltage Source | SFFM Voltage Source | Sinusoidal Voltage Source

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

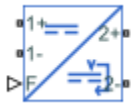
“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# DC-DC Converter

Behavioral model of power converter

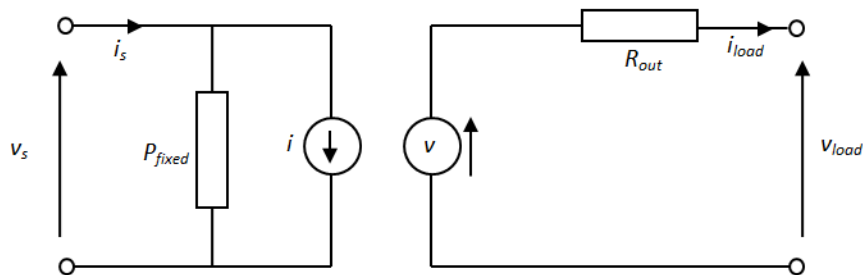
**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

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

This circuit illustrates the converter's behavior.



The  $P_{fixed}$  component draws a constant power and corresponds to converter losses that are independent of load current. The power drawn is set by the **Converter losses at zero output power** parameter value. The resistor  $R_{out}$  corresponds to losses that increase with load current, and is determined from the value you specify for the **Percentage efficiency at rated output power** parameter.

The voltage source is defined by the following equation:

$$v = v_{ref} - i_{load} D + i_{load} R_{out}$$

Where:

- $v_{ref}$  is the load side voltage set point, as defined by the value you specify for the **Output voltage reference demand** parameter. Alternatively, you can provide this value as input to the **Vref** port when the **Voltage reference** parameter is set to External.
- $D$  is the value you specify for the **Output voltage droop with output current** parameter. For more information, see "Voltage Droop" on page 1-406.

The current source value  $i$  is calculated so that the power flowing in to the converter equals the sum of the power flowing out plus the converter losses.

To specify the converter behavior when the voltage presented by the load is higher than the converter output voltage reference demand, use the **Power direction** parameter:

- Unidirectional power flow from supply to regulated side — Current is blocked by the off-state diode, and the current source current  $i$  is zero. Set the conductance of this diode using the **Diode off-state conductance** parameter.
- Bidirectional power flow — Power is transmitted to the supply side, and  $i$  becomes negative.

Optionally, the block can include voltage regulation dynamics. If you select **Specify voltage regulation time constant** for the **Dynamics** parameter, then a first-order lag is added to the equation defining the voltage source value. With the dynamics enabled, a load step change results in a transient change in output voltage, the time constant being defined by the **Voltage regulation time constant** parameter.

### Voltage Droop

Voltage droop is the loss in output voltage for a device that drives a load. Droop is important in voltage regulation circuits as it increases the tolerance of the systems towards load transients.

A separate value for the droop allows you to control how the output voltage varies with the load regardless of load-dependent losses. To enable droop, select the **Enable droop** parameter. To specify the droop, set the **Droop parameterization** parameter to:

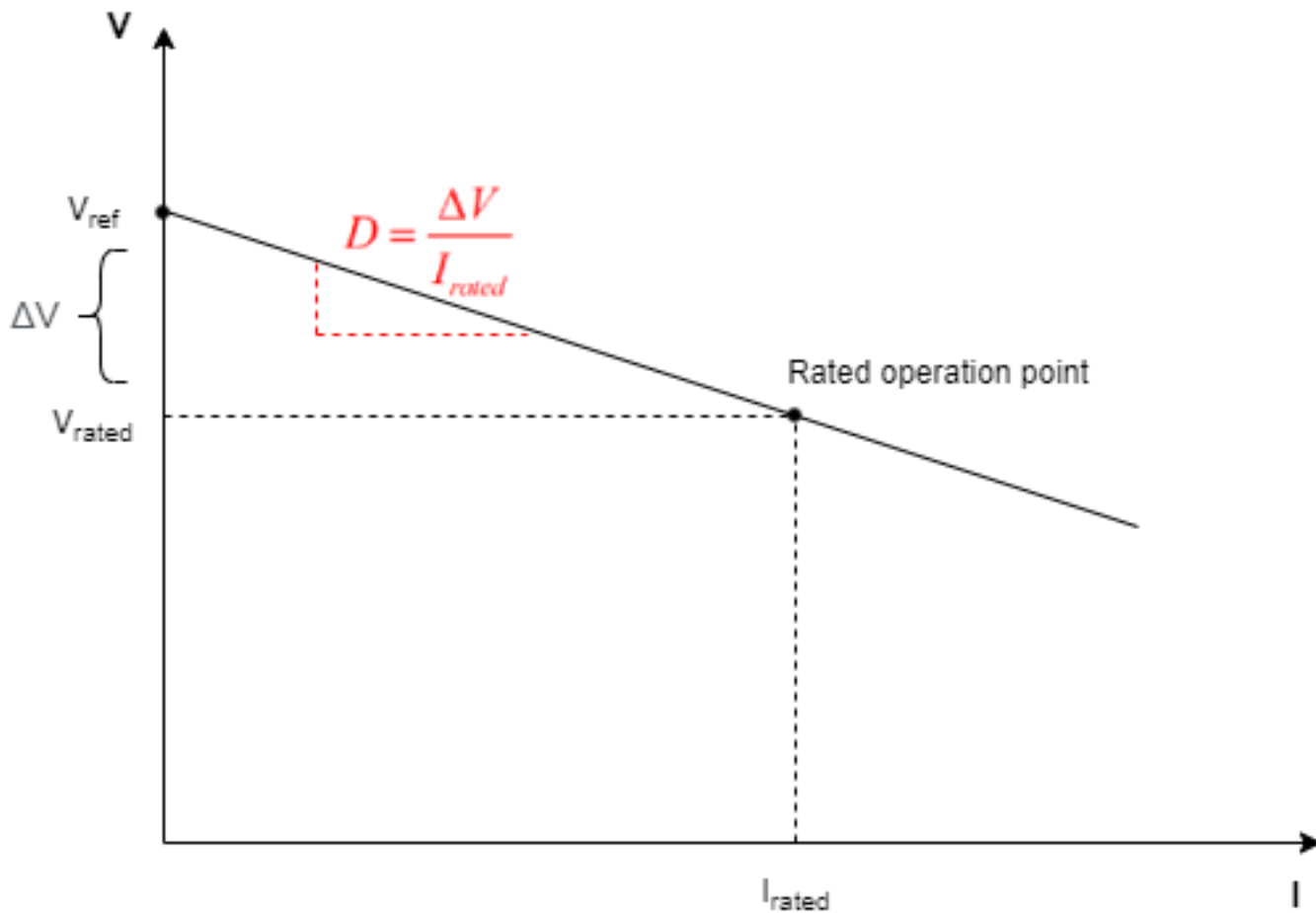
- By voltage droop with output current — Set the voltage droop directly by specifying the value of the **Output voltage droop with output current** parameter.
- By percent voltage droop at rated load — This equation calculates the voltage droop as a percent of the **Output voltage reference demand** parameter, based on the value of the **Voltage droop percent at rated load** parameter:

$$\Delta V = \frac{D_{\text{percent}}}{100} V_{\text{ref\_droop}}$$

$$D = \frac{\Delta V (V_{\text{ref\_droop}} - \Delta V)}{P_{\text{rated}}}$$

where  $V_{\text{ref\_droop}}$  is the value of the **Output voltage reference demand** parameter. If you set the **Voltage reference** parameter to **External**, the block uses the constant value of the **Voltage reference used to calculate droop percent** parameter to calculate the droop instead.

This plot shows how the block calculates the voltage droop:



The voltage droop,  $D$ , is the slope of the line, where  $I_{rated} = \frac{P_{rated}}{V_{rated}}$  and  $V_{rated} = V_{ref\_droop} - \Delta V$ .

### Tabulated efficiency

You can tabulate the DC-DC Converter block efficiency depending on the output current, input voltage, and temperature.

This equation defines the relationship between losses and efficiency:

$$loss(i_2, v_1, T) = \left[ 1 - \frac{eff(i_2, v_1, T)}{100} \right] abs(v_1 i_1) \text{ if } i_2 < 0 \text{ (power flow } 1 \rightarrow 2)$$

$$loss(i_2, v_1, T) = \left[ 1 - \frac{eff(i_2, v_1, T)}{100} \right] abs(v_2 i_2) \text{ if } i_2 > 0 \text{ (power flow } 2 \rightarrow 1)$$

where

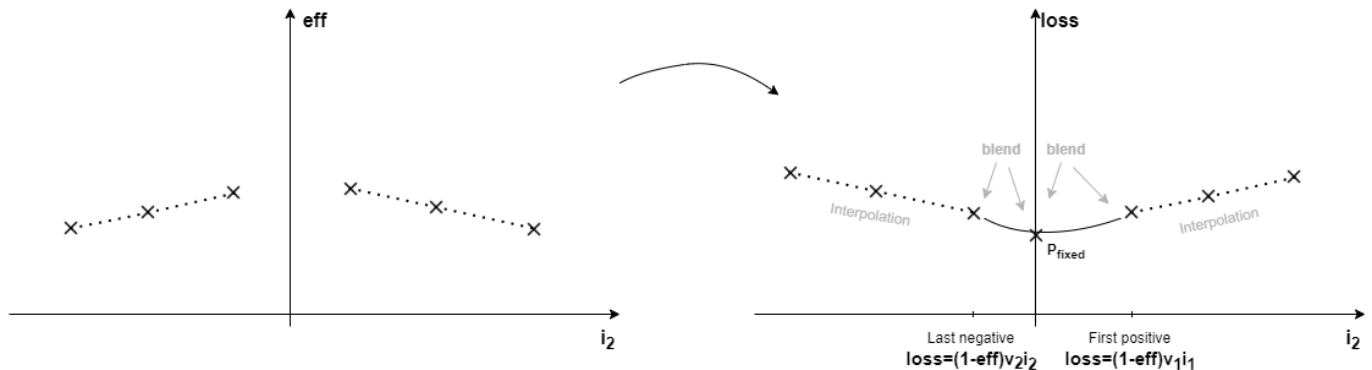
- $loss(i_2, v_1, T)$  are the DC-DC converter losses.
- $v_1$  is the input voltage.
- $i_1$  is the input current.

- $i_2$  is the output current.
- $T$  is the device temperature, if you expose the thermal port of this block.
- $eff(i_2, v_1, T)$  is the efficiency of the DC-DC converter as a function of the output current, input voltage, and temperature, as specified in **Percentage efficiency table,  $eff(I2, V1, T)$** .

If the output current is equal to 0, the converter losses are equal to **Converter losses at zero output power**.

These are the steps to compute the used power losses depending on the output current:

- 1 If the output current is less than the last negative current point or greater than the first positive current point in the **Vector of output currents for tabulated efficiencies, I2** parameter, the block uses the **Percentage efficiency table,  $eff(I2)$**  parameter to find the corresponding efficiency (with linear interpolation or nearest extrapolation) and then convert that efficiency to losses.
- 2 Else, the block blends the zero output current case to the last negative point or the first positive point in the table, as shown in this figure:



If you set the **Modeling option** parameter to Show thermal port, the block tabulates the efficiency depending on the temperature, in addition to the output current and input voltage. The losses calculation remains the same as the option with hidden thermal ports.

### Simulating When the Supply Cannot Match the Load

The DC-DC Converter block uses the **Minimum supply voltage to function** parameter to protect the network when the voltage supply cannot deliver the load requirements. This happens if you connect a switch in series with the supply and then turn it off, or if the supply connection has a series resistance which drops too much voltage. During the simulation, if the voltage falls below the value of the **Minimum supply voltage to function** parameter, the block relaxes the regulation of the output voltage. Once the voltage falls below 90% of the **Minimum supply voltage to function** parameter, the input of the converter behaves like a fixed resistance equal to the value of the **Input resistance if not functioning** parameter. Similarly, the output of the DC-DC Converter block behaves like a fixed resistance equal to the value of the **Output resistance if not functioning** parameter.

### Simulating Faults

You can use the physical signal input port **F** to simulate a converter failure.

To expose the fault port, **F**, in the **Faults** tab, set the **Enable output open-circuit fault** parameter to Yes.

You control the behavior in response to the physical signal fault input F by the parameters on the **Faults** tab of the block dialog box. With the default parameter settings:

- **Fault condition** is Output open circuit if  $F \geq \text{Fault threshold}$
- **Fault threshold** is 0.5

If a signal is connected to port F, then the block operates according to the parameter settings on the **Faults** tab. For example, if **Fault condition** is Output open circuit if  $F \geq \text{Fault threshold}$ , then when the signal at port F rises above the **Fault threshold** value, the converter stops operating. Zero current is taken from the supply side, and zero current is supplied to the load side.

### Model thermal effects

This block has one optional thermal port. To expose the thermal port, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain a thermal port.
- Show thermal port — The block contains one thermal conserving port.

The block transfers heat generated from electrical losses through a Controlled Heat Flow Rate Source to a Thermal Mass block. The electrical properties of the block do not change with temperature. Specify the thermal properties for this block using the parameters **Thermal mass** and **Initial temperature**.

### Assumptions

- The two electrical networks connected to the supply-side and regulated-side terminals must each have their own Electrical Reference block.
- The supply-side equation defines a power constraint on the product of the voltage,  $v_s$ , and the current,  $i_s$ . For simulation, the solver must be able to uniquely determine  $v_s$ . To ensure that the solution is unique, the block implements two assertions:
  - $v_s > 0$  — This assertion ensures that the sign of  $v_s$  is uniquely defined
  - $i_s < i_{max}$  — This assertion deals with the case when the voltage supply to the block has a series resistance

When there is a series resistance, there are two possible steady-state solutions for  $i_s$  that satisfy the power constraint, the one with the smaller magnitude being the desired one.

### Ports

#### Conserving

##### 1+ — Input positive terminal

electrical

Electrical conserving port associated with the positive terminal of the input side.

##### 1- — Input negative terminal

electrical

Electrical conserving port associated with the negative terminal of the input side.

**2+ – Output positive terminal**

electrical

Electrical conserving port associated with the positive terminal of the output side.

**2- – Output negative terminal**

electrical

Electrical conserving port associated with the negative terminal of the output side.

**Vref – Voltage reference**

physical

Physical signal input port that provides the external voltage reference signal.

**Dependencies**

To enable this port, set the **Voltage Reference** parameter to External.

**F – Fault signal**

physical

Physical signal input port that provides the external fault trigger signal.

**Dependencies**

To expose this port, in the **Faults** tab, set **Enable output open-circuit fault** to Yes.

**H – Thermal mass**

thermal

Thermal conserving port that represents the thermal mass. When you expose this port, provide additional parameters to define battery behavior at a second temperature. For more information, see the “Thermal” on page 1-416 parameters.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model thermal parameters.

**Main****Voltage reference – Voltage reference**

Internal (default) | External

Whether to model the voltage reference internally or externally.

**Output voltage reference demand – Voltage set point**

10 V (default)

Set point for the voltage regulator, and the output voltage value when there is no output current.



**Dependencies**

This port is visible only when the **Voltage Reference** parameter is set to `Internal`.

**Rated output power — Rated power**

10 W (default)

Output power for which the percentage efficiency value is given. This parameter is also used to calculate droop,  $D$ , if droop is specified as a percentage.

**Enable droop — Droop option**

On (default) | Off

Whether to enable voltage droop.

**Droop parameterization — Droop model**

By voltage droop with output current (default) | By percent voltage droop at rated load

Select one of the following methods for droop parameterization:

- By voltage droop with output current — Specify the absolute value of droop,  $D$ . This is the default option.
- By percent voltage droop at rated load — Specify droop,  $D$ , as a percentage at rated load.

**Dependencies**

To enable this parameter, select the **Enable droop** parameter.

**Output voltage droop with output current — Voltage droop at 1 A**

0.1 V/A (default)

Number of volts that the output voltage will drop from the set point for an output current of 1 A.

**Dependencies**

To enable this parameter, select the **Enable droop** parameter and set **Droop parameterization** to `By voltage droop with output current`.

**Voltage droop percent at rated load — Voltage droop percent at rated load**

2 (default)

Percentage by which the voltage drops compared to the nominal output voltage when supplying the rated load. If you set the **Voltage reference** parameter to `Internal`, the block uses the **Output voltage reference demand** parameter to calculate the droop percent. If you set the **Voltage reference** parameter to `External`, the block uses the **Voltage reference used to calculate droop percent** parameter instead.

**Dependencies**

To enable this parameter, select the **Enable droop** parameter and set **Droop parameterization** to `By percent voltage droop at rated load`.

**Voltage reference used to calculate droop percent — Voltage droop percent at rated load**

10 V (default)

Voltage reference that the **Voltage droop percent at rated load** parameter uses to calculate the voltage droop during the simulation.

**Dependencies**

To enable this parameter:

- Set **Voltage reference** to External.
- Select the **Enable droop** parameter.
- Set **Droop parameterization** to By percent voltage droop at rated load.

**Power direction – Power flow direction**

Unidirectional power flow from supply to regulated side (default) | Bidirectional power flow

Select one of the following methods for the direction of power conversion:

- **Unidirectional power flow from supply to regulated side** — Most small power regulators are unidirectional. This is the default option.
- **Bidirectional power flow** — Larger power converters can be bidirectional, for example, converters used in electric vehicles to allow regenerative braking.

**Diode off-state conductance – Unidirectional diode**

1e-8 1/Ohm (default)

Ideal diode incorporated on the output side to prevent current from being forced into the converter in the unidirectional configuration.

**Losses****Converter losses at zero output power – Constant losses**

1 W (default)

Power drawn by the  $P_{\text{fixed}}$  component in the equivalent circuit diagram, which corresponds to converter losses that are independent of load current.

**Converter losses – Converter losses**

Assume proportional to square of output current (default) | Tabulate efficiency

Converter losses parameterization.

**Percentage efficiency at rated output power – Rated efficiency**

80 (default)

Efficiency as defined by 100 times the output load power divided by the input supply power.

**Dependencies**

To enable this parameter, set **Converter losses** to Assume proportional to square of output current.

**Variables for tabulation – Tabulation variables**

Output current (default) | Output current and input voltage | Output current and temperature | Output current, input voltage, and temperature

Whether to tabulate efficiency on the output current, the input voltage, and the temperature. To enable the temperature options, you need to expose the thermal ports of the block.

**Vector of output currents for tabulated efficiencies, I2 – Output currents for tabulated efficiencies**

[-2, -1, 1, 2] A (default) | vector of positive ascending scalars

Vector of output currents for tabulated efficiencies. The size of this parameter must be equal to or greater than 3 and equal to the size of **Percentage efficiency table, eff(I)**.

**Dependencies**

To enable this parameter, set **Converter losses** to Tabulate efficiency.

**Percentage efficiency table, eff(I2) – Efficiency table in function of current**

[66, 76, 80, 74] (default) | vector of positive scalars

Efficiency table in function of the current, in percentage. The size of this parameter must be equal to the size of **Vector of output currents for tabulated efficiencies, I2**.

**Dependencies**

To enable this parameter, set **Converter losses** to Tabulate efficiency and **Variables for tabulation** to Output current.

**Vector of input voltages for tabulated efficiencies, V1 – Input voltages for tabulated efficiencies**

[15, 30, 45] V (default) | vector of positive ascending scalars

Vector of input voltages for tabulated efficiencies. The size of this parameter must be equal to or greater than 2.

**Dependencies**

To enable this parameter, set **Converter losses** to Tabulate efficiency and **Variables for tabulation** to Output current and input voltage or Output current, input voltage, and temperature.

**Percentage efficiency table, eff(I2,V1) – Efficiency table in function of current and voltage**

[66, 66, 66; 76, 76, 76; 80, 80, 80; 74, 74, 74] (default) | vector of positive scalars

Efficiency table in function of the current, in percentage.

**Dependencies**

To enable this parameter, set **Converter losses** to Tabulate efficiency and **Variables for tabulation** to Output current.

**Vector of temperatures for tabulated efficiencies, T – Temperatures for tabulated efficiencies**

[280, 300, 320] K (default) | vector of positive ascending scalars

Vector of temperatures for tabulated efficiencies. The size of this parameter must be equal to or greater than 2.

**Dependencies**

To enable this parameter, expose the thermal port of this block and set **Converter losses** to Tabulate efficiency.

**Percentage efficiency table,  $\text{eff}(I_2, T)$  — Efficiency table in function of current and temperature**

[66, 66, 66; 76, 76, 76; 80, 80, 80; 74, 74, 74] (default) | vector of scalars

Efficiency table in function of current and temperature, in percentage.

**Dependencies**

To enable this parameter, expose the thermal port of this block and set **Converter losses** to Tabulate efficiency and **Variables for tabulation** to Output current and temperature.

**Percentage efficiency table,  $\text{eff}(I_2, V_1, T)$  — Efficiency table in function of current, voltage, and temperature**

repmat([66; 76; 80; 74], 1, 3, 3) (default) | vector of scalars

Efficiency table in function of current, voltage, and temperature, in percentage.

**Dependencies**

To enable this parameter, expose the thermal port of this block and set **Converter losses** to Tabulate efficiency and **Variables for tabulation** to Output current, input voltage, and temperature.

**Dynamics****Dynamics — Dynamics model**

No dynamics (default) | Specify voltage regulation time constant

Specify whether to include voltage regulation dynamics:

- **No dynamics** — Do not consider the voltage regulation dynamics.
- **Specify voltage regulation time constant** — Add a first-order lag to the equation defining the voltage source value. With the dynamics enabled, a load step change results in a transient change in output voltage.

**Voltage regulation time constant — Dynamics time constant**

0.02 s (default)

Time constant associated with voltage transients when the load current is stepped.

**Dependencies**

To enable this parameter, set **Dynamics** to Specify voltage regulation time constant.

**Initial output voltage demand — Initial demand voltage**

10 V (default)

Value of  $v_{\text{ref}}$  at time zero. Normally,  $v_{\text{ref}}$  is defined by the **Output voltage reference demand** parameter. However, if you want to initialize the model with no transients when delivering a steady-state load current, you can set the initial  $v_{\text{ref}}$  value by using this parameter, and increase it accordingly to take account of output resistance and droop.

**Dependencies**

To enable this parameter, set **Dynamics** to Specify voltage regulation time constant.

**Off State****Minimum supply voltage to function – Minimum supply voltage to operate**

1 V (default)

Minimum supply voltage for the converter to operate and regulate the output voltage to the set point. When the supply voltage drops below 90% of the value of this parameter, the input to the DC-DC Converter block behaves like a fixed resistance equal to the value of the **Input resistance if not functioning** parameter. Similarly, the output of the DC-DC Converter block behaves like a fixed resistance equal to the value of the **Output resistance if not functioning** parameter.

**Input resistance if not functioning – Input resistance if not functioning**

1000 Ohm (default)

Input resistance when the supply voltage falls below 90% of the value of the **Minimum supply voltage to function** parameter.

**Output resistance if not functioning – Output resistance if not functioning**

1000 Ohm (default)

Output resistance when the supply voltage falls below 90% of the value of the **Minimum supply voltage to function** parameter.

**Faults****Enable output open-circuit fault – Option to enable faults**

No (default) | Yes

Whether to model output open-circuit faults in the block.

**Fault condition – Fault model**

Output open circuit if  $F \geq \text{Fault threshold}$  (default) | Output open circuit if  $F \leq \text{Fault threshold}$

Selects whether the converter is disabled by a signal that is high or low:

- Output open circuit if  $F \geq \text{Fault threshold}$  — Converter is disabled if the signal at port F rises above the threshold value. This is the default option.
- Output open circuit if  $F \leq \text{Fault threshold}$  — Converter is disabled if the signal at port F falls below the threshold value.

**Dependencies**

To enable this parameter, set **Enable output open-circuit fault** to Yes.

**Fault threshold – Fault limit**

0.5 (default)

Threshold value used to detect a fault.

### Dependencies

To enable this parameter, set **Enable output open-circuit fault** to Yes.

### Thermal

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Thermal mass — Thermal mass associated with the thermal port**

100 J/K (default)

Thermal mass associated with thermal port **H**. It represents the energy required to raise the temperature of the thermal port by one degree.

#### **Initial temperature — Initial temperature at the thermal port**

298.15 K (default)

Initial temperature associated with thermal port **H**.

## Version History

### Introduced in R2012b

#### **Support series-connected supply switch**

*Behavior changed in R2022b*

You can now connect a switch in series between the voltage supply and the DC-DC Converter block and toggle this switch on and off during the simulation.

To reflect this change, the **Maximum expected supply-side current** parameter is now called **Minimum supply voltage to function**. If the supply voltage drops below this value, both the input and the output sides of the converter behave like constant impedances and the converter ceases to convert.

If your model saved in an earlier release contains a DC-DC Converter block with dynamics enabled, and you configure this block to stop the power flowing from the load to the supply, the simulation results might behave slightly different. The voltage regulation time constant no longer lags the blocking diode that prevents reverse power flow and it is now instantaneous.

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

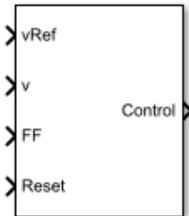
## See Also

Voltage Source | Current Source

# DC-DC Voltage Controller

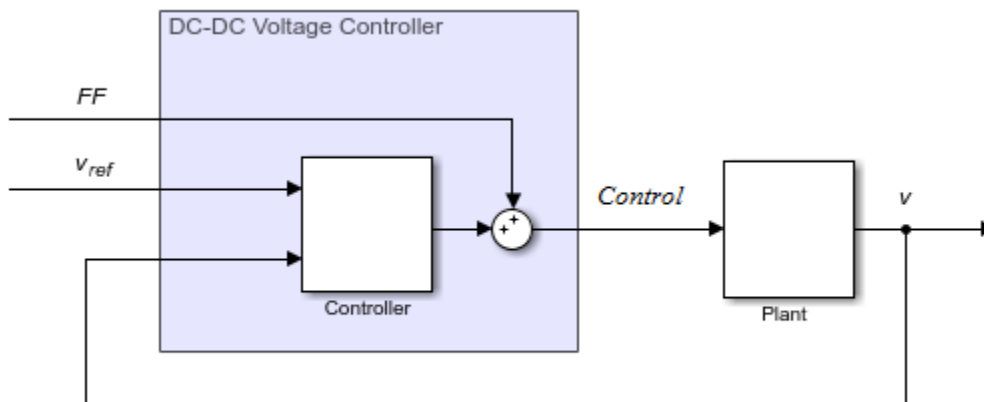
Discrete-time DC-DC voltage PI control with feedforward and optional integral anti-windup

**Library:** Simscape / Electrical / Control / General Machine Control



## Description

The DC-DC Voltage Controller block implements discrete-time proportional-integral (PI) DC-DC voltage control with feedforward,  $FF$ . The feedforward input optimizes the transient response. The block can output a duty cycle or a current control signal. To avoid saturation of the integral gain, the block can implement anti-windup gain.



## Equations

The equation that the DC-DC Voltage Controller block uses to calculate the control signal is

$$control = \left( K_p + K_i \frac{T_s z}{z-1} \right) (v_{ref} - v) + FF,$$

where:

- $control$  is the control signal, expressed as a duty cycle or a current.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the sample time.

- $v_{ref}$  is the reference voltage.
- $v$  is the measured voltage.
- $FF$  is the feedforward input.

To avoid saturation of the integrator output, the block uses an anti-windup mechanism. The integrator gain is then equal to

$$K_i + K_{aw}(control_{sat} - control_{unsat}),$$

where:

- $K_{aw}$  is the anti-windup gain.
- $control_{sat}$  is the saturated control signal, which the block calculates as  $control_{sat} = \min(\max(control_{unsat}, control_{min}), control_{max})$ ,

where:

- $control_{unsat}$  is the unsaturated control signal.
- $control_{min}$  is the lower limit for the control signal.
- $control_{max}$  is the upper limit for the control signal.

## Ports

### Input

#### **vRef — Reference DC voltage**

scalar

Desired DC output voltage for the plant.

Data Types: `single` | `double`

#### **v — Measured DC voltage**

scalar

Measured DC output voltage for the plant.

Data Types: `single` | `double`

#### **FF — Feedforward**

scalar

Feedforward term.

Data Types: `single` | `double`

#### **Reset — External reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `single` | `double`



## Output

### Control — Control signal

scalar

Control signal, *control*, expressed as a duty cycle or a current.

Data Types: `single` | `double`

## Parameters

### Proportional gain — Proportional gain, $K_p$

0.1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

### Integral gain — Integral gain, $K_i$

50 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.

### Anti-windup gain — Anti-windup gain, $K_{aw}$

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.

### Control action upper limit — Upper limit for the control signal, $control_{max}$

1 (default) | positive scalar

Upper limit for the **Control** output signal. The value must be greater than the value of the **Control action lower limit** parameter.

### Control action lower limit — Lower limit for the control signal, $control_{min}$

0 (default) | non-negative scalar

Lower limit for the **Control** output signal.

### Sample time (-1 for inherited) — Block sample time

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

### Time constant voltage filter — DC voltage filter time constant, $\tau$

0.001 (default) | positive scalar

Time constant,  $\tau$ , for the DC voltage filter.

**Dependencies**

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** checkbox is selected.

**Filter DC voltage — DC voltage filter option**

on (default) | off

To enable the filter on voltage measurement path, select the checkbox. To disable the filter, clear the check box.

**Dependencies**

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** check box is selected.

## Version History

Introduced in R2018a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

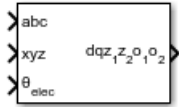
**Blocks**

DC Current Controller | DC Voltage Controller

## Decoupled Transform (Six-Phase)

Implement  $abc$  and  $xyz$  to  $dqz_1z_2o_1o_2$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



### Description

The Decoupled Transform (Six-Phase) block converts the time-domain components of a six-phase system in  $abc$  and  $xyz$  reference frames to direct, quadrature,  $z_1$ ,  $z_2$ ,  $o_1$ , and  $o_2$  components in a rotating reference frame. The three-phase systems  $abc$  and  $xyz$  are spatially shifted by 30 electrical degrees.

You can configure the block to align the phase  $a$ -axis of the six-phase system to either the  $d$ -axis or  $q$ -axis of the rotating reference frame at time  $t = 0$ .

$\theta = \omega t + \text{Offset}$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment, where:

- $\omega$  is the rotational speed of the rotational reference frame.
- $t$  is the time, in s, from the initial alignment.
- $\text{Offset} = 0$ ,  $d$  - axis alignment
- $\text{Offset} = -\frac{\pi}{2}$ ,  $q$  - axis alignment

### Equations

The Decoupled Transform (Six-Phase) block implements the transform as

$$\begin{bmatrix} d \\ q \\ z_1 \\ z_2 \\ o_1 \\ o_2 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{\pi}{6}\right) & \cos\left(\theta - \frac{5\pi}{6}\right) & \cos\left(\theta + \frac{\pi}{2}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{\pi}{6}\right) & -\sin\left(\theta - \frac{5\pi}{6}\right) & -\sin\left(\theta + \frac{\pi}{2}\right) \\ 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ x \\ y \\ z \end{bmatrix}$$

### Ports

#### Input

**abc** —  $a$ -,  $b$ -, and  $c$ -phase components  
vector

Components of the three-phase system in an *abc* reference frame, specified as a vector.

Data Types: `single` | `double`

### **xyz — x-, y-, and z-phase components**

vector

Components of the three-phase system in an *xyz* reference frame, specified as a vector.

Data Types: `single` | `double`

### **$\theta_{elec}$ — Rotational angle**

scalar

Angular position of the rotating reference frame, specified as a scalar.

Data Types: `single` | `double`

### **Output**

#### **dqz1z2o1o2 — Direct, Quadrature, $z_1$ , $z_2$ , $o_1$ , and $o_2$ components**

vector

Direct, quadrature,  $z_1$ -,  $z_2$ -,  $o_1$ -, and  $o_2$ -axis components of the system in the rotating reference frame, returned as a vector.

Data Types: `single` | `double`

### **Parameters**

#### **Phase-a axis alignment — $dqz_1z_2o_1o_2$ reference frame alignment**

Q-axis (default) | D-axis

Whether to align the *a*-phase vector of the *abc* reference frame to the *d*-axis or *q*-axis of the rotating reference frame.

## **Version History**

Introduced in R2021a

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

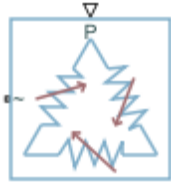
### **See Also**

Inverse Coordinate Transform (Five-Phase) | Coordinate Transform (Five-Phase) | Inverse Decoupled Transform (Six-Phase)

## Delta-Connected Variable Load

Three-phase variable load wired in delta configuration

**Library:** Simscape / Electrical / Passive / RLC Assemblies



### Description

The Delta-Connected Variable Load block models a three-phase variable load wired in a delta configuration. Each limb of the load contains a resistor. The block calculates the resistance required to draw the real power of the physical signal input **P** at the rated voltage that you specify. Therefore, the block can represent a real load.

To ensure that the resistance is always greater than zero, you specify the minimum real power that the load consumes. The minimum real power must be greater than zero.

### Equations

The resistance is defined by

$$R = 3 \frac{V_{Rated}^2}{P},$$

where:

- $R$  is the per-phase series resistance.
- $V_{Rated}$  is the RMS of rated line-line voltage.
- $P$  is the three-phase real power required. If the value of the **P** input port is less than the value of the **Minimum real power** parameter, the block uses the minimum real power to calculate the resistance.

### Ports

#### Input

##### **P** — Real power

physical signal

Physical signal input port associated with the real power.

#### Conserving

##### ~ — Three-phase port

electrical

Expandable three-phase electrical conserving port.

## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### **Rated voltage — Rated line-line voltage**

24e3 V (default) | positive scalar

RMS of rated line-line voltage for the resistance equation.

### **Minimum real power — Minimum real power**

1e3 W (default) | positive scalar

Minimum real power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0.

## Version History

**Introduced in R2021b**

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Delta-Connected Load

# Delta Reference (Three-Phase)

Internal reference point for delta-connected network

**Library:** Simscape / Electrical / Connectors & References



## Description

In a Simscape Electrical model, connect a Delta Reference (Three-Phase) block to any part of the three-phase system that is connected in a delta winding configuration. The block provides a reference point for the delta winding, representing the center of the line-line vector voltage triangle. The software calculates absolute node voltages relative to the voltage at this reference point.

For example, suppose you model a transmission system that consists of a generator connected in a wye configuration, a wye-delta transformer, a delta-wye transformer, and a load connected in wye. Connect a Delta Reference (Three-Phase) block to the part of the circuit between the two transformers.

## Ports

### Conserving

~ — **Expandable three-phase port**  
electrical

Expandable composite (**a**, **b**, **c**) three-phase port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Version History

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Delta-Connected Load | Electrical Reference

### **Topics**

“Expand and Collapse Three-Phase Ports on a Block”



## Delta-Connected Load

Three-phase load wired in delta configuration

**Library:** Simscape / Electrical / Passive / RLC Assemblies



### Description

The Delta-Connected Load block models a three-phase load wired in a delta configuration. Each limb of the load can include any combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

You can specify values for the R, L, and C components directly in terms of resistance, inductance, and capacitance, or by rated powers at a rated voltage and frequency.

- If you parameterize the block directly in terms of R, L, and C values, then for initialization provide a three-element row vector of initial voltages for a capacitor, and a three-element row vector of initial currents for an inductor.
- If you parameterize the block in terms of rated powers, then specify initial conditions in terms of an initial voltage, initial voltage phase, and initial frequency. For example, if the load is connected directly to a three-phase voltage source, then the initial conditions are identical to the source values for RMS line voltage, frequency, and phase shift. To specify zero initial voltage magnitude, set the initial voltage to 0.

For certain combinations of R, L, and C, for some circuit topologies, specify parasitic resistance or conductance values that help the simulation to converge numerically. These parasitic terms ensure that an inductor has a small parallel resistive path and that a capacitor has a small series resistance. When you parameterize the block in terms of rated powers, the rated power values do not account for these small parasitic terms. The rated powers represent only the R, L, and C values of the load itself.

### Block parameterization

The following two tables list the block parameters for each **Component structure**, based on the selected **Parameterization** option:

- Specify by rated power
- Specify component values directly

**Specify by Rated Power**

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
R	Rated voltage Real power	None	None
L	Rated voltage Rated electrical frequency Inductive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
C	Rated voltage Rated electrical frequency Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency
Series RL	Rated voltage Rated electrical frequency Real power Inductive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
Series RC	Rated voltage Rated electrical frequency Real power Capacitive reactive power	None	Terminal voltage magnitude Terminal voltage angle Frequency
Series LC	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency

Component Structure	Main Parameters	Parasitics Parameters	Initial Conditions Parameters
Series RLC	Rated voltage Rated electrical frequency Real power Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel RL	Rated voltage Rated electrical frequency Real power Inductive reactive power	None	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel RC	Rated voltage Rated electrical frequency Real power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel LC	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
Parallel RLC	Rated voltage Rated electrical frequency Real power Inductive reactive power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency

**Specify Component Values Directly**

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
R	Resistance	None	None
L	Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
C	Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Series RL	Resistance Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
Series RC	Resistance Capacitance	None	Initial capacitor voltage [ Va Vb Vc ]
Series LC	Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Series RLC	Resistance Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RL	Resistance Inductance	None	Initial inductor current [ Ia Ib Ic ]
Parallel RC	Resistance Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Parallel LC	Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RLC	Resistance Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]

**Variables**

Use the **Variables** tab to specify the priority and initial target values for the block variables before simulation. For more information, see “Set Priority and Initial Target for Block Variables”.

To enable the **Variables** tab, set the **Parameterization** parameter to Specify component values directly. The tab is not visible if you set **Component structure** to R.

## Ports

### Conserving

**~ — Three-phase port**  
electrical

Electrical expandable three-phase electrical conserving port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### Parameterization — Parameterization

Specify by rated power (default) | Specify component values directly

Select one of these values:

- **Specify by rated power** — Specify values for the R, L, and C components by rated powers at a rated voltage and frequency. This is the default.
- **Specify component values directly** — Specify values for the R, L, and C components directly in terms of resistance, inductance, and capacitance.

Switching the **Parameterization** value resets the **Component structure** value. Select the component parameterization option first, and then the component structure. If you later switch the **Parameterization** value, check the **Component structure** value and reselect it, if necessary.

#### Component structure — Desired combination

R (default) | L | C | Series RL | Series RC | Series LC | Series RLC | Parallel RL | Parallel RC | Parallel LC | Parallel RLC

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

#### Rated voltage — Rated voltage

2.4e4 V (default)

Voltage for which load powers are specified. This parameter is visible only when you specify values by rated power.

**Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter.

**Real power — Total real dissipated power**

1000 W (default) | positive scalar

Total real power dissipated by three-phase load when supplied at the rated voltage. The value must be greater than 0.

**Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes a resistor.

**Rated electrical frequency — Rated electrical frequency**

60 Hz (default)

Frequency for which reactive load powers are specified.

**Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter.

**Inductive reactive power — Inductive reactive power**

100 V\*A (default) | positive scalar

Total inductive reactive power taken by the three-phase load when supplied at the rated voltage. The value must be greater than 0.

**Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes an inductor.

**Capacitive reactive power — Total capacitive reactive power**

-100 V\*A (default) | negative scalar

Total capacitive reactive power taken by the three-phase load when supplied at the rated voltage. The value must be less than 0.

**Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes a capacitor.

**Resistance — Resistance**

1 Ohm (default)

The resistance of each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes a resistor.

**Inductance — Inductance**

0.001 H (default)

Inductance of each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes an inductor.

**Capacitance — Capacitance**

1e-6 F (default)

Capacitance in each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes a capacitor.

**Parasitics****Parasitic series resistance — Parasitic series resistance**

0 Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load.

**Parasitic parallel conductance — Parasitic parallel conductance**

0 1/Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load.

If you set **Component structure** to either **Series RL**, **Series LC**, or **Series RLC**, the parasitic parallel conductance is connected across the full branch of the block.

## Version History

Introduced in R2013b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

RLC (Three-Phase) | Wye-Connected Load

**Topics**

“Expand and Collapse Three-Phase Ports on a Block”



# Diffusion Resistor

Resistor model with velocity saturation and optional tolerance, operational limits, fault behavior, and noise

**Library:** Simscape / Electrical / Passive



## Description

The Diffusion Resistor block represents a resistor with velocity saturation, while letting you model the following effects:

- “Tolerances” on page 1-436
- “Operating Limits” on page 1-437
- “Faults” on page 1-437
- “Thermal Noise” on page 1-437
- “Model Thermal Effects” on page 1-438

You can turn these modeling options on and off independently of each other.

In its simplest form, the resistance of the Diffusion Resistor block is:

$$R = R_0 \left( 1 - p_2 - p_3 + p_2 \sqrt{1 + (\theta_2 v_{pn})^2} + p_3 \sqrt[3]{1 + |\theta_3 v_{pn}|^3} \right)$$

where:

- $R_0$  is zero-bias resistance.
- $p_2$  and  $p_3$  are the quadratic and linear voltage coefficients, respectively.
- $\theta_2$  and  $\theta_3$  are inverse voltages for quadratic and linear voltage activation, respectively.
- $v_{pn}$  is applied voltage across the resistor.

At low bias,

$$R \approx R_0 \left( 1 + \frac{p_2 \theta_2^2 v_{pn}^2}{2} \right)$$

and therefore  $p_2$  and  $\theta_2$  determine the low-bias quadratic behavior of the resistor.

At high bias,

$$R \approx R_0 (1 - p_2 - p_3 + |v_{pn}| (p_2 \theta_2 + p_3 \theta_3))$$

and therefore  $p_3$  and  $\theta_3$  impact only the high-bias linear behavior of the resistor.

You can use the voltage-dependence of the resistance to model velocity saturation in a diffused resistor. For sufficiently high voltage,

$$i_{sat} = \frac{1}{R_0(p_2\theta_2 + p_3\theta_3)}$$

where  $i_{sat}$  is saturation current.

### Simplified Parameterization

The simplified parameterization model assumes that the quadratic and linear coefficients are the same. This is one of the recommended assumptions for the r2\_cmc model provided by the Compact Model Coalition, as a reasonable initial guess when performing parameter extraction. With this assumption, it is possible to define two new parameters, **Critical voltage** and **Corner voltage**, which provide a simpler means for parameterizing models:

$$p_2 = p_3 = \frac{v_{co}}{2v_{crit}}$$

$$\theta_2 = \theta_3 = \frac{1}{2v_{co}}$$

where:

- $v_{crit}$  is critical voltage.
- $v_{co}$  is corner voltage.

At high voltage,

$$\frac{dR}{dv_{pn}} \approx \frac{R_0}{v_{crit}}$$

and therefore, critical voltage is the reciprocal of the slope of the increase of  $R/R_0$  with voltage.

With this parameterization, the saturation current is

$$i_{sat} = \frac{v_{crit}}{R_0}$$

### Tolerances

You can apply tolerances to the nominal value you provide for the **Resistance** parameter. Datasheets typically provide a tolerance percentage for a given resistor type. The table shows how the block applies tolerances and calculates resistance based on the selected **Tolerance application** option.

Option	Resistance Value
None – use nominal value	$R_0$
Random tolerance	Uniform distribution: $R_0 \cdot (1 - tol + 2 \cdot tol \cdot rand)$ Gaussian distribution: $R_0 \cdot (1 + tol \cdot randn / nSigma)$
Apply maximum tolerance value	$R_0 \cdot (1 + tol)$
Apply minimum tolerance value	$R_0 \cdot (1 - tol)$

In the table,

- $R_0$  is the **Resistance** parameter value, nominal zero-bias resistance.
- $tol$  is fractional tolerance, **Tolerance (%)** /100.
- $nSigma$  is the value you provide for the **Number of standard deviations for quoted tolerance** parameter.
- `rand` and `randn` are standard MATLAB functions for generating uniform and normal distribution random numbers.

---

**Note** If you choose the `Random tolerance` option and you are in "Fast Restart" mode, the random tolerance value is updated on every simulation if at least one between the fractional tolerance,  $tol$ , or the **Number of standard deviations for quoted tolerance**,  $nSigma$ , is set to Run-time and is defined with a variable (even if you do not modify that variable).

---

### Operating Limits

You can specify operating limits in terms of power and maximum working voltage. If you set the **Modeling option** parameter to `Show thermal port` (see "Model Thermal Effects" on page 1-438), you can also specify operating limits in terms of temperature.

When an operating limit is exceeded, the block can either generate a warning or stop the simulation with an error. For more information, see the "Operating Limits" on page 1-442 parameters section.

### Faults

The Diffusion Resistor block allows you to model an electrical fault as an instantaneous change in resistance. The block can trigger fault events:

- At a specific time
- When a current limit is exceeded for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault takes precedence. In other words, component fails no more than once per simulation.

When the resistor fails, its resistance is changed to the value you specify for the **Faulted zero-voltage resistance** parameter. You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

### Thermal Noise

The Diffusion Resistor block can generate thermal noise current. If you set the **Noise mode** parameter to `Enabled`, then the block includes a noise current source connected in parallel to the diffusion resistor.

If the sampling time is  $h$ , then the thermal noise is given by:

$$i_N = \sqrt{2kT/R} \frac{N(0, 1)}{\sqrt{h}}$$

where:

- $k$  is the Boltzmann constant,  $1.3806504e-23$  J/K.
- $T$  is temperature.
- $R$  is resistance.
- $N$  is a Gaussian random number with zero mean and standard deviation of one.
- $2kT/R$  is the double-sided thermal noise power distribution (the single-sided equivalent is  $4kT/R$ ).

The block generates Gaussian noise by using the PS Random Number source in the Simscape Foundation library. You can control the random number seed by setting the **Repeatability** parameter:

- **Not repeatable** — Every time you simulate your model, the block resets the random seed using the MATLAB random number generator:  

$$\text{seed} = \text{randi}(2^{32}-1);$$
- **Repeatable** — The block automatically generates a seed value and stores it inside the block, to always start the simulation with the same random number. This auto-generated seed value is set when you add a Diffusion Resistor block from the block library to the model. When you make a new copy of the Diffusion Resistor block from an existing one in a model, a new seed value is generated. The block sets the value using the MATLAB random number generator command shown above.
- **Specify seed** — If you select this option, the additional **Seed** parameter lets you directly specify the random number seed value.

### Model Thermal Effects

You can expose thermal ports to specify how the resistance value changes with temperature and to set the thermal mass. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains one thermal conserving port.

Use the **Variables** settings to set the initial temperature target.

If you set the **Modeling option** parameter to **Show thermal port**, the defining equation for the resistance is augmented with additional temperature scaling:

$$R = R_0 \left( 1 + T_{C1}^{eff} \Delta T + T_{C2}^{eff} (\Delta T)^2 \right) \left( 1 - p_2 - p_3 + p_2 \sqrt{1 + (\theta_2 v_{pn})^2} + p_3 \sqrt[3]{1 + |\theta_3 v_{pn}|^3} \right)$$

where  $T_{C1}^{eff}$  and  $T_{C2}^{eff}$  are the linear and quadratic temperature scaling coefficients, respectively.

$$\Delta T = T_{sim} - T_{meas}$$

where:

- $T_{sim}$  is simulation temperature.
- $T_{meas}$  is measurement temperature.

With the thermal port exposed, the generated noise uses the temperature at the thermal port when determining the instantaneous noise value. Exposing the thermal port also extends the options on the **Operating Limits** tab as follows:

- The **Power rating** parameter becomes temperature dependent. You define a temperature up to which the full power rating is available, plus a higher temperature for which the power rating is

reduced to zero. It is assumed that the power rating decreases linearly with temperature between these two values.

- An additional parameter, **Operating temperature range, [Tmin Tmax]**, lets you define the valid temperature range for block operation.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

This section appears only for the blocks with exposed thermal port. The **Temperature** variable lets you specify a high-priority target for the temperature at the start of simulation.

## Basic Assumptions and Limitations

Simulating with noise enabled slows down simulation. Choose the sample time ( $h$ ) so that noise is generated only at frequencies of interest, and not higher.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the resistor positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the resistor negative terminal.

#### H – Resistor thermal mass

thermal

Thermal conserving port that represents the resistor thermal mass.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and specify how the resistance value changes with temperature and to set the thermal mass.

**Main****Resistance — Nominal zero-bias resistance**

1 Ohm (default)

The zero-bias resistance, used as the nominal resistance value. Resistance value must be greater than zero. If you set **Modeling option** to Show thermal port, this is the zero-bias resistance at a temperature equal to the **Measurement temperature** parameter in the “Thermal” on page 1-446 section.

**Tolerance (%) — Resistor tolerance, in percent**

5 (default)

The resistor tolerance as defined on the manufacturer datasheet.

**Tolerance application — Select how to apply tolerance during simulation**

None — use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation:

- **None — use nominal value** — The block does not apply tolerance, uses the nominal resistance value. This is the default.
- **Random tolerance** — The block applies random offset to the resistance value, within the tolerance value limit. You can choose Uniform or Gaussian distribution for calculating the random number by using the **Tolerance distribution** parameter.
- **Apply maximum tolerance value** — The resistance is increased by the specified tolerance percent value.
- **Apply minimum tolerance value** — The resistance is decreased by the specified tolerance percent value.

**Tolerance distribution — Select the distribution type**

Uniform (default) | Gaussian

Select the distribution type for random tolerance:

- **Uniform** — Uniform distribution
- **Gaussian** — Gaussian distribution

**Dependencies**

Enabled when the **Tolerance application** parameter is set to Random tolerance.

**Number of standard deviations for quoted tolerance — Used for calculating the Gaussian random number**

4 (default)

Number of standard deviations for calculating the Gaussian random number.

**Dependencies**

Enabled when the **Tolerance distribution** parameter is set to Gaussian.

**Parameterization — Select parameterization method**

Simplified (default) | Advanced

Select how to apply tolerance during simulation:

- **Simplified** — Assume that the quadratic and linear coefficients are the same, and define block behavior using the **Critical voltage** and **Corner voltage** parameters.
- **Advanced** — Explicitly specify values for the quadratic and linear voltage coefficients and for the inverse voltages for quadratic and linear voltage activation.

#### **Critical voltage — Critical voltage for saturation**

4 V (default)

Critical voltage for the saturation mechanism. You can determine this parameter value by taking the reciprocal of the slope of the increase of  $R/R_0$  with voltage.

##### **Dependencies**

Enabled when the **Parameterization** parameter is set to **Simplified**.

#### **Corner voltage — Voltage at which the resistance increase starts to occur**

2 V (default)

Corner voltage, at which the resistance increase starts to occur. The **Corner voltage** must be less than the **Critical voltage**.

##### **Dependencies**

Enabled when the **Parameterization** parameter is set to **Simplified**.

#### **Quadratic voltage coefficient — Coefficient $p_2$**

0.25 (default)

Coefficient  $p_2$  from the defining equation.

##### **Dependencies**

Enabled when the **Parameterization** parameter is set to **Advanced**.

#### **Inverse voltage for quadratic voltage activation — Coefficient $\theta_2$**

0.25 1/V (default)

Coefficient  $\theta_2$  from the defining equation.

##### **Dependencies**

Enabled when the **Parameterization** parameter is set to **Advanced**.

#### **Linear voltage coefficient — Coefficient $p_3$**

0.25 (default)

Coefficient  $p_3$  from the defining equation.

##### **Dependencies**

Enabled when the **Parameterization** parameter is set to **Advanced**.

#### **Inverse voltage for linear voltage activation — Coefficient $\theta_3$**

0.25 1/V (default)

Coefficient  $\theta_3$  from the defining equation.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Advanced.

**Operating Limits****Enable operating limits — Select Yes to enable reporting when the operational limits are exceeded**

No (default) | Yes

Select Yes to enable reporting when the operational limits are exceeded. The associated parameters in the **Operating Limits** section become visible to let you select the reporting method and specify the operating limits in terms of power and maximum working voltage. Parameters that specify operating limits in terms of temperature are visible only for blocks with exposed thermal port (see “Model Thermal Effects” on page 1-438). The default value is No.

**Reporting when operating limits exceeded — Select the reporting method**

Warn (default) | Error

Select what happens when an operating limit is exceeded:

- Warn — The block issues a warning.
- Error — Simulation stops with an error.

**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Maximum working voltage — Maximum voltage allowed for normal block operation**

100 V (default)

Maximum voltage magnitude allowed for normal block operation.

**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Power rating — Maximum power allowed for normal block operation**

1 W (default)

Maximum power allowed for normal block operation.

If you expose the thermal port of the block, this parameter becomes temperature dependent. The value you specify for the **Power rating** parameter applies up to the temperature specified by the **Temperature below which full power rating is available** parameter value. Then the power rating decreases linearly with temperature, until it becomes 0 at temperature specified by the **Temperature above which power rating is reduced to zero** parameter value.

**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Temperature below which full power rating is available — Maximum temperature where full power rating still applies**

70 °C (default)



Maximum temperature where full power rating, specified by the **Power rating** parameter value, still applies.

#### Dependencies

To enable this parameter, set **Modeling option** to Show thermal port.

#### Temperature above which power rating is reduced to zero — Temperature where power rating becomes 0

155 °C (default)

Temperature where power rating becomes 0. Above this temperature, the simulation always issues an assertion regardless of dissipated power. This parameter value must be higher than **Temperature below which full power rating is available**.

#### Dependencies

To enable this parameter, set **Modeling option** to Show thermal port.

#### Operating temperature range, [Tmin Tmax] — Minimum and maximum temperature values allowed for normal block operation

[-50 150] °C (default)

A row vector of length 2 specifying minimum and maximum temperature values allowed for normal block operation. The first element is the lowest allowable operating temperature, and the second element is the largest allowable operating temperature.

#### Dependencies

To enable this parameter, set **Modeling option** to Show thermal port.

#### Faults

##### Enable faults — Select Yes to enable faults modeling

No (default) | Yes

Select Yes to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

##### Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

##### Faulted zero-voltage resistance — Resistance when block is in faulted state

inf Ohm (default)

Zero-voltage resistance between the + and - ports when the block is in the faulted state.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Enable temporal fault trigger – Select Yes to enable time-based fault triggering**

No (default) | Yes

Select Yes to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Simulation time for fault event – Time before entering faulted state**

1 s (default)

Set the simulation time at which you want the block to enter the faulted state.

**Dependencies**

Enabled when the **Enable temporal fault trigger** parameter is set to Yes.

**Enable behavioral fault trigger – Select Yes to enable behavioral fault triggering**

No (default) | Yes

Select Yes to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Maximum permissible current – Current threshold to fault transition**

1 A (default)

Specify the maximum permissible current value. If the current exceeds this value for longer than the **Time to fail when exceeding maximum permissible current** parameter value, then the block enters the faulted state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Time to fail when exceeding maximum permissible current – Maximum length of time the current exceeds the threshold**

1 s (default)

Set the maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

## Noise

### Noise mode — Select whether to model thermal noise current

Disabled (default) | Enabled

Select whether to model thermal noise current:

- **Disabled** — No noise is produced by the resistor.
- **Enabled** — Resistor generates thermal noise current, and the associated parameters become visible in the **Noise** section.

### Sample time — Rate at which the noise source is sampled

1e-3 s (default)

Defines the rate at which the noise source is sampled. Choose it to reflect the frequencies of interest in your model. Making the sample time too small will unnecessarily slow down your simulation.

#### Dependencies

Enabled when the **Noise mode** parameter is set to Enabled.

### Repeatability — Select the noise control option

Not repeatable (default) | Repeatable | Specify seed

Select the noise control option:

- **Not repeatable** — The random sequence used for noise generation is not repeatable.
- **Repeatable** — The random sequence used for noise generation is repeatable, with a system-generated seed.
- **Specify seed** — The random sequence used for noise generation is repeatable, and you control the seed by using the **Seed** parameter.

#### Dependencies

Enabled when the **Noise mode** parameter is set to Enabled.

### Auto-generated seed used for repeatable option — Auto-generated random number seed

random real number

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Seed** parameter.

#### Dependencies

Enabled when the **Repeatability** parameter is set to Repeatable.

### Seed — Random number seed

0 (default)

Seed used by the noise random number generator.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Specify seed.

**Device simulation temperature — Temperature of resistor at the start of the simulation**

25 °C (default)

The temperature of the resistor at the start of the simulation.

**Dependencies**

Enabled when the **Noise mode** parameter is set to Enabled.

For blocks with an exposed thermal port, this parameter is disabled. Instead, use the **Variables** tab to set the initial temperature target. For more information, see “Variables” on page 1-439.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Resistance linear temperature coefficient — Specifies how the resistance value changes with temperature**

0 1/K (default)

The coefficient  $T_{C1}^{eff}$  in the equation that describes resistance as a function of temperature. See “Model Thermal Effects” on page 1-438 for details.

**Resistance quadratic temperature coefficient — Specifies how the resistance value changes with temperature**

0 1/K<sup>2</sup> (default)

The coefficient  $T_{C2}^{eff}$  in the equation that describes resistance as a function of temperature. See “Model Thermal Effects” on page 1-438 for details.

**Measurement temperature — Temperature corresponding to nominal resistance**

25 °C (default)

The temperature  $T_0$ , for which the nominal resistance  $R$  is specified.

**Thermal mass — Thermal mass associated with port H**

100 J/K (default)

Thermal mass associated with the thermal port H. It represents the energy required to raise the temperature of the thermal port by one degree.

## Version History

Introduced in R2017b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Fault | Resistor

# Diode

Piecewise or exponential diode

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Diode block can represent either a piecewise linear diode, an exponential diode, or a diode with tabulated I-V curve. Use the **Fidelity level** parameter to select the level of detail in the dynamics model.

### Piecewise Linear Diode

The piecewise linear diode model is the same model as the **Simscape > Foundation Library > Electrical > Electrical Elements > Diode** block, with the addition of a fixed junction capacitance and optional charge dynamics. If the diode forward voltage exceeds the value specified in the **Forward voltage** parameter, the diode behaves as a linear resistor with the resistance specified in the **On resistance** parameter. Otherwise, the diode behaves as a linear resistor with the small conductance specified in the **Off conductance** parameter. Zero voltage across the diode results in zero current flowing.

For more information about the Diode block in the Simscape Foundation library, see Diode.

### Exponential Diode

The exponential diode model represents the following relationship between the diode current  $I$  and the diode voltage  $V$ :

$$I = IS \cdot \left( e^{\frac{qV}{NkT_{m1}}} - 1 \right) V > -BV$$

$$I = -IS \cdot \left( e^{\frac{-q(V+Vz)}{kT_{m1}}} - e^{\frac{qV}{NkT_{m1}}} \right) V \leq -BV$$

where:

- $q$  is the elementary charge on an electron (1.602176e-19 coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $BV$  is the **Reverse breakdown voltage** parameter value.
- $N$  is the emission coefficient.
- $IS$  is the saturation current.
- $T_{m1}$  is the temperature at which the diode parameters are specified, as defined by the **Measurement temperature** parameter value.

When  $(qV / NkT_{m1}) > 80$ , the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} - 79)e^{80}$ , which matches the gradient of the diode current at  $(qV / NkT_{m1}) = 80$  and extrapolates linearly. When  $(qV / NkT_{m1}) < -79$ ,

the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} + 80)e^{-79}$ , which also matches the gradient and extrapolates linearly. Typical electrical circuits do not reach these extreme values. The block provides this linear extrapolation to help convergence when solving for the constraints during simulation.

When you select Use parameters **IS** and **N** for the **Parameterization** parameter, you specify the diode in terms of the **Saturation current IS** and **Emission coefficient N** parameters. When you select Use two I-V curve data points for the **Parameterization** parameter, you specify two voltage and current measurement points on the diode I-V curve and the block derives the **IS** and **N** values. The block then calculates **IS** and **N** as follows:

- $N = ((V_1 - V_2)/V_t)/(\log(I_1) - \log(I_2))$
- $IS = (I_1/(\exp(V_1/(NV_t)) - 1) + I_2/(\exp(V_2/(NV_t)) - 1))/2$

where:

- $V_t = kT_{m1} / q$ .
- $V_1$  and  $V_2$  are the values in the **Voltages [V1 V2]** vector.
- $I_1$  and  $I_2$  are the values in the **Currents [I1 I2]** vector.

When you select Use an I-V data point and **IS** for the **Parameterization** parameter, then the block calculates **N** as follows:

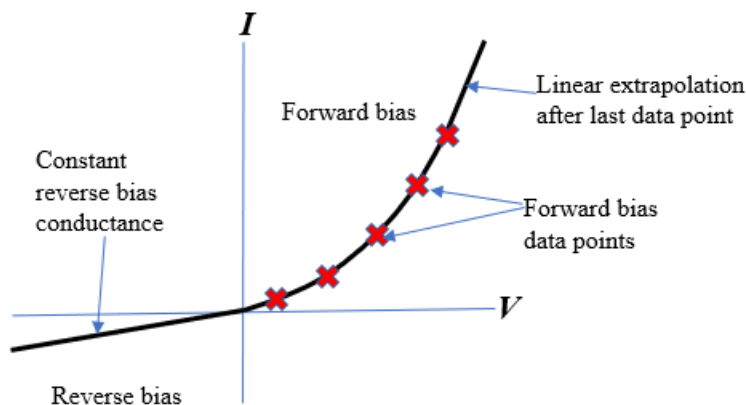
$$N = V_1 / \left( V_t \log \left( \frac{I_1}{IS} + 1 \right) \right)$$

When you select Use an I-V data point and **N** for the **Parameterization** parameter, then the block calculates **IS** as follows:

$$IS = I_1 / (\exp(V_1 / (NV_t)) - 1)$$

### Tabulated Diode

To model a tabulated diode, set the **Diode model** parameter to Tabulated I-V curve. This figure shows the implementation of the tabulated diode option:



When choosing this parameterization, you must provide the data for the forward bias only. If the diode is reverse biased, it is modeled as a constant off-state conductance specified in the **Off**

**conductance** parameter instead. The value of the **Off conductance** must be less than the gradient of the forward I-V curve for small positive voltages.

The block implements the diode using a smooth interpolation option. If the diode exceeds the provided tabulated data range, the block uses a linear extrapolation technique at the last voltage-current data point.

---

**Note** The tabulated diode does not model the reverse breakdown.

---

## Junction Capacitance

The Diode block models the junction capacitance only if you set the **Fidelity level** parameter to **Include capacitance and charge dynamics**. To include a junction capacitance:

- Select **Include fixed or zero junction capacitance** for the **Parameterization** parameter. The capacitance is fixed.
- Select **Use parameters CJO, VJ, M & FC** for the **Parameterization** parameter. The block uses the coefficients  $CJO$ ,  $VJ$ ,  $M$ , and  $FC$  to calculate a junction capacitance that depends on the junction voltage.
- Select **Use C-V curve data points** for the **Parameterization** parameter. The block uses three capacitance values on the C-V capacitance curve to estimate  $CJO$ ,  $VJ$ , and  $M$  and uses these values with the specified value of  $FC$  to calculate a junction capacitance that depends on the junction voltage. The block calculates  $CJO$ ,  $VJ$ , and  $M$  as follows:

- $$CJO = C_1((V_{R2} - V_{R1})/(V_{R2} - V_{R1}(C_2/C_1)^{-1/M}))^M$$
- $$VJ = -(-V_{R2}(C_1/C_2)^{-1/M} + V_{R1})/(1 - (C_1/C_2)^{-1/M})$$
- $$M = \log(C_3/C_2)/\log(V_{R2}/V_{R3})$$

where:

- $V_{R1}$ ,  $V_{R2}$ , and  $V_{R3}$  are the values in the **Reverse bias voltages [VR1 VR2 VR3]** vector.
- $C_1$ ,  $C_2$ , and  $C_3$  are the values in the **Corresponding capacitances [C1 C2 C3]** vector.

The reverse bias voltages (defined as positive values) should satisfy  $V_{R3} > V_{R2} > V_{R1}$ . This means that the capacitances should satisfy  $C_1 > C_2 > C_3$  as reverse bias widens the depletion region and hence reduces capacitance. Violating these inequalities results in an error. Voltages  $V_{R2}$  and  $V_{R3}$  should be well away from the Junction potential  $VJ$ . Voltage  $V_{R1}$  should be less than the Junction potential  $VJ$ , with a typical value for  $V_{R1}$  being 0.1 V.

The voltage-dependent junction capacitance is defined in terms of the capacitor charge storage  $Q_j$  as:

- For  $V < FC \cdot VJ$ :

$$Q_j = CJO \cdot (VJ/(M - 1)) \cdot ((1 - V/VJ)^{1-M} - 1)$$

- For  $V \geq FC \cdot VJ$ :

$$Q_j = CJO \cdot F_1 + (CJO/F_2) \cdot (F_3 \cdot (V - FC \cdot VJ) + 0.5(M/VJ) \cdot (V^2 - (FC \cdot VJ)^2))$$

where:



- $F_1 = (VJ/(1 - M)) \cdot (1 - (1 - FC)^{1 - M})$
- $F_2 = (1 - FC)^{1 + M}$
- $F_3 = 1 - FC \cdot (1 + M)$

These equations are the same as used in [2 on page 1-480], except that the temperature dependence of  $VJ$  and  $FC$  is not modeled.

### Charge Dynamics

The Diode block models the charge dynamics only if you set the **Fidelity level** parameter to **Include capacitance and charge dynamics**. For applications such as commutation diodes it can be important to model diode charge dynamics. When a forward-biased diode has a reverse voltage applied across it, it takes time for the charge to dissipate and hence for the diode to turn off. The time taken for the diode to turn off is captured primarily by the transit time parameter. Once the diode is off, any remaining charge then dissipates, the rate at which this happens being determined by the carrier lifetime.

The Diode block uses the model of Lauritzen and Ma [3 on page 1-480] to capture these effects. These are the defining equations.

$$i = \frac{q_E - q_M}{T_M} \quad (1-2)$$

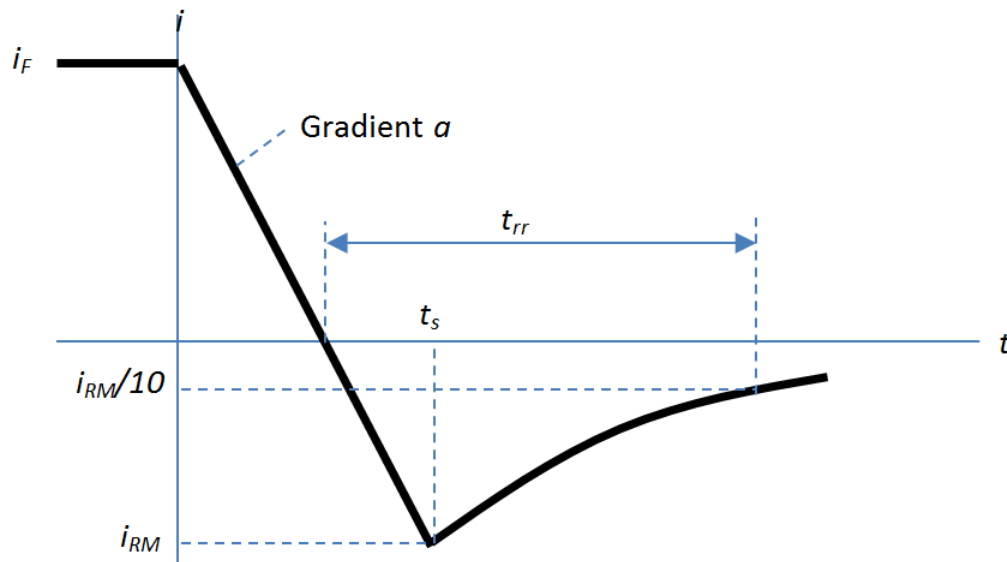
$$\frac{dq_M}{dt} + \frac{q_M}{\tau} - \frac{q_E - q_M}{T_M} = 0 \quad (1-3)$$

$$q_E = (\tau + T_M)i \quad (1-4)$$

where:

- $i$  is the diode current.
- $q_E$  is the junction charge.
- $q_M$  is the total stored charge.
- $T_M$  is the transit time.
- $\tau$  is the carrier lifetime.
- $v_D$  is the voltage across the diode.
- $v_F$  is the diode forward voltage.
- $R$  is the diode on resistance.
- $G$  is the diode off conductance.

This graphic shows a typical reverse-mode current characteristic for a diode device.



where:

- $i_{RM}$  is the peak reverse current.
- $i_F$  is the starting forward current when measuring  $i_{RM}$ .
- $a$  is the rate of change of current when measuring  $i_{RM}$ .
- $t_{rr}$  is the reverse recovery time.

Data sheets for diodes quote values for peak reverse current for an initial forward current and a steady rate of change of current. The data sheet might also provide values for reverse recovery time and total recovery charge.

#### How the Block Calculates $T_M$ and Tau

The block calculates transit time  $T_M$  and carrier lifetime  $\tau$  based on the values you enter for the **Charge Dynamics** parameters. The block uses  $T_M$  and  $\tau$  to solve the charge dynamics equations 1, 2, and 3.

During initial current drop in reverse mode, the diode is still on, and the rate of change of current is determined by an external test circuit.

First, the block uses equation 1 to perform this calculation.

$$i_F + at = \frac{qE - q_M}{T_M} \quad (1-5)$$

Then, it substitutes equation 4 into equation 2.

$$\frac{dq_M}{dt} + \frac{q_M}{\tau} = i_F + at \quad (1-6)$$

Then, it solves equation 5 for  $q_M$ ,

$$q_M = i_F \tau - a \tau^2 + \frac{k}{\exp\left(\frac{t}{\tau}\right)} + a \tau t, \quad (1-7)$$

where  $k$  is a constant.

When  $t$  is zero,  $i = i_F$  and  $q_M = \tau i_F$  because the system is in steady state.

Substituting these relationships into equation 6 and solving the equation gives  $k = a\tau^2$ .

Therefore,

$$q_M = i_F \tau + a\tau^2 \left( \frac{1}{\exp\left(\frac{t}{\tau}\right)} - 1 \right) + a\tau t. \quad (1-8)$$

At time  $t = t_s$ , the current is  $i_{RM}$  and the junction charge  $q_E$  is zero.

The block substitutes these values into equation 1.

$$i_{RM} = \frac{-q_M}{T_M} \quad (1-9)$$

The block rearranges equation 8 to solve for  $q_M$  and substitutes the result into equation 7.

$$-T_M i_{RM} = i_F \tau + a\tau^2 \left( \frac{1}{\exp\left(\frac{t_s}{\tau}\right)} - 1 \right) + a\tau t_s \quad (1-10)$$

Then, the block expresses time  $t_s$  in terms of  $i_{RM}$ ,  $i_F$ , and  $a$ .

$$t_s = \frac{i_{RM} - i_F}{a} \quad (1-11)$$

Consider the diode recovery, that is, when  $t > t_s$ . The diode is reverse biased, and current and junction charge are effectively zero.

The current is defined by this equation.

$$i = i_{RM} \exp\left[-\frac{(t - t_s)}{\tau_{rr}}\right], \quad (1-12)$$

where:

$$\frac{1}{\tau_{rr}} = \frac{1}{\tau} + \frac{1}{T_M}. \quad (1-13)$$

The block now relates the expression in equation 12 to the reverse recovery time  $t_{rr}$ .

When  $t = \frac{i_{RM}}{a} + t_{rr}$ , the current is  $\frac{i_{RM}}{10}$ .

Therefore,

$$\exp\left(-\frac{t - t_s}{\tau_{rr}}\right) = 0.1 \quad (1-14)$$

and

$$t_{rr} = \tau_{rr} \log(10) + \frac{i_{RM}}{a}. \quad (1-15)$$

The block uses equations 9 and 14 to calculate values for  $T_M$  and  $\tau$ . The calculation uses an iterative scheme because of the exponential term in Equation 9.

#### Alternatives to Specifying $t_{rr}$ Directly

In addition to allowing you to specify reverse recovery time  $t_{rr}$  directly, the block supports three alternative parameterizations. The block can derive  $t_{rr}$  from either of these parameters:

- Reverse recovery time stretch factor  $\lambda$
- Reverse recovery charge  $Q_{rr}$ , when the data sheet specifies this value instead of the reverse recovery time.
- Reverse recovery energy  $E_{rec}$ , when the data sheet specifies this value instead of the reverse recovery time.

The relationship between reverse recovery time stretch factor  $\lambda$  and  $t_{rr}$  is expressed by the equation

$$\lambda = \frac{t_{rr} a}{i_{RM}}.$$

Reverse recovery time must be greater than  $\frac{i_{RM}}{a}$  and a typical value is  $3(\frac{i_{RM}}{a})$ .

Therefore, a typical value for  $\lambda$  is 3.  $\lambda$  must be greater than 1.

Reverse recovery charge  $Q_{rr}$  is the integral over time of the reverse current from the point where the current goes negative until it decays back to zero.

The initial charge, to time  $t_s$  (as shown in the figure), is expressed by this equation:

$$Q_s = \frac{1}{2}(-i_{RM})\frac{i_{RM}}{a}. \quad (1-16)$$

Integrating equation 11 gives the charge between times  $t_s$  and inf. This charge is equal to

$$\tau_{rr} i_{RM}.$$

Therefore, total reverse recovery charge is given by this equation:

$$Q_{rr} = -\frac{i_{RM}^2}{2a} + \tau_{rr} i_{RM}. \quad (1-17)$$

Rearranging equation 16 to solve for  $\tau_{rr}$  and substituting the result into equation 14 gives an equation that expresses  $t_{rr}$  in terms of  $Q_{rr}$ :

$$t_{rr} = \left( \frac{Q_{rr}}{i_{RM}} + \frac{i_{RM}}{2a} \right) \log(10) + \frac{i_{RM}}{a}.$$

Alternatively, the block calculates  $\tau_{rr}$  by using the reverse recovery energy,  $E_{rec}$ . This equation defines the diode voltage curve:

$$v_d = v_R - (v_R - v_{RM}) \exp(-(t - t_1)/\tau_{rr}), \quad (1-18)$$

where  $v_{RM} = v_R - i_{RM}\left(R - \frac{L}{\tau_{rr}}\right)$  is the maximum diode reverse voltage.

If  $\frac{L}{\tau_{rr}} \gg R$ , which is a common condition for the reverse recovery test circuit, the block calculates the maximum diode reverse voltage as:

$$v_{RM} = v_R + i_{RM}\frac{L}{\tau_{rr}}.$$

Since the drop time value is small, the block assumes that the diode current drop is linear:

$$i_F + didt \cdot t = \frac{q_E - q_M}{T_M}. \quad (1-19)$$

Then, it substitutes equation 18 into equation 5:

$$\frac{dq_M}{dt} + \frac{q_M}{\tau} = i_F + didt \cdot t. \quad (1-20)$$

To obtain the total stored charge, it solves equation 19 :

$$q_M = i_F\tau + didt \cdot \tau^2 \left( \frac{1}{\exp\left(\frac{t}{\tau}\right)} - 1 \right) + didt \cdot \tau t, \quad (1-21)$$

where  $didt$  is the gradient of the current.

When  $t = t_1 = \frac{i_{RM} - i_F}{didt}$ , the peak reverse current is:

$$i_{RM} = \frac{-q_M}{T_M}. \quad (1-22)$$

The block now substitutes equation 21 into equation 20:

$$\begin{aligned} -i_{RM}T_M &= i_F\tau + didt \cdot \tau^2 \left( \exp\left(-\frac{i_{RM} - i_F}{didt \cdot \tau}\right) - 1 \right) + \tau(i_{RM} - i_F) \\ -(T_M + \tau)i_{RM} &= didt \cdot \tau^2 \left( \exp\left(-\frac{i_{RM} - i_F}{didt \cdot \tau}\right) - 1 \right) \end{aligned} \quad (1-23)$$

$$i_{RM} = F(didt, i_F, T_M, \tau).$$

Finally, the block solves equation 22 to obtain the reverse recovery energy:

$$\begin{aligned} E_{rec} &= \int_1^{t_2} i_d v_d dt \\ E_{rec} &= \frac{\tau_{rr} i_{RM} (v_R + v_{RM})}{2} \\ E_{rec} &= \frac{i_{RM} (2v_R \tau_{rr} + i_{RM} L)}{2}. \end{aligned} \quad (1-24)$$

### Charge dynamics with faults

When the device is faulted, the Diode block calculates the junction charge as

$$q_E = (\tau + T_M)i_{diode},$$

where:

- $i_{diode}$  is the diode current without the charge model.
- $q_E$  is the junction charge.
- $T_M$  is the transit time.
- $\tau$  is the carrier lifetime.

This equation then defines the value of the diode current:

$$\frac{dq_M}{dt} + \frac{q_M}{\tau} - \frac{q_E - q_M}{T_M} = 0$$

$$i = \frac{q_E - q_M}{T_M} Q_{scale} - (Q_{scale} - 1)i_{diode}$$

where:

- $i$  is the diode current.
- $q_M$  is the total stored charge.
- $Q_{scale}$  is the current value of the charge ratio.

### Temperature Dependence

The default behavior for the Diode block is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. The exponential diode model contains several options for modeling the dependence of the diode current-voltage relationship on temperature during simulation. Temperature dependence of the junction capacitance is not modeled because it has a much smaller effect.

When including temperature dependence, the diode defining equation remains the same. The measurement temperature value,  $T_{m1}$ , is replaced with the simulation temperature,  $T_s$ . The saturation current,  $IS$ , becomes a function of temperature according to the following equation:

$$IS_{T_s} = IS_{T_{m1}} \cdot (T_s/T_{m1})^{XTI/N} \cdot \exp\left(-\frac{EG}{NkT_s}(1 - T_s/T_{m1})\right)$$

where:

- $T_{m1}$  is the temperature at which the diode parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $IS_{T_{m1}}$  is the saturation current at measurement temperature.
- $IS_{T_s}$  is the saturation current at simulation temperature. This is the saturation current value used in the standard diode equation when temperature dependence is modeled.
- $EG$  is the energy gap for the semiconductor type measured in joules(J). The value for silicon is usually taken to be 1.11 eV, where 1 eV is 1.602e-19.
- $XTI$  is the saturation current temperature exponent. This is usually set to 3.0 for pn-junction diodes, and 2.0 for Schottky barrier diodes.
- $N$  is the emission coefficient.

- $k$  is the Boltzmann constant ( $1.3806503e-23$  J/K).

Appropriate values for  $XTI$  and  $EG$  depend on the type of diode and the semiconductor material used. Default values for particular material types and diode types capture approximate behavior with temperature. The block provides default values for common types of diode.

In practice, the values of  $XTI$  and  $EG$  need tuning to model the exact behavior of a particular diode. Some manufacturers quote these tuned values in a SPICE Netlist, and you can read off the appropriate values. Otherwise, you can determine improved estimates for  $EG$  by using a datasheet-defined current-voltage data point at a higher temperature. The block provides a parameterization option for this. It also gives the option of specifying the saturation current at a higher temperature  $IS_{Tm2}$  directly.

You can also tune the values of  $XTI$  and  $EG$  yourself, to match lab data for your particular device. You can use Simulink Design Optimization™ software to help tune the values for  $XTI$  and  $EG$ .

---

**Caution** Device temperature behavior is also dependent on the emission coefficient. An inappropriate value for the emission coefficient can give incorrect temperature dependence, because saturation current is a function of the ratio of  $EG$  to  $N$ .

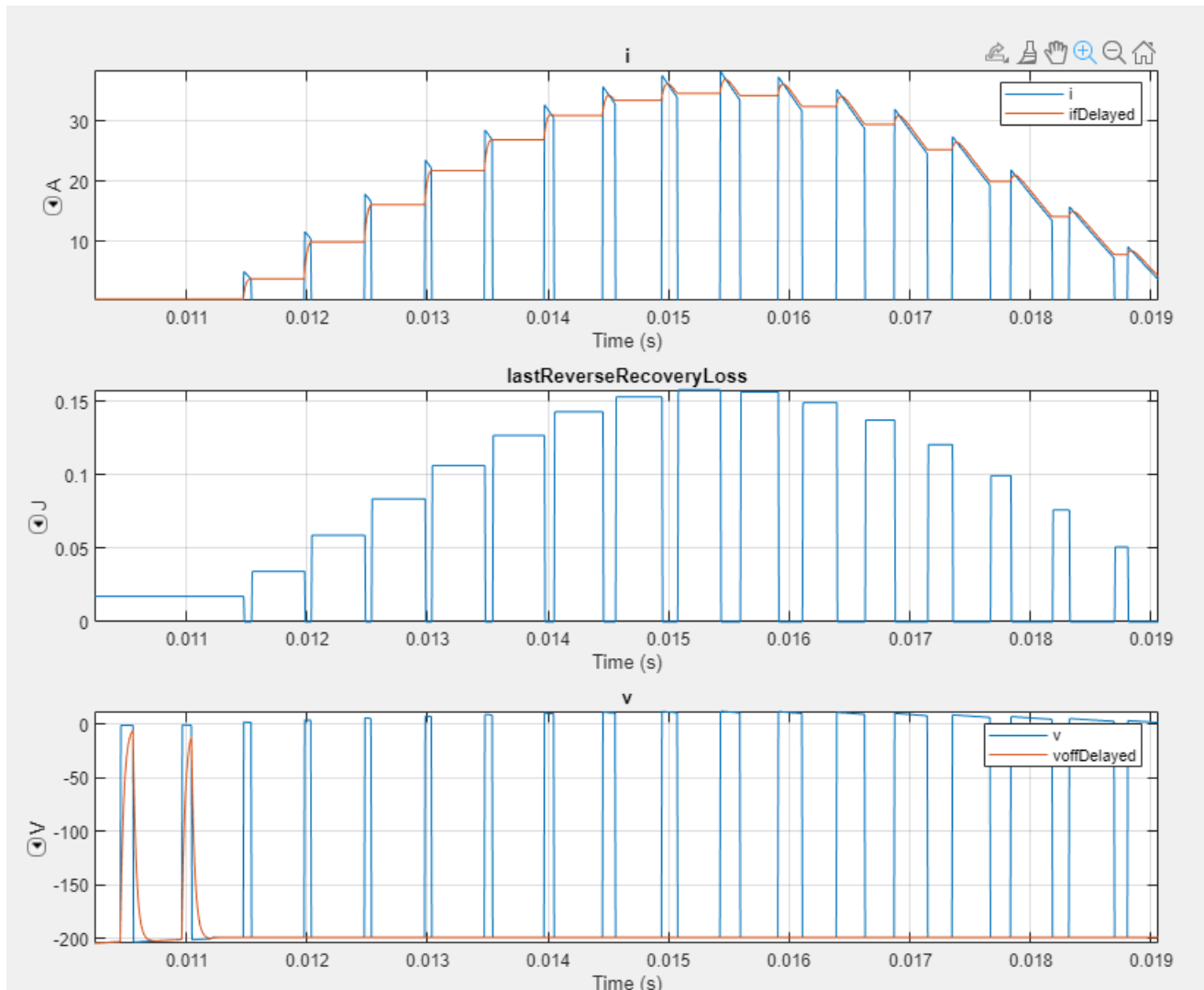
---

If defining a finite reverse breakdown voltage ( $BV$ ), then the value of the reverse  $BV$  is modulated by the reverse breakdown temperature coefficient  $TCV$  (specified using the **Reverse breakdown voltage temperature coefficient, dBV/dT** parameter):

$$BV_{T_s} = BV_{T_{m1}} - TCV \cdot (T_s - T_{m1}) \quad (1-25)$$

### Thermal Losses

This figure shows the voltage and current of the diode in a switching regime inside a half-bridge.



Reverse recovery loss is one of the main sources of thermal loss in diodes. The diode dissipates energy every time it turns off, from its conducting state to the open-circuit state.

In this block, reverse recovery losses are applied by stepping up the junction temperature with a value equal to the reverse recovery loss divided by the total thermal mass at the junction.

If you set the **Reverse recovery loss model** parameter to **Tabulated loss**, the value of the **Reverse recovery loss table, Erec(T<sub>j</sub>, I<sub>f</sub>)** parameter specifies the dissipated energy in function of the junction temperature and the forward current just before the switching event. The off-state voltage linearly scales the losses prior to the latest device turn-off event. The table uses delayed values for current and voltage. To ensure the value used in the lookup table is close enough to the instantaneous value, set the **Delay for voltage and current tabulation** parameter to a value that is lower than the fastest switching period.

If you set the **Reverse recovery loss model** parameter to **Fixed loss**, the value of the **Reverse recovery loss** parameter specifies the dissipated energy in each turn-off event, regardless of the state of the diode before or after switching event.



---

**Note** The `lastReverseRecoveryLoss` variable in the simlog includes the reverse recovery losses as a pulse with amplitude equal to the energy loss. If you use a script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

---

## Faults

The Diode block allows you to model three types of faults:

- `Open` — Failure due to metallization burnout.
- `Short` — Failure due to punch-through.
- `Parameter shift` — Failure due to aging.

The block can trigger fault events:

- At a specific time
- When a current limit, a voltage limit, or a temperature limit are exceeded for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault takes precedence. In other words, the component fails no more than once per simulation.

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Modeling Thermal Effects

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain a thermal port and does not simulate heat generation in the device.
- `Show thermal port` — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

- When you select Use two I-V curve data points for the **Parameterization** parameter, choose a pair of voltages near the diode turn-on voltage. Typically, this is in the range from 0.05 to 1 V. Using values outside of this region may lead to numerical issues and poor estimates for  $I_S$  and  $N$ .
- The block does not account for temperature-dependent effects on the junction capacitance.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.
- You cannot use the Tabulated I-V curve parameterization to model the reverse breakdown.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the anode.

#### - – Negative terminal

electrical

Electrical conserving port associated with the cathode.

#### H – Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

#### Fidelity level – Level of fidelity

Include capacitance and charge dynamics (default) | Ideal switching

Level of detail of the dynamics model of the diode. If you select **Ideal** switching, the block only models the reverse recovery losses. If you select **Include capacitance and charge dynamics**, the block models both junction capacitance and charge dynamics.

### Diode model — Diode model

Piecewise Linear (default) | Exponential | Tabulated I-V curve

Select one of these diode models:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448. This is the default method.
- **Exponential** — Use a standard exponential model for the diode, as described in “Exponential Diode” on page 1-448.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data plus fixed reverse bias off conductance, as described in “Tabulated Diode” on page 1-449.

### Table type — Tabulated function

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

#### Dependencies

To enable this parameter, set **Diode model** to Tabulated I-V curve.

### Forward voltage — Forward voltage

0.6 V (default)

Minimum voltage that needs to be applied for the diode to become forward-biased.

#### Dependencies

To enable this parameter, set **Diode model** to Piecewise linear.

### On resistance — On resistance

0.3 Ohm (default)

Resistance of the diode when it is forward biased.

#### Dependencies

To enable this parameter, set **Diode model** to Piecewise linear.

### Off conductance — Off conductance

$1e-8$  1/Ohm (default)

Conductance of the diode when it is reverse biased.

#### Dependencies

To enable this parameter, set **Diode model** to Piecewise linear or Tabulated I-V curve.

### Parameterization — Model parameterization

Use two I-V curve data points (default) | Use parameters IS and N | Use an I-V data point and IS | Use an I-V data point and N

Select one of the following methods for model parameterization:

- Use two I-V curve data points — Specify measured data at two points on the diode I-V curve. This is the default method.
- Use parameters *IS* and *N* — Specify saturation current and emission coefficient.
- 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.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential.

**Currents [I1 I2] — Vector of current values at two points**

[.0137, .545] A (default)

Vector of the current values at the two points on the diode I-V curve that the block uses to calculate *IS* and *N*.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use two I-V curve data points.

**Voltages [V1 V2] — Vector of voltage values at two points**

[.6, .7] V (default)

Vector of the voltage values at the two points on the diode I-V curve that the block uses to calculate *IS* and *N*.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use two I-V curve data points.

**Saturation current, IS — Saturation current**

1e-12 A (default)

Magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use parameters *IS* and *N* or Use an I-V data point and *IS*.

**Emission coefficient, N — Diode emission coefficient**

1 (default)

Diode emission coefficient or ideality factor.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use parameters *IS* and *N* or Use an I-V data point and *IS*.

**Current I1 — Current value**

0.0137 A (default)

Current value at the point on the diode I-V curve that the block uses for calculations. Depending on the **Parameterization** value, the block uses this parameter to calculate either  $N$  or  $IS$ .

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use an I-V data point and  $IS$  or Use an I-V data point and  $N$ .

**Voltage V1 — Voltage value**

0.6 V (default)

Voltage value at the point on the diode I-V curve that the block uses for calculations.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential and **Parameterization** to Use an I-V data point and  $IS$  or Use an I-V data point and  $N$ .

**Ohmic resistance, RS — Ohmic resistance**

0 Ohm (default)

Series diode connection resistance.

**Dependencies**

To enable this parameter, set **Diode model** to Exponential.

**Measurement temperature — Measurement temperature**

25 degC (default)

Temperature  $T_{m1}$  at which  $IS$  or the I-V curve was measured.

**Forward currents, If(Tj,Vf) — Vector of forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, set **Table type** to Table in  $If(Tj,Vf)$ .

**Forward voltages, Vf(Tj,If) — Vector of forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, set **Table type** to Table in  $Vf(Tj,If)$ .

**Junction temperatures, Tj — Vector of junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures.

If the size of the vector is equal to 1, the diode characteristics does not depend on the temperature.

**Dependencies**

To enable this parameter, set **Diode model** to Tabulated I-V curve.

**Forward voltages, Vf — Vector of forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, set **Table type** to Table in If(Tj,Vf).

**Forward currents, If — Vector of forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, set **Table type** to Table in Vf(Tj,If).

**Number of series diodes — Number of diodes connected in series**

1 (default)

Number of diodes connected in series between the + and - block ports. Multiple diodes are not modeled. Rather, each diode has all voltage-related quantities scaled by the factor that you specify.

**Number of parallel diodes — Number of parallel diodes**

1 (default)

Number of parallel diodes, or number of parallel paths formed by series-connected diodes, between the + and - block ports. Multiple diodes are not modeled. Rather, each diode has all current-related quantities scaled by the factor that you specify.

**Breakdown****Zener resistance — Zener resistance**

0.3 Ohm (default)

Resistance of the diode when the voltage is less than the **Reverse breakdown voltage** value.

**Dependencies**

To enable this parameter, set **Diode model** to Piecewise linear.

**Reverse breakdown voltage — Reverse breakdown voltage**

inf V (default)

Reverse voltage below which to model the rapid increase in conductance that occurs at diode breakdown. The default value is Inf V, which effectively omits reverse breakdown from the model.

**Dependencies**

To enable this parameter, set **Diode model** to Piecewise linear or Exponential.

**Capacitance**

To enable these parameters, either set **Modeling option** to No thermal port or **Fidelity level** to Include capacitance and charge dynamics.

**Parameterization — Modeling junction capacitance**

Fixed or zero junction capacitance (default) | Use C-V curve data points | Use parameters CJ0, VJ, M & FC

Method for modeling the junction capacitance:

- Fixed or zero junction capacitance — Model the junction capacitance as a fixed value.
- Use C-V curve data points — Specify measured data at three points on the diode C-V curve.
- Use parameters CJ0, VJ, M & FC — Specify zero-bias junction capacitance, junction potential, grading coefficient, and forward-bias depletion capacitance coefficient.

**Junction capacitance — Junction capacitance**

0 pF (default)

Fixed junction capacitance value.

**Dependencies**

To enable this parameter, set **Parameterization** to Fixed or zero junction capacitance.

**Reverse bias voltages [VR1 VR2 VR3] — Vector of reverse bias voltage values**

[.1, 10, 100] V (default)

Vector of the reverse bias voltage values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

To enable this parameter, set **Parameterization** to Use C-V curve data points.

**Corresponding capacitances [C1 C2 C3] — Vector of capacitance values**

[3.5, 1, .4] pF (default)

Vector of the capacitance values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

To enable this parameter, set **Parameterization** to Use C-V curve data points.

**Zero-bias junction capacitance, CJ0 — Zero-bias junction capacitance**

5 pF (default)

Value of the capacitance placed in parallel with the conduction current term.

**Dependencies**

To enable this parameter, set **Parameterization** to Use parameters CJ0, VJ, M & FC.

**Junction potential, VJ — Junction potential**

1 V (default)

The junction potential.

**Dependencies**

To enable this parameter, set **Parameterization** to Use parameters CJ0, VJ, M & FC.

**Grading coefficient, M — Grading coefficient**

0.5 (default)

Grading coefficient.

**Dependencies**

To enable this parameter, set **Parameterization** to Use parameters CJ0, VJ, M & FC.

**Capacitance coefficient, FC — Capacitance coefficient**

0.5 (default)

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Use parameters CJ0, VJ, M & FC.

**Charge dynamics — Charge dynamics parameterization**

Do not model charge dynamics (default) | Use peak reverse current and stretch factor | Use peak reverse current and reverse recovery time | Use peak reverse current and reverse recovery charge | Use peak reverse current and reverse recovery energy | Use transit time and carrier lifetime

Select one of the following methods for charge dynamics parameterization:

- Do not model charge dynamics — Do not include charge dynamics modeling. This is the default method.
- Use peak reverse current and stretch factor — Model charge dynamics by providing values for peak reverse current  $i_{RM}$  and stretch factor  $\lambda$  plus information on the initial forward current and rate of change of current used in the test circuit when measuring  $i_{RM}$  and  $t_{rr}$ .
- Use peak reverse current and reverse recovery time — Model charge dynamics by providing values for peak reverse current  $i_{RM}$  and reverse recovery time  $t_{rr}$  plus information on the initial forward current and rate of change of current used in the test circuit when measuring  $i_{RM}$  and  $t_{rr}$ . Use this option if the manufacturer datasheet does not provide values for transit time  $TT$  and carrier lifetime  $\tau$ .
- Use peak reverse current and reverse recovery charge — Model charge dynamics by providing values for peak reverse current  $i_{RM}$  and reverse recovery charge  $Q_{rr}$  plus information on the initial forward current and rate of change of current used in the test circuit when measuring  $i_{RM}$  and  $t_{rr}$ .
- Use peak reverse current and reverse recovery energy — Model charge dynamics by providing values for peak reverse current  $i_{RM}$  and reverse recovery energy  $E_{rec}$  plus information on the initial forward current and rate of change of current used in the test circuit when measuring  $i_{RM}$ .



- Use transit time and carrier lifetime — Model charge dynamics by providing values for transit time  $TT$  and carrier lifetime  $\tau$ .

### Peak reverse current, $i_{RM}$ — Peak reverse current

-7.15 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Use peak reverse current and stretch factor, Use peak reverse current and reverse recovery time, Use peak reverse current and reverse recovery charge, or Use peak reverse current and reverse recovery energy.

### Initial forward current when measuring $i_{RM}$ — Initial forward current when measuring $i_{RM}$

4 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Use peak reverse current and stretch factor, Use peak reverse current and reverse recovery time, Use peak reverse current and reverse recovery charge, or Use peak reverse current and reverse recovery energy.

### Rate of change of current when measuring $i_{RM}$ — Rate of change of current when measuring $i_{RM}$

-750 A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Use peak reverse current and stretch factor, Use peak reverse current and reverse recovery time, Use peak reverse current and reverse recovery charge, or Use peak reverse current and reverse recovery energy.

### Reverse recovery time stretch factor — Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. The default value is 3.

Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

To enable this parameter, set **Charge dynamics** to Use peak reverse current and stretch factor.

**Reverse recovery time, trr – Reverse recovery time**

115 (default)

Time between the point where the current initially goes to zero when the diode turns off, and the point where the current falls to less than ten percent of the peak reverse current. The default value is 115 ns.

The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, iRM** parameter divided by the value of the **Rate of change of current when measuring iRM** parameter.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use peak reverse current and reverse recovery time.

**Reverse recovery charge, Qrr – Reverse recovery charge**

1500 s\*μA (default)

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use peak reverse current and reverse recovery charge.

**Diode turn-off voltage when measuring Erec – Diode turn-off voltage when measuring Erec**

-600 V (default)

Voltage between the diode in steady-state.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use peak reverse current and reverse recovery energy.

**Stray inductance when measuring Erec – Stray inductance when measuring Erec**

150 nH (default)

Total unintended inductance in the measurement circuit. The block uses this value to calculate **Reverse recovery energy, Erec**.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use peak reverse current and reverse recovery energy.

**Reverse recovery energy, Erec — Reverse recovery energy**

0.03 J (default)

Total switching losses due to the diode reverse recovery.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use peak reverse current and reverse recovery energy.

**Transit time, TT — Transit time**

50 ns (default)

Measure of how long it takes carriers to cross the diode junction.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use transit time and carrier lifetime.

**Carrier lifetime, tau — Carrier lifetime**

100 ns (default)

Measure of how long it takes for the carriers to dissipate once the diode is no longer conducting. The default value is 100 ns.

**Dependencies**

To enable this parameter, set **Charge dynamics** to Use transit time and carrier lifetime.

**Reverse Recovery Loss**

To enable these parameters, set **Modeling option** to Show thermal port and **Fidelity level** to Ideal switching.

**Reverse recovery loss model — Model for reverse recovery loss**

Fixed loss (default) | Tabulated loss

Whether to model fixed or tabulated reverse recovery losses.

**Reverse recovery loss — Reverse recovery loss**

0 J (default)

Dissipated energy in each turn-off event, regardless of the state of the diode before or after the switching event.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Fixed loss.

**Reverse recovery loss table, Erec(Tj, If) — Reverse recovery loss**

zeros(2, 3) J (default)

Dissipated energy in function of the forward current,  $I_f$ , just before the switching event, and the final off-state voltage once the diode is in off state.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Tabulated loss.

**Temperature vector for recovery loss table,  $T_j$  – Temperature vector for recovery loss table**

[298.15, 398.15] K (default)

Temperature vector used in the tabulation of the reverse recovery loss.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Tabulated loss.

**Forward current vector for recovery loss table,  $I_f$  – Forward current vector for recovery loss table**

[.1, 1, 10] A (default)

Forward current vector used in the tabulation of the reverse recovery loss.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Tabulated loss.

**Turn-off voltage when measuring recovery loss,  $V_{rec}$  – Turn-off voltage when measuring recovery loss**

10 V (default)

Voltage between the diode in steady-state when measuring recovery loss.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Tabulated loss.

**Filter time constant for voltage and current values in recovery loss table – Filter time constant for voltage and current values in recovery loss table**

1 / 1000 / 100 s (default)

Filter time constant for voltage and current values in the recovery loss table. To ensure the value used in the lookup table is close enough to the instantaneous value, specify a value that is lower than the fastest switching period.

**Dependencies**

To enable this parameter, set **Reverse recovery loss model** to Tabulated loss.

**Temperature Dependence**

This section is applicable to Exponential and Tabulated I-V curve diode models only.

This table shows how the visibility of **Temperature Dependence** setting depends on how you configure the **Diode model** parameter in the **Main** setting and whether you expose the thermal port. To learn how to read this table, see “Parameter Dependencies” on page A-2

## Visibility of the Temperature Dependence Setting

Parameters, options, and visibility of the setting					
Diode model					
<i>Piecewise linear</i>		<i>Exponential</i>		<i>Tabulated I-V curve</i>	
Thermal Port		Thermal Port		Thermal Port	
<i>Not exposed</i>	<i>Exposed</i>	<i>Not exposed</i>	<i>Exposed</i>	<i>Not exposed</i>	<i>Exposed</i>
Hidden	Hidden	Visible	Visible	Visible	Hidden

### Parameterization — Temperature dependence parameterization

None - Use characteristics at parameter measurement temperature (default) | Use an I-V data point at second measurement temperature | Specify saturation current at second measurement temperature | Specify the energy gap EG

Select one of the following methods for temperature dependence parameterization:

- None - Use characteristics at parameter measurement temperature — Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Use an I-V data point at second measurement temperature T2 — If you select this option, you specify a second measurement temperature  $T_{m2}$ , and the current and voltage values at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify saturation current at second measurement temperature T2 — If you select this option, you specify a second measurement temperature  $T_{m2}$ , and saturation current value at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify the energy gap EG — Specify the energy gap value directly.

### Current I1 at second measurement temperature — Current I1 at second measurement temperature

0.0245 A (default)

Specify the diode current  $I1$  value when the voltage is  $V1$  at the second measurement temperature.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature.

### Voltage V1 at second measurement temperature — Voltage V1 at second measurement temperature

0.5 V (default)

Specify the diode voltage  $V1$  value when the current is  $I1$  at the second measurement temperature.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature.

**Saturation current,  $I_S$ , at second measurement temperature — Saturation current,  $I_S$ , at second measurement temperature**

1.25e-7 A (default)

Specify the saturation current  $I_S$  value at the second measurement temperature.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify saturation current at second measurement temperature.

**Second measurement temperature — Second measurement temperature**

125 °C (default)

Specify the value for the second measurement temperature.

**Dependencies**

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature or Specify saturation current at second measurement temperature.

**Energy gap parameterization — Energy gap parameterization**

Use nominal value for silicon (EG=1.11eV) (default) | Use nominal value for 4H-SiC silicon carbide (EG=3.23eV) | Use nominal value for 6H-SiC silicon carbide (EG=3.00eV) | Use nominal value for germanium (EG=0.67eV) | Use nominal value for gallium arsenide (EG=1.43eV) | Use nominal value for selenium (EG=1.74eV) | Use nominal value for Schottky barrier diodes (EG=0.69eV) | Specify a custom value

Select a value for the energy gap from a list of predetermined options, or specify a custom value:

- Use nominal value for silicon (EG=1.11eV) — This is the default.
- Use nominal value for 4H-SiC silicon carbide (EG=3.23eV)
- Use nominal value for 6H-SiC silicon carbide (EG=3.00eV)
- Use nominal value for germanium (EG=0.67eV)
- Use nominal value for gallium arsenide (EG=1.43eV)
- Use nominal value for selenium (EG=1.74eV)
- Use nominal value for Schottky barrier diodes (EG=0.69eV)
- Specify a custom value — If you select this option, the **Energy gap, EG** parameter appears in the dialog box, to let you specify a custom value for  $EG$ .

**Dependencies**

To enable this parameter, set **Parameterization** to Specify the energy gap  $EG$ .

**Energy gap,  $EG$  — Energy gap**

1.11 eV (default)

Specify a custom value for the energy gap,  $EG$ .

**Dependencies**

To enable this parameter, set **Energy gap parameterization** to Specify a custom value.

### Saturation current temperature exponent parameterization — Saturation current temperature exponent parameterization

Use nominal value for pn-junction diode (XTI=3) (default) | Use nominal value for Schottky barrier diode (XTI=2) | Specify a custom value

Select one of the following options to specify the saturation current temperature exponent value:

- Use nominal value for pn-junction diode (XTI=3) — This is the default.
- Use nominal value for Schottky barrier diode (XTI=2)
- Specify a custom value — If you select this option, the **Saturation current temperature exponent, XTI** parameter appears in the dialog box, to let you specify a custom value for *XTI*.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature, Specify saturation current at second measurement temperature, or Specify the energy gap, EG.

### Saturation current temperature exponent, XTI — Saturation current temperature exponent

3 (default)

Specify a custom value for the saturation current temperature exponent, *XTI*.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature, Specify saturation current at second measurement temperature, or Specify the energy gap, EG and **Saturation current temperature exponent parameterization** to Specify a custom value.

### Reverse breakdown voltage temperature coefficient, dBV/dT — Reverse breakdown voltage temperature coefficient

0 V/K (default)

Modulate the reverse breakdown voltage *BV*. If you define the reverse breakdown voltage *BV* as a positive quantity, a positive value for *TCV* implies that the magnitude of the reverse breakdown voltage decreases with temperature.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature, Specify saturation current at second measurement temperature, or Specify the energy gap, EG.

### Device simulation temperature — Device simulation temperature

25 °C (default)

Specify the value for the temperature  $T_s$ , at which the device is to be simulated.

#### Dependencies

To enable this parameter, set **Parameterization** to Use an I-V data point at second measurement temperature, Specify saturation current at second measurement temperature, or Specify the energy gap, EG.

## Faults

### Enable faults — Whether to enable faults

Off (default) | On

Select On to enable faults modeling and expose parameters that allow you to select the reporting method and specify the trigger mechanism.

### Reporting when a fault occurs — Reporting when a fault occurs

None (default) | Warn | Error

Simulation reporting when a fault occurs:

- None — Does not generate a warning or an error.
- Warn — Generates a warning.
- Error — The simulation stops and generates an error.

### Dependencies

To enable this parameter, set **Enable faults** to On.

### Fault mode — Failure category

Open (default) | Short | Parameter shift

Specify the failure mode of the Diode block:

- Open — Failure due to metallization burnout.
- Short — Failure due to punch-through.
- Parameter shift — Failure due to aging.

### Dependencies

To enable this parameter, set **Enable faults** to On.

### Fault transition time constant — Fault transition time constant

1e-3 s (default)

Time constant for transitioning to faulted state.

### Dependencies

To enable this parameter, set **Enable faults** to On.

### Faulted series resistance — Faulted series resistance

1000 Ohm (default)

Series resistance in faulted state.

### Dependencies

To enable this parameter, set **Enable faults** to On and **Fault mode** to Open.

### Faulted parallel conductance — Faulted series resistance

1000 1/Ohm (default)

Parallel conductance in faulted state.



**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault mode** to Short.

**Faulted forward voltage — Faulted forward voltage**

0.6 V (default)

Forward voltage in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Piecewise linear.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted on resistance — Faulted on resistance**

0.3 Ohm (default)

On resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Piecewise linear.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted off conductance — Faulted off conductance**

1e-8 S (default)

Off conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Piecewise linear or Tabulated I-V curve.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted zener resistance — Faulted zener resistance**

0.3 Ohm (default)

Off conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Piecewise linear.
- **Enable faults** to On.

- **Fault mode** to Parameter shift.

**Faulted reverse breakdown voltage — Faulted reverse breakdown voltage**

$\infty$  V (default)

Reverse breakdown voltage in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Piecewise linear or Exponential.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted currents, [I1 I2] — Faulted currents**

[.0137, .545] A (default)

$I1$  and  $I2$  currents in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Parameterization** to Use two I-V curve data points.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted voltages, [V1 V2] — Faulted voltages**

[.6, .7] V (default)

$V1$  and  $V2$  voltages in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Parameterization** to Use two I-V curve data points.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted ohmic resistance,  $R_S$  — Faulted voltages**

0 Ohm (default)

Ohmic resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Enable faults** to On.

- **Fault mode** to Parameter shift.

#### **Faulted saturation current, IS — Faulted saturation current**

1e-12 A (default)

Saturation current in faulted state.

##### **Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Parameterization** to Use parameters IS and N or Use an I-V data point and IS.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

#### **Faulted emission coefficient, N — Faulted emission coefficient**

1 (default)

Emission coefficient in faulted state.

##### **Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Parameterization** to Use parameters IS and N or Use an I-V data point and N.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

#### **Faulted current, I1 — Faulted current**

0.0137 A (default)

Current in faulted state.

##### **Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.
- **Parameterization** to Use an I-V data point and IS or Use an I-V data point and N.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

#### **Faulted voltage, V1 — Faulted voltage**

0.6 V (default)

Voltage in faulted state.

##### **Dependencies**

To enable this parameter, set:

- **Diode model** to Exponential.

- **Parameterization** to Use an I-V data point and IS or Use an I-V data point and N.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted forward currents,  $I_f(T_j, V_f)$  — Faulted forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12]  
A (default)

Forward currents,  $I_f(T_j, V_f)$ , in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Tabulated I-V curve.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted forward voltages,  $V_f$  — Faulted forward currents**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Forward currents,  $V_f$ , in faulted state.

**Dependencies**

To enable this parameter, set:

- **Diode model** to Tabulated I-V curve.
- **Enable faults** to On.
- **Fault mode** to Parameter shift.

**Faulted capacitance and charge as percentage of unfaulted — Faulted capacitance and charge as percentage of unfaulted**

100 (default)

Capacitance and charge in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Enable temporal fault trigger — Whether to enable a temporal fault trigger**

Off (default) | On

Whether to enable a temporal fault trigger.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Simulation time for fault event — Time before entering faulted state**

inf s (default)

Simulation time at which you want the block to enter the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

**Enable behavioral fault trigger — Whether to enable a behavioral fault trigger**

Off (default) | On

Whether to enable a behavioral fault trigger.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Maximum permissible reverse voltage — Maximum permissible reverse voltage**

inf V (default) | positive scalar

Maximum reverse voltage above which a fault may be triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Enable behavioral fault trigger** to On.

**Number of events to fail when exceeding voltage — Number of events to fail when exceeding voltage**

1 (default) | positive scalar integer

Number of times the diode must exceed the reverse voltage threshold before a fault is triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Enable behavioral fault trigger** to On.

**Maximum permissible current — Current threshold to fault transition**

inf A (default) | positive scalar

Maximum permissible current value. If the current exceeds this value for longer than the value of the **Time to fail when exceeding current** parameter, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Enable behavioral fault trigger** to On.

**Time to fail when exceeding current — Time to fail when exceeding current**

1 s (default) | positive scalar

Amount of time that the current must continuously exceed the maximum permissible current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Enable behavioral fault trigger** to On.

**Maximum diode recovery, dV/dt — Diode recovery threshold**

inf V/s (default) | positive scalar

Maximum diode recovery.

### Dependencies

To enable this parameter, set **Enable faults** to On and **Enable behavioral fault trigger** to On.

### Maximum permissible temperature — Temperature threshold to fault transition

inf K (default) | positive scalar

Maximum permissible temperature value. If the temperature exceeds this value for longer than the value of the **Time to fail when exceeding temperature** parameter, then the block enters the faulted state.

### Dependencies

To enable this parameter, set:

- **Diode model** to Piecewise linear.
- **Enable faults** to On.
- **Enable behavioral fault trigger** to On.

### Time to fail when exceeding temperature — Time to fail when exceeding current

1 s (default) | positive scalar

Amount of time that the temperature must continuously exceed the maximum permissible temperature before a behavioral fault is triggered.

### Dependencies

To enable this parameter, set:

- **Diode model** to Piecewise linear.
- **Enable faults** to On.
- **Enable behavioral fault trigger** to On.

### Thermal Port

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Version History

Introduced in R2008a

### References

- [1] MH. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition. Cambridge, UK: Cambridge University Press, 1984.
- [2] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.
- [3] Lauritzen, P.O. and C.L. Ma. “A Simple Diode Model with Reverse Recovery.” *IEEE® Transactions on Power Electronics*. Vol. 6, No. 2, April 1991, pp. 188-191.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

GTO | Ideal Semiconductor Switch | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | N-Channel MOSFET | P-Channel MOSFET | Thyristor (Piecewise Linear) | Diode

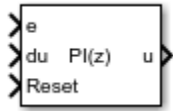
### Topics

“Simulating Thermal Effects in Semiconductors”

## Discrete PI Controller

Discrete-time PI controller with external anti-windup input

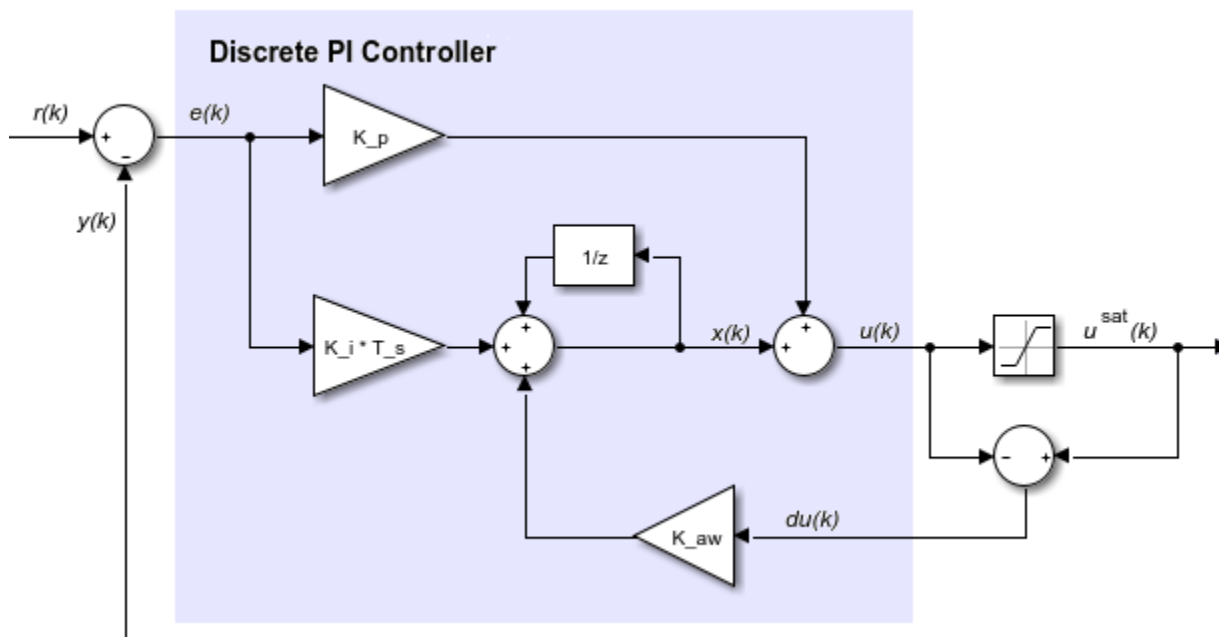
**Library:** Simscape / Electrical / Control / General Control



### Description

The Discrete PI Controller block implements discrete PI control with external anti-windup input.

This diagram is the equivalent circuit for the controller with external anti-windup input.



### Equations

The Discrete PI Controller block calculates the control signal using the backward Euler discretization method:

$$u(k) = \left[ K_p + (K_i + du(k)K_{aw}) \frac{T_s z}{z-1} \right] e(k),$$

where

- $u$  is the control signal.
- $K_p$  is the proportional gain coefficient.
- $K_i$  is the integral gain coefficient.



- $K_{aw}$  is the anti-windup gain coefficient.
- $T_s$  is the sampling period.
- $e$  is the error signal.

To prevent excessive overshoot, the block can use back calculation to implement an external anti-windup mechanism. It inputs  $du(k)$ , the difference between the saturated control signal,  $u^{sat}(k)$ , and the calculated unsaturated control signal,  $u(k)$ . It then multiplies the difference by the anti-windup coefficient and adds the amplified signal from the integral gain.

## Ports

### Input

#### **e — Error signal**

scalar

Error signal,  $e(k)$ , obtained as the difference between the reference,  $r(k)$ , and measurement,  $y(k)$ , signals.

Data Types: single | double

#### **du — Control signal saturation**

scalar

Difference,  $du(k)$ , between the saturated  $u^{sat}(k)$  and the unsaturated control signals,  $u(k)$ . If  $du(k)$  is zero, the anti-windup is disabled.

Description

Data Types: single | double

#### **Reset — Integrator gain reset**

scalar

External reset (rising edge) signal for the integrator.

Data Types: single | double

### Output

#### **u — Control signal**

scalar

Control signal,  $u(k)$ .

Data Types: single | double

## Parameters

#### **Proportional gain — $K_p$**

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the PI controller.

**Integral gain –  $K_i$** 

1 (default) | positive scalar

Integral gain,  $K_i$ , of the PI controller.**Anti-windup gain –  $K_{aw}$** 

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the PI controller.**Integrator initial condition – Initial integrator value**

0 (default) | scalar

Value of the integrator at simulation start time.

**Sample time (-1 for inherited) – Sampling interval**

-1 (default) | default value or a positive number

Time interval between samples. If the block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

## Version History

**Introduced in R2017b**

## References

[1] Åström, K. and T. Hägglund. *Advanced PID Control*. Research Triangle Park, NC: ISA, 2005.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

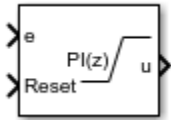
**Blocks**

Discrete PI Controller with Integral Anti-Windup

# Discrete PI Controller with Integral Anti-Windup

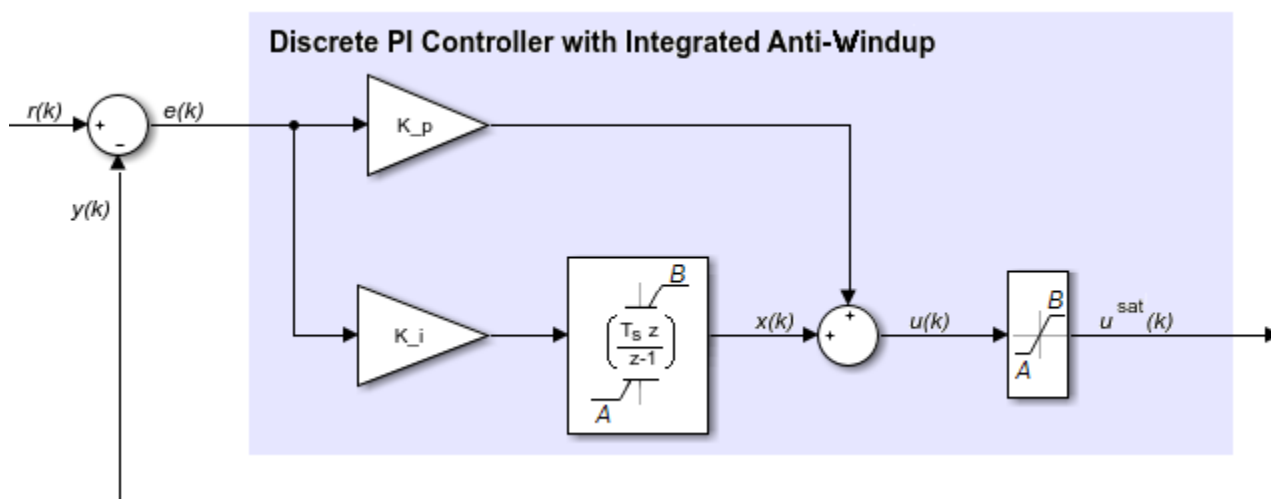
Discrete-time PI control with integral anti-windup

**Library:** Simscape / Electrical / Control / General Control



## Description

The Discrete PI Controller with Integral Anti-Windup block implements discrete PI control with internal anti-windup. The figure shows the equivalent circuit for the controller with internal anti-windup.



## Equations

The block calculates the control signal using the backward Euler discretization method:

$$u(k) = \text{sat}\left(K_p e(k) + \text{sat}\left(K_i \frac{T_s z}{z-1} e(k), A, B\right), A, B\right),$$

$$\text{sat}(x, A, B) = \min(\max(x, A), B),$$

where:

- $u$  is the control signal.
- $K_p$  is the proportional gain coefficient.
- $e$  is the error signal.
- $K_i$  is the integral gain coefficient.
- $T_s$  is the sampling period.

- $A$  is the lower limit for saturation.
- $B$  is the upper limit for saturation.

## Ports

### Input

#### **e – Error signal**

scalar

Error signal,  $e(k)$ , obtained as the difference between the reference,  $r(k)$ , and measurement  $y(k)$  signals.

Data Types: `single` | `double`

#### **Reset – Integrator gain reset**

scalar

External reset (rising edge) signal for the integrator.

Data Types: `single` | `double`

### Output

#### **u – Control signal**

scalar

Control signal,  $u(k)$ .

Data Types: `single` | `double`

## Parameters

#### **Proportional gain – $K_p$**

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the PI controller.

#### **Integral gain – $K_i$**

1 (default) | positive scalar

Integral gain,  $K_i$ , of the PI controller.

#### **Upper saturation limit – $B$**

5 (default) | scalar greater than the value of the **Lower saturation limit** parameter

Upper limit,  $B$ , of the output for the PI controller.

#### **Lower saturation limit – $A$**

-5 (default) | scalar

Lower limit,  $A$ , of the output for the PI controller.

#### **Integrator initial condition – Initial integrator value**

0 (default) | scalar

Value of the integrator at simulation start time.

### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2017b**

## **References**

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5/D39. Piscataway, NJ: IEEE-SA, 2015.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Discrete PI Controller

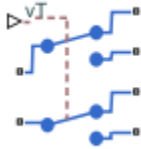
### **Topics**

“Promote Block Parameters on a Mask”

## DPDT Switch

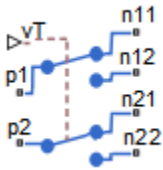
Double-pole double-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



### Description

The DPDT Switch block models a double-pole double-throw switch:



- When the switch is closed, ports **p1** and **p2** are connected to ports **n12** and **n22**, respectively.
- When the switch is open, ports **p1** and **p2** are connected to ports **n11** and **n21**, respectively.

Closed connections are modeled by a resistor with value equal to the **Closed resistance** parameter value. Open connections are modeled by a resistor with value equal to the reciprocal of the **Open conductance** parameter value.

If the **Threshold width** parameter is set to zero, the switch is closed if the voltage presented at the **vT** 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:

$$G = \frac{x}{R_{closed}} + (1 - x)G_{open}$$

$$\lambda = \frac{vT - \text{Threshold}}{\text{Threshold width}}$$

$$x = \begin{cases} 0 & \text{for } \lambda \leq 0 \\ 3\lambda^2 - 2\lambda^3 & \text{for } 0 < \lambda < 1 \\ 1 & \text{for } \lambda \geq 1 \end{cases}$$

The block uses the function  $3\lambda^2 - 2\lambda^3$  because its derivative is zero for  $\lambda = 0$  and  $\lambda = 1$ .

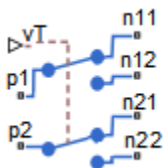
Defining a small positive **Threshold width** can help solver convergence in some models, particularly if the control port signal  $vT$  varies continuously as a function of other network variables. However,

defining a nonzero threshold width precludes the solver making use of switched linear optimizations. Therefore, if the rest of your network is switched linear, set **Threshold width** to zero.

Optionally, you can add a delay between the point at which the voltage at **vT** passes the threshold and the switch opening or closing. To enable the delay, on the **Dynamics** tab, set the **Model dynamics** parameter to Model turn-on and turn-off times.

## Ports

Refer to the figure for port locations.



### Input

#### **vT — Switch control port**

physical signal

Physical port associated with the signal that opens and closes the switch.

### Conserving

#### **p1 — Electrical port**

electrical

Electrical conserving port.

#### **p2 — Electrical port**

electrical

Electrical conserving port.

#### **n11 — Electrical port**

electrical

Electrical conserving port.

#### **n12 — Electrical port**

electrical

Electrical conserving port.

#### **n21 — Electrical port**

electrical

Electrical conserving port.

#### **n22 — Electrical port**

electrical

Electrical conserving port.

## Parameters

### Main

#### **Closed resistance — Closed resistance**

0.01 Ohm (default)

Resistance between the **p** and **n** electrical ports when the switch is closed.

#### **Open conductance — Open conductance**

1e-6 S (default)

Conductance between the **p** and **n** electrical ports when the switch is open. The value must be greater than zero.

#### **Threshold — Threshold voltage**

0.5 V (default)

The threshold voltage for the control physical signal input  $vT$  above which the switch will turn on.

#### **Threshold width — Threshold width**

0 V (default)

The minimum increase in the control signal  $vT$  above the threshold value that will move the switch from fully open to fully closed.

### Dynamics

#### **Model dynamics — Model dynamics**

No dynamics (default) | Model turn-on and turn-off times

Select whether the block models a switching delay:

- **No dynamics** — Do not model the delay. This is the default option.
- **Model turn-on and turn-off times** — Use additional parameters to model a delay between the point at which the voltage at  $vT$  passes the threshold and the switch opening or closing.

#### **Turn-on delay — Turn-on delay**

1e-3 (default) | positive scalar

Time between the input voltage exceeding the threshold voltage and the switch closing. The value must be greater than zero.

### Dependencies

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

#### **Turn-off delay — Turn-off delay**

1e-3 (default) | positive scalar

Time between the input voltage falling below the threshold voltage and the switch opening. The value must be greater than zero.



**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

**Initial input value,  $vT$  — Initial input value,  $vT$** 

0 (default)

The value of the physical signal input  $vT$  at time zero. This value is used to initialize the delayed control voltage parameter internally.

**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

## Version History

**Introduced in R2012b**

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

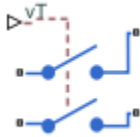
**See Also**

[DPST Switch](#) | [SPDT Switch](#) | [SPDT Switch \(Three-Phase\)](#) | [SPST Switch](#) | [SPST Switch \(Three-Phase\)](#)

## DPST Switch

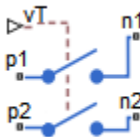
Double-pole single-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



### Description

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



When the switch is closed, ports **p1** and **p2** are connected to ports **n1** and **n2**, respectively.

Closed connections are modeled by a resistor with value equal to the **Closed resistance** parameter value. Open connections are modeled by a resistor with value equal to the reciprocal of the **Open conductance** parameter value.

If the **Threshold width** parameter is set to zero, the switch is closed if the voltage presented at the **vT** 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:

$$G = \frac{x}{R_{closed}} + (1 - x)G_{open}$$

$$\lambda = \frac{vT - \text{Threshold}}{\text{Threshold width}}$$

$$x = \begin{cases} 0 & \text{for } \lambda \leq 0 \\ 3\lambda^2 - 2\lambda^3 & \text{for } 0 < \lambda < 1 \\ 1 & \text{for } \lambda \geq 1 \end{cases}$$

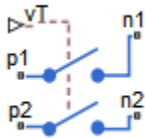
The block uses the function  $3\lambda^2 - 2\lambda^3$  because its derivative is zero for  $\lambda = 0$  and  $\lambda = 1$ .

Defining a small positive **Threshold width** can help solver convergence in some models, particularly if the control port signal  $vT$  varies continuously as a function of other network variables. However, defining a nonzero threshold width precludes the solver making use of switched linear optimizations. Therefore, if the rest of your network is switched linear, set **Threshold width** to zero.

Optionally, you can add a delay between the point at which the voltage at **vT** passes the threshold and the switch opening or closing. To enable the delay, on the **Dynamics** tab, set the **Model dynamics** parameter to **Model turn-on and turn-off times**.

## Ports

Refer to the figure for port locations.



## Input

### **vT — Switch control port**

physical signal

Physical port associated with the signal that opens and closes the switch.

## Conserving

### **p1 — Electrical port**

electrical

Electrical conserving port.

### **p2 — Electrical port**

electrical

Electrical conserving port.

### **n1 — Electrical port**

electrical

Electrical conserving port.

### **n2 — Electrical port**

electrical

Electrical conserving port.

## Parameters

### Main

#### **Closed resistance — Closed resistance**

0.01 Ohm (default)

Resistance between the **p** and **n** electrical ports when the switch is closed.

**Open conductance — Open conductance**

1e-6 S (default)

Conductance between the **p** and **n** electrical ports when the switch is open. The value must be greater than zero.

**Threshold — Threshold voltage**

0.5 V (default)

The threshold voltage for the control physical signal input  $vT$  above which the switch will turn on.

**Threshold width — Threshold width**

0 V (default)

The minimum increase in the control signal  $vT$  above the threshold value that will move the switch from fully open to fully closed.

**Dynamics****Model dynamics — Model dynamics**

No dynamics (default) | Model turn-on and turn-off times

Select whether the block models a switching delay:

- **No dynamics** — Do not model the delay. This is the default option.
- **Model turn-on and turn-off times** — Use additional parameters to model a delay between the point at which the voltage at  $vT$  passes the threshold and the switch opening or closing.

**Turn-on delay — Turn-on delay**

1e-3 (default) | positive scalar

Time between the input voltage exceeding the threshold voltage and the switch closing. The value must be greater than zero.

**Dependencies**

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

**Turn-off delay — Turn-off delay**

1e-3 (default) | positive scalar

Time between the input voltage falling below the threshold voltage and the switch opening. The value must be greater than zero.

**Dependencies**

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

**Initial input value,  $vT$  — Initial input value,  $vT$** 

0 (default)

The value of the physical signal input  $vT$  at time zero. This value is used to initialize the delayed control voltage parameter internally.

**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

**Version History**

**Introduced in R2012b**

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

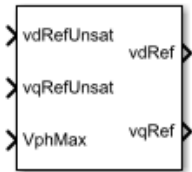
**See Also**

[DPDT Switch](#) | [SPDT Switch](#) | [SPDT Switch \(Three-Phase\)](#) | [SPST Switch](#) | [SPST Switch \(Three-Phase\)](#)

## d-q Voltage Limiter

Limit voltage in the rotor direct-quadrature reference frame

**Library:** Simscape / Electrical / Control / Protection

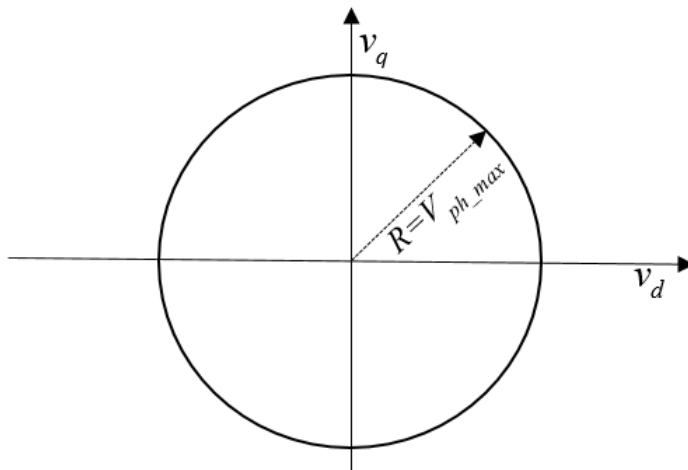


### Description

The d-q Voltage Limiter block implements a voltage limiter in the rotor direct-quadrature ( $d$ - $q$ ) reference frame.

### Equations

The figure shows the circle that limits the  $d$ - $q$  voltage vector.



That is,

$$\sqrt{v_d^2 + v_q^2} \leq V_{ph\_max}$$

where:

- $v_d$  is the  $d$ -axis voltage.
- $v_q$  is the  $q$ -axis voltage.
- $V_{ph\_max}$  is the maximum phase voltage.

Three cases of voltage limiting are possible:

- $d$ -axis prioritization
- $q$ -axis prioritization
- $d$ - $q$  equivalence

If one axis is prioritized over the other axis, the constrained or saturated voltages are defined as

$$v_1^{sat} = \min(\max(v_1^{unsat}, -V_{ph\_max}), V_{ph\_max})$$

and

$$v_2^{sat} = \min(\max(v_2^{unsat}, -V_{2\_max}), V_{2\_max}),$$

where:

- $v_{2\_max} = \sqrt{(V_{ph\_max})^2 - (v_1^{sat})^2}$
- $v_1$  is voltage of the prioritized axis.
- $v_2$  is voltage of the nonprioritized axis.

If neither axis is prioritized, the constrained voltages are defined as

$$v_d^{sat} = \min(\max(v_d^{unsat}, -V_{d\_max}), V_{d\_max})$$

and

$$v_q^{sat} = \min(\max(v_q^{unsat}, -V_{q\_max}), V_{q\_max}),$$

where:

- $V_{d\_max} = \frac{V_{ph\_max}|v_d^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$
- $V_{q\_max} = \frac{V_{ph\_max}|v_q^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$

## Ports

### Input

**vdRefUnsat** —  $v_d^{unsat}$

scalar

Unsaturated direct-axis reference voltage.

Example: Example

Data Types: single | double

**VqRefUnsat** —  $v_q^{unsat}$

scalar

Unsaturated quadrature-axis reference voltage.

Example: Example

Data Types: single | double

**VphMax** —  $V_{ph\_max}$

scalar

Maximum phase voltage.

Data Types: single | double

## Output

**vdRef** —  $v_d^{sat}$

scalar

Saturated direct-axis reference voltage.

Data Types: single | double

**vqRef** —  $v_q^{sat}$

scalar

Saturated quadrature-axis reference voltage.

Example: Example

Data Types: single | double

## Parameters

**Axis prioritization** — Prioritize the *d*- or *q* axis

Q-axis (default) | D-axis | D-Q equivalence

Prioritize the direct-axis, the quadrature-axis, or neither axis.

**Sample time (-1 for inherited)** — Sampling interval

-1 (default) | default value or a positive number

Time interval between samples. If the block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.



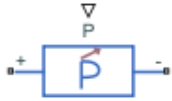
## See Also

**Simscape Blocks**  
Voltage Source

## Dynamic Load

Dynamic load for DC or AC supply

**Library:** Simscape / Electrical / Passive



### Description

The Dynamic Load block implements a dynamic load for a DC or AC supply.

If you set the **Load type** parameter to DC:

- The consumed power of the block is equal to the **P** port as long as it is greater than the value of the **Minimum consumed power** parameter and the voltage from the DC supply is equal to or greater than the value specified for the **Minimum supply voltage** parameter.
- When the voltage from the DC supply drops below the **Minimum supply voltage**, the load behaviour changes and the block acts as a resistive load. If the supply voltage becomes negative, the block acts as an open circuit conductance.

To ensure smooth transitions between these behaviours, the block uses a third-order polynomial spline with continuous derivatives. You can specify the width of this transition using the **Transition voltage width** parameter.

If you set the **Load type** parameter to AC, this table shows the relationship between the voltage from the AC supply,  $V$ , the active power at the **P** port,  $P_{in}$ , and the consumed active power of the block,  $P_{actual}$ :

Applicable Range of $V$ Values	Applicable Range of $P_{in}$ Values	Corresponding Consumed Active Power
$V > V_{min}$	$P_{in} > P_{min}$	$P_{actual} = P_{in}$
	$P_{in} < P_{min}$	$P_{actual} = P_{min}$
$V < V_{min}$	$P_{in} > P_{min}$	The block models a load with an impedance equal to: $R = V_{min}^2 \frac{P_{actual}}{P_{actual}^2 + Q_{actual}^2}$ $X = V_{min}^2 \frac{Q_{actual}}{P_{actual}^2 + Q_{actual}^2}$ $Z = \sqrt{R^2 + X^2}$
	$P_{in} < P_{min}$	$P_{actual} = P_{min}$

Where:

- $V_{min}$  is the value of the “Minimum supply voltage (RMS)” on page 1-0 parameter.
- $P_{min}$  is the value of the “Minimum active power” on page 1-0 parameter.
- $P_{actual}$  is the consumed active power of the Dynamic Load block.
- $Q_{actual}$  is the consumed reactive power of the Dynamic Load block.

The consumed reactive power of the block is equal to the **Q** port as long as the voltage from the AC supply is equal to or greater than the value specified for the **Minimum supply voltage (RMS)** parameter. Otherwise, the block models a load with an impedance equal to:

$$R = V_{min}^2 \frac{P_{actual}}{P_{actual}^2 + Q_{actual}^2}$$

$$X = V_{min}^2 \frac{Q_{actual}}{P_{actual}^2 + Q_{actual}^2}$$

$$Z = \sqrt{R^2 + X^2}$$

.

## Faults

The Dynamic Load block allows you to model an electrical fault as an open circuit. The block can trigger fault events at a specific time.

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Load-Flow Analysis

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Ports

### Input

#### **P — Real power**

physical signal

Physical signal port associated with real power, specified as a scalar.

#### **Q — Reactive power**

physical signal

Physical signal port associated with reactive power, specified as a scalar.

### Conserving

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the dynamic load positive terminal.

**- – Negative terminal**

electrical

Electrical conserving port associated with the dynamic load negative terminal.

## Parameters

### Main

**Load type – Load type**

DC (default) | AC

Type of load. Select DC to use this load with a DC supply or AC to use this load with an AC supply.

**Minimum supply voltage – Minimum DC supply voltage**

20 V (default) | scalar

Minimum voltage the DC supply needs to give the block to act as a constant power load.

**Dependencies**

To enable this parameter, set **Load Type** to DC.

**Minimum consumed power – Minimum consumed power**

1e-3 W (default) | scalar

Minimum amount of power consumed by the block.

**Dependencies**

To enable this parameter, set **Load Type** to DC.

**Reverse conductance – Reverse conductance**

1e-6 S (default) | scalar

Value of the reverse conductance when the supply voltage becomes negative.

**Dependencies**

To enable this parameter, set **Load Type** to DC.

**Transition voltage width – Transition voltage width**

1e-3 V (default) | scalar

Transition voltage width. This value is used to ensure smooth transitions between the block's behaviours when the supply voltage drops below the **Minimum supply voltage** or becomes negative.

**Dependencies**

To enable this parameter, set **Load Type** to DC.

**Rated frequency – Rated frequency**

60 Hz (default) | scalar

Rated frequency.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Minimum supply voltage (RMS) — Minimum AC supply voltage

200 V (default) | scalar

Minimum voltage, in RMS, that the AC supply needs to give the block to act as a dynamic load.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Minimum active power — Minimum active power

1e-3 W (default) | scalar

Minimum active power of the block.

#### Dependencies

To enable this parameter, set **Load Type** to AC.

#### Faults

#### Enable open circuit fault — Whether to enable fault modeling

Off (default) | On

Select **On** to enable fault modeling and expose the associated parameters in the **Faults** section, which enables you to select the reporting method and specify the time at which the fault triggers.

#### Reporting when a fault occurs — Whether to issue assertion when fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- **None** — The block does not issue an assertion.
- **Warn** — The block issues a warning.
- **Error** — Simulation stops with an error.

#### Dependencies

To enable this parameter, set **Enable open circuit fault** to On.

#### Simulation time for fault event — Time before entering faulted state

1 s (default) | scalar

Simulation time when the block enters the faulted state.

#### Dependencies

To enable this parameter, set **Enable open circuit fault** to On.

#### Open-circuit conductance — Open-circuit conductance

1e-4 S (default) | scalar

Conductance during the open-circuit state.

### **Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

## **Version History**

**Introduced in R2020b**

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Dynamic Load (Three-Phase)

#### **Topics**

“Perform a Load-Flow Analysis Using Simscape Electrical”

“Frequency and Time Simulation Mode”

# Dynamic Load (Three-Phase)

Balanced three-phase dynamic load

**Library:** Simscape / Electrical / Passive



## Description

The Dynamic Load (Three-Phase) block models a balanced dynamic load for balanced three-phase supplies.

This table shows the relationship between the voltage from the supply,  $V$ , the active power at the **P** port,  $P_{in}$ , and the consumed active power of the block,  $P_{actual}$ .

Applicable Range of $V$ Values	Applicable Range of $P_{in}$ Values	Corresponding Consumed Active Power
$V > V_{min}$	$P_{in} > P_{min}$	$P_{actual} = P_{in}$
	$P_{in} < P_{min}$	$P_{actual} = P_{min}$
$V < V_{min}$	$P_{in} > P_{min}$	The block models a load with an impedance equal to: $R = V_{min}^2 \frac{P_{actual}}{P_{actual}^2 + Q_{actual}^2}$ $X = V_{min}^2 \frac{Q_{actual}}{P_{actual}^2 + Q_{actual}^2}$ $Z = \sqrt{R^2 + X^2}$
	$P_{in} < P_{min}$	$P_{actual} = P_{min}$

Where:

- $V_{min}$  is the value of the “Minimum supply voltage (phase-to-phase RMS)” on page 1-0 parameter.
- $P_{min}$  is the value of the “Minimum active power” on page 1-0 parameter.
- $P_{actual}$  is the consumed active power of the Dynamic Load (Three-Phase) block.
- $Q_{actual}$  is the consumed reactive power of the Dynamic Load (Three-Phase) block.

The consumed reactive power of the block is equal to the **Q** port as long as the voltage from the supply is equal to or greater than the value specified for the **Minimum supply voltage (phase-to-phase RMS)** parameter. Otherwise, the block models a load with an impedance equal to:

$$R = V_{\min}^2 \frac{P_{\text{actual}}}{P_{\text{actual}}^2 + Q_{\text{actual}}^2}$$

$$X = V_{\min}^2 \frac{Q_{\text{actual}}}{P_{\text{actual}}^2 + Q_{\text{actual}}^2}$$

$$Z = \sqrt{R^2 + X^2}$$

## Faults

The Dynamic Load (Three-Phase) block allows you to model an electrical fault as an open circuit. The block can trigger fault events at a specific time.

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Load-Flow Analysis

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Limitations

If you connect this block to an inductive node or a current source, you must provide a large parasitic.

## Ports

### Input

#### **P — Real power**

physical signal

Physical signal port associated with real power, specified as a scalar.

#### **Q — Reactive power**

physical signal

Physical signal port associated with reactive power, specified as a scalar.

### Conserving

#### **~ — Three-phase voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase voltage.



## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### Minimum supply voltage (phase-to-phase RMS) — Minimum supply voltage

100 V (default) | scalar

Minimum voltage, in phase-to-phase RMS, that the supply needs to give the block so that it acts as a dynamic load.

#### Minimum active power — Minimum active power

1e-3 W (default) | scalar

Minimum active power.

### Faults

#### Enable open circuit fault — Whether to enable fault modeling

Off (default) | On

Select On to enable fault modeling and expose the associated parameters in the **Faults** section, which enables you to select the reporting method and specify the time at which the fault triggers.

#### Reporting when a fault occurs — Whether to issue assertion when fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

### Dependencies

To enable this parameter, set **Enable open circuit fault** to On.

#### Simulation time for fault event — Time before entering faulted state

1 s (default) | scalar

Simulation time when the block enters the faulted state.

### Dependencies

To enable this parameter, set **Enable open circuit fault** to On.

**Open-circuit conductance — Open-circuit conductance**

1e-4 S (default) | scalar

Conductance during the open-circuit state.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

## Version History

**Introduced in R2019a**

**Library Path Change**

*Behavior changed in R2020b*

The Dynamic Load (Three-Phase) block has been moved from the **Simscape > Electrical > Passive > RLC Assemblies** library to the **Simscape > Electrical > Passive** library.

If you have a script that depends on the library path or the Simscape component path of this block, you need to update it to reflect these changes.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

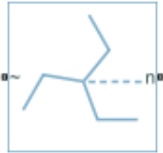
## See Also

Delta-Connected Load | Wye-Connected Variable Load | Wye-Connected Variable Load (lagging)

# Earthing Transformer

Three-phase earthing transformer in zigzag configuration

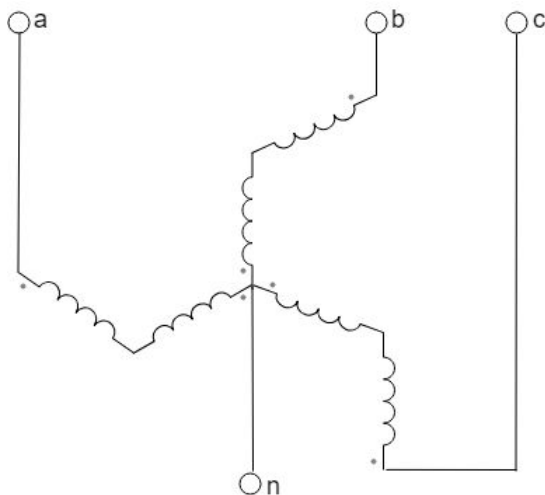
**Library:** Simscape / Electrical / Passive / Transformers



## Description

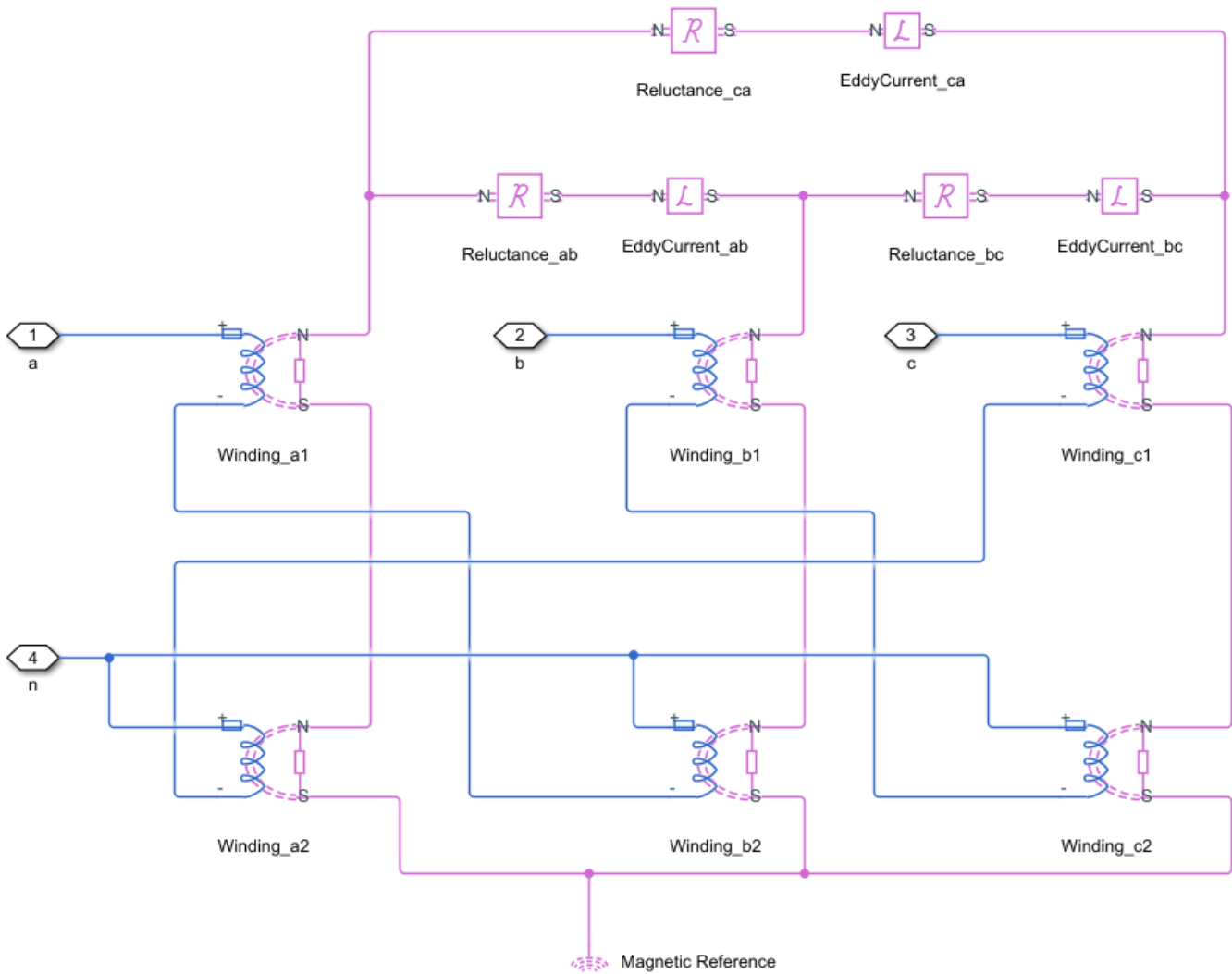
The Earthing Transformer block models a three-phase zigzag-connection earthing or grounding transformer. The block allows you to provide an artificial neutral in an ungrounded three-phase power system. The artificial neutral creates a low-impedance path for the zero-sequence current and high-impedance path for the positive-sequence current. To accommodate a phase-to-neutral load, the return path allows a delta-connected supply.

For more information on using a three-phase zigzag-connection earthing or grounding transformer, see “Engineering Applications” on page 1-513



## Equations

This block is implemented in both electrical and magnetic domains, representing the physics of phase windings and magnetic core. The equivalent circuit for the block includes windings, reluctances, and eddy currents.



For a zigzag configuration in magnetic circuit, the  $a$ -,  $b$ -, and  $c$ -phases represent each leg of the magnetic core.

You can obtain the relationship between the electrical and the magnetic properties by applying several mathematical steps based on the magnetic flux conservation rule and on the Hopkinson's law.

For earthing transformers, the couplings between the phases and all the windings are identical. In the equations below:

- $n$  is the number of turns of the winding.
- $R$  is the magnetizing reluctance between phases.
- $R_L$  is the leakage reluctance.
- $L_{eddy}$  is the magnetic inductance due to eddy current losses between phases.

Then the relationship between the inductance matrix in electrical domain and the reluctance matrix in the magnetic domain can be written as:

$$L = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} = \begin{bmatrix} n^2\left(\frac{2}{Rl} + \frac{6}{R}\right) & -3\frac{n^2}{R} & -3\frac{n^2}{R} \\ -3\frac{n^2}{R} & n^2\left(\frac{2}{Rl} + \frac{6}{R}\right) & -3\frac{n^2}{R} \\ -3\frac{n^2}{R} & -3\frac{n^2}{R} & n^2\left(\frac{2}{Rl} + \frac{6}{R}\right) \end{bmatrix}$$

The diagonal elements represent the sum of a leakage and magnetizing inductance for each phase, such that:

$$L_l = \frac{2n^2}{Rl}$$

Therefore:

$$R_L = \frac{2n^2}{L_l}$$

$$R = \frac{9n^2}{L_m}$$

$$L_{eddy} = \frac{9n^2}{R_m}$$

## Ports

### Conserving

#### ~ – Three-phase voltage

electrical

Expandable three-phase electrical conserving port associated with three-phase voltage.

#### n – Neutral

electrical

Electrical three-phase electrical conserving port associated with the neutral point.

## Parameters

### Modeling option – Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Rated apparent power — Rated apparent power**

100e6 V\*A (default) | positive scalar

Rated apparent power in SI units. The value must be greater than 0.

**Rated voltage — Rated voltage**

24e3 V (default) | positive scalar

Rated voltage in SI units. The value must be greater than 0.

**Rated electrical frequency — Rated electrical frequency**

60 Hz (default) | positive scalar

Rated electrical frequency in SI units. The value must be greater than 0.

**Units — Parameterization unit system**

Per unit (default) | SI

System of units for impedance parameterization.

**Dependencies**

The visibility of other parameters depends on the value of this parameter.

**Zero-sequence resistance (pu) — Zero-sequence resistance**

0.01 (default) | positive scalar

Per-unit zero-sequence resistance. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to Per unit.

**Zero-sequence reactance (pu) — Zero-sequence reactance**

0.3 (default) | positive scalar

Per-unit zero-sequence reactance. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to Per unit.

**Shunt magnetizing resistance (pu) — Shunt magnetizing resistance**

500 (default) | positive scalar

Per-unit shunt magnetizing resistance. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to Per unit.

**Shunt magnetizing reactance (pu) — Shunt magnetizing reactance**

500 (default) | positive scalar

Per-unit shunt magnetizing reactance. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to Per unit.

**Zero-sequence resistance – Zero-sequence resistance**

0.0576 Ohm (default) | positive scalar

Zero-sequence resistance in SI units. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to SI.

**Zero-sequence inductance – Zero-sequence inductance**

1.728 H (default) | positive scalar

Zero-sequence inductance in SI units. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to SI.

**Shunt magnetizing resistance – Shunt magnetizing resistance**

2880 Ohm (default) | positive scalar

Shunt magnetizing resistance in SI units. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to SI.

**Shunt magnetizing inductance – Shunt magnetizing inductance**

2880 H (default) | positive scalar

Shunt magnetizing inductance in SI units. The value must be greater than 0.

**Dependencies**

This parameter is only visible when the **Units** parameter is set to SI.

**More About****Engineering Applications**

To facilitate grounding and detection of an earth fault, earthing transformers provide a ground path to either an ungrounded "Y" or a delta-connected system. They are typically used to:

- Provide a relatively low impedance path to ground, thereby maintaining the system neutral at or near ground potential.
- Limit the magnitude of transient overvoltages when re-striking ground faults occur.
- Provide a source of ground fault current during line-to-ground faults.
- Permit the connection of phase-to-neutral loads when desired.

**Version History**

**Introduced in R2019a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Ideal Transformer | Nonlinear Transformer | Tap-Changing Transformer | Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer



# Eddy Current

Eddy current loss representation

**Library:** Simscape / Electrical / Passive



## Description

The Eddy Current block models the effect of eddy currents by generating a magnetomotive force (MMF) that opposes changes in the magnetic flux. In addition, the block models parasitic effects using a series reluctance and a parallel leakage permeance, and thermal effects using an optional thermal port, **H**.

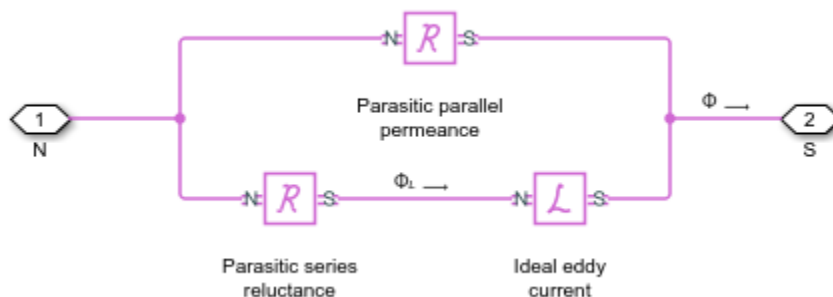
Eddy currents in magnetic core materials are caused by time-varying magnetic fields. These changing field densities induce voltage potentials within the material, causing current to flow in closed loops. Such currents are usually undesired and cause heating of the magnetic material.

Use this block to add eddy current losses in the magnetic domain to a custom transformer or other magnetic component.

The eddy current component is sometimes referred to as magnetic inductance because it is the magnetic-domain analog to an inductor in the electrical domain.

## Equations

This is the equivalent magnetic circuit for the block, including the eddy current path (bottom) and parallel leakage path (top).



In the diagram:

- $\Phi$  is the total flux at the terminals. This flux is the summation of the eddy current loop and the parallel leakage path.
- $\Phi_L$  is the flux through the eddy current loop.

The block calculates the terminal MMF,  $\mathcal{F}$ , and flux,  $\Phi$ , as:

$$\mathcal{F} = G_{\text{eddy}} \frac{d\Phi_L}{dt} + \mathcal{R}\Phi_L$$

$$\Phi = \mathcal{F}P + \Phi_L$$

where:

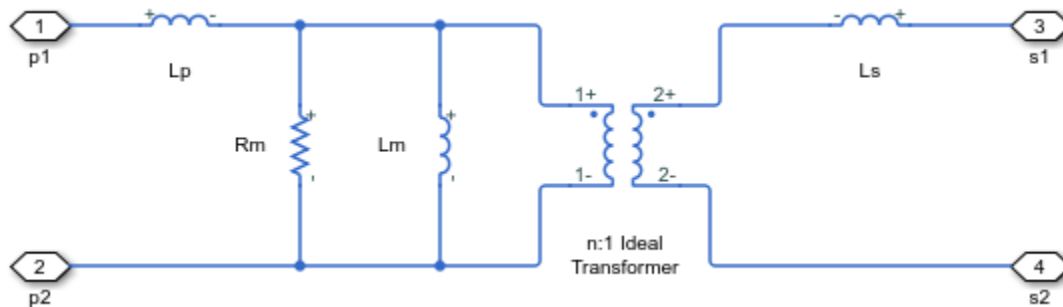
- $G_{\text{eddy}}$  is the conductance of the eddy current loop.
- $\mathcal{R}$  is the parasitic series reluctance of the eddy current path.
- $P$  is the parallel permeance.

Because the parasitic series reluctance and parallel permeance are lossless, the total dissipated power over the block is:

$$P_{\text{diss}} = G_{\text{eddy}} \left( \frac{d\Phi_L}{dt} \right)^2$$

### Relating Transformer Electrical and Magnetic Models

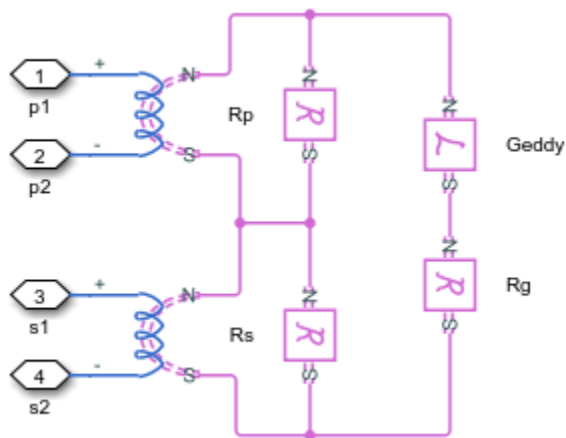
In the electrical domain, eddy current losses are modeled using a parallel resistance across the primary winding. This is the equivalent circuit for a nonideal two-winding transformer.



In the diagram:

- $L_p$  and  $L_s$  are the self-inductances of the primary and secondary windings, respectively.
- $L_m$  is the mutual inductance between the two windings.
- $R_m$  is the mutual resistance between the two windings, caused by the eddy current losses.

This two-winding transformer can similarly be represented in the magnetic domain. This is the equivalent circuit.



In the diagram:

- $\mathcal{R}_p$  and  $\mathcal{R}_s$  are the reluctances associated with the primary and secondary windings, respectively.
- $\mathcal{R}_g$  is the reluctance associated with the magnetic coupling of the two windings.
- $n$  is the turns ratio between the two windings.

The electrical- and magnetic-domain circuits are equivalent if:

$$\mathcal{R}_g = \frac{n^2}{L_m}$$

$$\mathcal{R}_p = \frac{n^2}{L_p}$$

$$\mathcal{R}_s = \frac{1}{L_s}$$

$$G_{\text{eddy}} = \frac{n^2}{R_m}$$

You can use these relationships to calculate the equivalent two-winding transformer properties for one domain from the other.

## Ports

### Conserving

#### N – North terminal

magnetic

Magnetic conserving port associated with the north terminal of the block.

#### S – South terminal

magnetic

Magnetic conserving port associated with the south terminal of the block.

**H – Thermal port**

thermal

Thermal conserving port associated with the thermal port of the block.

**Dependencies**

To enable this port, select the **Show thermal port** parameter.

**Parameters****Conductance of eddy current loop – Eddy current conductance**

1 1/Ohm (default) | positive number

Conductance,  $G_{eddy}$ , of the eddy current loop. This component is the magnetic-domain analog of inductance in the electrical domain.

**Parasitic series reluctance – Series reluctance**

1e-3 1/H (default) | nonnegative number

Parasitic reluctance,  $\mathcal{R}$ , of the eddy current loop.

**Parasitic parallel permeance – Parallel leakage permeance**

1e-6 H (default) | nonnegative number

Parasitic permeance,  $P$ , of the parallel leakage path. To aid simulation convergence in some circuit topologies, set this parameter to a small value.

**Show thermal port – Thermal port visibility**

Off (default) | On

Whether to enable the thermal port of the block.

**Thermal mass – Thermal mass**

0.1 J/K (default) | positive number

Thermal mass of the block. This parameter models the ability of the Eddy Current block to absorb and store thermal energy.

**Dependencies**

To enable this parameter, select the **Show thermal port** parameter.

**Version History****Introduced in R2018a****References**

- [1] Brown, A. D., J. N. Ross, and K. G. Nichols. "Time-domain simulation of mixed nonlinear magnetic and electronic systems." *IEEE Transactions on Magnetics*. Vol. 37, Number 1, 2001, pp. 522-532.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

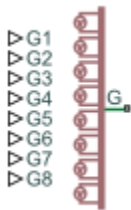
### **See Also**

Nonlinear Transformer | Mutual Inductor

## Eight-Pulse Gate Multiplexer

Multiplex gate voltage signals to Bidirectional DC-DC Converter block

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



### Description

The Eight-Pulse Gate Multiplexer block routes gate voltage signals to the eight switching devices in a Bidirectional DC-DC Converter block. The block multiplexes the eight separate gate signals into a single vector.

When you set the **Gate-control port** parameter to `Electrical`, the block exposes conserving ports that represent electrical reference points. These ports are associated with the individual phase voltages that connect to the high-side switching devices and the negative DC voltage common to each low-side switching device in the Bidirectional DC-DC Converter block.

### Ports

#### Input

##### **G1 — Gate-control voltage signal 1, V**

physical signal

Physical signal port associated with the gate terminal of the first switching device in a Bidirectional DC-DC Converter block.

#### Dependencies

To enable this port, set **Gate-control port** to `PS`.

Data Types: `double`

##### **G2 — Gate-control voltage signal 2, V**

physical signal

Physical signal port associated with the gate terminal of the second switching device in a Bidirectional DC-DC Converter block.

#### Dependencies

To enable this port, set **Gate-control port** to `PS`.

Data Types: `double`

**G3 — Gate-control voltage signal 3, V**

physical signal

Physical signal port associated with the gate terminal of the third switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G4 — Gate-control voltage signal 4, V**

physical signal

Physical signal port associated with the gate terminal of the fourth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G5 — Gate-control voltage signal 5, V**

physical signal

Physical signal port associated with the gate terminal of the fifth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G6 — Gate-control voltage signal 6, V**

physical signal

Physical signal port associated with the gate terminal of the sixth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G7 — Gate-control voltage signal 7, V**

physical signal

Physical signal port associated with the gate terminal of the seventh switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G8 — Gate-control voltage signal 8, V**

physical signal

Physical signal port associated with the gate terminal of the eighth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**Conserving****G1 — Gate-control voltage signal 1**

electrical

Electrical conserving port associated with the gate terminal of the first switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**a — A-phase reference point**

electrical

Electrical conserving port associated with the A-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G3 — Gate-control voltage signal 3**

electrical

Electrical conserving port associated with the gate terminal of the third switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**b — B-phase reference point**

electrical

Electrical conserving port associated with the B-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G5 — Gate-control voltage signal 5**

electrical

Electrical conserving port associated with the gate terminal of the fifth switching device in a Bidirectional DC-DC Converter block.



**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**c — C-phase reference point**

electrical

Electrical conserving port associated with the *C*-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G7 — Gate-control voltage signal 7**

electrical

Electrical conserving port associated with the gate terminal of the seventh switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**d — D-phase reference point**

electrical

Electrical conserving port associated with the *D*-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G2 — Gate-control voltage signal 2**

electrical

Electrical conserving port associated with the gate terminal of the second switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G4 — Gate-control voltage signal 4**

electrical

Electrical conserving port associated with the gate terminal of the fourth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G6 — Gate-control voltage signal 6**

electrical

Electrical conserving port associated with the gate terminal of the sixth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to `Electrical`.

**G8 — Gate-control voltage signal 8**

`electrical`

Electrical conserving port associated with the gate terminal of the eighth switching device in a Bidirectional DC-DC Converter block.

**Dependencies**

To enable this port, set **Gate-control port** to `Electrical`.

**L — DC reference point**

`electrical`

Electrical conserving port associated with the DC negative connection for the low-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to `Electrical`.

**Parameters****Gate-control port — Gate control port**

`PS` (default) | `Electrical`

Whether to expose physical signal or electrical control ports.

**Version History**

Introduced in R2022a

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

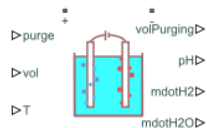
**See Also**

[Bidirectional DC-DC Converter](#) | [Two-Pulse Gate Multiplexer](#) | [Four-Pulse Gate Multiplexer](#) | [Six-Pulse Gate Multiplexer](#) | [Twelve-Pulse Gate Multiplexer](#)

# Electrolyzer

Electric load for electrolyzer

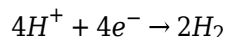
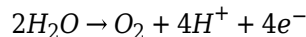
**Library:** Simscape / Electrical / Sources



## Description

The Electrolyzer block represents the electric load associated with an electrolyzer. An electrolyzer consists of an anode and a cathode separated by an electrolyte. The block calculates the amount of produced hydrogen based on the electrical energy provided and the water temperature in the tank.

In polymer electrolyte membrane (PEM) electrolyzers, these chemical reactions define the anode and cathode conversions:



An electrolysis stack comprises several series-connected individual electrolyzer cells. Specify the **Assumption pH** parameter to model this block either in a pH-dependent state or in an ideal state with constant pH.

## Equations

The Electrolyzer block calculates the electrical power by using

$$P_E = \frac{v^2}{R_T},$$

where  $v$  is the voltage and  $R_T$  is the total resistance. The total resistance,  $R_T$ , is defined by

$$\frac{1}{R_T} = \frac{1}{R_{el}} + \frac{1}{R},$$

where:

- $R$  is the external resistance.
- $R_{el} = \frac{\rho(pH)x}{A}$ . This value is a resistive term obtained from the cross-sectional transport area,  $A$ , the distance anode-cathode,  $x$ , and the electrical resistivity of the solution,  $\rho(pH)$ .

The molar energy stored across the electrical load enables the process of water electrolysis and is calculated by this equation:

$$G_E = \frac{P_E}{1 \text{ mol/s}}.$$

To accomplish electrolysis, the electrolyzer requires a minimum amount of energy

$$G_{\min} = \Delta H - \frac{T}{T_0} T_{\Delta S},$$

where:

- $\Delta H$  is the water enthalpy.
- $T_{\Delta S}$  is the reaction entropy.
- $T$  is the water temperature.
- $T_0$  is the reference temperature.

If the electrical energy is not sufficient to dissociate the water, the electrolyzer does not produce hydrogen. Therefore, the Electrolyzer block calculates the mole rate of electron,  $mol_e$ , and the mole rate of hydrogen,  $mol_{H_2}$

$$mol_e = \begin{cases} \frac{\mu(G_E - G_{\min})}{e_V N_A} & G_E > G_{\min} \\ 0 & G_E \leq G_{\min} \end{cases}$$

$$mol_{H_2} = \begin{cases} \frac{N_c mol_e}{2}, & G_E > G_{\min} \\ 0, & G_E \leq G_{\min} \end{cases}$$

where:

- $N_A$  is the Avogadro constant.
- $e_V$  is the energy per electron.
- $N_c$  is the number of cells.
- $\mu$  is the temperature-dependent efficiency of the electrolysis.

The current that flows through the electrolyzer tank is then obtained as

$$i = mol_e F,$$

where  $F$  is the Faraday constant.

The Electrolyzer block calculates the mass rates of consumed water and generated hydrogen by using these equations:

$$\dot{m}_{H_2O} = mol_{H_2} M_{H_2O}$$

$$\dot{m}_{H_2} = mol_{H_2} M_{H_2}$$

where  $M_{H_2O}$  is the molar mass of water and  $M_{H_2}$  is the molar mass of hydrogen.

To compute the pH of the water in the tank, the block uses

$$pH = -\ln\left(\frac{x_{hp}}{mol_{H_2\_acc}}\right),$$

where:

- $$x_{hp} = \begin{cases} \int mol_e, & pH \text{ assumed constant} \\ \int (1 - purge)mol_e, & pH \text{ assumed dynamic} \end{cases}$$
 captures the floating hydrons (H+) in the solution.
- $mol_{H_2\_acc} = \frac{V\rho_{H_2O}(T)}{M_{H_2O}}$  is the number of moles of water.
- $V$  is the volume.
- $\rho_{H_2O}(T)$  is the density of the water in the tank.

The residual protons in the solution inside the electrolyzer naturally increases the pH and conductivity of the water in the tank over time. The increase in conductivity leads to a run-away phenomenon. To prevent the run-away, you must purge the water in the tank by taking out the volume of the tank and fill it with clean water.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the anode of the electrolyzer tank.

#### - – Negative terminal

electrical

Electrical conserving port associated with the cathode of the electrolyzer tank.

### Input

#### purge – Purge logic input, unitless

physical signal

Purge logic input, specified as a physical signal. This signal enables the substitution of the water in the tank.

#### vol – Water volume, m<sup>3</sup>

physical signal

Water volume in the electrolyzer tank, in m<sup>3</sup>, specified as a physical signal.

#### T – Water temperature, K

physical signal

Water temperature, in K, specified as a physical signal.

### Output

#### volPurging – Purged water volume, m<sup>3</sup>

physical signal

Volume of water consumed for purging, in m<sup>3</sup>, specified as a physical signal.

**pH — Water pH, unitless**

physical signal

Water pH in the electrolyzer tank, specified as a physical signal.

**mdotH2 — Mass flow rate of hydrogen, g/s**

physical signal

Mass flow rate of the hydrogen, in g/s, specified as a physical signal.

**mdotH2O — Mass flow rate of consumed water, g/s**

physical signal

Mass flow rate of the consumed water, in g/s, specified as a physical signal.

**Parameters****Assumption pH — Constant or dynamic pH**

Constant (default) | Dynamic

Whether to assume constant or dynamic pH in the electrolyzer tank.

**Resistance — External resistance**

10 Ohm (default) | nonnegative scalar

Amount of external resistance.

**Cross-sectional transport area — Cross-sectional transport area**0.04 m<sup>2</sup> (default) | positive scalar

Normal area to the vector that links the anode and the cathode. Positively charged ions travel across this area from anode to cathode, where they get reduced and the hydrogen is produced.

**Distance anode-cathode — Distance between anode and cathode**

0.01 m (default) | positive scalar

Distance between the anode and the cathode, submerged in the electrolytic solution.

**Number of electrode pairs — Electrode pairs**

4 (default) | positive scalar

Number of electrode pairs.

**Number of cells — Number of cells**

50 (default) | positive scalar

Number of cells in the electrolyzer.

**Initial pH — Initial pH**

0 (default) | scalar

Initial pH value of the water in the tank.

**Constant pH — Constant pH**

6 (default) | scalar

Constant pH value of the water in the tank.

#### Dependencies

To enable this parameter, set **Assumption pH** to Constant.

#### pH vector, pH\_vec – pH vector

[5, 5.5, 6, 6.5, 7] (default) | row vector of positive scalars

Vector of pH values. This parameter must have the same number of elements as the **Electrical resistivity vector, rho\_vec** parameter.

#### Dependencies

To enable this parameter, set **Assumption pH** to Dynamic

#### Electrical resistivity vector, rho\_vec – Electrical resistivity vector

[.2, .75, 2.5, 8, 18.2] MOhm\*cm (default) | vector of scalars

Vector of electrical resistivity values. This parameter must have the same number of elements as the **pH vector, pH\_vec** parameter.

#### Dependencies

To enable this parameter, set **Assumption pH** to Dynamic

#### Temperature vector for efficiency, T\_vec – Efficiency temperature vector

[273, 298, 323, 348, 373] K (default) | vector of scalars

Vector of temperatures for efficiency. This parameter must have the same number of elements as the **Electrolysis efficiency vector, efficiency(T)** parameter.

#### Electrolysis efficiency vector, efficiency(T) – Electrolysis efficiency vector

ones(5, 1) (default) | vector of scalars

Vector of electrolysis efficiency values. This parameter must have the same number of elements as the **Temperature vector for efficiency, T\_vec** parameter.

## Version History

Introduced in R2022a

## References

- [1] Maximilian Schalenbach et al. "A Perspective on Low-Temperature Water Electrolysis - Challenges in Alkaline and Acidic Technology". *International Journal of Electrochemical Science* 13, (2018): 1173-1226.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

**See Also**

Fuel Cell



# Environment Parameters

Set parameters that apply to all connected SPICE-compatible blocks

**Library:** Simscape / Electrical / Utilities



## Description

The Environment Parameters block lets you set parameters that apply to all SPICE-compatible blocks in an electrical network:

- Circuit temperature
- Minimum conductance

If your model does not contain an Environment Parameters block, all blocks use the default values of these parameters. To override the default values, connect every network in the system to an Environment Parameters block.

---

**Note** The simple semiconductor models in the Semiconductors sublibrary are not temperature dependent, so the Environment Parameters block only changes the minimum conductance parameter used by the exponential diode and bipolar transistor models.

---

## Ports

### Conserving

#### OUT — Output port

electrical

Electrical conserving port associated with the environment parameters output

## Parameters

### Temperature — SPICE temperature

300.15 K (default)

The temperature of the connected SPICE-compatible blocks.

### GMIN — Minimum conductance

1e-12 1/Ohm (default)

The minimum conductance for applicable connected SPICE-compatible blocks.

## Version History

**Introduced in R2008a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Pulse Current Source | Pulse Voltage Source | Piecewise Linear Current Source | Piecewise Linear Voltage Source

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Exponential Current Source

Exponential pulse current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Exponential Current Source block represents a current source whose output current value is an exponential pulse as a function of time and is independent of the voltage across the terminals of the source. The following equations describe the current through the source as a function of time:

$$I_{out}(0 \leq \text{Time} \leq \text{TDR}) = I1$$

$$I_{out}(\text{TDR} < \text{Time} \leq \text{TDF}) = I1 + (I2 - I1) * (1 - e^{-(\text{Time} - \text{TDR})/\text{TR}})$$

$$I_{out}(\text{TDF} < \text{Time}) = I1 + (I2 - I1) * (e^{-(\text{Time} - \text{TDF})/\text{TF}} - e^{-(\text{Time} - \text{TDR})/\text{TR}})$$

where:

- $I1$  is the **Initial value**, **I1** parameter value.
- $I2$  is the **Pulse value**, **I2** parameter value.
- $\text{TDR}$  is the **Rise delay time**, **TDR** parameter value.
- $\text{TR}$  is the **Rise time**, **TR** parameter value.
- $\text{TDF}$  is the **Fall delay time**, **TDF** parameter value.
- $\text{TF}$  is the **Fall time**, **TF** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

**+ – Positive voltage**

electrical

Electrical conserving port associated with the DC current source positive voltage

**- — Negative voltage**

electrical

Electrical conserving port associated with the DC current source negative voltage

**Parameters****Initial value, I1 — Output current initial value**

0 A (default)

Value of the output current at time zero, in A.

**Pulse value, I2 — Output current asymptotic value**

0 A (default)

Asymptotic value of the output current when the output is high, in A.

**Rise delay time, TDR — Rise delay time**

0 s (default)

Rise time delay, in s.

**Rise time, TR — Rise time**

1e-9 s (default) | positive scalar

Time it takes the output current to rise from the **Initial Value, I1** value to the **Pulse Value, I2** value.

**Fall delay time, TDF — Fall delay time**

0 s (default)

Fall time delay. The default value is 0s, which differs from the SPICE default value.

**Fall time, TF — Fall time**

1e-9 s (default) | positive scalar

Time it takes the output current to fall from the **Pulse value, I2** value to the **Initial value, I1** value.

**Include parallel conductance, GMIN — Whether to include parallel conductance**

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

**Values of parallel conductance, GMIN**

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

**Version History**

Introduced in R2008a

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also****Simscape Blocks**

Exponential Voltage Source | Environment Parameters

**Functions**

subcircuit2ssc

**Topics**

"Additional Parameterization Workflows"

"Converting a SPICE Netlist to Simscape Blocks"

"Parameterize an Exponential Diode from SPICE Netlist"

# Exponential Voltage Source

Exponential pulse voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Exponential Voltage Source block represents a voltage source whose output voltage value is an exponential pulse as a function of time and is independent of the current through the source. The following equations describe the output current as a function of time:

$$V_{out}(0 \leq Time \leq TDR) = V1$$

$$V_{out}(TDR < Time \leq TDF) = V1 + (V2 - V1) * (1 - e^{-(Time - TDR)/TR})$$

$$V_{out}(TDF < Time) = V1 + (V2 - V1) * (e^{-(Time - TDF)/TF} - e^{-(Time - TDR)/TR})$$

where:

- $V1$  is the **Initial value**, **V1** parameter value.
- $V2$  is the **Pulse value**, **V2** parameter value.
- $TDR$  is the **Rise delay time**, **TDR** parameter value.
- $TR$  is the **Rise time**, **TR** parameter value.
- $TDF$  is the **Fall delay time**, **TDF** parameter value.
- $TF$  is the **Fall time**, **TF** parameter value.

## Ports

### Conserving

#### + – Positive voltage

electrical

Electrical conserving port associated with the DC current source positive voltage

#### - – Negative voltage

electrical

Electrical conserving port associated with the DC current source negative voltage

## Parameters

### Initial value, V1 – Output voltage initial value

0 V (default)

Value of the output voltage at time zero, in V.

**Pulse value, V2 – Output voltage asymptotic value**

0 V (default)

Asymptotic value of the output voltage when the output is high, in V.

**Rise delay time, TDR – Rise delay time**

0 s (default)

Rise time delay, in s.

**Rise time, TR – Rise time**

1e-9 s (default) | positive scalar

Time it takes the output current to rise from the **Initial Value, V1** value to the **Pulse Value, V2** value.

**Fall delay time, TDF – Fall delay time**

0 s (default)

Fall time delay.

**Fall time, TF – Fall time**

1e-9 s (default) | positive scalar

Time it takes the output current to fall from the **Pulse value, V2** value to the **Initial value, V1** value.

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Exponential Current Source | Environment Parameters

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Fault

Electrical fault with temporal, behavioral, or external trigger

**Library:** Simscape / Electrical / Utilities



## Description

The Fault block allows you to represent an electrical fault as an instantaneous change in resistance. You can use it to replicate both open-circuit and short-circuit fault behaviors. The block can trigger fault events:

- At a specific time. The block can also stop imposing a temporal fault after a specified time duration.
- When a predefined voltage range or current range is exceeded
- When an external trigger signal goes high or low

Optionally, the external trigger option also permits the fault to reset when the trigger signal reverts. You can enable or disable all three trigger mechanisms separately. If more than one trigger mechanism is required in a simulation, you can use them together.

When no fault is triggered, the resistance between the two electrical ports is defined by the **Unfaulted resistance** parameter value. The default value for this parameter is `inf` ohms, that is the ports are open-circuit. When a fault is triggered, the block changes the resistance between the two electrical ports to the **Faulted resistance** value. The default value for this parameter is `1e-3` ohms, that is the ports are short-circuited.

You can choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

The physical output  $X$  represents the fault state; if the block is faulted, the output is 1, otherwise, it is 0. The physical signal input  $F$  is the external fault trigger signal and is used only if **Enable external fault trigger** is set to Yes.

## Ports

### Input

**F — External fault trigger signal**

physical

Physical signal input port associated with the external fault trigger signal.

### Output

**X — Fault state**

physical



Physical signal output port associated with the fault state. It outputs 1 if the block is faulted, 0 otherwise.

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the fault positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the fault negative terminal.

## Parameters

### Main

#### Unfaulted resistance – Resistance with no fault

inf Ohm (default)

Resistance between the + and - ports when there is no fault.

#### Faulted resistance – Resistance with fault

1e-3 Ohm (default)

Resistance between the + and - ports when the block is in the faulted state.

#### Reporting when a fault occurs – Fault occurrence report

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None – The block does not issue an assertion.
- Warn – The block issues a warning.
- Error – Simulation stops with an error.

### Temporal Trigger

#### Enable temporal fault trigger – Temporal fault trigger option

No (default) | Yes

Select Yes to enable time-based fault triggering.

#### Simulation time for fault event – Simulation time for fault event

1 s (default)

Set the simulation time that the block enters the fault state.

### Dependencies

This parameter is visible only when you set the **Enable temporal fault trigger** parameter to Yes.

**Fault duration — Duration time for the fault**

inf s (default)

Set the simulation duration time for the fault. The block exits the fault state when the duration time is over.

**Dependencies**

This parameter is visible only when you set the **Enable temporal fault trigger** parameter to Yes.

**Behavioral Trigger****Enable behavioral fault trigger — Behavioral fault trigger option**

No (default) | Yes

Select Yes to enable behavioral fault triggering.

**Permissible voltage range — Permissible voltage range**

[-100, 100] V (default)

Specify a vector of length 2 that defines the permissible voltage range. If the voltage is outside this range for longer than the **Time to fail when exceeding voltage range** parameter value, then the block enters the fault state.

**Dependencies**

This parameter is visible only when you set the **Enable behavioral fault trigger** parameter to Yes.

**Time to fail when exceeding voltage range — Time to fail when exceeding voltage range**

1 s (default)

Set the maximum length of time that the voltage can be outside the permissible voltage range without triggering the fault.

**Dependencies**

This parameter is visible only when you set the **Enable behavioral fault trigger** parameter to Yes.

**Permissible current range — Permissible current range**

[-1, 1] A (default)

Specify a vector of length 2 that defines the permissible current range. If the current is outside this range for longer than the **Time to fail when exceeding current range** parameter value, then the block enters the fault state.

**Dependencies**

This parameter is visible only when you set the **Enable behavioral fault trigger** parameter to Yes.

**Time to fail when exceeding current range — Time to fail when exceeding current range**

1 s (default)

Set the maximum length of time that the current can be outside the permissible current range without triggering the fault.

**Dependencies**

This parameter is visible only when you set the **Enable behavioral fault trigger** parameter to Yes.

**External Trigger****Enable external fault trigger – External fault trigger option**

No (default) | Yes

Select Yes to enable external fault triggering. The physical signal input F provides the external fault trigger signal.

**External fault trigger – External fault trigger condition**

Faulted if  $F \geq$  Fault threshold (default) | Faulted if  $F \leq$  Fault threshold

Choose the fault condition:

- Faulted if  $F \geq$  Fault threshold — The fault occurs when the external signal value becomes greater than, or equal to, the **Fault threshold** parameter value.
- Faulted if  $F \leq$  Fault threshold — The fault occurs when the external signal value becomes less than, or equal to, the **Fault threshold** parameter value.

**Dependencies**

This parameter is visible only when you set the **Enable external fault trigger** parameter to Yes.

**Fault threshold – Fault threshold**

0.5 (default)

The threshold value that triggers the fault when the external signal crosses it in the direction, specified by the fault condition.

**Dependencies**

This parameter is visible only when you set the **Enable external fault trigger** parameter to Yes.

**Fault resets when fault trigger reverts – Fault resets when fault trigger reverts**

No (default) | Yes

Select Yes to have the fault reset when the trigger signal reverts.

**Dependencies**

This parameter is visible only when you set the **Enable external fault trigger** parameter to Yes.

**Version History**

Introduced in R2015b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Resistor | Fault (Three-Phase)

## Fault (Three-Phase)

Signal-enabled or time-based, single-phase, two-phase, or three-phase, grounded or ungrounded fault

**Library:** Simscape / Electrical / Utilities



### Description

The Fault (Three-Phase) models any permutation of a single-phase, two-phase, or three-phase grounded or ungrounded fault. The block can impose both time- and event-based faults. The block can impose one or both type of faults, consecutively, at the same time, or for an overlapping time period.

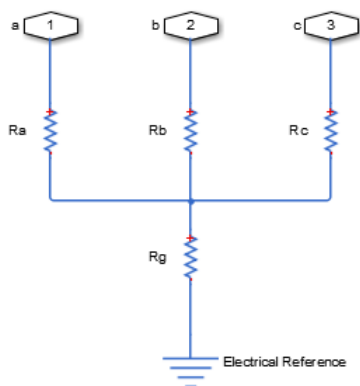
For a time-based or temporal fault, specify the fault activation time using the block **Fault start time** parameter. The fault becomes inactive when the fault duration that you specify elapses.

For an event-based fault, specify the fault activation threshold using the block **Threshold** parameter. An external control signal  $F$  enables the fault. The fault is active when  $F$  is greater than the threshold. The fault is inactive when  $F$  is less than or equal to the threshold.

The block can represent these permutations:

- Single-phase-to-ground fault ( $a-g$ ,  $b-g$ , or  $c-g$ )
- Two-phase fault ( $a-b$ ,  $b-c$ , or  $c-a$ )
- Two-phase-to-ground fault ( $a-b-g$ ,  $b-c-g$ , or  $c-a-g$ )
- Three-phase fault ( $a-b-c$ )
- Three-phase-to-ground fault ( $a-b-c-g$ )

The figure shows the equivalent circuit diagram for the block.



To determine the resistance in the equivalent circuit, apply the relevant equations from the table.

Failure mode		Value of $R_a$	Value of $R_b$	Value of $R_c$	Value of $R_g$
None / inactive		$\frac{1}{G_{pn}}$	$\frac{1}{G_{pn}}$	$\frac{1}{G_{pn}}$	Infinity / open circuit
Single phase to ground	$a-g$	$R_{pn}$	$\frac{1}{G_{pn}}$	$\frac{1}{G_{pn}}$	$R_{ng}$
	$b-g$	$\frac{1}{G_{pn}}$	$R_{pn}$	$\frac{1}{G_{pn}}$	$R_{ng}$
	$c-g$	$\frac{1}{G_{pn}}$	$\frac{1}{G_{pn}}$	$R_{pn}$	$R_{ng}$
Two phase	$a-b$	$R_{pn}$	$R_{pn}$	$\frac{1}{G_{pn}}$	Infinity / open circuit
	$b-c$	$\frac{1}{G_{pn}}$	$R_{pn}$	$R_{pn}$	Infinity / open circuit
	$c-a$	$R_{pn}$	$\frac{1}{G_{pn}}$	$R_{pn}$	Infinity / open circuit
Two phase to ground	$a-b-g$	$R_{pn}$	$R_{pn}$	$\frac{1}{G_{pn}}$	$R_{ng}$
	$b-c-g$	$\frac{1}{G_{pn}}$	$R_{pn}$	$R_{pn}$	$R_{ng}$
	$c-a-g$	$R_{pn}$	$\frac{1}{G_{pn}}$	$R_{pn}$	$R_{ng}$
Three phase	$a-b-c$	$R_{pn}$	$R_{pn}$	$R_{pn}$	Infinity / open circuit
Three phase to ground	$a-b-c-g$	$R_{pn}$	$R_{pn}$	$R_{pn}$	$R_{ng}$

where:

- $R_a$  is the resistance between the  $a$ -phase and the neutral point of a wye connection.
- $R_b$  is the resistance between the  $b$ -phase and the neutral point of a wye connection.
- $R_c$  is the resistance between the  $c$ -phase and the neutral point of a wye connection.
- $R_g$  is the resistance between the neutral point of a wye connection and electrical reference.
- $R_{pn}$  is the value of the **Faulted phase-neutral resistance** parameter.
- $R_{ng}$  is the value of the **Faulted neutral-ground resistance** parameter.
- $G_{pn}$  is the value of the **Unfaulted phase-neutral conductance** parameter.

## Ports

### Conserving

#### ~ – Three-phase port

electrical

Expandable electrical conserving three-phase port for connecting the fault to the system.

## Input

### F — Externally triggered fault control input

physical

Physical signal scalar control input port for enabling an externally triggered fault.

#### Dependencies

To enable this port, on the **External Trigger** settings, set the **Enable external fault trigger** parameter to Yes.

## Output

### X — Fault state

physical

Physical signal scalar output port for reporting the fault state:

- 1 — Faulted
- 0 — Unfaulted

#### Dependencies

To enable this port, on the **Main** settings, set the **State port** parameter to Visible.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Main

### Failure mode — Type of fault

None (default) | Single-phase to ground (a-g) | Single-phase to ground (b-g) | Single-phase to ground (c-g) | Two-phase (a-b) | Two-phase (b-c) | Two-phase (c-a) | Two-phase to ground (a-b-g) | Two-phase to ground (b-c-g) | Two-phase to ground (c-a-g) | Three-phase (a-b-c) | Three-phase to ground (a-b-c-g)

Type of fault to model. The visibility of related parameters depends on the fault model. Options are:

- None — Specifies that the fault is not active.
- Single-phase to ground (a-g)
- Single-phase to ground (b-g)

- Single-phase to ground (c-g)
- Two-phase (a-b)
- Two-phase (b-c)
- Two-phase (c-a)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase (a-b-c)
- Three-phase to ground (a-b-c-g)

**Faulted phase-neutral resistance – Faulted phase-neutral resistance**

1e-3 Ohm (default)

Resistance between the phase connection and the neutral point when the fault is active.

**Dependencies**

To enable this parameter set the **Failure mode** parameter to:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)
- Single-phase to ground (c-g)
- Two-phase (a-b)
- Two-phase (b-c)
- Two-phase (c-a)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase (a-b-c)
- Three-phase to ground (a-b-c-g)

**Faulted neutral-ground resistance – Faulted neutral-ground resistance**

1e-3 Ohm (default)

Resistance between the neutral point and the electrical reference when fault is active.

**Dependencies**

To enable this parameter set the **Failure mode** parameter to:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)
- Single-phase to ground (c-g)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)



- Three-phase to ground (a-b-c-g)

### State port — State port visibility

Hidden (default) | Visible

Option to hide or expose port **x**, a physical signal output port that reports the fault state:

- Hidden — Port **x** is not visible if you select this option.
- Visible — Port **x** is visible only if you select this option.

### External Trigger

#### Enable external fault trigger — External fault model

No (default) | Yes

External fault model. Choose between

- No — Do not model an external fault. Physical signal input port **F** is not visible if you select this option.
- Yes — Model an external fault. Related parameters and physical signal input port **F** are visible only if you select this option.

#### Fault threshold — Threshold value that triggers the fault

0.5 (default)

The threshold value that triggers the fault when the external signal crosses it in the direction, specified by the fault condition. This parameter is visible only if the **Enable external fault trigger** parameter is set to Yes. The default value is 0.5.

#### Dependencies

To enable this parameter, set the **Enable external fault trigger** parameter to Yes.

#### External fault trigger — Fault condition

Faulted if  $F \geq \text{Fault threshold}$  (default) | Faulted if  $F \leq \text{Fault threshold}$

Choose the fault condition:

- Faulted if  $F \geq \text{Fault threshold}$  — The fault occurs when the external signal value becomes greater than, or equal to, the **Fault threshold** parameter value.
- Faulted if  $F \leq \text{Fault threshold}$  — The fault occurs when the external signal value becomes less than, or equal to, the **Fault threshold** parameter value.

#### Dependencies

To enable this parameter, set the **Enable external fault trigger** parameter to Yes.

#### Fault resets when fault trigger reverts — Fault resets when fault trigger reverts

No (default) | Yes

Select Yes to have the fault reset when the trigger signal reverts. The default value is No.

#### Dependencies

To enable this parameter, set the **Enable external fault trigger** parameter to Yes.

## Temporal Trigger

### Enable temporal fault trigger — Temporal fault model

No (default) | Yes

Temporal fault model. Choose between

- No — Do not model a temporal fault.
- Yes — Model a temporal fault. Related parameters are visible only if you select this option.

### Fault start time — Fault start time

1 s (default)

Simulation time that the block enters the fault state.

#### Dependencies

To enable this parameter, set the **Enable temporal fault trigger** parameter to Yes.

### Fault duration — Fault duration

0.1 s (default)

Simulation duration time for the fault. The block exits the fault state when the duration time is over.

#### Dependencies

To enable this parameter, set the **Enable temporal fault trigger** parameter to Yes.

#### Parasitics

### Unfaulted phase-neutral conductance — Unfaulted phase-neutral conductance

1e-6 1/Ohm (default)

Conductance between the phase connections and the neutral point when a phase is not involved in the fault.

## Version History

Introduced in R2019b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

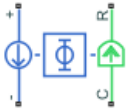
## See Also

Fault

# FEM-Parameterized Linear Actuator

Linear actuator defined in terms of magnetic flux

**Library:** Simscape / Electrical / Electromechanical / Mechatronics Actuators



## Description

The FEM-Parameterized Linear Actuator block implements a model of a linear actuator defined in terms of magnetic flux. Use this block to model custom solenoids and linear motors where magnetic flux depends on both distance and current. You parameterize the block using data from a third-party magnetic finite-element method (FEM) package.

---

**Note** To use this block to represent a solenoid with air gap  $x$  between case **C** and plunger **R**, then the tabulated force should be negative to represent the pulling force bringing **R** towards **C**.

---

The block has two options for the electrical equation. The first, **Define in terms of**  $d\Phi(i, x)/dx$  and  $d\Phi(i, x)/di$ , defines the current in terms of partial derivatives of the magnetic flux ( $\Phi$ ) with respect to distance ( $x$ ) and current ( $i$ ), the equations for which are:

$$\frac{di}{dt} = \left( v - iR - \frac{\partial \Phi}{\partial x} \frac{dx}{dt} \right) / \frac{\partial \Phi}{\partial i}$$

The second option, **Define in terms of**  $\Phi(i, x)$ , defines the voltage across the component directly in terms of the flux, the equation for which is:

$$v = iR + \frac{d}{dt} \Phi(x, i)$$

Numerically, defining the electrical equation in terms of flux partial derivatives is better because the back-emf is piecewise continuous. If using the flux directly, using a finer grid size for current and position will improve results, as will selecting cubic or spline interpolation.

In both cases, you have an option to either directly specify the force as a function of current and position, by using the **Force matrix,  $F(i, x)$**  parameter, or have the block automatically calculate the force matrix.

If entering the electromagnetic force data directly, you can either use data supplied by the finite element magnetic package (which you used to determine the flux) or calculate the force from the flux with following equation:

$$F = \int_0^i \frac{\partial \Phi(x, i)}{\partial x} di$$

For an example that shows how to implement this type of integration in MATLAB, see “Solenoid Parameterized with FEM Data”. The related file `ee_solenoid_fem_params.m` contains the code that calculates and plots the flux data.

Alternatively, the block can automatically calculate the force matrix from the flux information that you provide. To select this option, set the **Calculate force matrix?** parameter to **Yes**. The force matrix calculation occurs at model initialization based on current block flux linkage information. The force is calculated by numerically integrating the rate of change of flux linkage with respect to position over current, according to the preceding equation. If the **Electrical model** parameter is set to **Define in terms of  $\Phi(i,x)$** , then the block must first estimate the **Flux partial derivative wrt displacement,  $d\Phi(i,x)/dx$**  parameter value from the flux linkage data. When doing this, the block uses the interpolation method specified by the **Interpolation method** parameter. Typically, the **Smooth** option is most accurate, but the **Linear** option is most robust.

You can define  $\Phi$  and its partial derivatives for just positive, or positive and negative currents. If defining for just positive currents, then the block assumes that  $\Phi(-i,x) = -\Phi(i,x)$ . Therefore, if the current vector is positive only:

- The first current value must be zero.
- The flux corresponding to zero current must be zero.
- The partial derivative of flux with respect to displacement must be zero for zero current.

To model a linear motor with a repeated flux pattern, set the **Flux dependence on displacement** parameter to **Cyclic**. When selecting this option, the force and flux (or force and flux partial derivatives depending on the option chosen) must have identical first and last columns.

---

**Note** The actuated motion direction of this block can be swapped by flipping the block and swapping the R and C connections.

---

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Assumptions and Limitations

- You must supply a consistent set of force and flux data. There is no check that ensures that the force matrix is consistent with the flux data.
- When driving the FEM-Parameterized Linear Actuator block via a series inductor, you may need to include a parallel conductance in the inductor component.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the actuator positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the actuator negative terminal.

#### C – Case

mechanical

Mechanical translational conserving port associated with the actuator case.

#### R – Plunger

mechanical

Mechanical translational conserving port associated with the plunger.

#### H – Thermal port

thermal

Thermal port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

### Magnetic Force

#### Electrical model – Electrical model parameterization

Define in terms of  $d\Phi(i,x)/dx$  and  $d\Phi(i,x)/di$  (default) | Define in terms of  $\Phi(i,x)$

Select one of the following parameterization options, based on the underlying electrical model:

- Define in terms of  $d\Phi(i,x)/dx$  and  $d\Phi(i,x)/di$  – Define the current through the block in terms of partial derivatives of the magnetic flux with respect to distance and current.
- Define in terms of  $\Phi(i,x)$  – Define the voltage across the block terminals directly in terms of the flux.

**Current vector, i – Current vector**

[0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1] A (default)

Specify a vector of monotonically increasing current values corresponding to your force-flux data. If you specify positive currents only, the first element must be zero.

**Displacement vector, x – Displacement vector**

[0, .05, .1, .15, .2] mm (default)

Specify a vector of monotonically increasing displacement values corresponding to your force-flux data.

**Flux partial derivative wrt current,  $d\Phi(i,x)/di$  – Flux partial derivative with respect to current**

[.104, .098, .091, .085, .078; .095, .089, .084, .079, .073; .085, .081, .077, .073, .069; .076, .073, .07, .067, .064; .067, .065, .063, .061, .06; .057, .057, .056, .056, .055; .048, .049, .049, .05, .05; .038, .04, .042, .044, .046; .029, .032, .035, .038, .041; .02, .024, .028, .033, .037; .01, .016, .021, .027, .032] Wb/A (default)

Specify a matrix of the flux partial derivatives with respect to current. The default value, in Wb/A, is:

```
[ 0.104 0.098 0.091 0.085 0.078;
0.095 0.089 0.084 0.079 0.073;
0.085 0.081 0.077 0.073 0.069;
0.076 0.073 0.07 0.067 0.064;
0.067 0.065 0.063 0.061 0.06;
0.057 0.057 0.056 0.056 0.055;
0.048 0.049 0.049 0.05 0.05;
0.038 0.04 0.042 0.044 0.046;
0.029 0.032 0.035 0.038 0.041;
0.02 0.024 0.028 0.033 0.037;
0.01 0.016 0.021 0.027 0.032 ]
```

**Dependencies**

This parameter is visible only when you set the **Electrical model** parameter to Define in terms of  $d\Phi(i,x)/dx$  and  $d\Phi(i,x)/di$ .

**Flux partial derivative wrt displacement,  $d\Phi(i,x)/dx$  – Flux partial derivative with respect to displacement**

[0, 0, 0, 0, 0; -11.94, -10.57, -9.19, -7.81, -6.43; -21.17, -19.92, -18.67, -17.42, -16.16; -27.99, -26.87, -25.75, -24.62, -23.5; -32.42, -31.43, -30.43, -29.43, -28.44; -34.46, -33.59, -32.72, -31.85, -30.98; -34.09, -33.35, -32.61, -31.87, -31.12; -31.33, -30.72, -30.1, -29.49, -28.87; -26.17, -25.68, -25.2, -24.71, -24.22; -18.62, -18.26, -17.9, -17.54, -17.18; -8.66, -8.43, -8.2, -7.97, -7.73] Wb/m (default)

Specify a matrix of the flux partial derivatives with respect to displacement. The default value, in Wb/m, is:

```
[ 0 0 0 0 0;
-11.94 -10.57 -9.19 -7.81 -6.43;
-21.17 -19.92 -18.67 -17.42 -16.16;
-27.99 -26.87 -25.75 -24.62 -23.5;
-32.42 -31.43 -30.43 -29.43 -28.44;
```

```
-34.46 -33.59 -32.72 -31.85 -30.98;
-34.09 -33.35 -32.61 -31.87 -31.12;
-31.33 -30.72 -30.1 -29.49 -28.87;
-26.17 -25.68 -25.2 -24.71 -24.22;
-18.62 -18.26 -17.9 -17.54 -17.18;
-8.66 -8.43 -8.2 -7.97 -7.73 ]
```

### Dependencies

This parameter is visible only when you set the **Electrical model** parameter to **Define** in terms of  $d\Phi(i,x)/dx$  and  $d\Phi(i,x)/di$ .

### Flux linkage matrix, $\Phi(i,x)$ — Flux linkage matrix

```
[0, 0, 0, 0,
0; .0085, .0079, .0075, .0071, .0067; .0171, .016, .0151, .0143, .0137; .0254
, .0239, .0226, .0215, .0206; .033, .0312, .0297, .0283, .0271; .0396, .0377,
.036, .0345, .0331; .0452, .0433, .0415, .0399, .0384; .0495, .0478, .0461, .
0446, .0431; .0526, .0512, .0498, .0485, .0472; .0545, .0537, .0528, .0519, .
0508; .0554, .0553, .0551, .0548, .0542] Wb (default)
```

Specify a matrix of the total flux linkage, that is, flux times the number of turns. The default value, in  $Wb$ , is:

```
[ 0 0 0 0 0;
0.0085 0.0079 0.0075 0.0071 0.0067;
0.0171 0.016 0.0151 0.0143 0.0137;
0.0254 0.0239 0.0226 0.0215 0.0206;
0.033 0.0312 0.0297 0.0283 0.0271;
0.0396 0.0377 0.036 0.0345 0.0331;
0.0452 0.0433 0.0415 0.0399 0.0384;
0.0495 0.0478 0.0461 0.0446 0.0431;
0.0526 0.0512 0.0498 0.0485 0.0472;
0.0545 0.0537 0.0528 0.0519 0.0508;
0.0554 0.0553 0.0551 0.0548 0.0542 ]
```

### Dependencies

This parameter is visible only when you set the **Electrical model** parameter to **Define** in terms of  $\Phi(i,x)$ .

### Calculate force matrix? — Electromagnetic force data specification

No (default) | Yes

Specify the way of providing the electromagnetic force data:

- **No** — Enter the electromagnetic force data directly, by using the **Force matrix,  $F(i,x)$**  parameter. This is the default option.
- **Yes** — The block calculates the force from the flux linkage information, as a function of current and displacement.

### Force matrix, $F(i,x)$ — Force matrix

```
[0, 0, 0, 0, 0; -.6, -.5, -.4, -.3, -.3; -2.3, -2, -1.7, -1.4, -1.2; -4.9,
-4.3, -3.7, -3.2, -2.7; -8.3, -7.3, -6.4, -5.5, -4.7; -12.2, -10.7, -9.4,
-8.2, -7.2; -16.2, -14.4, -12.7, -11.3, -10; -20, -17.9, -15.9, -14.3, -12.9;
-23.3, -20.9, -18.8, -17.1, -15.7; -25.7, -23.1, -21.1, -19.4, -18.2; -26.5,
-24.1, -22.2, -20.9, -20.1] N (default)
```

Specify a matrix of the electromagnetic force applied to the plunger or moving part. The default value, in N, is:

```
[ 0 0 0 0 0;
-0.6 -0.5 -0.4 -0.3 -0.3;
-2.3 -2 -1.7 -1.4 -1.2;
-4.9 -4.3 -3.7 -3.2 -2.7;
-8.3 -7.3 -6.4 -5.5 -4.7;
-12.2 -10.7 -9.4 -8.2 -7.2;
-16.2 -14.4 -12.7 -11.3 -10;
-20 -17.9 -15.9 -14.3 -12.9;
-23.3 -20.9 -18.8 -17.1 -15.7;
-25.7 -23.1 -21.1 -19.4 -18.2;
-26.5 -24.1 -22.2 -20.9 -20.1 ]
```

### Dependencies

This parameter is visible only when you set the **Calculate force matrix?** parameter to **No**.

### Flux dependence on displacement — Flux pattern

Unique (default) | Cyclic

Specify the flux pattern:

- **Unique** — No flux pattern present.
- **Cyclic** — Select this option to model a linear motor with a repeated flux pattern. The force and flux (or force and flux partial derivatives, depending on the **Electrical model** option chosen) must have identical first and last columns.

### Interpolation method — Interpolation method

Linear (default) | Smooth

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- **Linear** — Select this option to get the best performance.
- **Smooth** — Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (2D) block reference page.

### Extrapolation method — Extrapolation method

Linear (default) | Nearest | Error

Select one of the following extrapolation methods for determining the output value when the input value is outside the range specified in the argument list:

- **Linear** — Select this option to produce a surface with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- **Nearest** — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- **Error** — Issues an error if the input signal is outside the range of the table. Select this option to avoid getting into the extrapolation mode when you want your data to be within the table range.



For more information on extrapolation algorithms, see the PS Lookup Table (2D) block reference page.

### **Winding resistance — Electrical winding total resistance**

14 Ohm (default)

Total resistance of the electrical winding.

### **Magnetizing resistance — Magnetizing resistance for iron losses**

inf Ohm (default)

Magnetizing resistance. The value must be greater than zero. The default value is inf, which implies that there are no iron losses.

## **Mechanical**

### **Damping — Damping**

1 N/(m/s) (default)

Linear damping. The value can be zero.

### **Plunger mass — Mass of the moving part**

0.05 kg (default)

Mass of the moving part, which corresponds to mechanical translational port **R**. The value can be zero.

### **Minimum stroke — Minimum stroke**

0 mm (default)

The stroke at which the lower mechanical end stop is applied. The value can be -Inf.

### **Maximum stroke — Maximum stroke**

0.2 mm (default)

The stroke at which the upper mechanical end stop is applied. The value can be Inf.

### **Contact stiffness — Contact stiffness**

1e8 N/m (default)

Contact stiffness between plunger and end stops.

### **Contact damping — Contact damping**

1e4 N/(m/s) (default)

Contact damping between plunger and end stops.

## **Temperature Dependence**

This tab appears only for blocks with exposed thermal port.

### **Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Resistance temperature coefficient.

**Measurement temperature – Measurement temperature**

25 degC (default)

Temperature for which the actuator parameters are defined.

**Thermal Port**

This tab appears only for blocks with exposed thermal port.

**Thermal mass – Thermal mass**

100 J/K (default)

Thermal mass is the energy required to raise the temperature by one degree.

## **Version History**

**Introduced in R2010a**

## **Extended Capabilities**

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

FEM-Parameterized Rotary Actuator | Solenoid

## FEM-Parameterized PMSM

Permanent magnet synchronous motor defined in terms of magnetic flux linkage

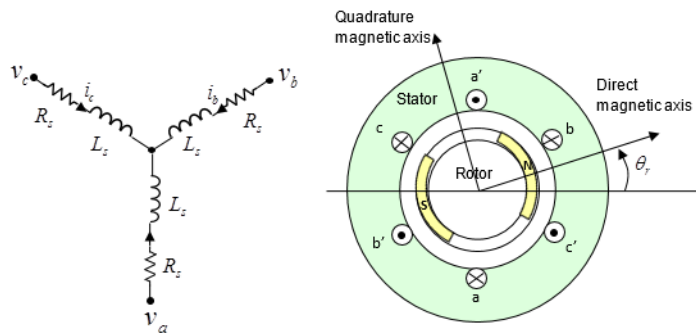
**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



### Description

The FEM-Parameterized PMSM block 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. This is the way third-party magnetic finite-element method (FEM) packages usually export flux information. Because of the tabulated form, the flux can vary in a nonlinear way on both rotor angle and current. You can therefore use this block to model PMSM with trapezoidal back-emf profile, sometimes called brushless DC motor, as well as regular PMSM.

The figure shows the equivalent circuit for a wye-connected PMSM. The rotor angle is zero when the permanent magnet flux aligns with the A-phase magnetic axis.



In practice, the flux linking each of the three windings depends on all three currents and rotor angle. Tabulating flux as a function of four independent variables might lead to simulation inefficiency and significant memory requirements to manage the data. The block, therefore, lets you select between the following parameterization methods for flux and torque:

- **2-D partial derivative data** — 2-D table lookup, with options to tabulate in terms of current and rotor angle, or in terms of  $d$ -axis and  $q$ -axis currents. The first option assumes constant mutual inductance and supports nonsinusoidal back emf profiles. The second option assumes a sinusoidal back emf and captures saturation effects for interior PMSMs (IPMSMs).
- **3-D partial derivative data** — 3-D table lookup, based on direct current, quadrature current, and rotor angle. You provide the flux lookup data for the  $a$  phase. The block uses Park transform 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 partial derivative data** — 4-D table lookup, based on the three stator winding currents and the rotor angle. You provide the flux lookup data for the *a* phase. This model has the best fidelity of the three, but also is the most costly in terms of simulation performance and memory requirements.
- **3-D flux linkage data** — 3-D table lookup, based on the flux linkage data. You can provide the flux linkage data in a variety of formats. The block uses Park transform 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.

By default, the blocks implements a wye-wound configuration for the stator windings for all these parameterizations. However, it is possible to switch to a delta-wound configuration, selectable using the **Winding type** parameter. When in the delta-wound configuration, the *a* phase is connected between ports **a** and **b**, the *b* phase between ports **b** and **c** and the *c* phase between ports **c** and **a**.

To access these parameterization methods, double-click the block in your model and set the **Modeling option** parameter to the desired option, with or without thermal ports. By default, the thermal ports are not exposed. For more information, see “Model Thermal Effects” on page 1-567.

## 2-D Data Model with Constant Mutual Inductance

In this 2-D flux data model, the flux linking each winding is assumed to depend nonlinearly only on the current in that same winding, plus the rotor angle. In practice, this is a reasonable assumption for many permanent magnet synchronous motors; however, it is less accurate for switched reluctance motors. Given this assumption, the fluxes in the three windings are:

$$\begin{bmatrix} \phi_a \\ \phi_b \\ \phi_c \end{bmatrix} = \begin{bmatrix} 0 & -M_s & -M_s \\ -M_s & 0 & -M_s \\ -M_s & -M_s & 0 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \phi(i_a, \theta_r) \\ \phi(i_b, \theta_r - 2\pi/(3N)) \\ \phi(i_c, \theta_r - 4\pi/(3N)) \end{bmatrix}$$

where  $\phi(\theta_r, i_a)$  is the flux linkage for the A-phase winding as a function of rotor angle and A-phase current.  $\theta_r = 0$  corresponds to the rotor *d*-axis aligning with the A-phase positive magnetic flux direction.  $M_s$  is the stator-stator mutual inductance.

For improved numerical performance, the equations implemented in the block actually work with the partial derivatives of flux linkage with respect to current,  $\partial\phi(i, \theta_r)/\partial i$ , and rotor angle,  $\partial\phi(i, \theta_r)/\partial\theta_r$ , rather than the flux directly. If your FEM package does not export these partial derivatives, you can determine them using a MATLAB script. See the Solenoid Parameterized with FEM Data example model and its supporting MATLAB script for an example of how to do this.

The electrical equations for the block, defined in terms of flux partial derivatives, are:

$$\begin{aligned} v_a &= \frac{\partial\phi}{\partial i_a} \frac{di_a}{dt} + \frac{\partial\phi}{\partial\theta_r} \frac{d\theta_r}{dt} - M_s \left( \frac{di_b}{dt} + \frac{di_c}{dt} \right) + R_s i_a \\ v_b &= \frac{\partial\phi}{\partial i_b} \frac{di_b}{dt} + \frac{\partial\phi}{\partial\theta_r} \frac{d\theta_r}{dt} - M_s \left( \frac{di_a}{dt} + \frac{di_c}{dt} \right) + R_s i_b \\ v_c &= \frac{\partial\phi}{\partial i_c} \frac{di_c}{dt} + \frac{\partial\phi}{\partial\theta_r} \frac{d\theta_r}{dt} - M_s \left( \frac{di_a}{dt} + \frac{di_b}{dt} \right) + R_s i_c \end{aligned}$$

where

- $v_a, v_b, v_c$  are the voltages applied to the A, B, and C stator windings.

- $i_a, i_b, i_c$  are the stator currents in each of the three windings.
- $R_s$  is the resistance of each of the stator windings.
- $M_s$  is the stator-stator mutual inductance.
- $\partial\phi/\partial i_a, \partial\phi/\partial i_b, \partial\phi/\partial i_c$  are the partial derivatives of flux linkage with respect to stator current in each of the three windings.
- $\partial\phi/\partial\theta_r$  is the partial derivative of flux linkage with respect to rotor angle.

The block can automatically calculate the torque matrix from the flux information that you provide. Alternatively, you can set the **Calculate torque matrix?** parameter to **No** and directly specify the torque as a function of current and rotor angle. See the FEM-Parameterized Rotary Actuator block reference page for more information.

### 2-D Data Model with Sinusoidal Back EMF

In this 2-D flux data model, the flux linking each winding is assumed to depend nonlinearly on all stator winding currents, plus it is assumed that the permanent magnet flux linkage is sinusoidal. Interior magnet PMSMs (or IPMSMs) usually fit this assumption well. The equations are:

$$\begin{bmatrix} \phi_d \\ \phi_q \end{bmatrix} = \begin{bmatrix} L_d(i_d, i_q) & \\ & L_q(i_d, i_q) \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \phi_m(i_d, i_q) \\ 0 \end{bmatrix}$$

$$T = \frac{3}{2}N(i_q(i_d L_d(i_d, i_q) + \phi_m(i_d, i_q)) - i_d i_q L_q(i_d, i_q))$$

where

- $i_d$  and  $i_q$  are the  $d$ -axis and  $q$ -axis currents, respectively.
- $\phi_d$  and  $\phi_q$  are the  $d$ -axis and  $q$ -axis flux linkages, respectively.
- $\phi_m$  is the permanent magnet flux linkage.
- $L_d$  and  $L_q$  are the  $d$ -axis and  $q$ -axis inductances, respectively. They are assumed to depend on the  $d$ -axis and  $q$ -axis currents.
- $N$  is the number of pole pairs.
- $T$  is the electrical torque.

### 3-D Partial Derivative Data Model Using Park Transform

Working with four-dimensional data has both a simulation performance cost and a memory cost. To reduce the table dimension to three-dimensional, the 3-D data model uses Park transform to map the three currents to direct and quadrature currents:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(N\theta_r) & \cos\left(N\theta_r - \frac{2\pi}{3}\right) & \cos\left(N\theta_r + \frac{2\pi}{3}\right) \\ -\sin(N\theta_r) & -\sin\left(N\theta_r - \frac{2\pi}{3}\right) & -\sin\left(N\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

In the general case, Park transform maps to direct, quadrature, and zero-sequence currents. However, the zero-sequence current is typically small under normal operating conditions. Therefore, the model neglects the dependence of the flux linkage terms on zero-sequence current, and determines the flux linkage in terms of just direct and quadrature currents plus rotor angle. The flux equation for the 3-D data model is:

$$\begin{bmatrix} \phi_a \\ \phi_b \\ \phi_c \end{bmatrix} = \begin{bmatrix} \phi(i_d, i_q, \theta_r) \\ \phi(i_d, i_q, \theta_r - 2\pi/(3N)) \\ \phi(i_d, i_q, \theta_r - 4\pi/(3N)) \end{bmatrix}$$

The electrical equations for the block are also defined in terms of flux partial derivatives, similar to the 4-D data model. You can calculate 3-D flux linkage partial derivative data from 4-D flux linkage data using `ee_calculateFluxPartialDerivatives`.

#### 4-D Partial Derivative Data Model

The flux linking each of the windings is a function of the current in that winding, the currents in the other two windings, and the rotor angle. For full accuracy, the 4-D flux data model assumes that the flux linkage is a function of the three currents and the rotor angle, therefore performing four-dimensional table lookups. The flux equation is:

$$\begin{bmatrix} \phi_a \\ \phi_b \\ \phi_c \end{bmatrix} = \begin{bmatrix} \phi(i_a, i_b, i_c, \theta_r) \\ \phi(i_b, i_c, i_a, \theta_r - 2\pi/(3N)) \\ \phi(i_c, i_a, i_b, \theta_r - 4\pi/(3N)) \end{bmatrix}$$

where

- $\phi_a, \phi_b, \phi_c$  are the flux linkages for the A, B, and C stator windings.
- $i_a, i_b, i_c$  are the stator currents in each of the three windings.
- $\theta_r$  is the rotor angle.  $\theta_r = 0$  corresponds to the case where the permanent magnet flux is aligned with the A-phase stator winding flux.
- $N$  is the number of pole pairs.

Flux linkage data is assumed cyclic with  $\theta_r$ . If, for example, the motor has six pole pairs, then the range for the data is  $0 \leq \theta_r \leq 60^\circ$ . You must provide data both at 0 and 60 degrees, and because the data is cyclic, the flux linkage partial derivatives must be the same at these two end points.

The torque equation is:

$$\tau = T(i_a, i_b, i_c, \theta_r)$$

The 4-D data model does not have an option for the block to determine torque from flux linkage. Because of the increased numerical overhead in the 4-D case, it is better to precalculate the torque just once, rather than calculate it every time you run the simulation.

For improved numerical performance, the equations implemented in the block actually work with the partial derivatives of flux linkage with respect to the three currents and the rotor angle, rather than the flux directly. If your FEM package does not export these partial derivatives, you can determine them using `ee_calculateFluxPartialDerivatives`.

The electrical equations for the block, defined in terms of flux partial derivatives, are:

$$\begin{aligned} v_a &= \frac{\partial \phi_a}{\partial i_a} \frac{di_a}{dt} + \frac{\partial \phi_a}{\partial i_b} \frac{di_b}{dt} + \frac{\partial \phi_a}{\partial i_c} \frac{di_c}{dt} + \frac{\partial \phi_a}{\partial \theta_r} \frac{d\theta_r}{dt} + R_s i_a \\ v_b &= \frac{\partial \phi_b}{\partial i_a} \frac{di_a}{dt} + \frac{\partial \phi_b}{\partial i_b} \frac{di_b}{dt} + \frac{\partial \phi_b}{\partial i_c} \frac{di_c}{dt} + \frac{\partial \phi_b}{\partial \theta_r} \frac{d\theta_r}{dt} + R_s i_b \\ v_c &= \frac{\partial \phi_c}{\partial i_a} \frac{di_a}{dt} + \frac{\partial \phi_c}{\partial i_b} \frac{di_b}{dt} + \frac{\partial \phi_c}{\partial i_c} \frac{di_c}{dt} + \frac{\partial \phi_c}{\partial \theta_r} \frac{d\theta_r}{dt} + R_s i_c \end{aligned}$$

where

- $v_a, v_b, v_c$  are the voltages applied to the A, B, and C stator windings.
- $i_a, i_b, i_c$  are the stator currents in each of the three windings.
- $R_s$  is the resistance of each of the stator windings.

### 3-D Flux Linkage Data Models

The 3-D flux linkage data options let you work with raw flux linkage data exported from your finite-element (FE) motor design tool. This is in contrast to the 3-D partial derivative data options, for which you need to determine the partial derivatives. You can provide flux linkage data in a variety of formats, to support different FE tool conventions:

- Tabulate DQ-axes flux linkage data or A-phase flux linkage data — Some tools support working with flux linkage resolved into direct (D) and quadrature (Q) axes. An advantage of this approach is that data for rotor angles in the range 0 to  $360/N/3$  degrees is required (where  $N$  is the number of pole pairs). Other tools work directly with A-, B-, and C-phase flux linkages, and for this you can import just the A-phase flux linkage, for which the rotor angle range must be in the range 0 to  $360/N$  degrees. The implicit assumption of importing just the A-phase data is that the B and C phase data is the same except shifted in phase.
- Tabulate using cartesian or polar current coordinates — Cartesian tabulation implies the flux linkage is tabulated in terms of D-axis current and Q-axis current (plus rotor angle). Alternatively, polar tabulation involves tabulating flux linkages in terms of current magnitude, current advance angle relative to the Q-axis, and rotor angle. The advantage of polar coordinates is that it more naturally reflects the permitted operating currents, thereby avoiding unused table data points.

These conventions result in four **Flux linkage data format** parameterization options:

- D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ )
- D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ )
- A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ )
- A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ )

Besides selecting the flux linkage data format used by your FE tool, you have to select the version of Park transform used by the tool. The four conventions are described below and correspond to the four options for the **Park's convention for tabulated data** drop-down menu.

---

**Note** When looking at logged values for D- and Q-axis currents, keep in mind that for each of these options, the format is converted, as needed, so that internally the FEM-Parameterized PMSM block consistently uses Option 1.

---



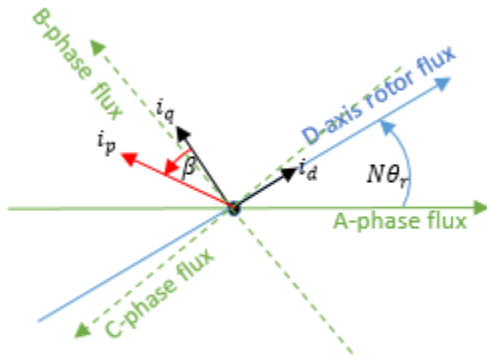
---

**Note** To preserve higher-order harmonics due to the permanent magnet back EMF, provide the A-phase flux data rather than the D-axis and Q-axis flux data. Conversely, if you provide D-axis and Q-axis fluxes, the Park's transform removes the higher-order harmonics. Working with D and Q fluxes is sufficient for many applications, particularly if you set the **Winding type** parameter to Wye-wound and there aren't connections to the neutral port of this block. For this particular case, the constraint that the sum of the currents is zero at the neutral port removes all third-order harmonics.

---

**Option 1. Q leads D, rotor angle measured from A-phase to D-axis**

This is the Park's convention used internally by Simscape Electrical motor and machine blocks. All other options are converted into this format.



- $N$ : number of pole pairs
- $\theta_r$ : rotor angle
- $i_d, i_q$ : D-axis and Q-axis currents
- $i_p$ : Current magnitude =  $\sqrt{i_d^2 + i_q^2}$
- $\beta$ : Current advance angle =  $\tan^{-1}(-i_d/i_q)$

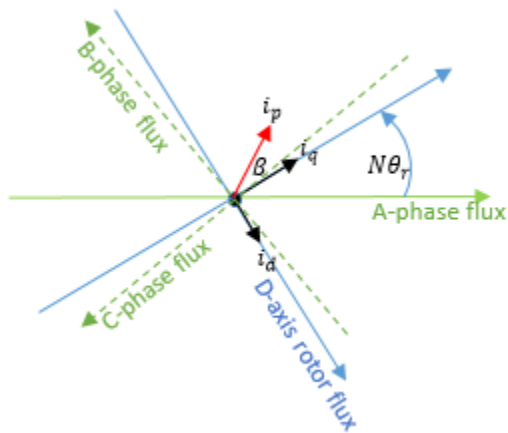
Corresponding Park transform is

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(N\theta_r) & \cos\left(N\theta_r - \frac{2\pi}{3}\right) & \cos\left(N\theta_r + \frac{2\pi}{3}\right) \\ -\sin(N\theta_r) & -\sin\left(N\theta_r - \frac{2\pi}{3}\right) & -\sin\left(N\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

where  $i_a, i_b,$  and  $i_c$  are the A-phase, B-phase, and C-phase currents, respectively.



**Option 2. Q leads D, rotor angle measured from A-phase to Q-axis**



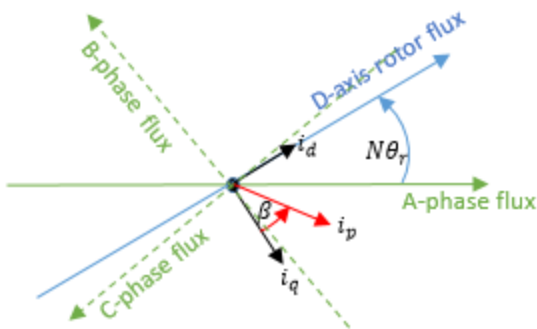
- $N$ : number of pole pairs
- $\theta_r$ : rotor angle
- $i_d, i_q$ : D-axis and Q-axis currents
- $i_p$ : Current magnitude =  $\sqrt{i_d^2 + i_q^2}$
- $\beta$ : Current advance angle =  $\tan^{-1}(-i_d/i_q)$

Corresponding Park transform is

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin(N\theta_r) & \sin\left(N\theta_r - \frac{2\pi}{3}\right) & \sin\left(N\theta_r + \frac{2\pi}{3}\right) \\ \cos(N\theta_r) & \cos\left(N\theta_r - \frac{2\pi}{3}\right) & \cos\left(N\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

where  $i_a$ ,  $i_b$ , and  $i_c$  are the A-phase, B-phase, and C-phase currents, respectively.

**Option 3. D leads Q, rotor angle measured from A-phase to D-axis**



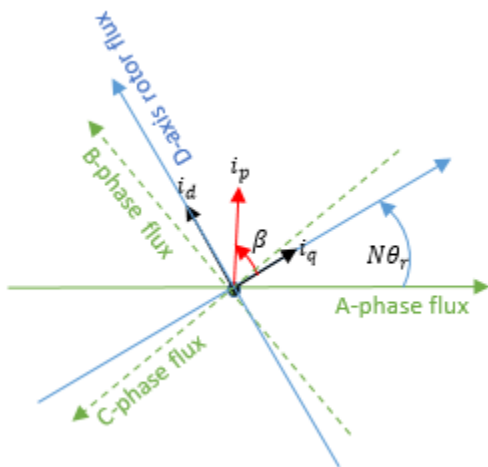
- $N$ : number of pole pairs
- $\theta_r$ : rotor angle
- $i_d, i_q$ : D-axis and Q-axis currents
- $i_p$ : Current magnitude =  $\sqrt{i_d^2 + i_q^2}$
- $\beta$ : Current advance angle =  $\tan^{-1}(-i_d/i_q)$

Corresponding Park transform is

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(N\theta_r) & \cos\left(N\theta_r - \frac{2\pi}{3}\right) & \cos\left(N\theta_r + \frac{2\pi}{3}\right) \\ \sin(N\theta_r) & \sin\left(N\theta_r - \frac{2\pi}{3}\right) & \sin\left(N\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

where  $i_a, i_b,$  and  $i_c$  are the A-phase, B-phase, and C-phase currents, respectively.

**Option 4. D leads Q, rotor angle measured from A-phase to Q-axis**



- $N$ : number of pole pairs
- $\theta_r$ : rotor angle
- $i_d, i_q$ : D-axis and Q-axis currents
- $i_p$ : Current magnitude =  $\sqrt{i_d^2 + i_q^2}$
- $\beta$ : Current advance angle =  $\tan^{-1}(-i_d/i_q)$

Corresponding Park transform is

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} -\sin(N\theta_r) & -\sin\left(N\theta_r - \frac{2\pi}{3}\right) & -\sin\left(N\theta_r + \frac{2\pi}{3}\right) \\ \cos(N\theta_r) & \cos\left(N\theta_r - \frac{2\pi}{3}\right) & \cos\left(N\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

where  $i_a$ ,  $i_b$ , and  $i_c$  are the A-phase, B-phase, and C-phase currents, respectively.

### Calculating Iron Losses

The FEM-Parameterized PMSM block models iron losses according to the parameterization methods you choose for flux and torque.

#### Specify open-circuit and short-circuit loss data (permanent magnet motors only)

For the **2-D partial derivative data**, **3-D partial derivative data**, and **4-D partial derivative data** options, with or without thermal ports, the iron loss model is based on the work of Mellor [1]. 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.

The term representing the main magnetizing path depends on the induced RMS stator voltage,  $V_{m_{rms}}$ :

$$P_{OC}(V_{m_{rms}}) = \frac{a_h}{k} V_{m_{rms}} + \frac{a_j}{k^2} V_{m_{rms}}^2 + \frac{a_{ex}}{k^{1.5}} V_{m_{rms}}^{1.5}$$

This is the dominant term during no-load operation.  $k$  is the back emf constant relating RMS volts per Hz. It is defined as  $k = V_{m_{rms}}/f$ , where  $f$  is the electrical frequency. The first term on the right-hand side is the magnetic hysteresis loss, the second is the eddy current loss and the third is the excess loss. The three coefficients appearing on the numerators are derived from the values that you provide for the open-circuit hysteresis, eddy, and excess losses.

The term representing the cross-tooth tip path becomes important when a demagnetizing field is set up and can be determined from a finite element analysis short-circuit test. It depends on the RMS emf associated with the cross-tooth tip flux,  $V_{d_{rms}}^*$ :

$$P_{SC}(V_{d_{rms}}^*) = \frac{b_h}{k} V_{d_{rms}}^* + \frac{b_j}{k^2} V_{d_{rms}}^{*2} + \frac{b_{ex}}{k^{1.5}} V_{d_{rms}}^{*1.5}$$

The three numerator terms are derived from the values you provide for the short-circuit hysteresis, eddy, and excess losses.

#### Steinmetz parameterization method

For the **3-D flux linkage data** with or without thermal ports, you can also model the iron losses based on the Steinmetz equation. The Steinmetz method scales for different motor speeds or electrical frequencies so that iron loss data is only required as function of motor currents.

If, in the **Iron Losses** setting, you set the **Model** parameter to **Specify open and short circuit loss data (permanent magnet motors only)**, the block uses the Steinmetz method, but assumes constant coefficients and no dependency on peak current and current phase advance. Select this option if you have dyno measurements or if you perform an FE analysis to get the iron losses from just open-circuit and short-circuit simulations.

Conversely, if you set the **Flux linkage data format** parameter to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ), then the block tabulated the coefficients with the **Peak current magnitude vector, I** and **Current advance angle vector, B** parameters, so that the iron losses are given by:

$$P_{rotor}(f) = k_{hr}(I_p, \beta)f + k_{Jr}(I_p, \beta)f^2 + k_{er}(I_p, \beta)f^{1.5}$$

$$P_{stator}(f) = k_{hs}(I_p, \beta)f + k_{Js}(I_p, \beta)f^2 + k_{es}(I_p, \beta)f^{1.5}$$

where:

- $f$  is the electrical frequency, in Hz.
- $k_{hr}(I_p, \beta)$  is the **Rotor hysteresis loss coefficient,  $k_{hr}(I, B)$ .**
- $k_{Jr}(I_p, \beta)$  is the **Rotor eddy current loss coefficient,  $k_{Jr}(I, B)$ .**
- $k_{er}(I_p, \beta)$  is the **Rotor excess current loss coefficient,  $k_{er}(I, B)$ .**
- $k_{hs}(I_p, \beta)$  is the **Stator hysteresis loss coefficient,  $k_{hs}(I, B)$ .**
- $k_{Js}(I_p, \beta)$  is the **Stator eddy current loss coefficient,  $k_{Js}(I, B)$ .**
- $k_{es}(I_p, \beta)$  is the **Stator excess current loss coefficient,  $k_{es}(I, B)$ .**

Similarly, if you set the **Flux linkage data format** parameter to either D and Q axes flux linkages as a function of D-axis current ( $i_d$ ), Q-axis current ( $i_q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_d$ ), Q-axis current ( $i_q$ ), and rotor angle ( $\theta$ ), then the iron losses are given by:

$$P_{loss, rotor}(f) = k_{hr}(i_d, i_q)f + k_{Jr}(i_d, i_q)f^2 + k_{er}(i_d, i_q)f^{1.5}$$

$$P_{loss, stator}(f) = k_{hs}(i_d, i_q)f + k_{Js}(i_d, i_q)f^2 + k_{es}(i_d, i_q)f^{1.5}$$

where:

- $k_{hr}(i_d, i_q)$  is the **Rotor hysteresis loss coefficient,  $k_{hr}(id, iq)$ .**
- $k_{Jr}(i_d, i_q)$  is the **Rotor eddy current loss coefficient,  $k_{Jr}(id, iq)$ .**
- $k_{er}(i_d, i_q)$  is the **Rotor excess current loss coefficient,  $k_{er}(id, iq)$ .**
- $k_{hs}(i_d, i_q)$  is the **Stator hysteresis loss coefficient,  $k_{hs}(id, iq)$ .**
- $k_{Js}(i_d, i_q)$  is the **Stator eddy current loss coefficient,  $k_{Js}(id, iq)$ .**
- $k_{es}(i_d, i_q)$  is the **Stator excess current loss coefficient,  $k_{es}(id, iq)$ .**

Select this option only in conjunction with a motor design tool that can calculate the coefficients. For more information, see the “Import IPMSM Flux Linkage Data from Motor-CAD” example.

#### **Tabulate with current and speed**

For the **3-D flux linkage data** with or without thermal ports, you can also model the iron losses by tabulating them independently for rotor and stator.

If you set the **Flux linkage data format** parameter to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ), the block tabulated the iron losses with the **Peak current magnitude vector,  $I$**  and **Current advance angle vector,  $B$**  parameters, so that the iron losses are given by:

$$P_{loss}(I_p, \beta, w) = \text{tablelookup}(P_{table}, I_p, \beta, w)$$

where  $w$  is the **Rotor speed vector,  $w$**  parameter.

Similarly, if you set the **Flux linkage data format** parameter to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ), then the iron losses are given by:

$$P_{\text{loss}}(i_d, i_q, w) = \text{tablelookup}(P_{\text{table}}, i_d, i_q, w).$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- 2-D partial derivative data | No thermal port — 2-D table lookup parameterization, with options to tabulate in terms of current and rotor angle, but without thermal ports.
- 2-D partial derivative data | Show thermal port — 2-D table lookup parameterization, with options to tabulate in terms of current and rotor angle, and with thermal ports.
- 3-D partial derivative data | No thermal port — 3-D table lookup parameterization, based on direct current, quadrature current, and rotor angle, but without thermal ports.
- 3-D partial derivative data | Show thermal port — 3-D table lookup parameterization, based on direct current, quadrature current, and rotor angle, and with thermal ports.
- 4-D partial derivative data | No thermal port — 4-D table lookup parameterization, based on the three stator winding currents and the rotor angle, but without thermal ports.
- 4-D partial derivative data | Show thermal port — 4-D table lookup parameterization, based on the three stator winding currents and the rotor angle, and with thermal ports.
- 3-D flux linkage data | No thermal port — 3-D table lookup parameterization, based on the flux linkage data without thermal ports.
- 3-D flux linkage data | Show thermal port — 3-D table lookup parameterization, based on the flux linkage data with thermal ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Assumptions and Limitations

This block has the following limitations:

- For the 2-D data model, the stator-stator mutual inductance, defined by the **Stator mutual inductance,  $M_s$**  parameter value, is constant during simulation and does not vary with rotor angle. This means that the block is suitable for modeling most PMSM and brushless DC motors, but not switched reluctance motors.
- The 3-D and 4-D data models assume symmetry, so that the flux linkage dependency on currents and rotor angle for windings B and C can be determined from that for winding A.
- For the 4-D data model, consider memory requirements when fixing the independent parameter values (three currents and rotor angles). The linear interpolation option uses less memory, but the smooth interpolation option is more accurate for a given independent parameter spacing.
- The iron losses model assumes sinusoidal currents.

## Ports

### Conserving

#### **a – A-phase connection**

electrical

Electrical conserving port associated with the A-phase connection.

#### **b – B-phase connection**

electrical

Electrical conserving port associated with the B-phase connection.

#### **c – C-phase connection**

electrical

Electrical conserving port associated with the C-phase connection.

#### **n – Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

### Dependencies

To enable this port, set **Modeling option** parameter to either

- 2-D partial derivative data | No thermal port, 2-D partial derivative data | Show thermal port, 3-D partial derivative data | No thermal port, 3-D partial derivative data | Show thermal port, 4-D partial derivative data | No thermal port, or 4-D partial derivative data | Show thermal port
- 3-D flux linkage data | No thermal port or 3-D flux linkage data | Show thermal port and set the **Expose neutral port** parameter to Yes.

#### **C – Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

#### **R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the motor rotor.

#### **HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### **HB – Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### HC — Winding C thermal port

thermal

Thermal conserving port associated with winding C.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### HR — Rotor thermal port

thermal

Thermal conserving port associated with the rotor.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

#### Modeling option — Flux and torque parameterization methods

2-D partial derivative data | No thermal port (default) | 2-D partial derivative data | Show thermal port | 3-D partial derivative data | No thermal port | 3-D partial derivative data | Show thermal port | 4-D partial derivative data | No thermal port | 4-D partial derivative data | Show thermal port | 3-D flux linkage data | No thermal port | 3-D flux linkage data | Show thermal port

Parameterization method for the flux and torque. To model the effects of copper resistance and iron losses that convert electrical power to heat, choose the options with thermal ports.

#### Electrical (2-D Partial Derivative Data Parameterization)

This configuration of the **Electrical** parameters corresponds to the 2-D Partial Derivative Data block parameterizations, with or without thermal ports. If you are using the 3-D Partial Derivative Data, 4-D Partial Derivative Data, or 3-D Flux Linkage Data parameterization of the block, see “Electrical (3-D Partial Derivative Data Parameterization)” on page 1-573, “Electrical (4-D Partial Derivative Data Parameterization)” on page 1-575, or “Electrical (3-D Flux Linkage Data Parameterization)” on page 1-577 respectively.

#### Parameterization — Parameterization method

Assume constant mutual inductance - tabulate with phase current and rotor angle (default) | Assume sinusoidal back emf - tabulate with d- and q-axis currents

Select the parameterization method:

- Assume constant mutual inductance - tabulate with phase current and rotor angle — This method assumes that the flux linking each winding depends nonlinearly only on the current in that same winding, plus the rotor angle.

- Assume sinusoidal back emf - tabulate with d- and q-axis currents — This method 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).

### Winding type — Stator windings configuration

Wye-wound (default) | Delta-wound

Select the configuration for the stator windings:

- Wye-wound — The stator windings are wye-wound.
- Delta-wound — The stator windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

### Current vector, **i** — Vector of currents

[-2, 0, 2] A (default)

Vector of currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle.

### Rotor angle vector, **theta** — Vector of rotor angles

[0, 20, 40, 60] deg (default)

Vector of rotor angles corresponding to the provided flux linkage partial derivatives. The vector must start at zero. This value corresponds to the angle where the A-phase magnetic flux aligns with the rotor permanent magnetic peak flux direction (the direct-axis, or *d*-axis). The last value,  $\theta_{\max}$ , must be the rotor angle where the flux linkage pattern peaks again. Therefore, the number of pole pairs is  $360/\theta_{\max}$  if  $\theta_{\max}$  is expressed in degrees. The default value corresponds to a 6 pole-pair motor.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle.

### Flux linkage partial derivative wrt current, $d\Phi(i, \theta)/di$ — Flux linkage partial derivative with respect to current

0.0002\*ones(3,4) Wb/A (default)

Matrix of the flux linkage partial derivatives with respect to current, defined as a function of current vector and rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns. The default value corresponds to the special case where stator inductance does not depend on stator current or on rotor angle.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle.

### Flux linkage partial derivative wrt angle, $d\Phi(i, \theta)/d\theta$ — Flux linkage partial derivative with respect to angle

[0, -0.16, 0.16, 0; 0, -0.16, 0.16, 0; 0, -0.16, 0.16, 0] Wb/rad (default)



Matrix of the flux linkage partial derivatives with respect to rotor angle, defined as a function of current vector and rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns. The default value is  $[0, -0.16, 0.16, 0; 0, -0.16, 0.16, 0; 0, -0.16, 0.16, 0]$  Wb/rad, which corresponds to the special case where stator inductance does not depend on stator current.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle.

#### Direct axis current vector, $i_d$ — Direct axis current vector

$[-200, 0, 200]$  A (default)

Vector of  $d$ -axis currents corresponding to the provided inductances. The current vector must be two-sided (have positive and negative values).

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with  $d$ - and  $q$ -axis currents.

#### Quadrature axis current vector, $i_q$ — Quadrature axis current vector

$[-200, 0, 200]$  A (default)

Vector of  $q$ -axis currents corresponding to the provided inductances. The current vector must be two-sided (have positive and negative values).

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with  $d$ - and  $q$ -axis currents.

#### Ld matrix, $L_d(i_d, i_q)$ — Ld matrix

$0.0002 * \text{ones}(3, 3)$  H (default)

Matrix of the  $d$ -axis inductances with respect to current, defined as a function of  $d$ -axis and  $q$ -axis current vectors.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with  $d$ - and  $q$ -axis currents.

#### Lq matrix, $L_q(i_d, i_q)$ — Lq matrix

$0.0002 * \text{ones}(3, 3)$  H (default)

Matrix of the  $q$ -axis inductances with respect to current, defined as a function of  $d$ -axis and  $q$ -axis current vectors.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with  $d$ - and  $q$ -axis currents.

#### Permanent magnet flux linkage, $PM(i_d, i_q)$ — Permanent magnet flux linkage

$0.1 * \text{ones}(3, 3)$  Wb (default)

Matrix of the permanent magnet flux linkages with respect to current, defined as a function of  $d$ -axis and  $q$ -axis current vectors. Flux linkage is the flux multiplied by the number of winding turns.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with d- and q-axis currents.

#### Number of pole pairs — Number of pole pairs

6 (default)

Number of the permanent magnet motor pole pairs.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with d- and q-axis currents.

#### Calculate torque matrix? — Electromagnetic torque data specification

Yes (default) | No — specify directly

Specify the way of providing the electromagnetic torque data:

- Yes — The block calculates the torque from the flux linkage information, as a function of current and rotor angle.
- No — specify directly — Enter the electromagnetic torque data directly, by using the **Torque matrix,  $T(i, \theta)$**  parameter.

#### Dependencies

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle. If you set the **Parameterization** parameter to Assume sinusoidal back emf - tabulate with d- and q-axis currents, the equation for torque is explicit in terms of the provided matrices.

#### Torque matrix, $T(i, \theta)$ — Torque matrix

zeros(3, 3, 3, 4) N\*m (default)

Specify a matrix of the electromagnetic torque applied to the rotor, as a function of current and rotor angle. This parameter is visible only if **Calculate torque matrix?** is set to No — specify directly.

#### Dependencies

This parameter is visible only when you set the **Calculate torque matrix?** parameter to No — specify directly.

#### Interpolation method — Interpolation method

Linear Smooth (default)

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- Linear — Uses an extension of linear algorithm for multidimensional interpolation. Select this option to get the best performance.

- **Smooth** — Uses a modified Akima interpolation algorithm. Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation methods, see the PS Lookup Table (2D) block reference page.

### **Stator resistance per phase, $R_s$ — Stator resistance per phase**

0.013 Ohm (default)

Resistance of each of the stator windings.

### **Stator mutual inductance, $M_s$ — Stator mutual inductance**

0.00002 H (default)

Stator-stator mutual inductance, which is assumed to be independent of both current and rotor angle.

#### **Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Assume constant mutual inductance - tabulate with phase current and rotor angle.

### **Stator zero-sequence inductance, $L_0$ — Stator zero-sequence inductance**

0.00016 H (default)

Zero-sequence inductance.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Assume sinusoidal back emf - tabulate with d- and q-axis currents.

### **Electrical (3-D Partial Derivative Data Parameterization)**

This configuration of the **Electrical** parameters corresponds to the 3-D Partial Derivative Data block variants, with or without thermal ports. If you are using the 2-D Partial Derivative Data, 4-D Partial Derivative Data, or 3-D Flux Linkage Data parameterization of the block, see “Electrical (2-D Partial Derivative Data Parameterization)” on page 1-569, “Electrical (4-D Partial Derivative Data Parameterization)” on page 1-575, or “Electrical (3-D Flux Linkage Data Parameterization)” on page 1-577 respectively.

### **Winding type — Stator windings configuration**

Wye-wound (default) | Delta-wound

Select the configuration for the stator windings:

- **Wye-wound** — The stator windings are wye-wound.
- **Delta-wound** — The stator windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

### **Direct-axis current vector, $i_D$ — Direct-axis current vector**

[-200, 0, 200] A (default)

Vector of direct-axis currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

### **Quadrature-axis current vector, $i_Q$ — Quadrature-axis current vector**

[-200, 0, 200] A (default)

Vector of quadrature-axis currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

**Rotor angle vector, theta — Rotor angle vector**

[0, 20, 40, 60] deg (default)

Vector of rotor angles corresponding to the provided flux linkage partial derivatives. The vector must start at zero. This value corresponds to the angle where the A-phase magnetic flux aligns with the rotor permanent magnetic peak flux direction (the direct-axis, or  $d$ -axis). The last value,  $\theta_{\max}$ , must be the rotor angle where the flux linkage pattern peaks again. Therefore, the number of pole pairs is  $360/\theta_{\max}$  if  $\theta_{\max}$  is expressed in degrees. The default value corresponds to a 6 pole-pair motor.

**A-phase flux linkage partial derivative wrt iA, dPhiA(id,iq,theta)/diA — A-phase flux linkage partial derivative with respect to iA**

zeros(3, 3, 4) Wb/A (default)

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding A, defined as a function of the two current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt iB, dPhiA(id,iq,theta)/diB — A-phase flux linkage partial derivative with respect to iB**

zeros(3, 3, 4) Wb/A (default)

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding B, defined as a function of the two current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt iC, dPhiA(id,iq,theta)/diC — A-phase flux linkage partial derivative with respect to iC**

zeros(3, 3, 4) Wb/A (default)

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding C, defined as a function of the two current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt angle, dPhiA(id,iq,theta)/dtheta — A-phase flux linkage partial derivative with respect to angle**

zeros(3, 3, 4) Wb/rad (default)

Matrix of the A-phase flux linkage partial derivatives with respect to rotor angle, defined as a function of the two current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**Torque matrix, T(id,iq,theta) — Torque matrix**

zeros(3, 3, 4) N\*m (default)

Specify a matrix of the electromagnetic torque applied to the rotor, as a function of the two currents and the rotor angle.

**Interpolation method — Interpolation method**

Linear (default) | Smooth

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- **Linear** — Uses an extension of linear algorithm for multidimensional interpolation. Select this option to get the best performance.
- **Smooth** — Uses a modified Akima interpolation algorithm. Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation methods, see the PS Lookup Table (3D) block reference page.

### Stator resistance per phase, $R_s$ — Stator resistance per phase

0.013 Ohm (default)

Resistance of each of the stator windings.

### Electrical (4-D Partial Derivative Data Parameterization)

This configuration of the **Electrical** parameters corresponds to the 4-D Partial Derivative Data block variants, with or without thermal ports. If you are using the 2-D Partial Derivative Data, 3-D Partial Derivative Data, or 3-D Flux Linkage Data parameterization of the block, see “Electrical (2-D Partial Derivative Data Parameterization)” on page 1-569, “Electrical (3-D Partial Derivative Data Parameterization)” on page 1-573, or “Electrical (3-D Flux Linkage Data Parameterization)” on page 1-577 respectively.

### Winding type — Stator windings configuration

Wye-wound (default) | Delta-wound

Select the configuration for the stator windings:

- **Wye-wound** — The stator windings are wye-wound.
- **Delta-wound** — The stator windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

### A-phase current vector, $i_A$ — A-phase current vector

[-200, 0, 200] A (default)

Vector of A-phase currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

### B-phase current vector, $i_B$ — B-phase current vector

[-200, 0, 200] A (default)

Vector of B-phase currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

### C-phase current vector, $i_C$ — C-phase current vector

[-200, 0, 200] A (default)

Vector of C-phase currents corresponding to the provided flux linkage partial derivatives. The current vector must be two-sided (have positive and negative values).

### Rotor angle vector, $\theta$ — Rotor angle vector

[0, 20, 40, 60] deg (default)

Vector of rotor angles corresponding to the provided flux linkage partial derivatives. The vector must start at zero. This value corresponds to the angle where the A-phase magnetic flux aligns with the rotor permanent magnetic peak flux direction (the direct-axis, or *d*-axis). The last value,  $\theta_{\max}$ , must be

the rotor angle where the flux linkage pattern peaks again. Therefore, the number of pole pairs is  $360/\theta_{\max}$  if  $\theta_{\max}$  is expressed in degrees. The default value corresponds to a 6 pole-pair motor.

**A-phase flux linkage partial derivative wrt  $i_A$ ,  $d\Phi_A(i_A, i_B, i_C, \theta)/di_A$  – A-phase flux linkage partial derivative with respect to  $i_A$**

`zeros(3, 3, 3, 4) Wb/A (default)`

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding A, defined as a function of the three current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt  $i_B$ ,  $d\Phi_A(i_A, i_B, i_C, \theta)/di_B$  – A-phase flux linkage partial derivative with respect to  $i_B$**

`zeros(3, 3, 3, 4) Wb/A (default)`

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding B, defined as a function of the three current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt  $i_C$ ,  $d\Phi_A(i_A, i_B, i_C, \theta)/di_C$  – A-phase flux linkage partial derivative with respect to  $i_C$**

`zeros(3, 3, 3, 4) Wb/A (default)`

Matrix of the A-phase flux linkage partial derivatives with respect to current in winding C, defined as a function of the three current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**A-phase flux linkage partial derivative wrt angle,  $d\Phi_A(i_A, i_B, i_C, \theta)/d\theta$  – A-phase flux linkage partial derivative with respect to angle**

`zeros(3, 3, 3, 4) Wb/rad (default)`

Matrix of the A-phase flux linkage partial derivatives with respect to rotor angle, defined as a function of the three current vectors and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

**Torque matrix,  $T(i_A, i_B, i_C, \theta)$  – Torque matrix**

`zeros(3, 3, 3, 4) N*m (default)`

Specify a matrix of the electromagnetic torque applied to the rotor, as a function of the three currents and the rotor angle.

**Interpolation method – Interpolation method**

`Linear (default) | Smooth`

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- **Linear** — Uses an extension of linear algorithm for multidimensional interpolation. Select this option to get the best performance.
- **Smooth** — Uses a modified Akima interpolation algorithm. Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation methods, see the PS Lookup Table (3D) block reference page.

**Stator resistance per phase,  $R_s$  — Stator resistance per phase**

0.013 Ohm (default)

Resistance of each of the stator windings.

**Electrical (3-D Flux Linkage Data Parameterization)**

This configuration of the **Electrical** parameters corresponds to the 3-D Flux Linkage Data block variants, with or without thermal ports. If you are using the 2-D Partial Derivative Data, 3-D Partial Derivative Data, or 4-D Partial Derivative Data parameterization of the block, see “Electrical (2-D Partial Derivative Data Parameterization)” on page 1-569, “Electrical (3-D Partial Derivative Data Parameterization)” on page 1-573, or “Electrical (4-D Partial Derivative Data Parameterization)” on page 1-575, respectively.

**Flux linkage data format — Flux linkage data format**

D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) (default) | D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) | A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) | A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ )

Select the flux linkage data format used by your FE tool:

- D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ )
- D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ )
- A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ )
- A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ )

**Winding type — Stator windings configuration**

Wye-wound (default) | Delta-wound

Select the configuration for the stator windings:

- Wye-wound — The stator windings are wye-wound.
- Delta-wound — The stator windings are delta-wound. The  $a$ -phase is connected between ports **a** and **b**, the  $b$ -phase between ports **b** and **c** and the  $c$ -phase between ports **c** and **a**.

**Expose neutral port — Neutral port visibility**

No (default) | Yes

Whether to expose the neutral port of the block.

If you expose the neutral port, the block also models the zero sequence currents. The zero sequence is parameterized by the **Stator zero-sequence inductance,  $L_0$**  parameter. When you tabulate in terms of  $i_D$  and  $i_Q$ , or in terms of current magnitude and phase advance, there is no information about non-linear dependence on the zero-sequence current, and the block introduces a degree of approximation. For more accurate results, particularly for large zero-sequence currents, use the **4-D partial derivative data** parameterization option.

**Dependencies**

To enable this parameter, set **Winding type** parameter to Wye-wound.

**Number of pole pairs — Number of pole pairs**

4 (default)

Number of the permanent magnet motor pole pairs. The default value is 4.

**Park's convention for tabulated data — Park's convention for tabulated data**

Q leads D, rotor angle measured from A-phase to D-axis (default) | Q leads D, rotor angle measured from A-phase to Q-axis | D leads Q, rotor angle measured from A-phase to D-axis | D leads Q, rotor angle measured from A-phase to Q-axis

Select the order and reference angle for the Park transform mapping the given  $dq$  data to the three windings.

- Q leads D, rotor angle measured from A-phase to D-axis — Quadrature-direct transformation with angle measured with respect to  $d$  axis.
- Q leads D, rotor angle measured from A-phase to Q-axis — Quadrature-direct transformation with angle measured with respect to  $q$  axis.
- D leads Q, rotor angle measured from A-phase to D-axis — Direct-quadrature transformation with angle measured with respect to  $d$  axis.
- D leads Q, rotor angle measured from A-phase to Q-axis — Direct-quadrature transformation with angle measured with respect to  $q$  axis.

**Direct-axis current vector,  $i_D$  — Direct-axis current vector**

[-200, 0, 200] A (default)

Vector of direct-axis currents at which the flux linkage is tabulated. The current vector must be two-sided (have positive and negative values).

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Quadrature-axis current vector,  $i_Q$  — Quadrature-axis current vector**

[-200, 0, 200] A (default)

Vector of quadrature-axis currents at which the flux linkage is tabulated. The current vector must be two-sided (have positive and negative values).

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Peak current magnitude vector,  $I$  — Peak current magnitude vector**

[0, 100, 200] A (default)



Row vector of current magnitudes at which the flux linkage is tabulated. The first element must be zero. The adjacent current value should be small relative to the current values at which magnetic saturation begins to occur. This is because derived flux partial derivatives are ill-defined at zero current, and so are calculated at this first nonzero current instead.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

#### Current advance angle, B — Current advance angle

[-180, -90, 0, 90, 180] deg (default)

Row vector of current advance angle values at which the flux linkage is tabulated. Current advance angle is defined as the angle by which the current leads the quadrature (Q) axis.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

#### Rotor angle vector, theta — Rotor angle vector

[[0, 5, 10, 15, 20, 25, 30] deg (default)

Vector of rotor angles at which the flux linkage is tabulated. The vector must start at zero. This value corresponds to the angle where the A-phase magnetic flux aligns with the rotor permanent magnetic peak flux direction (the direct-axis, or *d*-axis). The last value,  $\theta_{\max}$ , must be the rotor angle where the flux linkage pattern peaks again. Therefore, the number of pole pairs is  $360/\theta_{\max}$  if  $\theta_{\max}$  is expressed in degrees. The default value corresponds to a 4 pole-pair motor.

If **Flux linkage data format** is D and Q axes flux linkages as a function of D-axis current (*i*<sub>D</sub>), Q-axis current (*i*<sub>Q</sub>), and rotor angle (theta) or D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) (that is, if you tabulate D and Q flux linkage data), then the rotor angle vector must have four or more points and a range from 0 to  $120/N$  degrees, where *N* is the number of pole pairs. If **Flux linkage data format** is A-phase flux linkage as a function of D-axis current (*i*<sub>D</sub>), Q-axis current (*i*<sub>Q</sub>), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) (that is, if you tabulate A-phase flux linkage data), then the rotor angle vector must have  $3n+1$  points, where  $n \geq 2$ , and the range must be from 0 to  $360/3/N$  degrees.

#### D-axis flux linkage, Fd(*i*<sub>D</sub>,*i*<sub>Q</sub>,theta) — D-axis flux linkage

zeros(3, 3, 7) Wb (default)

Matrix of the *d*-axis flux linkage, defined as a function of the *dq* currents, and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of D-axis current (`iD`), Q-axis current (`iQ`), and rotor angle (`theta`).

#### Q-axis flux linkage, $F_q(iD, iQ, \theta)$ – Q-axis flux linkage

`zeros(3, 3, 7) Wb (default)`

Matrix of the  $q$ -axis flux linkage, defined as a function of the  $dq$  currents, and the rotor angle vector. Flux linkage is the flux multiplied by the number of winding turns.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of D-axis current (`iD`), Q-axis current (`iQ`), and rotor angle (`theta`).

#### D-axis flux linkage, $F_d(I, B, \theta)$ – D-axis flux linkage

`zeros(3, 5, 7) Wb (default)`

3-D matrix of  $d$ -axis flux linkage values as a function of **Peak current magnitude vector, I**, **Current advance angle, B**, and **Rotor angle vector, theta**.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of peak current magnitude (`I`), current advance angle (`B`), and rotor angle (`theta`).

#### Q-axis flux linkage, $F_q(I, B, \theta)$ – Q-axis flux linkage

`zeros(3, 5, 7) Wb (default)`

3-D matrix of  $q$ -axis flux linkage values as a function of **Peak current magnitude vector, I**, **Current advance angle, B**, and **Rotor angle vector, theta**.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

#### Dependencies

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of peak current magnitude (`I`), current advance angle (`B`), and rotor angle (`theta`).

**A-phase flux linkage,  $F(i_D, i_Q, \theta)$  – A-phase flux linkage**

zeros(3, 3, 7) Wb (default)

3-D matrix of A-phase flux linkage values, as a function of the  $dq$  currents and the rotor angle.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**A-phase flux linkage,  $F(I, B, \theta)$  – A-phase flux linkage**

zeros(3, 5, 7) Wb (default)

3-D matrix of A-phase flux linkage values, as a function of **Peak current magnitude vector, I, Current advance angle, B,** and **Rotor angle vector, theta.**

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Torque matrix,  $T(i_D, i_Q, \theta)$  – Torque matrix**

zeros(3, 3, 7) N\*m (default)

3-D matrix of the electromagnetic torque applied to the rotor, as a function of the  $dq$  currents and the rotor angle.

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Torque matrix,  $T(I, B, \theta)$  – Torque matrix**

zeros(3, 5, 7) N\*m (default)

3-D matrix of the electromagnetic torque applied to the rotor, as a function of **Peak current magnitude vector, I, Current advance angle, B,** and **Rotor angle vector, theta.**

If your flux data is given in a different order, you can use the `permute` function to reorder it. For an example of this reordering, see the associated MATLAB script in “Import IPMSM Flux Linkage Data from ANSYS Maxwell”.

**Dependencies**

This parameter is visible only when you set the **Flux linkage data format** parameter to D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

**Interpolation method — Interpolation method**

Linear (default) | Smooth

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- **Linear** — Uses an extension of linear algorithm for multidimensional interpolation. Select this option to get the best performance.
- **Smooth** — Uses a modified Akima interpolation algorithm. Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation methods, see the PS Lookup Table (4D) block reference page.

**Stator resistance per phase, Rs — Stator resistance per phase**

0.013 Ohm (default)

Resistance of each of the stator windings.

**Stator zero-sequence inductance, L0 — Stator zero-sequence inductance**

0.00016 H (default)

Zero-sequence inductance.

**Dependencies**

To enable this parameter, set **Winding type** to Delta-Wound or **Expose neutral port** to Yes.

**Iron Losses**

For more information on calculating iron losses, see “Calculating Iron Losses” on page 1-565.

**Model — Iron losses modeling option**

None (default) | Specify open and short circuit loss data (permanent magnet motors only) | Specify tabulated Steinmetz coefficients | Tabulate with current and speed

Whether to enable iron losses modeling. The Specify tabulated Steinmetz coefficients and Tabulate with current and speed options are only available if you choose the **3-D flux linkage data** | **No thermal port** or **3-D flux linkage data** | **Show thermal port** variants.

**Open-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] — Open-circuit iron losses**

[0.0, 0.0, 0.0] W (default)

Row vector, of length 3, of the open-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

**Dependencies**

To enable this parameter, set **Model** to Specify open and short circuit loss data (permanent magnet motors only)

**Short-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] — Short-circuit iron losses**

[0.0, 0.0, 0.0] W (default)

Row vector, of length 3, of the short-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

**Dependencies**

To enable this parameter, set **Model** to Specify open and short circuit loss data (permanent magnet motors only)

**Electrical frequency at which losses determined — Electrical frequency at which losses determined**

60 Hz (default)

Electrical frequency at which the open-circuit and short-circuit iron losses were measured.

**Dependencies**

To enable this parameter, set **Model** to Specify open and short circuit loss data (permanent magnet motors only).

**Short-circuit RMS current for short-circuit iron losses — Short-circuit RMS current for short-circuit iron losses**

95 A (default)

The resulting short-circuit RMS phase current when measuring the short-circuit losses.

**Dependencies**

To enable this parameter, set **Model** to Specify open and short circuit loss data (permanent magnet motors only)

**Rotor hysteresis loss coefficient, k\_hr(id,iq) — Rotor hysteresis loss coefficient**

zeros(3, 3) W/Hz (default)

Hysteresis loss coefficient of the rotor depending on the *d*-axis and *q*-axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current (*i*<sub>D</sub>), Q-axis current (*i*<sub>Q</sub>), and rotor angle (*theta*) or A-phase flux linkage as a function of D-axis current (*i*<sub>D</sub>), Q-axis current (*i*<sub>Q</sub>), and rotor angle (*theta*).

**Rotor eddy current loss coefficient, k\_Jr(id,iq) — Rotor eddy current loss coefficient**

zeros(3, 3) W/Hz<sup>2</sup> (default)

Eddy current loss coefficient of the rotor depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

#### Dependencies

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

#### Rotor excess current loss coefficient, $k_{er}(i_D, i_Q)$ – Rotor excess current loss coefficient

`zeros(3, 3) W/Hz^1.50000` (default)

Excess current loss coefficient of the rotor depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

#### Dependencies

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

#### Stator hysteresis loss coefficient, $k_{hs}(i_D, i_Q)$ – Stator hysteresis loss coefficient

`zeros(3, 3) W/Hz` (default)

Hysteresis loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

#### Dependencies

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

#### Stator eddy current loss coefficient, $k_{Js}(i_D, i_Q)$ – Stator eddy current loss coefficient

`zeros(3, 3) W/Hz^2` (default)

Eddy current loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

#### Dependencies

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Stator excess current loss coefficient,  $k_{es}(i_d, i_q)$  – Stator excess current loss coefficient**

$\text{zeros}(3, 3)$  W/Hz<sup>1.50000</sup> (default)

Excess current loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_d$ ), Q-axis current ( $i_q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_d$ ), Q-axis current ( $i_q$ ), and rotor angle ( $\theta$ ).

**Rotor hysteresis loss coefficient,  $k_{hr}(I, B)$  – Rotor hysteresis loss coefficient**

$\text{zeros}(3, 5)$  W/Hz (default)

Hysteresis loss coefficient of the rotor depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ).

**Rotor eddy current loss coefficient,  $k_{Jr}(I, B)$  – Rotor eddy current loss coefficient**

$\text{zeros}(3, 5)$  W/Hz<sup>2</sup> (default)

Eddy current loss coefficient of the rotor depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ).

**Rotor excess current loss coefficient,  $k_{er}(I, B)$  – Rotor excess current loss coefficient**

$\text{zeros}(3, 5)$  W/Hz<sup>1.50000</sup> (default)

Excess current loss coefficient of the rotor depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ).

**Stator hysteresis loss coefficient,  $k_{hs}(I,B)$  – Stator hysteresis loss coefficient**  
zeros(3, 5) W/Hz (default)

Hysteresis loss coefficient of the stator depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

**Stator eddy current loss coefficient,  $k_{Js}(I,B)$  – Stator eddy current loss coefficient**  
zeros(3, 5) W/Hz<sup>2</sup> (default)

Eddy current loss coefficient of the stator depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

**Stator excess current loss coefficient,  $k_{es}(I,B)$  – Stator excess current loss coefficient**  
zeros(3, 5) W/Hz<sup>1.50000</sup> (default)

Excess current loss coefficient of the stator depending on the current magnitude and current phase advance. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients and, in the **Electrical** tab, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta) or A-phase flux linkage as a function of peak current magnitude (I), current advance angle (B), and rotor angle (theta).

**Rotor speed vector,  $w$  – Rotor speed**  
[1000, 2000] rpm (default)

Vector of rotor speed at which the iron loss is tabulated.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed and, in the **Electrical** setting, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current (iD), Q-axis current (iQ), and rotor angle (theta) or A-phase flux linkage as a function of D-axis current (iD), Q-axis current (iQ), and rotor angle (theta).



**Rotor iron loss table,  $P_r(i_d, i_q, w)$  – Rotor iron loss table**

zeros(3, 3, 2) W (default)

Iron losses of the rotor tabulated on the  $d$ -axis and  $q$ -axis currents, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed and, in the **Electrical** setting, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Stator iron loss table,  $P_s(i_d, i_q, w)$  – Stator iron loss table**

zeros(3, 3, 2) W (default)

Iron losses of the stator tabulated on the  $d$ -axis and  $q$ -axis currents, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed and, in the **Electrical** setting, set **Flux linkage data format** to either D and Q axes flux linkages as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of D-axis current ( $i_D$ ), Q-axis current ( $i_Q$ ), and rotor angle ( $\theta$ ).

**Rotor iron loss table,  $P_r(I, B, w)$  – Rotor iron loss table**

zeros(3, 5, 2) W (default)

Iron losses of the rotor tabulated on the current magnitude, current phase advance, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed and, in the **Electrical** setting, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ).

**Stator iron loss table,  $P_s(I, B, w)$  – Stator iron loss table**

zeros(3, 5, 2) W (default)

Iron losses of the stator tabulated on the current magnitude, current phase advance, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed and, in the **Electrical** setting, set **Flux linkage data format** to either D and Q axes flux linkages as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ) or A-phase flux linkage as a function of peak current magnitude ( $I$ ), current advance angle ( $B$ ), and rotor angle ( $\theta$ ).

**Mechanical****Rotor inertia – Rotor inertia**0.01 kg\*m<sup>2</sup> (default)

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

**Rotor damping — Rotor damping**

0 N\*m/(rad/s) (default)

Rotary damping.

**Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature — Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Permanent magnet flux temperature coefficient — Permanent magnet flux temperature coefficient**

-0.001 (default)

The fractional rate of change of permanent magnet flux density with temperature. It is used to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

**Thermal mass for each stator winding — Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Initial stator winding temperatures — Initial stator winding temperatures**

[298.15, 298.15, 298.15] K (default)

A 1-by-3 row vector defining the temperature of the A, B, and C thermal ports at the start of simulation.

**Rotor thermal mass — Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Rotor initial temperature — Rotor initial temperature**

298.15 K (default)

The temperature of the rotor at the start of simulation.

**Percentage of main flux path iron losses associated with the rotor –  
Percentage of main flux path iron losses associated with the rotor**

90 (default)

The percentage of the main flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

**Percentage of cross-tooth flux path iron losses associated with the rotor –  
Percentage of cross-tooth flux path iron losses associated with the rotor**

30 (default)

The percentage of the cross-tooth flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

## Version History

Introduced in R2019b

## References

- [1] Mellor, P.H., R. Wrobel, and D. Holliday. "A computationally efficient iron loss model for brushless AC machines that caters for rated flux and field weakened operation." *IEEE Electric Machines and Drives Conference*. May 2009.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

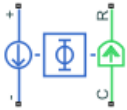
## See Also

FEM-Parameterized Linear Actuator | FEM-Parameterized Rotary Actuator |  
ee\_calculateFluxPartialDerivatives

# FEM-Parameterized Rotary Actuator

Rotary actuator defined in terms of magnetic flux

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



## Description

The FEM-Parameterized Rotary Actuator block implements a model of a rotary actuator defined in terms of magnetic flux. Use this block to model custom rotary actuators and motors where magnetic flux depends on both rotor angle and current. You parameterize the block using data from a third-party Finite Element Magnetic (FEM) package.

The block has two options for the electrical equation. The first, **Define in terms of  $d\Phi(i, \theta)/d\theta$  and  $d\Phi(i, \theta)/di$** , defines the current in terms of partial derivatives of the magnetic flux ( $\Phi$ ) with respect to rotor angle ( $\theta$ ) and current ( $i$ ), the equations for which are:

$$\frac{di}{dt} = \left( v - iR - \frac{\partial \Phi}{\partial \theta} \frac{d\theta}{dt} \right) / \frac{\partial \Phi}{\partial i}$$

The second option, **Define in terms of  $\Phi(i, \theta)$** , defines the voltage across the component directly in terms of the flux, the equation for which is:

$$v = iR + \frac{d}{dt} \Phi(\theta, i)$$

Numerically, defining the electrical equation in terms of flux partial derivatives is better because the back-emf is piecewise continuous. If using the flux directly, using a finer grid size for current and position will improve results, as will selecting cubic or spline interpolation.

In both cases, you have an option to either directly specify the torque as a function of current and rotor angle, by using the **Torque matrix,  $T(i, \theta)$**  parameter, or have the block automatically calculate the torque matrix.

If entering the electromagnetic torque data directly, you can either use data supplied by the finite element magnetic package (which you used to determine the flux) or calculate the torque from the flux with following equation:

$$T = \int_0^i \frac{\partial \Phi(\theta, i)}{\partial \theta} di$$

See the “Solenoid Parameterized with FEM Data” example model for an example of how to implement this type of integration in MATLAB.

Alternatively, the block can automatically calculate the torque matrix from the flux information that you provide. To select this option, set the **Calculate torque matrix?** parameter to Yes. The torque

matrix calculation occurs at model initialization based on current block flux linkage information. The torque is calculated by numerically integrating the rate of change of flux linkage with respect to angle over current, according to the preceding equation. If the **Electrical model** parameter is set to **Define in terms of  $\Phi(i, \theta)$** , then the block must first estimate the **Flux partial derivative wrt angle,  $\Phi(i, \theta)/d\theta$**  parameter value from the flux linkage data. When doing this, the block uses the interpolation method specified by the **Interpolation method** parameter. Typically, the **Smooth** option is most accurate, but the **Linear** option is most robust.

You can define  $\Phi$  and its partial derivatives for just positive, or positive and negative currents. If defining for just positive currents, then the block assumes that  $\Phi(-i, x) = -\Phi(i, x)$ . Therefore, if the current vector is positive only:

- The first current value must be zero.
- The flux corresponding to zero current must be zero.
- The partial derivative of flux with respect to rotor angle must be zero for zero current.

To model a rotary motor with a repeated flux pattern, set the **Flux dependence on displacement** parameter to **Cyclic**. When selecting this option, the torque and flux (or torque and flux partial derivatives depending on the option chosen) must have identical first and last columns.

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Assumptions and Limitations

- You must supply a consistent set of torque and flux data. There is no check to ensure that the torque matrix is consistent with the flux data.
- When driving the FEM-Parameterized Rotary Actuator block via a series inductor, you may need to include a parallel conductance in the inductor component.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the actuator positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the actuator negative terminal.

**C – Case**

mechanical

Mechanical rotational conserving port associated with the actuator case.

**R – Rotor**

mechanical

Mechanical rotational conserving port associated with the rotor.

**H – Thermal port**

thermal

Thermal port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

**Magnetic Force****Electrical model – Electrical model parameterization**Define in terms of  $d\Phi(i,\theta)/d\theta$  and  $d\Phi(i,\theta)/di$  (default) | Define in terms of  $\Phi(i,\theta)$ 

Select one of the following parameterization options, based on the underlying electrical model:

- Define in terms of  $d\Phi(i,\theta)/d\theta$  and  $d\Phi(i,\theta)/di$  — Define the current through the block in terms of partial derivatives of the magnetic flux with respect to distance and current.
- Define in terms of  $\Phi(i,\theta)$  — Define the voltage across the block terminals directly in terms of the flux.

**Current vector,  $i$  – Current vector**

[0, .2, .4, .6, .8, 1] A (default)

Specify a vector of monotonically increasing current values corresponding to your torque-flux data. If you specify positive currents only, the first element must be zero.

**Angle vector,  $\theta$  – Angle vector**

[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180] deg (default)

Specify a vector of monotonically increasing rotor angle values corresponding to your torque-flux data.







```
2.0265, 1.5041, .8003, 0, -.8003, -1.5041, -2.0265, -2.3045, -2.3045,
-2.0265, -1.5041, -.8003, 0; 0, 1.4228, 2.674, 3.6027, 4.0968, 4.0968,
3.6027, 2.674, 1.4228, 0, -1.4228, -2.674, -3.6027, -4.0968, -4.0968,
-3.6027, -2.674, -1.4228, 0; 0, 2.2231, 4.1781, 5.6292, 6.4013, 6.4013,
5.6292, 4.1781, 2.2231, 0, -2.2231, -4.1781, -5.6292, -6.4013, -6.4013,
-5.6292, -4.1781, -2.2231, 0] mN*m (default)
```

Specify a matrix of the electromagnetic torque applied to the rotor. This parameter is visible only if **Calculate torque matrix?** is set to No. The default value, in mN\*m, is:

```
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
 0 0.0889 0.1671 0.2252 0.2561 0.2561 0.2252 0.1671 0.0889 ...
 0 -0.0889 -0.1671 -0.2252 -0.2561 -0.2561 -0.2252 -0.1671 -0.0889 0;
 0 0.3557 0.6685 0.9007 1.0242 1.0242 0.9007 0.6685 0.3557 ...
 0 -0.3557 -0.6685 -0.9007 -1.0242 -1.0242 -0.9007 -0.6685 -0.3557 0;
 0 0.8003 1.5041 2.0265 2.3045 2.3045 2.0265 1.5041 0.8003 ...
 0 -0.8003 -1.5041 -2.0265 -2.3045 -2.3045 -2.0265 -1.5041 -0.8003 0;
 0 1.4228 2.674 3.6027 4.0968 4.0968 3.6027 2.674 1.4228 ...
 0 -1.4228 -2.674 -3.6027 -4.0968 -4.0968 -3.6027 -2.674 -1.4228 0;
 0 2.2231 4.1781 5.6292 6.4013 6.4013 5.6292 4.1781 2.2231 ...
 0 -2.2231 -4.1781 -5.6292 -6.4013 -6.4013 -5.6292 -4.1781 -2.2231 0 ]
```

### Dependencies

This parameter is visible only when you set the **Calculate torque matrix?** parameter to No.

### Flux dependence on displacement — Flux pattern

Unique (default) | Cyclic

Specify the flux pattern:

- **Unique** — No flux pattern present.
- **Cyclic** — Select this option to model a linear motor with a repeated flux pattern. The force and flux (or force and flux partial derivatives, depending on the **Electrical model** option chosen) must have identical first and last columns.

### Interpolation method — Interpolation method

Linear (default) | Smooth

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- **Linear** — Select this option to get the best performance.
- **Smooth** — Select this option to produce a continuous surface with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (2D) block reference page.

### Extrapolation method — Extrapolation method

Linear (default) | Nearest | Error

Select one of the following extrapolation methods for determining the output value when the input value is outside the range specified in the argument list:

- **Linear** — Select this option to produce a surface with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- **Nearest** — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

- **Error** — Issues an error if the input signal is outside the range of the table. Select this option to avoid going into the extrapolation mode when you want your data to be within the table range.

For more information on extrapolation algorithms, see the PS Lookup Table (2D) block reference page.

### **Dependencies**

This parameter is visible only when you set the **Flux dependence on displacement** parameter to Unique.

### **Winding resistance — Electrical winding total resistance**

14 Ohm (default)

Total resistance of the electrical winding.

### **Magnetizing resistance — Magnetizing resistance for iron losses**

inf Ohm (default)

Magnetizing resistance. The value must be greater than zero. The default value is inf, which implies that there are no iron losses.

### **Mechanical**

#### **Damping — Rotary damping**

0.0001 N\*m/(rad/s) (default)

Rotary damping. The value can be zero.

#### **Rotor inertia — Rotor inertia**

0.00005 kg\*m<sup>2</sup> (default)

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

#### **Minimum rotor angle — Minimum rotor angle**

-Inf deg (default)

The rotor angle at which the lower mechanical end stop is applied.

#### **Maximum rotor angle — Maximum rotor angle**

Inf deg (default)

The rotor angle at which the upper mechanical end stop is applied.

#### **Contact stiffness — Contact stiffness**

1e8 N\*m/rad (default)

Contact stiffness between rotor and end stops.

#### **Contact damping — Contact damping**

1e4 N\*m/(rad/s) (default)

Contact damping between rotor and end stops.

### **Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

**Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Resistance temperature coefficient.

**Measurement temperature – Measurement temperature**

25 degC (default)

Temperature for which the actuator parameters are defined.

**Thermal Port**To enable this parameter, set **Modeling option** to Show thermal port.**Thermal mass – Thermal mass**

100 J/K (default)

Thermal mass is the energy required to raise the temperature by one degree.

## Version History

Introduced in R2010a

### Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

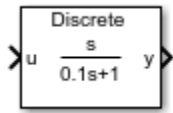
**See Also**

FEM-Parameterized Linear Actuator | FEM-Parameterized PMSM | Solenoid

## Filtered Derivative (Discrete or Continuous)

Discrete-time or continuous-time filtered derivative

**Library:** Simscape / Electrical / Control / General Control



### Description

The Filtered Derivative (Discrete or Continuous) block implements a filtered derivative in conformance with IEEE 421.5-2016<sup>[1]</sup>.

You can switch between continuous and discrete implementations of the derivative using the **Sample time** parameter.

### Equations

#### Continuous

To configure the filtered derivative for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{Ks}{Ts + 1},$$

where:

- $K$  is the gain.
- $T$  is the time constant.

From the preceding transfer function, the derivative defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}(Ku(t) - x(t)) \\ y(t) = \frac{1}{T}(Ku(t) - x(t)) \end{cases} \quad x(0) = u_0, \quad y(0) = 0,$$

where:

- $u$  is the block input.
- $x$  is the state.
- $y$  is the block output.
- $t$  is the simulation time.
- $u_0$  is the initial input to the block.

## Discrete

To configure the filtered derivative for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$\left(\frac{K}{T}\right)\frac{z-1}{z+T_s/T-1},$$

where:

- $K$  is the gain.
- $T$  is the time constant.
- $T_s$  is the sample time.

From the discrete transfer function, the derivative equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = \left(1 - \frac{T_s}{T}\right)x(n) + \left(\frac{T_s}{T}\right)u(n) \\ y(n) = \frac{K}{T}(u(n) - x(n)) \end{cases} \quad x(0) = u_0, y(0) = 0,$$

where:

- $u$  is the block input.
- $x$  is the block state.
- $y$  is the block output.
- $n$  is the simulation time step.
- $u_0$  is the initial input to the block.

## Initial Conditions

To specify the initial conditions of this block, set **Initialization** to:

- **Inherited from block input** — The block sets the state initial condition to the initial input, making the initial output zero.
- **Specify as parameter** — The block sets the state initial condition to the value of **Initial state**.

## Limiting the Output

Limit the filtered derivative output by setting the **Upper saturation limit** and **Lower saturation limit** parameters to finite values.

Unlike other common blocks given in IEEE 421.5-2016, there is no difference between the windup and anti-windup saturation methods for the filtered derivative. The output can respond immediately to a reversal of the input sign when the output is saturated.

## Ports

### Input

#### **u — Derivative input**

vector

Filtered derivative input signal. The block uses the input initial value to determine the state initial value.

Data Types: `single` | `double`

### Output

#### **y — Derivative output**

vector

Filtered derivative output signal.

Data Types: `single` | `double`

## Parameters

#### **Gain — Derivative gain**

1 (default) | positive number

Filtered derivative gain.

#### **Time constant — Derivative time constant**

0.1 (default) | positive number

Filtered derivative time constant. For acceptable accuracy, set this value at least 10 times greater than the **Sample time**.

#### **Upper saturation limit — Output upper limit**

`inf` (default) | real number

Filtered derivative upper output limit. Set this to `inf` for an unsaturated upper limit.

#### **Lower saturation limit — Output lower limit**

`-inf` (default) | real number

Filtered derivative lower output limit. Set this to `-inf` for an unsaturated lower limit.

#### **Minimum sample time to time constant ratio — Discrete ratio**

10 (default) | real number

Minimum acceptable sample time to time constant ratio. As the sample time approaches the time constant, the accuracy of the block decreases. Use this parameter to set the tolerance of this ratio.

#### **Initialization — Initial state specification**

Inherited from block input (default) | Specify as parameter

Specify the initial state condition for this block. For more information, see “Initial Conditions” on page 1-599.

**Initial state – Initial state**

0 (default) | real number

Block initial state.

**Dependencies**To enable this parameter, set **Initialization to Specify** as parameter.**Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive number

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

For acceptable accuracy, set this value at least 10 times smaller than the **Time constant** parameter.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Version History****Introduced in R2017b****References**

[1] IEEE. 2016. *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also****Blocks**

Lead-Lag (Discrete or Continuous) | Washout (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

## Finite-Gain Op-Amp

Gain-limited operational amplifier model with optional noise

**Library:** Simscape / Electrical / Integrated Circuits



### Description

The Finite-Gain Op-Amp block models a gain-limited operational amplifier. If the voltages at the positive and negative ports are  $V_p$  and  $V_m$ , respectively, the output voltage is:

$$V_{out} = A(V_p - V_m) - I_{out} * R_{out}$$

where:

- $A$  is the gain.
- $R_{out}$  is the output resistance.
- $I_{out}$  is the output current.

The input current is:

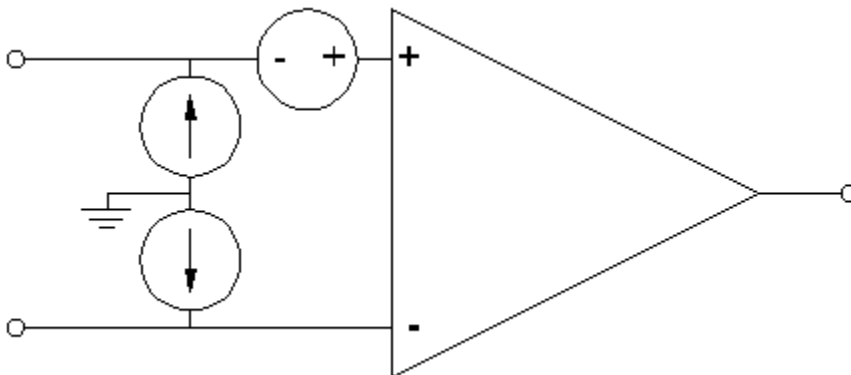
$$\frac{V_p - V_m}{R_{in}}$$

where  $R_{in}$  is the input resistance.

The output voltage is limited by the minimum and maximum output values you specify in the block dialog box.

### Thermal Noise

The Finite-Gain Op-Amp block can generate thermal noise. If you set the **Noise mode** parameter to Enabled, then the equivalent circuit for the block includes a noise current source attached to each of the inputs, and a noise voltage source attached to the noninverting input. These three noise sources are independent and uncorrelated.





The block generates noise voltage and current according to:

$$v_{noise} = \frac{v_{density}}{\sqrt{2h}}N(0, 1)$$

$$i_{noise} = \frac{i_{density}}{\sqrt{2h}}N(0, 1)$$

where:

- $v_{noise}$  is the noise voltage.
- $v_{density}$  is the single-sided, spectral amplitude density of the voltage noise.
- $i_{noise}$  is the noise current at an input.
- $i_{density}$  is the single-sided, spectral amplitude density of the current noise applied to that input.
- $h$  is the sampling time.
- $N$  is a Gaussian random number with zero mean and standard deviation of one.

The block generates Gaussian noise by using the PS Random Number source in the Simscape Foundation library. You can control the random number seed by setting the **Repeatability** parameter:

- **Not repeatable** — Every time you simulate your model, the block resets the random seed using the MATLAB random number generator:

```
seed = randi(2^32-1);
```

- **Repeatable** — The block automatically generates a seed value and stores it inside the block, to always start the simulation with the same random number. This auto-generated seed value is set when you add a Finite-Gain Op-Amp block from the block library to the model. When you make a new copy of the Finite-Gain Op-Amp block from an existing one in a model, a new seed value is generated. The block sets the value using the MATLAB random number generator command shown above.
- **Specify seed** — If you select this option, additional parameters let you directly specify the random number seed values for input voltage, noninverting input current, and inverting input current.

## Ports

### Conserving

#### + — Positive electrical voltage

electrical

Electrical conserving port associated with the op-amp noninverting input.

#### - — Negative electrical voltage

electrical

Electrical conserving port associated with the op-amp inverting input.

#### out — Output voltage

electrical

Electrical conserving port associated with the op-amp output. The port name is hidden on the block icon, but you can see it in simulation data logs.

## Parameters

### Main

#### Gain, A — Open-loop gain

1000 (default)

The open-loop gain of the operational amplifier.

#### Input resistance, Rin — Resistance at block input

1e6 Ohm (default)

The resistance at the input of the operational amplifier that the block uses to calculate the input current.

#### Output resistance, Rout — Resistance at block output

100 Ohm (default)

The resistance at the output of the operational amplifier that the block uses to calculate the drop in output voltage due to output current.

#### Minimum output, Vmin — Output voltage lower limit

-15 V (default)

The lower limit on the operational amplifier output voltage.

#### Maximum output, Vmax — Output voltage upper limit

15 V (default)

The upper limit on the operational amplifier output voltage.

### Noise

#### Noise mode — Select whether op-amp generates thermal noise

Disabled (default) | Enabled

Select whether to model thermal noise effects:

- Disabled — Op-amp does not generate thermal noise.
- Enabled — Op-amp generates thermal noise voltage and current, and the associated parameters become visible in the **Noise** section.

#### Input noise voltage density — Density of voltage noise

30e-9 V/Hz<sup>0.5</sup> (default)

Single-sided, spectral amplitude density of the voltage noise applied to the noninverting input.

### Dependencies

Enabled when the **Noise mode** parameter is set to Enabled.

**Noise current parameterization — Select whether noise current density is different or the same for both inputs**

Apply same density function to both inputs (default) | Apply different density function to each input

Select whether current density values applied to block inputs are different or the same.

**Dependencies**

Enabled when the **Noise mode** parameter is set to Enabled.

**Input noise current density — Current noise at both inputs**

0.5e-12 A/Hz<sup>0.5</sup> (default)

Single-sided, spectral amplitude density of the current noise applied to both inputs. Note that even though the density function is the same for both inputs, the actual noise current also depends on the random number seed. If the seeds used for the random number generation are different, then the actual noise currents at the inputs are also different.

**Dependencies**

Enabled when the **Noise current parameterization** parameter is set to Apply same density function to both inputs.

**Noninverting input noise current density — Current noise at noninverting input**

0.5e-12 A/Hz<sup>0.5</sup> (default)

Single-sided, spectral amplitude density of the current noise applied to noninverting input.

**Dependencies**

Enabled when the **Noise current parameterization** parameter is set to Apply different density function to each input.

**Inverting input noise current density — Current noise at inverting input**

0.5e-12 A/Hz<sup>0.5</sup> (default)

Single-sided, spectral amplitude density of the current noise applied to inverting input.

**Dependencies**

Enabled when the **Noise current parameterization** parameter is set to Apply different density function to each input.

**Sample time — Rate at which the noise source is sampled**

1e-3 s (default)

Defines the rate at which the noise source is sampled. Choose it to reflect the frequencies of interest in your model. Making the sample time too small will unnecessarily slow down your simulation.

**Dependencies**

Enabled when the **Noise mode** parameter is set to Enabled.

**Repeatability — Select the noise control option**

Not repeatable (default) | Repeatable | Specify seed

Select the noise control option:

- **Not repeatable** — The random sequence used for noise generation is not repeatable.
- **Repeatable** — The random sequence used for noise generation is repeatable, with a system-generated seed.
- **Specify seed** — The random sequence used for noise generation is repeatable, and you control the seed by using the seed parameters. You specify seed values separately for input noise voltage, noninverting input noise current, and inverting input noise current.

**Dependencies**

Enabled when the **Noise mode** parameter is set to Enabled.

**Input noise voltage auto-generated seed used for repeatable option — Auto-generated random number seed for voltage noise**

random real number

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Input noise voltage seed** parameter.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Repeatable.

**Noninverting input noise current auto-generated seed used for repeatable option — Auto-generated random number seed for current noise at noninverting input**

random real number

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Noninverting input noise current seed** parameter.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Repeatable.

**Inverting input noise current auto-generated seed used for repeatable option — Auto-generated random number seed for current noise at inverting input**

random real number

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Inverting input noise current seed** parameter.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Repeatable.

**Input noise voltage seed — Random number seed for voltage noise**

0 (default)

Seed used by the noise random number generator.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Specify seed.

**Noninverting input noise current seed — Random number seed for current noise at noninverting input**

0 (default)

Seed used by the noise random number generator.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Specify seed.

**Inverting input noise current seed — Random number seed for current noise at inverting input**

0 (default)

Seed used by the noise random number generator.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Specify seed.

## Version History

Introduced in R2008b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Band-Limited Op-Amp | Fully Differential Op-Amp | Op-Amp

## Floating Neutral (Three-Phase)

Internal floating neutral point for wye-connected network

**Library:** Simscape / Electrical / Connectors & References



### Description

The Floating Neutral (Three-Phase) block connects the individual phases of a three-phase system to form a floating neutral point.

---

**Note** If you want to create a neutral point that you can connect to other blocks, use the Neutral Port (Three-Phase) block. If you want to create a neutral point that is connected to ground, use the Grounded Neutral (Three-Phase) block.

---

### Ports

#### Conserving

~ — **Three-phase port**

electrical

Expandable composite (**a,b, c**) three-phase port.

### Parameters

**Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Parasitic ground conductance — Parasitic ground conductance**

1e-12 1/0hm (default)

Parasitic conductance to ground. A nonzero value is required for the simulation of some circuit topologies.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

[Grounded Neutral \(Three-Phase\)](#) | [Open Circuit \(Three-Phase\)](#) | [Neutral Port \(Three-Phase\)](#)

## Foster Thermal Model

Heat transfer through a semiconductor module

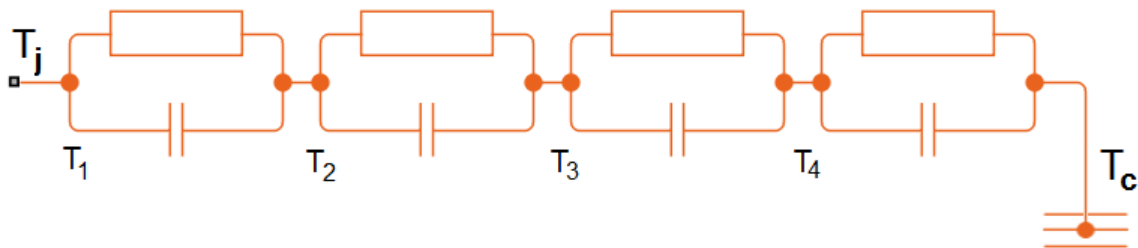
**Library:** Simscape / Electrical / Passive / Thermal



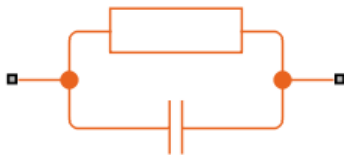
### Description

**Note** The Foster Thermal Model block represents an end-to-end behavioral model for temperature drop across the  $T_j$  and  $T_c$  ports. This block does not predict the heatflow rate at port **B**. You cannot extend this model with additional blocks either. To predict the heatflow rate at port B and to extend the model with additional blocks, use the Causer Thermal Model block and select the **Parameterize from Foster coefficients** parameter.

The Foster Thermal Model block represents heat transfer through a semiconductor module. The figure shows an equivalent circuit for a fourth-order Foster Thermal Model block.  $T_j$  is the junction temperature and  $T_c$  is the base plate or ambient temperature.



A Foster thermal model contains one or more instances of Foster thermal model elements. The figure shows an equivalent circuit for a Foster thermal model element.



The number of thermal elements is equal to the order of representation. For a first order model, use scalar block parameters. For an  $n$ th order model, use row vectors of length  $n$ . Other terms that describe a Foster thermal model are:

- Partial fraction circuit
- Pi model

The defining equations for a first-order Foster thermal model element are:



$$C_{thermal} = \frac{\tau}{R_{thermal}}$$

and

$$Q_{AB} = \frac{T_{AB}}{R_{thermal}} + C_{thermal} \frac{dT_{AB}}{dt},$$

where:

- $C_{thermal}$  is the thermal capacity.
- $\tau$  is the thermal time constant.
- $R_{thermal}$  is the thermal resistance.
- $Q_{AB}$  is the heat flow through the material.
- $T_{AB}$  is the temperature difference between the material layers.

### Initial Conditions

You can initialize the model in steady state or specify the initial node temperatures. To initialize the block in steady state, set the **Start from steady state** parameter to On. To specify the initial node temperatures and enable the **Initial Targets** section, set **Start from steady state** to Off. The first node temperature corresponds to the temperature in port A.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Ports

#### Conserving

##### A — Left junction

*thermal*

Thermal conserving port associated with the left junction.

##### B — Right junction

*thermal*

Thermal conserving port associated with the right junction.

### Parameters

#### Thermal resistance data — Thermal resistance values

[ 0.0016 0.0043 0.0013 0.0014 ] K/W (default)

Thermal resistance values,  $R_{thermal}$ , of the semiconductor module, specified as a vector.

**Thermal time constant data – Thermal time constant data**

[ 0.0068 0.064 0.32 2 ] s (default)

Thermal time constant values,  $\tau$ , of the semiconductor module, specified as a vector.

**Start from steady state – Thermal time constant data**

On (default) | Off

Whether to initialize the block in steady state.

## Version History

Introduced in R2016a

## References

[1] Schütze, T. *AN2008-03: Thermal equivalent circuit models*. Application Note. V1.0. Germany: Infineon Technologies AG, 2008.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Cauer Thermal Model | Thermal Resistor

### Functions

ee\_getCauerFromFoster

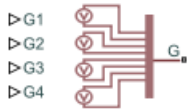
### Topics

“Simulating Thermal Effects in Semiconductors”

# Four-Pulse Gate Multiplexer

Multiplex gate input signals to four quadrant

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Four-Pulse Gate Multiplexer block multiplexes four separate voltage signals into a single vector. The vectorized signal can control the gates of four switching devices in a converter, such as a Four-Quadrant Chopper block.

## Model

This block can model a four-pulse gate multiplexer through either physical signal ports or electrical ports:

- PS ports — Four-pulse gate multiplexer with physical signal ports. Set the **Gate-control port** parameter to PS to control switching device gates in a converter block using Simulink gate-control voltage signals. To multiplex and connect Simulink signals to the gate-control inport of a converter block:
  - 1 Convert each voltage signal using a Simulink-PS Converter block.
  - 2 Multiplex the converted gate signals into a single vector using the multiplexer block.
  - 3 Connect the vector signal to the **G** port of the converter.
- Electrical ports — Four-pulse gate multiplexer with electrical conserving ports. To control switching device gates in a converter block using Simscape Electrical Electronics and Mechatronics blocks, set the **Gate-control port** parameter to **Electrical**. The electrical ports include pairs of electrical connections. Each pair corresponds to the gate and cathode of a switching device in the connected converter block.

## Ports

### Input

#### G1 — Gate-control voltage signal 1

physical signal

Physical signal port associated with the gate terminal of the first switching device in a connected converter block.

### Dependencies

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G2 — Gate-control voltage signal 2**

physical signal

Physical signal port associated with the gate terminal of the second switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G3 — Gate-control voltage signal 3**

physical signal

Physical signal port associated with the gate terminal of the third switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**G4 — Gate-control voltage signal 4**

physical signal

Physical signal port associated with the gate terminal of the fourth switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**Conserving****G1 — Gate-control voltage signal 1**

electrical

Electrical conserving port associated with the gate terminal of the first switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**a — A-phase AC reference point**

electrical

Electrical conserving port associated with the A-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G2 — Gate-control voltage signal 2**

electrical

Electrical conserving port associated with the gate terminal of the second switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**b — B-phase AC reference point**

electrical

Electrical conserving port associated with the *B*-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G3 — Gate-control voltage signal 3**

electrical

Electrical conserving port associated with the gate terminal of the third switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G4 — Gate-control voltage signal 4**

electrical

Electrical conserving port associated with the gate terminal of the fourth switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**L — DC reference point**

electrical

Electrical conserving port associated with the DC negative connection for the low-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Parameters****Gate-control port — Gate control port**

PS (default) | Electrical

Whether to expose PS or electrical control ports.

**Version History**

Introduced in R2018a

**Physical and electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between physical and electrical ports, set the **Gate-control port** parameter to either PS or Electrical.

As a result of these changes, inside a model saved in an earlier release, review the **Gate-control port** parameter of this block.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Four-Quadrant Chopper | Six-Pulse Gate Multiplexer | Twelve-Pulse Gate Multiplexer | Two-Pulse Gate Multiplexer

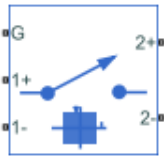
**Topics**

Switch Between Physical Signal and Electrical Ports

# Four-Quadrant Chopper

Controller-driven four quadrant DC-DC chopper

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters

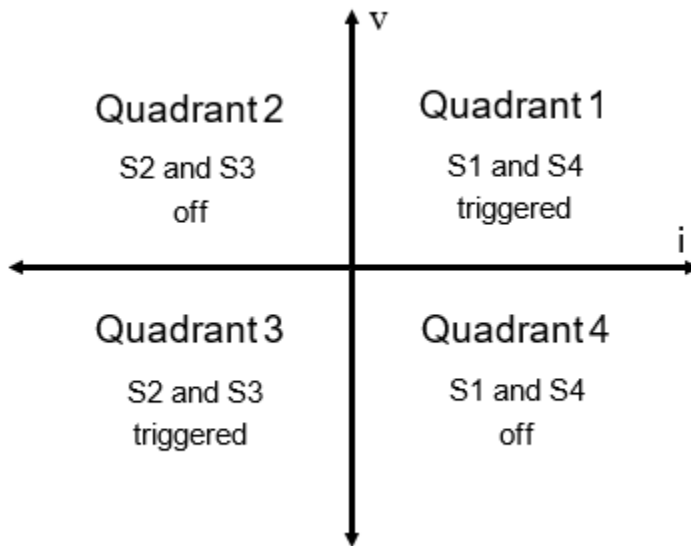
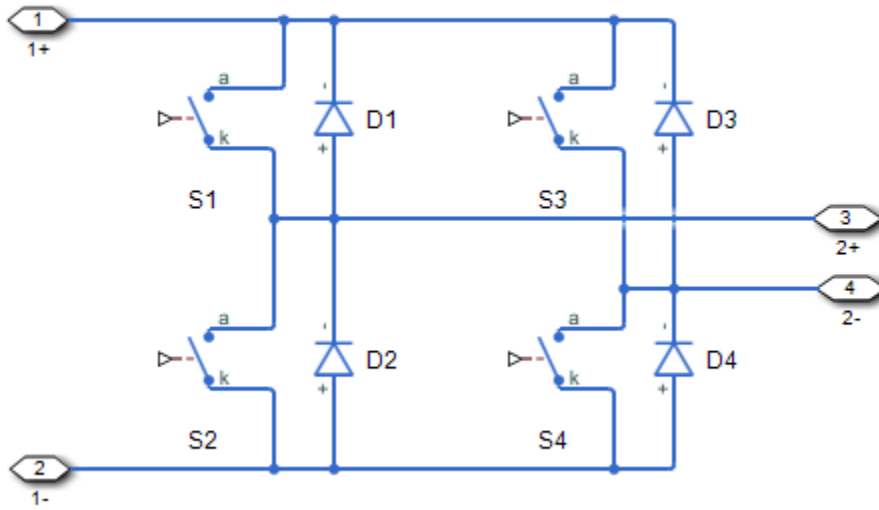


## Description

The Four-Quadrant Chopper block represents a four-quadrant controlled chopper for converting a fixed DC input to a variable DC output. The block contains two bridge arms. Each bridge arm each has two switching devices. Options for the type of switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

The figures show the equivalent circuit and the operation for the block.



**Protection**

The block contains an integral protection diode for each switching device. The integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

To configure the internal protection diode block, use the **Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.



Goals	Value to Select	Integral Protection Diode
Prioritize simulation speed.	Diode with no dynamics	The Diode block
Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

### Gate Control

To connect Simulink gate-control voltage signals to the gate ports of the internal switching devices:

- 1 Convert each voltage signal using a Simulink-PS Converter block.
- 2 Multiplex the converted gate signals into a single vector using a Four-Pulse Gate Multiplexer block.
- 3 Connect the vector signal to the **G** port.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

### Assumptions and Limitations

If, in the Solver Configuration block inside your model, you set the **Solver type** parameter to *Partitioning*, the averaged switches introduce instability during dead time, when all gate inputs are set to 0. Where possible, use the open-zero state by setting all high-side switches to 0 and all low-side switches to 1.

## Ports

### Conserving

#### **G — Switching device gate control**

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

Data Types: double

#### **1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

#### **1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

#### **2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

#### **2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

## Parameters

### Switching Devices

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

## Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

### Switching device – Switch type

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the converter.

### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage – Voltage

0.8 Ohm (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

### Dependencies

See the Switching Devices Parameter Dependencies table.

### On-state resistance – Resistance

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on

- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 0hm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance**

1e-5 1/0hm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate trigger voltage, Vgt — Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off – Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Holding current – Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Integer for piecewise constant approximation of gate input (0 for disabled) – Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Protection Diode**

The visibility of **Diode** parameters depends on how you configure the protection diode **Model dynamics** and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

## Protection Diode Parameter Dependencies

Parameters and Options			
<b>Model dynamics</b>			
None	<i>Diode with no dynamics</i>		<i>Diode with charge dynamics</i>
	Forward voltage		Forward voltage
	On resistance		On resistance
	Off conductance		Off conductance
			Junction capacitance
			Peak reverse current, iRM
			Initial forward current when measuring iRM
			Rate of change of current when measuring iRM
	<b>Reverse recovery time parameterization</b>		
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>
	Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr

### Model dynamics – Diode model

Diode with no dynamics (default) | None | Diode with charge dynamics

Diode type. The options are:

- None.
- Diode with no dynamics — Select this option to prioritize simulation speed using the Diode block.
- Diode with charge dynamics — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation model of the Diode block.

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Devices** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Forward voltage – Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### On resistance – Resistance

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

**Off conductance – Conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Junction capacitance – Capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Peak reverse current,  $i_{RM}$  – Current**

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Initial forward current when measuring  $i_{RM}$  – Current**

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Rate of change of current when measuring  $i_{RM}$  – Current change rate**

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery time parameterization – Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select **Specify stretch factor** or **Specify reverse recovery charge**, you can specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time, trr**. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery time, trr – Time

15 us (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, iRM** parameter divided by the value of the **Rate of change of current when measuring iRM** parameter.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Reverse recovery charge, Qrr – Charge

1500 s\*uA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

#### Dependencies

See the Protection Diode Parameter Dependencies table.

#### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.



## Snubbers Parameter Dependencies

Snubbers Parameter Dependencies	
<b>Snubber</b>	
None	<i>RC Snubber</i>
	Snubber resistance
	Snubber capacitance

### Snubber – Snubber model

None (default) | RC snubber

Switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber resistance – Resistance

0.1 Ohm (default) | scalar

Resistance of the switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber capacitance – Capacitance

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

## Version History

Introduced in R2018a

## References

[1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics, 2nd Edition*. Hoboken, NJ: John Wiley & Sons Inc., 2010.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

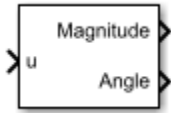
**See Also**

Average-Value Chopper | Four-Pulse Gate Multiplexer | One-Quadrant Chopper | Two-Quadrant Chopper

# Fourier Analysis

Discrete or continuous time Fourier analysis

**Library:** Simscape / Electrical / Control / General Control



## Description

The Fourier Analysis block performs a Fourier analysis on the input signal in either discrete or continuous time.

## Equations

A periodic function  $x(t)$  can be decomposed to an infinite sum of sine and cosine functions as

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nt) + b_n \sin(nt)]$$

where:

- $a_0$  is the DC component.
- $a_n$  and  $b_n$  are constant Fourier coefficients.
- $n$  is the harmonic number.

The coefficients  $a_n$  and  $b_n$  are defined as

$$a_n = \frac{2}{T_0} \int_{t_0}^{t_0 + T_0} x(t) \cos(n\Omega_0 t) dt, \quad n = 0, 1, \dots$$

$$b_n = \frac{2}{T_0} \int_{t_0}^{t_0 + T_0} x(t) \sin(n\Omega_0 t) dt, \quad n = 1, 2, \dots$$

$$\Omega_0 = \frac{2\pi}{T_0}$$

$$T_0 = \frac{1}{f}$$

where  $f$  is the fundamental frequency.

The magnitude and angle corresponding to the harmonic number are defined as:

$$|X_n| = \sqrt{a_n^2 + b_n^2}$$

$$\theta_n = -\tan^{-1} \left[ \frac{b_n}{a_n} \right]$$

---

**Note** If you use this block for continuous-time operations and set the **Sample time (-1 for inherited)** parameter to 0, you should also specify the value of the **Buffer size** parameter to ensure it covers the moving time window.

---

## Ports

### Input

#### **u — Fourier analysis input**

scalar | vector

Input signal to be analyzed. The input can be a single signal or multiple multiplexed signals. Input signals can be AC currents or voltages in an electrical system.

Data Types: single | double

### Output

#### **Magnitude — Signal magnitude**

scalar | vector

Signal magnitude corresponding to the harmonic number.

Data Types: single | double

#### **Angle — Signal angle**

scalar | vector

Signal angle corresponding to the harmonic number.

Data Types: single | double

## Parameters

#### **Fundamental frequency (Hz) — Signal fundamental frequency**

60 (default) | positive scalar | vector with positive values

Fundamental frequency of the signal, in Hz. If you specify the fundamental frequency using a vector, it must match the input vector dimensions.

#### **Harmonic numbers — Signal harmonic numbers**

[1 2] (default) | vector with elements  $\geq 0$

Specify signal harmonic numbers. Vector elements must be greater than or equal to 0.

#### **Initial magnitude — Initial signal magnitude**

1 (default) | vector with elements  $\geq 0$

Specify the initial magnitude of the signal. Vector elements must be greater than or equal to 0.

#### **Initial phase (rad) — Initial signal phase**

0 (default) | vector

Initial phase angle of the signal, in rad.

**Buffer size – Buffer size**

8192 (default) | scalar

Size of the buffer.

**Dependencies**To enable this parameter, set **Sample time (-1 for inherited)** to 0.**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2018b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

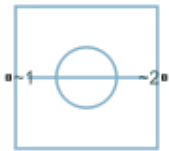
**Blocks**

Moving Average

## Frequency-Dependent Overhead Line (Three-Phase)

Three-phase overhead line which includes effects that vary as a function of frequency

**Library:** Simscape / Electrical / Passive / Lines

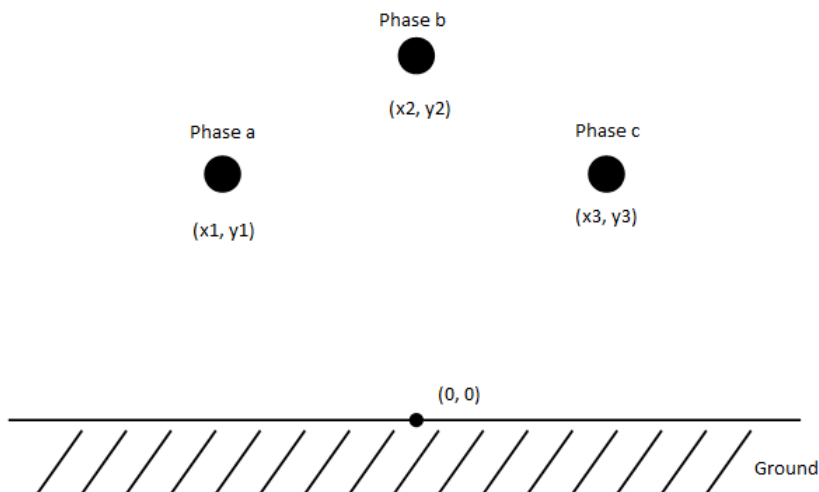


### Description

The Frequency-Dependent Overhead Line (Three-Phase) block represents a high-fidelity frequency-dependent overhead line that offers accurate transient response simulation from 0.01 Hz to 100 kHz.

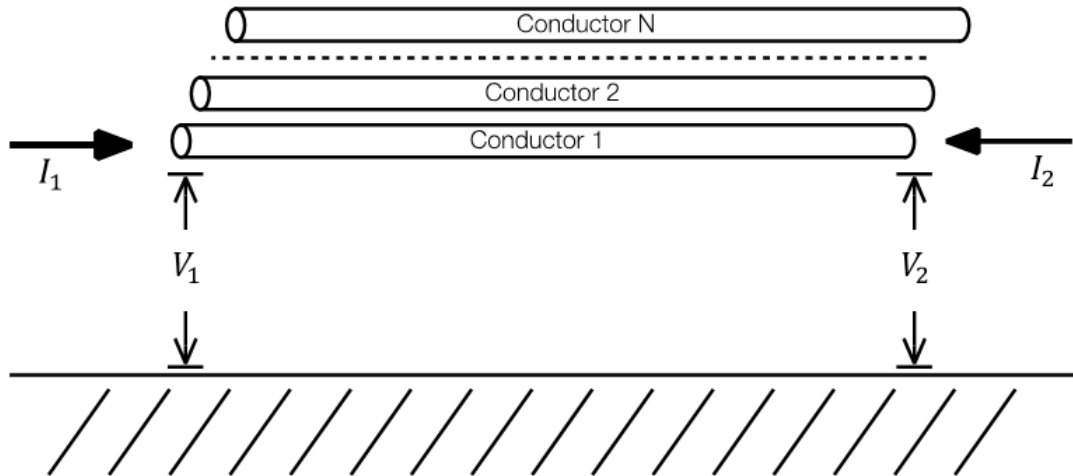
The block computes frequency-dependent impedance and admittance matrices. The calculations also depend on the loss, inductance, and capacitance of the return path.

For more information on using a frequency-dependent overhead line (three-phase), see “Engineering Applications” on page 1-636.



## Equations

### Line Model



The electromagnetic behavior of a multiconductor transmission line is described by the telegrapher's equation.

$$-\frac{dV}{dx} = ZI$$

$$-\frac{dI}{dx} = YV$$

Where:

- $V$  is the vector of line phase voltages.
- $I$  is the vector of phase currents.
- $Z$  is the series impedance matrix in per unit length.
- $Y$  is the shunt admittance matrix in per unit length.

These are general solutions for the vector of currents and voltages.

$$I(x) = e^{-\Psi x}C_1 + e^{\Psi x}C_2 \quad (1-26)$$

$$Y_c V(x) = e^{-\Psi x}C_1 - e^{\Psi x}C_2 \quad (1-27)$$

Where  $\Psi = \sqrt{YZ}$  is the propagation matrix and  $Y_c = (\sqrt{YZ})^{-1}Y$  is the characteristic admittance.

Consider now a transmission line segment of length  $x = l$ . At the beginning of one of its ends, when  $x = 0$ , the equations 1 and 2 are evaluated as follows.

$$I_1 = C_1 + C_2 \quad (1-28)$$

$$Y_c V_1 = C_1 - C_2 \quad (1-29)$$

The integration constant vectors  $C_1$  and  $C_2$  can be expressed in terms of  $I_0$  and  $V_0$ .

$$C_1 = \frac{(I_1 + Y_c V_1)}{2} \quad (1-30)$$

$$C_2 = \frac{(I_1 - Y_c V_1)}{2} \quad (1-31)$$

Similarly, at  $x = l$ , equations 1 and 2 are evaluated as follows.

$$I_2 = e^{-\Psi l} C_1 + e^{\Psi l} C_2 \quad (1-32)$$

$$Y_c V_2 = e^{-\Psi l} C_1 - e^{\Psi l} C_2 \quad (1-33)$$

To obtain the two fundamental equations for the transmission line model, first you use equations 7 and 8 to perform this calculation.

$$I_2 - Y_c V_2 = -2e^{-\Psi l} C_1 \quad (1-34)$$

Then you substitute equation 5 into equation 9.

$$I_2 - Y_c V_2 = -H(I_1 + Y_c V_1) \quad (1-35)$$

Where  $H = e^{-\Psi l}$  is the propagation factor matrix. Equation 10 establishes the relation between voltages and currents at the terminals of a multi-conductor line section.

You can obtain a companion expression providing the model for the terminal 1 at  $x = 0$ .

$$I_1 - Y_c V_1 = -H(I_2 + Y_c V_2) \quad (1-36)$$

Define:

- $I_{sh,1} = Y_c V_1$  — Shunt current vector produced at terminal 1 by injected voltages  $V_1$
- $I_{sh,2} = Y_c V_2$  — Shunt current vector produced at terminal 2 by injected voltages  $V_2$
- $I_{rfl,1} = \frac{1}{2}(I_1 + Y_c V_1)$  — Reflected currents of terminal 1
- $I_{rfl,2} = \frac{1}{2}(I_2 + Y_c V_2)$  — Reflected currents of terminal 2

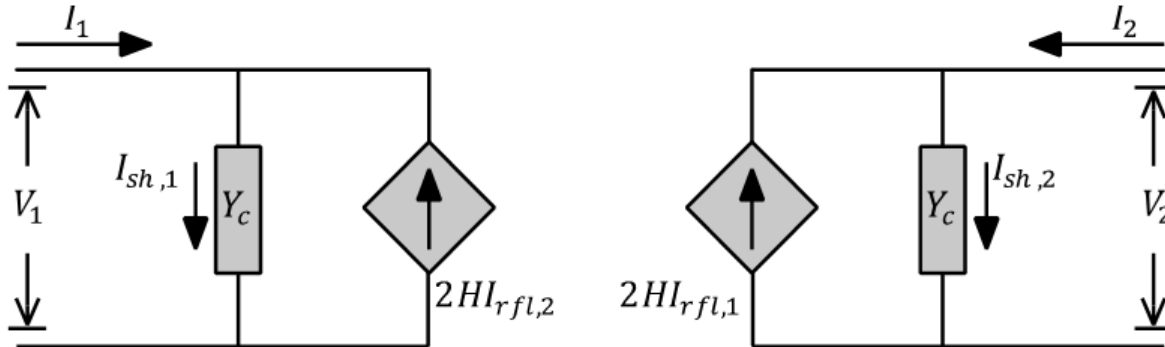
And finally rewrite equations 10 and 11 as follows:

$$I_1 = I_{sh,1} - 2H I_{rfl,2}$$

$$I_2 = I_{sh,2} - 2H I_{rfl,1}$$

These equations constitute a traveling wave line model for the segment of length  $L$ .





Particularly for transmission lines with ground return, the parameters are highly dependent on frequency. Model solutions are then carried out directly in the phase domain. The block computes automatically the impedance and the characteristic admittance matrices on the full frequency range, performing an approximation through rational fitting.

Rational fitting for this model is performed by using the vector fitting (VF) procedure. For more information on the phase domain line model and the state-space analysis, see *Wide-Band Line Model Implementation in Matlab for EMT Analysis* [1].

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Ports

#### Conserving

##### ~1 — Three-phase AC voltage 1

electrical

Expandable electrical conserving three-phase port associated with AC voltage 1.

##### ~2 — Three-phase AC voltage 2

electrical

Expandable electrical conserving three-phase port associated with AC voltage 2.

### Parameters

#### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**X position of the line – Horizontal line position**

`[-6, 0, 6] m (default) | vector of three elements`

Position of the line in terms of the x-axis of the reference frame.

**Y position of the line – Vertical line position**

`[16, 26, 16] m (default) | vector of three positive elements`

Position of the line in terms of the y-axis of the reference frame. The elements of the vector need to be greater than 0.

**Resistivity of the conductor – Conductor resistivity**

`2.7e-8 m*Ohm (default) | positive scalar`

Conductor resistivity. The value must be greater than 0.

**Resistivity of the earth – Earth return resistivity**

`100 m*Ohm (default) | positive scalar`

Earth return resistivity. The value must be greater than 0.

**Radius of the conductor – Conductor radius**

`0.017 m (default) | positive scalar`

Conductor radius. The value must be greater than 0.

**Length of the line – Line length**

`150e3 m (default) | positive scalar`

Line length. The value must be greater or equal than 1 and less or equal than 3e5.

**Ground configuration – Ground node accessibility**

`Internally grounded (default) | Accessible ground nodes`

Configuration of the ground nodes.

**More About****Engineering Applications**

Three-phase frequency-dependent overhead lines are typically used in:

- Simulations of transient phenomena in electrical power systems
- Design and operation of power systems and power apparatuses

## Version History

Introduced in R2019a

## References

- [1] Ramos-Leanos, O, Iracheta R.. *Wide-Band Line Model Implementation in Matlab for EMT Analysis*. Arlington, TX: IEEE North American Power Symposium (NAPS), 2010.
- [2] Ramos-Leanos, O. *Wideband Line/Cable Models for Real-Time and Off-Line Simulations of Electromagnetic Transients*. Diss. École Polytechnique de Montréal, 2013.
- [3] Ramos-Leanos, O., J. L. Naredo, J. Mahseredjian, I. Kocar, C. Dufour, and J. A. Gutierrez-Robles. *A wideband line/cable model for real-time simulations of power system transients*. IEEE Transactions on Power Delivery, 27.4 (2012): 2211-2218.
- [4] Iracheta, R., and O. Ramos-Leanos. *Improving computational efficiency of FD line model for real-time simulation of EMTS*. Arlington, TX: IEEE North American Power Symposium (NAPS), 2010.
- [5] Dommel, H. W. *Electromagnetic transients program (EMTP) theory book*. Portland OR: Bonneville Power Administration, 1986

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

AC Cable (Three-Phase) | Transmission Line (Three-Phase) | Transmission Line | Coupled Lines (Three-Phase)

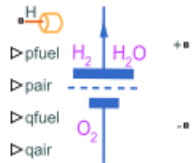
### Topics

“Expand and Collapse Three-Phase Ports on a Block”

# Fuel Cell

Fuel cell electrical system

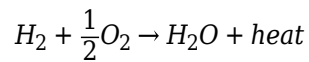
**Library:** Simscape / Electrical / Sources



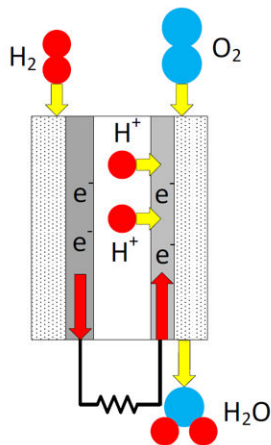
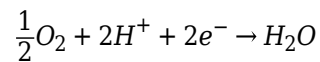
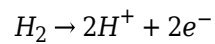
## Description

The Fuel Cell block models a fuel cell that converts the chemical energy of hydrogen into electrical energy.

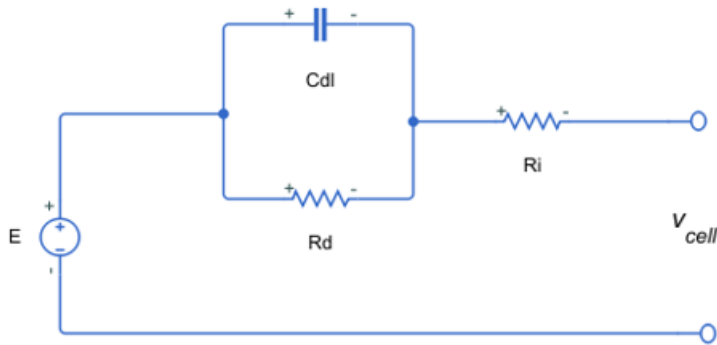
This chemical reaction defines the electrical conversion:



which occurs due to these anode and cathode half-reactions:



A fuel cell stack comprises several series-connected fuel cells. This figure shows the equivalent circuit of a single fuel cell that this block uses



where:

- $V_{cell}$  is the cell voltage.
- $R_i$  is the **Internal resistance**.
- $R_d$  is the **Sum of activation and concentration resistances**.
- $C_{dl}$  is the parallel RC capacitance that accounts for time dynamics in the cell.

### Equations

You can use the **Model fidelity** parameter to set the Fuel Cell block to two different levels of fidelity:

- **Simplified - nominal conditions** — The block calculates the Nernst voltage at the nominal condition of temperature and pressure.
- **Detailed with physical inputs** — The block calculates the Nernst voltage by considering the pressures and flow rates of fuel and air.

### Simplified Electrical Model

When **Model fidelity** is set to **Simplified - nominal conditions**, the Fuel Cell block calculates the Nernst voltage,  $E$ , at the nominal condition of temperature and pressure, as defined by these equations:

$$E = E_{oc} - NA \ln \left( \frac{i_{FC}}{i_0} \right)$$

$$v_{FC} = N_{unit} E - R_i i_{FC} - v_d$$

$$\frac{1}{R_d} \left( \tau \frac{dv_d}{dt} + v_d \right) = i_{FC}$$

where:

- $E_{oc}$  is the **Open-circuit voltage**.
- $N$  is the **Number of cells per module**.
- $i_{FC}$  is the current that the fuel cell generates.
- $v_{FC}$  is the voltage across the fuel cell terminals.
- $N_{unit}$  is the **Module units (Series)**.
- $v_d$  is the voltage drop that accounts for fuel cell dynamics.

- $A$  is the **Tafel slope**, in volts.
- $i_0$  is the **Nominal exchange current**.
- $\tau = \frac{C_{dl}}{R_d}$ .

**Detailed Electrical Model**

When **Model fidelity** is set to Detailed with physical inputs, the Fuel Cell block calculates the Nernst voltage,  $E$ , by considering the pressures and flow rates of fuel and air.

These equations determine the rates of utilization of hydrogen,  $U_{H_2}$ , and oxygen,  $U_{O_2}$

$$U_{H_2} = \frac{60000Ni_{FC}V_e}{p_{fuel}q_{fuel}x_{H_2}}$$

$$U_{O_2} = \frac{60000Ni_{FC}V_e}{2p_{air}q_{air}x_{O_2}}$$

where:

- $V_e$  is the thermal voltage at room temperature.
- $p_{fuel}$  is the supply pressure of the fuel, in bar.
- $q_{fuel}$  is the fuel flow rate.
- $x_{H_2}$  is the concentration of hydrogen in the fuel, in percent.
- $p_{air}$  is the supply pressure of air, in bar.
- $q_{air}$  is the air flow rate.
- $x_{O_2}$  is the concentration of oxygen in the air, in percent.

These equations define the partial pressures:

$$p_{H_2} = p_{fuel}x_{H_2} - U_{H_2}$$

$$p_{O_2} = p_{air}x_{O_2} - U_{O_2}$$

$$p_{H_2O} = p_{air}x_{H_2O} - 2U_{O_2}$$

where  $x_{H_2O}$  is the concentration of vapor in air, in percent.

Then, the block computes the Nernst voltage as

$$E = \begin{cases} \left[ \begin{array}{c} \text{Reactant} \\ K_c N \left( 1.229 + K_z \frac{RT}{ZF} \ln \left( \frac{p_{H_2} p_{O_2}^{\frac{1}{2}}}{p_{H_2O}} \right) \right) \end{array} \right] - E_{Tafel} - E_{conc}, T < 373.15 K \\ \left[ \begin{array}{c} \text{Activation} \\ K_c N \left( 1.229 + K_z \frac{RT}{ZF} \ln \left( \frac{p_{H_2} p_{O_2}^{\frac{1}{2}}}{p_{H_2O}} \right) \right) \end{array} \right] - E_{Tafel} - E_{conc}, T \geq 373.15 K \end{cases}$$

where:

- $$K_z = \frac{\frac{E_{oc\_adm}}{K_c N} - 1.229}{(T_{nom} - 298) \frac{-44.43}{z_0 F} + \frac{RT_{nom}}{z_0 F} \ln \left( p_{nH_2} p_{nO_2}^{\frac{1}{2}} \right)}$$
- $E_{Tafel} = N A_T \ln \left( \frac{i_{FC}}{i_0} \right)$  is the electrokinetic term for the activation.
- $E_{conc} = V_e \frac{T}{298} \ln \left( \frac{i_{lim}}{i_{lim} - i_{FC}} \right)$  is the electrokinetic term for the concentration.
- $$E_{oc\_adm} = \max \left( E_{oc}, N \left( 1.229 + \frac{RT_{nom}}{2F} \ln \left( p_{nH_2} p_{nO_2}^{\frac{1}{2}} \right) \right) \right)$$
- $K_c$  is the voltage constant at nominal condition of operation.
- $T$  is the operating temperature of the fuel cell.
- $T_{nom}$  is the value of the **Nominal temperature** parameter.
- $z$  is the number of moving electrons per second.
- $z_0$  is the number of moving electrons per second at the value of the **Nominal exchange current** parameter.
- $F$  is the Faraday constant.
- $R$  is the universal gas constant.
- $p_{nH_2}$  is the nominal hydrogen pressure, in bar.
- $p_{nO_2}$  is the nominal oxygen pressure, in bar.
- $A_T = \frac{AT}{\ln(10) * 298}$  is the Tafel slope as a function of the temperature.
- $i_{lim}$  is the value of the **Collapse current** parameter.
- The voltage 1.229 represents the standard cell potential for the Nernst equation.

The block computes the power dissipated, or the heat released in the fuel cell, by using this equation

$$P_{dissipated} = Z_{e\_agg}(T\Delta S) + R_i i_{FC} + v_d^2 \frac{1}{R_d}$$

where:

- $Z_{e\_agg} = \frac{(N_{unit}E - R_i i_{FC} - v_d) i_{FC}}{2\Delta G}$  is the total electron circulation rate, in mol/s.
- $T\Delta S$  is the change in entropy of the fuel cell reaction, in kJ/(mol\*K), at the operating temperature of the fuel cell.
- $\Delta G$  is the change in Gibbs free energy of the full fuel cell reaction, in kJ/mol, at the operating temperature of the fuel cell.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Limitations

The Fuel Cell block does not allow electrolysis.

## Ports

### Input

#### **pfuel** — Supply absolute pressure of fuel, bar

physical signal

Physical signal port associated with the supply absolute pressure of the fuel, in bar, specified as a physical signal.

#### Dependencies

To enable this port, set **Model fidelity** to Detailed with physical inputs.

#### **pair** — Supply gauge pressure of air, bar

physical signal

Physical signal port associated with the supply gauge pressure of air, in bar, specified as a physical signal.

#### Dependencies

To enable this port, set **Model fidelity** to Detailed with physical inputs.

#### **qfuel** — Fuel flow rate, l/min

physical signal

Physical signal port associated with the fuel flow rate, specified as a physical signal.

#### Dependencies

To enable this port, set **Model fidelity** to Detailed with physical inputs.

#### **qair** — Air flow rate, l/min

physical signal

Physical signal port associated with the air flow rate, specified as a physical signal.

#### Dependencies

To enable this port, set **Model fidelity** to Detailed with physical inputs.

## Conserving

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the fuel cell positive terminal.



**- — Negative terminal**

electrical

Electrical conserving port associated with the fuel cell negative terminal.

**H — Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Model fidelity** to Detailed with physical inputs.

**Parameters****Main****Model fidelity — Fuel cell fidelity**

Detailed with physical inputs (default) | Simplified - nominal conditions

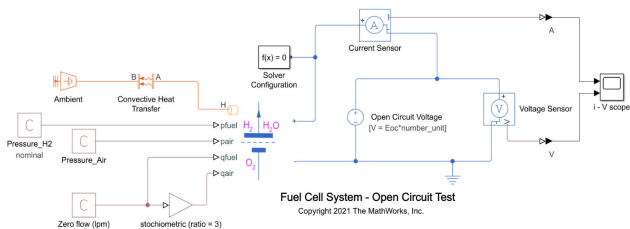
Level of fidelity of the fuel cell model.

**Open-circuit voltage — Open-circuit voltage**

65 V (default) | positive scalar

Open-circuit voltage.

When the amount of flow is very low or close to zero and the values of the fuel and air pressures are equal to their nominal values, the output voltage of the fuel cell is equal to the open-circuit voltage multiplied by the number of module units. The current that flows out of the fuel cell is negligible. This figure shows how to set the open-circuit test:

**Tafel slope — Tafel slope**

0.23 V (default) | positive scalar

Amount of overpotential required to increase the reaction rate by a factor ten.

**Internal resistance — Internal resistance**

0.05 Ohm (default) | positive scalar

Internal resistance.

**Nominal exchange current — Nominal exchange current**

80 A (default) | positive scalar

Exchange current at the nominal temperature.

At the nominal exchange current, the fuel cell leaves the region of activation polarization and enters the region of ohmic polarization.

**Collapse current — Collapse current**

200 A (default) | positive scalar

Value of current at which the fuel cell voltage becomes zero. As the fuel cell enters the region of concentration polarization and the current continues to increase, the voltage starts decreasing at a faster rate.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Number of cells per module — Number of cells per module**

65 (default) | positive scalar

Number of cells per module.

The cell count value in this block is consistent with the fuel cell delivering its maximum output power for specified set points of flow and H<sub>2</sub> pressure.

**Module units (Series) — Stack of module units in series**

10 (default) | positive scalar

Stack of module units in series.

Stack module in series to scale up voltage. For example, 10 module units in series with an open-circuit voltage of 65 V deliver a voltage of 650V.

**Supply**

To enable these parameters, in the **Main** tab, set **Model fidelity** to Detailed with physical inputs.

**Nominal H<sub>2</sub> pressure — Nominal hydrogen pressure**

1.5e5 Pa (default) | positive scalar

Hydrogen gauge pressure at nominal temperature.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Nominal O<sub>2</sub> pressure — Nominal oxygen pressure**

1.0e5 Pa (default) | positive scalar

Oxygen gauge pressure at nominal temperature.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Concentration H<sub>2</sub> in fuel (%) — Hydrogen concentration in fuel**

99 (default) | integer between 0 and 100

Molar fraction concentration of hydrogen in fuel, in percent.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Concentration O<sub>2</sub> in air (%) – Oxygen concentration in fuel**

21 (default) | integer between 0 and 100

Molar fraction concentration of oxygen in fuel, in percent.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Concentration vapor in air (%) – Vapor concentration in air**

1 (default) | integer between 0 and 100

Molar fraction concentration of vapor in air, in percent.

**Dependencies**

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

**Dynamics****Model activation delay – Activation delay modeling option**

No (default) | Yes

Whether to model the activation delay of the fuel cell.

**Sum of activation and concentration resistance – Sum of activation and concentration resistance**

0.005 Ohm (default) | positive scalar

Sum of the activation resistance and concentration resistance.

**Dependencies**

To enable this parameter, set **Model activation delay** to Yes.

**Time constant – Time constant**

10 s (default) | positive scalar

Time constant.

**Dependencies**

To enable this parameter, set **Model activation delay** to Yes.

**Thermal**

To enable these parameters, in the **Main** tab, set **Model fidelity** to Detailed with physical inputs.

**Nominal temperature – Nominal temperature**

293.15 K (default) | scalar

Temperature at which the nominal parameters are measured.

### Dependencies

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

### Thermal mass — Thermal mass associated with thermal port H

30000 J/K (default) | positive scalar

Thermal mass associated with the thermal port, **H**. This value represents the energy required to raise the temperature of the thermal port by one degree.

### Dependencies

To enable this parameter, set **Model fidelity** to Detailed with physical inputs.

## Version History

Introduced in R2021a

## References

- [1] Do, T.C., et al. "Energy Management Strategy of a PEM Fuel Cell Excavator with a Supercapacitor/Battery Hybrid Power Source". *Energies* 12, no. 22, (November 2019). DOI.org (Crossref), doi:10.3390/en13010136.
- [2] Motapon, Souleman N., O. Tremblay and L. Dessaint, "A generic fuel cell model for the simulation of fuel cell vehicles." *2009 IEEE Vehicle Power and Propulsion Conference*, Dearborn, MI, 2009, pp. 1722-1729, doi: 10.1109/VPPC.2009.5289692
- [3] Hirschenhofer, J. H., D.B. Stauffer, R.R. Engleman, and M.G. Klett. *Fuel Cell Handbook (4th Ed)*. U.S. Department of Energy Office of Fossil Energy, 1988.
- [4] Larminie, James, and Andrew Dicks. *Fuel Cell Systems Explained*. West Sussex, England: John Wiley & Sons, Ltd., 2003. <https://doi.org/10.1002/9781118878330>.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Solar Cell | Electrolyzer

# Fully Differential Op-Amp

Operational amplifier with fully differential output, that is, not referenced to ground

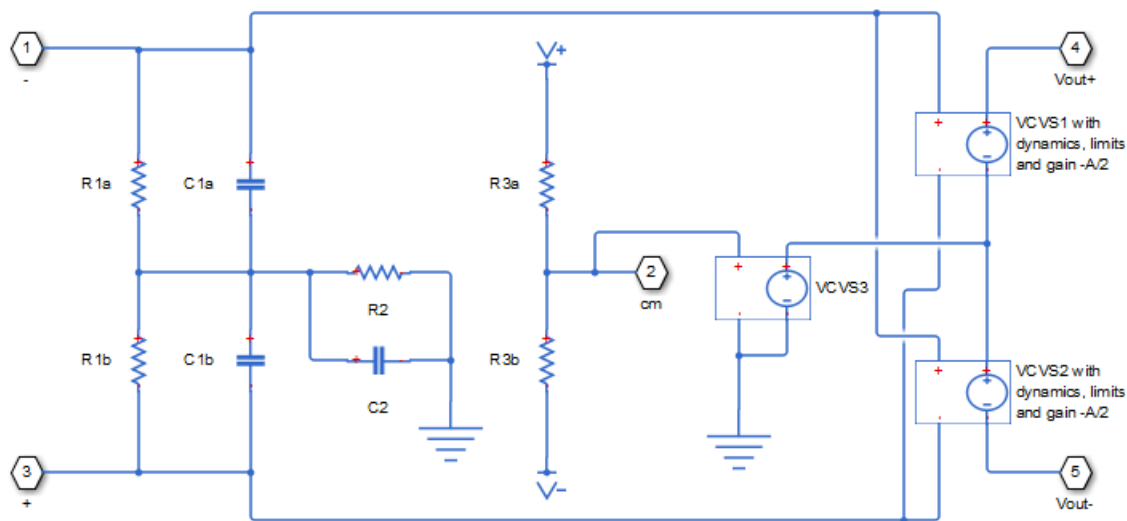
**Library:** Simscape / Electrical / Integrated Circuits



## Description

The Fully Differential Op-Amp block models a fully differential operational amplifier. Differential signal transmission is better than single-ended transmission due to reduced susceptibility to external noise sources. Applications include data acquisition where inputs are differential, for example, sigma-delta converters.

The following diagram shows the internal representation of the amplifier.



Parameters for the circuit components are derived from the block parameters that you provide. The gain of the two voltage-controlled voltage sources (VCVS1 and VCVS2) is set to half of the differential gain value. Similarly the slew rate of each of the voltage sources is set to half of the differential maximum slew rate value. The voltages of the two output ports  $V_{out+}$  and  $V_{out-}$  are both limited to be within the minimum and maximum output voltages that you specify.

The output voltage for zero differential input voltage is controlled by the common-mode port,  $cm$ . If no current is drawn from the  $cm$  port by the external circuit, then the output voltage is set to be the average of the positive and negative supply voltages by the resistor ladder of  $R_{3a}$  and  $R_{3b}$ . Note that the negative supply voltage can be zero, which corresponds to operation when a split supply is not available. The values for the minimum and maximum output voltages that you provide must be consistent with the values for the supply voltages that you provide. So, for example, the maximum output high voltage will be less than the positive supply voltage, the difference corresponding to the number of p-n junction voltage drops in the circuit.

## Assumptions and Limitations

- This 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.

## Ports

### Conserving

#### + – Non-inverting input

electrical

Electrical conserving port associated with the op-amp non-inverting input.

#### - – Inverting input

electrical

Electrical conserving port associated with the op-amp inverting input.

#### cm – Common-mode port

electrical

Electrical conserving port associated with the op-amp common-mode port. If you want to leave this pin open-circuit, connect a voltage sensor between the cm port and a reference port.

#### Vout+ – Non-inverting output

electrical

Electrical conserving port associated with the op-amp non-inverting output.

#### Vout- – Inverting output

electrical

Electrical conserving port associated with the op-amp inverting output.

## Parameters

### Gain

#### Differential gain – Differential gain

1e3 (default)

The gain applied to a voltage difference between the + and - inputs.

#### Bandwidth – Bandwidth

1.5 GHz (default)

The frequency at which the differential voltage gain drops by 3 dB from its dc value.

## Input Impedance

### Differential input resistance – Differential input resistance

1.3 M $\Omega$  (default)

The input resistance seen by a voltage source applied across the + and - inputs. The default value is 1.3 M $\Omega$ .

### Differential input capacitance – Differential input capacitance

1.8 pF (default)

The input capacitance seen by a current source applied across the + and - inputs.

### Common-mode input resistance – Common-mode input resistance

1.0 M $\Omega$  (default)

The input resistance seen by a voltage source applied between ground and the + input, or between ground and the - input.

### Common-mode input capacitance – Common-mode input capacitance

2.3 pF (default)

The input capacitance seen by a current source applied between ground and the + input, or between ground and the - input.

## Output Limits

### Output resistance – Output resistance

1  $\Omega$  (default)

The output resistance of either of the outputs with respect to the common-mode voltage reference. Differential output resistance is therefore twice the value of the output resistance  $R_{out}$ .

### Minimum output voltage low – Minimum output voltage low

-1.4 V (default)

The minimum output voltage for either of the two output pins with respect to ground.

### Maximum output voltage low – Maximum output voltage low

1.4 V (default)

The maximum output voltage for either of the two output pins with respect to ground.

### Differential maximum slew rate – Differential maximum slew rate

5000 V/ $\mu$ s (default)

The maximum slew rate of the differential output voltage.

## Output Bias

### Common-mode port input resistance – Common-mode port input resistance

23.0 k $\Omega$  (default)

The input resistance seen by a voltage source applied between ground and the common mode port.

**Negative supply voltage — Negative supply voltage**

-5 V (default)

The value of the negative supply voltage connected to common-mode bias resistor R3b (see diagram).

**Positive supply voltage — Positive supply voltage**

5 V (default)

The value of the positive supply voltage connected to common-mode bias resistor R3a (see diagram).

**Initial Conditions****Initial differential output voltage — Initial differential output voltage**

0 V (default)

The initial differential voltage across the two outputs if the output current is zero.

## Version History

Introduced in R2012a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Op-Amp | Band-Limited Op-Amp | Finite-Gain Op-Amp



# Fuse

Fuse that protects against excessive current

**Library:** Simscape / Electrical / Switches & Breakers



## Description

The Fuse block breaks the circuit, to protect it against excessive current. The block provides the following modeling options:

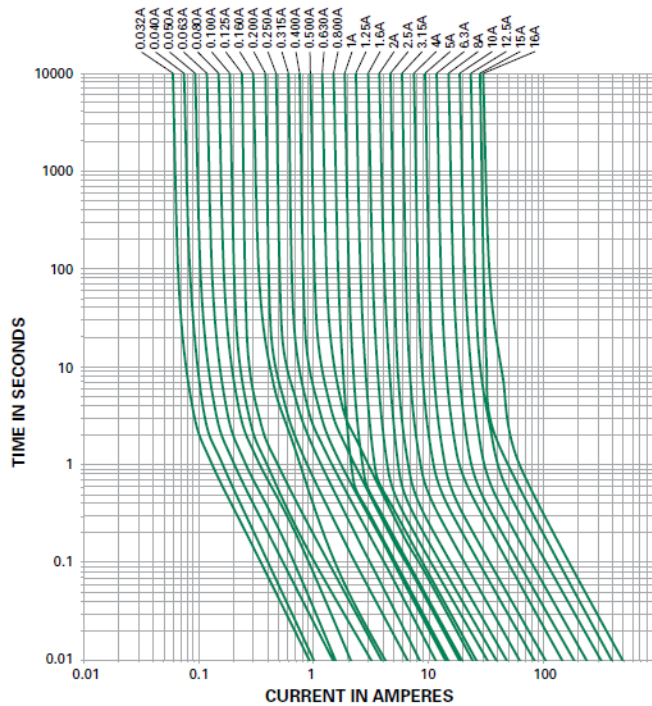
- **Opening time is independent of current** — The fuse opens when the current through the fuse exceeds the threshold current continuously for longer than the specified amount of time. This option assumes that the time to open is independent of the current. It also implicitly assumes that heat in the fuse is instantaneously dissipated once the current drops below a certain threshold.
- **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. If you select this option, the fuse blows when:

$$\int_{t_{\text{over}}}^t i^2 dt \geq I^2 t$$

where  $t_{\text{over}}$  is the time when the current crosses over the threshold current,  $i$  is instantaneous current through the fuse, and  $I^2 t$  is the nominal melting value from the datasheet. 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.

For example, these are the time-current curves from a Littelfuse® 218 Series fuse datasheet.



Select the appropriate curve and enter the corresponding current and time values into the **Tabulated currents** and **Tabulated opening time** parameter vectors, respectively. Use this option for small currents.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the fuse positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the fuse negative terminal.

### Output

#### x – Fuse state

physical signal

Outputs the fuse state as a unitless physical signal. The port outputs 0 if the fuse is intact, and 1 if the fuse is blown.

### Dependencies

Exposed when the **PS output for fuse state** parameter is set to Visible.

## Parameters

### Parameterization — Model dependence of opening time on current

Opening time is independent of current (default) | Opening time depends on  $I^2 \cdot t$  | Opening time is tabulated

Select whether opening time depends on current:

- **Opening time is independent of current** — The fuse opens when the current through the fuse exceeds the threshold current continuously for longer than the specified amount of time.
- **Opening time depends on  $I^2 \cdot t$**  — Makes fuse opening dependent on the energy dissipated in the fuse. Use this option for high-current range of operation.
- **Opening time is tabulated** — Makes fuse opening dependent on the average time-current tabulated data. Use this option for small currents.

### Rated current, $I_{\text{rated}}$ — Nominal operating current

1 A (default)

The fuse rating from the datasheet, or nominal operating current.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Opening time is independent of current or Opening time depends on  $I^2 \cdot t$ .

### Ratio of minimum melting current to rated current, $I_{\text{melt}}/I_{\text{rated}}$ — Defined threshold current for melting

1 (default)

Minimum melting current defines the fuse operating threshold. If the current through the fuse does not exceed this threshold, the fuse stays in low-ohmic state. With the default parameter value, 1, the threshold current is the same as the rated current. In most cases, this parameter value should be greater than 1. You can also set it to a value less than 1, for situations where the fuse may blow at a current level below the rated current (for example, due to high temperature, or due to repetitive pulses). The parameter value must be greater than 0.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Opening time is independent of current or Opening time depends on  $I^2 \cdot t$ .

### Time to fuse — Time needed for the fuse to blow

0 s (default)

The amount of time needed for the fuse to blow. If the current through the fuse exceeds the threshold current continuously for longer than this amount of time, the fuse enters the high-ohmic state and breaks the circuit. If the current falls below the threshold before the time elapsed, the fuse stays in the low-ohmic state. If the current exceeds the threshold again, the time counter restarts. With the default parameter value, 0, the fuse blows immediately as soon as the current exceeds the minimum melting current. The parameter value must be greater than or equal to 0.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Opening time is independent of current.

**Nominal melting,  $I^2t$  — Dissipated energy needed for the fuse to blow**

1 s\*A<sup>2</sup> (default)

The amount of energy dissipated in the fuse that is needed for the fuse to blow. This is the nominal melting, or  $I^2t$ , parameter from the datasheet. The value must be greater than or equal to 0.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Opening time depends on  $I^2t$ .

**Tabulated currents — Currents data from the time-current curves**

[1.9, 2, 2.5, 30] A (default)

The tabulated current values from a time-current curve on the manufacturer datasheet. Every element in this vector must be greater than 0, and the vector must be strictly ascending.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Opening time is tabulated.

**Tabulated opening times — Times data from the time-current curves**

[10000, 1, 0.3, 0.001] s (default)

The tabulated time values from a time-current curve on the manufacturer datasheet. Every element in this vector must be greater than or equal to 0, and the vector must be strictly descending. The vector must be of the same size as the **Tabulated currents** vector.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Opening time is tabulated.

**Fuse resistance R — Resistance of the fuse**

0.01  $\Omega$  (default)

The fuse resistance in low-ohmic state. The parameter value must be greater than 0.

**Open-circuit conductance G — Open-circuit fuse conductance**

1e-8 1/ $\Omega$  (default)

The open-circuit fuse conductance in high-ohmic state, when the fuse has blown. The parameter value must be greater than 0.

**PS output for fuse state — Controls the visibility of the output port for fuse state**

Hidden (default) | Visible

When set to Visible, a PS output port **x** appears on the block icon. The port outputs 0 if the fuse is intact, and 1 if the fuse is blown.

## Version History

Introduced in R2008b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

# Gate Driver

Behavioral model of gate driver integrated circuit

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Gate Driver block provides an abstracted representation of a gate driver integrated circuit. The block models input hysteresis, propagation delay, and turn-on/turn-off dynamics. Unless modeling a gate driver circuit explicitly, always use this block or the Half-Bridge Driver block 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.

You can model electrical or physical signal input ports by setting the **Modeling option** parameter to either:

- **PS input** — The driver output state is controlled by a physical signal input  $u$ . Use this modeling option if all of your controller, including PWM waveform generation, is determined by Simulink blocks. This modeling option is the default.
- **Electrical input ports** — The driver output state is controlled by two electrical input connections, PWM and REF. Use this option if your model has upstream analog components, such as the Controlled PWM Voltage source.

When the input rises above the logic 1 input level, the transition of the output state from off to on is initiated after a delay equal to the turn-on propagation delay. The demanded output voltage across the **G** and **S** ports steps in value from the off-state output voltage to the on-state output voltage, but the actual output voltage is set by the RC time constant associated with the **On-state gate drive resistance** value and the total load capacitance. Similarly, when the input falls below the logic 0 input value, the transition of the output state from on to off is initiated after a delay equal to the turn-off propagation delay and with dynamics now set by the **Off-state gate drive resistance** value.

## Faults

You can insert a fault into the output of the gate driver at a specified simulation time, to make the connected semiconductor device either permanently off or permanently on. Use this feature to represent a failed semiconductor device as failed at open-circuit or at normal on-state conditions.

## Ports

### Input

**u — Control signal, unitless**

physical signal

Input physical signal that specifies the input control value.

**Dependencies**

To enable this port, set **Modeling option** to PS input.

**Conserving****PWM — Pulse-width modulated signal**

electrical

Electrical conserving port associated with the pulse-width modulated signal.

**Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

**REF — Floating zero volt reference**

electrical

Electrical conserving port associated with the floating zero volt reference.

**Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

**G — Gate connection**

electrical

Electrical conserving port associated with the gate. Connect this port to the gate of a MOSFET or IGBT block.

**S — Source or emitter connection**

electrical

Electrical conserving port associated with the source or emitter. Connect this port to the source of a MOSFET block or the emitter of an IGBT block.

**Parameters****Modeling option — Whether to model electrical or physical signal input ports**

PS input (default) | Electrical input ports

Whether to model physical signal or electrical input ports.

**Input Logic****Logic 1 input value — Signal value for logic 1**

0.7 (default)

Value of the input signal corresponding to the logic 1 level.

**Dependencies**

To enable this port, set **Modeling option** to PS input.

**Logic 0 input value — Signal value for logic 0**

0.3 (default)

Value of the input signal corresponding to the logic 0 level.

**Dependencies**

To enable this port, set **Modeling option** to PS input.

**Logic 1 input voltage – Voltage value for logic 1**

2.0 V (default)

Value of the input voltage corresponding to the logic 1 level.

**Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

**Logic 0 input voltage – Voltage value for logic 0**

0.8 V (default)

Value of the input voltage corresponding to the logic 0 level.

**Dependencies**

To enable this port, set **Modeling option** to Electrical input ports.

**Outputs****On-state gate-source voltage – Demanded output voltage for on state**

15 V (default)

Demanded output voltage when the driver is in on state.

**Off-state gate-source voltage – Demanded output voltage for off state**

0 V (default)

Demanded output voltage when the driver is in off state.

**Timing****Propagation delay (logic 0->logic 1) – Turn-on propagation delay**

50 ns (default)

When the input rises above the logic 1 input level, the transition of the output state from off to on is initiated after a delay equal to the turn-on propagation delay.

**Propagation delay (logic 1->logic 0) – Turn-off propagation delay**

50 ns (default)

When the input falls below the logic 0 input value, the transition of the output state from on to off is initiated after a delay equal to the turn-off propagation delay.

**Dynamics****Parameterization – Select driver parameterization**

Output impedance (default) | Rise and fall times

Select the type of driver parameterization:



- **Output impedance** — Specify on-state and off-state gate drive resistances.
- **Rise and fall times** — Specify rise time, fall time, and load capacitance.

**On-state gate drive resistance — Gate drive resistance for on state**  
2  $\Omega$  (default)

Gate drive resistance when the driver is in on state.

**Dependencies**

Enabled when the **Parameterization** parameter is set to **Output impedance**.

**Off-state gate drive resistance — Gate drive resistance for off state**  
2  $\Omega$  (default)

Gate drive resistance when the driver is in off state.

**Dependencies**

Enabled when the **Parameterization** parameter is set to **Output impedance**.

**Rise time — Driver rise time**  
20 ns (default)

Driver rise time from 10% to 90%.

**Dependencies**

Enabled when the **Parameterization** parameter is set to **Rise and fall times**.

**Fall time — Driver fall time**  
20 ns (default)

Driver fall time from 90% to 10%.

**Dependencies**

Enabled when the **Parameterization** parameter is set to **Rise and fall times**.

**Load capacitance for rise and fall times — Driver load capacitance**  
10 nF (default)

Driver load capacitance.

**Dependencies**

Enabled when the **Parameterization** parameter is set to **Rise and fall times**.

**Faults**

**Enable faults — Select Yes to enable faults modeling**  
No (default) | Yes

Select **Yes** to enable faults modeling. The associated parameters in the **Faults** section become visible to let you specify time to fail and the failure mode.

**Simulation time for fault event — Time before entering faulted state**  
1 s (default)

Set the simulation time at which you want the block to enter the faulted state.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Failure mode — State of driver after failure**

Fail off (default) | Fail on

Select whether driver fails by making the connected semiconductor device permanently turned off or on.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

## **Version History**

**Introduced in R2017b**

## **Extended Capabilities**

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Controlled PWM Voltage | Half-Bridge Driver | N-Channel IGBT | N-Channel MOSFET | P-Channel MOSFET

# Generic Linear Actuator

Generic linear actuator driven from DC voltage source or PWM driver

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



## Description

The Generic Linear Actuator block implements a model of a generic linear actuator designed to be driven from a DC voltage source or a PWM driver. Define force-speed characteristics in terms of tabulated values for powering the motor at the rated voltage. This functionality enables you to model a motor without referencing an equivalent circuit.

The motor or actuator architecture determines the way in which electrical losses depend on force. For example, a DC motor has losses that are proportional to the square of the current. As force is proportional to current, losses are also proportional to mechanical force. Most motors have an electrical loss term that is proportional to the square of mechanical force. The Generic Linear Actuator block calculates this loss term using the **Motor efficiency (percent)** and **Speed at which efficiency is measured** parameters that you provide.

Some motors also have a loss term that is independent of force. An example is a shunt motor where the field winding draws a constant current regardless of load. The **Force-independent electrical losses** parameter accounts for this effect.

The motor efficiency is the mechanical power divided by the sum of the mechanical power and both electrical loss terms. The block assumes that the speed at which the motor efficiency is defined is in the motoring quadrant and, therefore, positive.

You can operate the block in the reverse direction by changing the sign of the voltage applied. The H-Bridge block, for example, reverses motor direction if the voltage at the REV port is greater than the **Reverse threshold voltage** parameter. However, if you are using the block in reverse, specify the force-speed data for forward operation:

- Positive forces and positive speeds in the motoring quadrant.
- Positive force and negative speeds in the generating counterclockwise quadrant.
- Negative force and positive speed in the generating clockwise quadrant.

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain a thermal port.
- Show thermal port — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Assumptions and Limitations

- The force-speed curve data corresponds only to the rated voltage, so the block produces accurate results only when driven by plus or minus the rated voltage.
- The block requires you to provide force-speed data for the full range over which you use the actuator. To use the actuator in the generating and braking regions, provide additional data outside of the normal motoring region.
- Model behavior is sensitive to force-speed data. For example, no-load speed is correctly defined and finite only when the data crosses the speed axis.
- To drive the block from the H-Bridge block:
  - Do not place any other blocks between the H-Bridge and the Generic Linear Actuator blocks.
  - In the H-Bridge block dialog box, set the **Freewheeling mode** to **Via one semiconductor switch and one freewheeling diode**. Selecting **Via two freewheeling diodes** does not set the bridge output voltage to zero when the PWM input signal is low.
  - In the H-Bridge, Generic Linear Actuator, and Controlled PWM Voltage block dialog boxes, ensure that the **Simulation mode** is the same for all three blocks.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the actuator positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the actuator negative terminal.

#### C – Case

mechanical

Mechanical translational conserving port associated with the actuator case.

#### R – Plunger

mechanical

Mechanical translational conserving port associated with the plunger.

#### H – Thermal port

thermal

Thermal port.

### Dependencies

To enable this port, set **Modeling option** to **Show thermal port**.

## Parameters

### Modeling option — Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

### Electrical Force

#### Speed values — Speed values

[-15, -10, -5, 0, 5, 10, 15, 20, 25, 30] m/s (default)

Specify a vector of speeds, including their units, for your force-speed data.

#### Force values — Force values

[4, 3.5, 3, 2.5, 2, 1.5, 1, .5, 0, -.5] N (default)

Specify a vector of forces, including their units, for your force-speed data.

#### Rated voltage — Rated voltage

12 V (default)

Indicate the voltage for which the device you are modeling is rated.

#### Motor efficiency (percent) — Motor efficiency

70 (default)

Efficiency that the block uses to calculate force-dependent electrical losses.

#### Speed at which efficiency is measured — Speed at which efficiency is measured

20 m/s (default)

Speed that the block uses to calculate force-dependent electrical losses.

#### Force-independent electrical losses — Force-independent electrical losses

2 W (default)

Fixed electrical loss associated with the actuator when the force is zero.

#### Simulation mode — Simulation mode

PWM (default) | Averaged

If you set the **Simulation mode** parameter to **PWM**, apply a PWM waveform switching between zero and rated volts to the block electrical terminals. The current drawn from the electrical supply is equal to the amount required to deliver the mechanical power and to compensate for electrical losses. If the applied voltage exceeds the rated voltage, the resultant force scales proportionately. However, applying anything other than the rated voltage can provide unrepresentative results.

If you set the **Simulation mode** parameter to **Averaged**, the force generated in response to an applied voltage  $V_{av}$  is

$$\frac{V_{av}}{V_{rated}} \times F(v)$$

where  $F(v)$  is the force value at speed  $v$ . The current drawn from the supply is such that the product of the current and  $V_{av}$  is equal to the average power that is consumed.

### **Mechanical**

#### **Plunger mass — Plunger mass**

0.1 kg (default)

Mass of the moving part of the motor. The value can be zero.

#### **Linear damping — Linear damping**

1e-5 N/(m/s) (default)

Linear damping. The value can be zero.

### **Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Resistance temperature coefficient.

#### **Measurement temperature — Measurement temperature**

25 degC (default)

The temperature for which the actuator parameters are defined.

### **Thermal Port**

To enable this parameter, set **Modeling option** to Show thermal port.

#### **Thermal mass — Thermal mass**

100 J/K (default)

The thermal mass is the energy required to raise the temperature by one degree.

## **Version History**

**Introduced in R2009b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Generic Rotary Actuator | H-Bridge

# Generic Rotary Actuator

Generic rotary actuator driven from DC voltage source or PWM driver

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



## Description

The Generic Rotary Actuator block implements a model of a generic rotary actuator designed to be driven from a DC voltage source or PWM driver. You define torque-speed characteristics in terms of tabulated values for powering the motor at the rated voltage. This functionality allows you to model a motor without referencing an equivalent circuit.

The motor or actuator architecture determines the way in which electrical losses depend on torque. For example, a DC motor has losses that are proportional to the square of the current. As torque is proportional to current, losses are also proportional to mechanical torque. Most motors have an electrical loss term that is proportional to the square of mechanical torque. The Generic Rotary Actuator block calculates this loss term using the **Motor efficiency (percent)** and **Speed at which efficiency is measured** parameters that you provide.

Some motors also have a loss term that is independent of torque. An example is a shunt motor where the field winding draws a constant current regardless of load. The **Torque-independent electrical losses** parameter accounts for this effect.

The motor efficiency is the mechanical power divided by the sum of the mechanical power and both electrical loss terms. The block assumes that the speed at which the motor efficiency is defined is in the motoring quadrant and, therefore, positive.

You can operate the block in the reverse direction by changing the sign of the voltage that you apply. The H-Bridge block, for example, reverses motor direction if the voltage at the REV port is greater than the **Reverse threshold voltage** parameter. However, if you are using the block in reverse, specify the torque-speed data for forward operation:

- Positive torques and positive speeds in the motoring quadrant.
- Positive torque and negative speeds in the generating counterclockwise quadrant.
- Negative torque and positive speed in the generating clockwise quadrant.

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain a thermal port.
- Show thermal port — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Assumptions and Limitations

- The torque-speed curve data corresponds only to the rated voltage, so the block produces accurate results only when driven by plus or minus the rated voltage.
- In this block requires, you must provide torque-speed data for the full range over which you use the actuator. To use the actuator in the generating and braking regions, provide additional data outside of the normal motoring region.
- Model behavior is sensitive to torque-speed data. For example, no-load speed is correctly defined and finite only when the data crosses the speed axis.
- To drive the block from the H-Bridge block:
  - Do not place any other blocks between the H-Bridge and the Generic Rotary Actuator blocks.
  - In the H-Bridge block dialog box, set the **Freewheeling mode** to **Via one semiconductor switch and one freewheeling diode**. Selecting **Via two freewheeling diodes** does not set the bridge output voltage to zero when the PWM input signal is low.
  - In the H-Bridge, Generic Rotary Actuator, and Controlled PWM Voltage block dialog boxes, ensure that the **Simulation mode** is the same for all three blocks.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the actuator positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the actuator negative terminal.

#### C – Case

mechanical

Mechanical rotational conserving port associated with the actuator case.

#### R – Rotor

mechanical

Mechanical rotational conserving port associated with the rotor.

#### H – Thermal port

thermal

Thermal port.

### Dependencies

To enable this port, set **Modeling option** to **Show thermal port**.



## Parameters

### Modeling option — Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

### Electrical Torque

#### Speed values — Vector of speeds

[-1500, -1000, -500, 0, 500, 1000, 1500, 2000, 2500, 3000] rpm (default)

Specify a vector of speeds, including their units, for your torque-speed data.

#### Torque values — Vector of torques

[.04, .035, .03, .025, .02, .015, .01, .005, 0, -.005] Nm (default)

Specify a vector of torques, including their units, for your torque-speed data.

#### Rated voltage — Rated voltage

12 V (default)

Indicate the voltage for which the device you are modeling is rated.

#### Motor efficiency (percent) — Motor efficiency

80 (default)

The efficiency that the block uses to calculate torque-dependent electrical losses.

#### Speed at which efficiency is measured — Speed at which efficiency is measured

2000 rpm (default)

The speed that the block uses to calculate torque-dependent electrical losses.

#### Torque-independent electrical losses — Fixed electrical loss

0.1 W (default)

Fixed electrical loss associated with the actuator when the torque is zero.

#### Simulation mode — Simulation mode

PWM (default) | Averaged

If you set the **Simulation mode** parameter to PWM, apply a PWM waveform switching between zero and rated volts to the block electrical terminals. The current drawn from the electrical supply is equal to the amount required to deliver the mechanical power and to compensate for electrical losses. If the applied voltage exceeds the rated voltage, the resultant torque scales proportionately. However, applying anything other than the rated voltage can provide unrepresentative results.

If you set the **Simulation mode** parameter to Averaged, the torque generated in response to an applied voltage  $V_{av}$  is

$$\frac{V_{av}}{V_{rated}} \times T(\omega)$$

where  $T(\omega)$  is the torque value at speed  $\omega$ . The current drawn from the supply is such that the product of the current and  $V_{av}$  is equal to the average power that is consumed.

## **Mechanical**

### **Rotor inertia — Rotor inertia**

0.0001 kg\*m<sup>2</sup> (default)

Rotor resistance to change in motor motion. The value can be zero.

### **Rotor damping — Rotor damping**

1e-8 N\*m/(rad/s) (default)

Rotor damping. The value can be zero.

## **Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

### **Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Resistance temperature coefficient.

### **Measurement temperature — Measurement temperature**

25 degC (default)

The temperature for which the actuator parameters are defined.

## **Thermal Port**

To enable this parameter, set **Modeling option** to Show thermal port.

### **Thermal mass — Thermal mass**

100 J/K (default)

The thermal mass is the energy required to raise the temperature by one degree.

## **Version History**

**Introduced in R2009b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

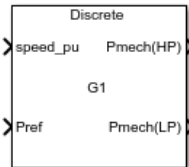
## **See Also**

Generic Linear Actuator | H-Bridge

# Governor Type 1

IEEE type 1 linearized speed-governing steam turbine model

**Library:** Simscape / Electrical / Control / Turbine-Governors



## Description

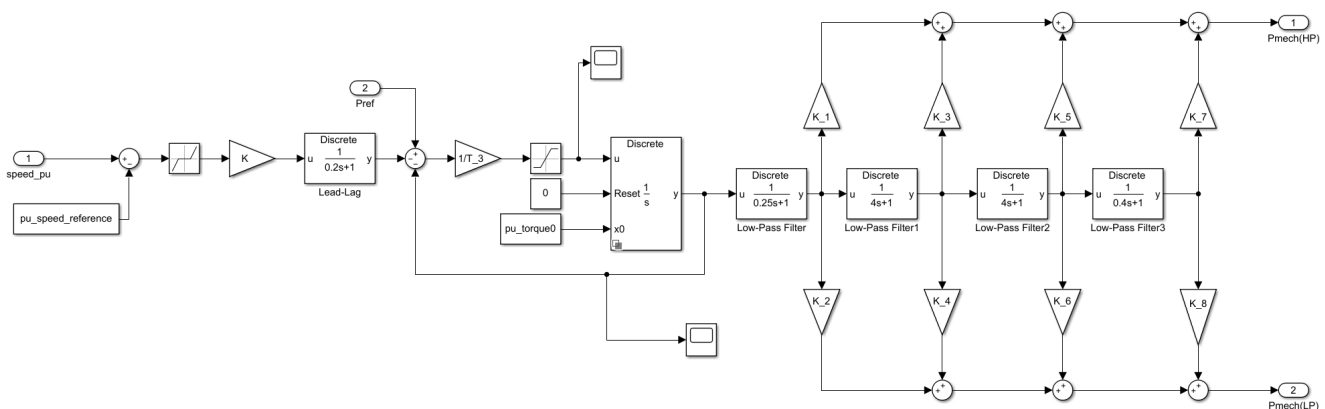
The Governor Type 1 block models a model IEEEG1 steam turbine-governor model IEEEG1.

This block allows the modeling of cross-compound units and it has two sets of turbine fractions:

- **LP Fraction, K\_2, LP Fraction, K\_4, LP Fraction, K\_6, LP Fraction, K\_8** for low-pressure (LP).
- **HP Fraction, K\_1, HP Fraction, K\_3, HP Fraction, K\_5, HP Fraction, K\_7** for high-pressure (HP).

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the governor for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the governor for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the block:



## Ports

### Input

#### **speed\_pu — Turbine speed**

scalar

Speed of the turbine, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **Pref — Speed reference bias**

scalar

Bias to the speed load reference, specified as a scalar. Connect this port to the **Pref** output port of a turbine load controller block, such as the Controller LCFB1 block.

Data Types: `single` | `double`

### Output

#### **Pmech(HP) — High-pressure mechanical power**

scalar

High-pressure mechanical power, returned as a scalar.

Data Types: `single` | `double`

#### **Pmech(LP) — Low-pressure mechanical power**

scalar

Low-pressure mechanical power, returned as a scalar.

Data Types: `single` | `double`

## Parameters

### General

#### **Speed reference (pu) — Speed reference**

1 (default) | scalar

Per-unit speed reference.

#### **Initial torque (pu) — Initial torque**

0.45 (default) | scalar

Per-unit initial torque at the start of the simulation.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see "Promote Block Parameters on a Mask".

## **Governor**

### **Governor gain (1/droop), K (pu) – Gain of governor**

20 (default) | scalar

Per-unit gain of the governor.

### **Lag time constant, T\_1 (s) – Governor lag time constant**

0.2 (default) | scalar

Equivalent lag time constant. This parameter is associated with the block labeled "Lead-Lag" in the diagram. Set this parameter to 0 when the additional lag dynamics are negligible.

### **Lead time constant, T\_2 (s) – Governor lead time constant**

0 (default) | scalar

Equivalent lead time constant. This parameter is associated with the block labeled "Lead-Lag" in the diagram. Set this parameter to 0 when the additional lead dynamics are negligible.

### **Valve position time constant, T\_3 (s) – Time constant of valve position**

0.1 (default) | scalar

Time constant of the valve position, in seconds.

### **Maximum valve opening rate, U\_o (pu/s) – Maximum opening rate of the valve**

0.1 (default) | scalar

Per-unit maximum opening rate of the valve.

### **Maximum valve closing rate, U\_c (pu/s) – Maximum closing rate of the valve**

-0.1 (default) | scalar

Per-unit maximum closing rate of the valve.

### **Maximum valve opening, on MW capability, P\_max (pu) – Maximum opening of the valve on MW capability**

0.98 (default) | scalar

Per-unit maximum opening of the valve, on MW capability.

### **Minimum valve opening, on MW capability, P\_min (pu) – Minimum opening of the valve, on MW capability**

0 (default) | scalar

Per-unit minimum opening of the valve, on MW capability.

### **Time constant for steam inlet, T\_4 (s) – Time constant of the inlet steam**

0.25 (default) | scalar

Time constant of the inlet steam bowl, in seconds. This parameter is associated with the block labeled "Low-Pass Filter" in the diagram.

**HP fraction, K\_1 — Fraction of high-pressure shaft power**

0.22 (default) | scalar

Fraction of the high-pressure shaft power after the first boiler pass.

**LP fraction, K2 — Fraction of low-pressure shaft power**

0 (default) | scalar

Fraction of the low-pressure shaft power after the first boiler pass.

**Time constant for second boiler pass, T\_5 (s) — Time constant for second boiler pass**

4 (default) | scalar

Time constant of the second boiler pass, in seconds. This parameter is associated with the block labeled "Low-Pass Filter1" in the diagram.

**HP fraction, K\_3 — Fraction of high-pressure shaft power**

0.22 (default) | scalar

Fraction of the high-pressure shaft power after the second boiler pass.

**LP fraction, K\_4 — Fraction of low-pressure shaft power**

0 (default) | scalar

Fraction of the low-pressure shaft power after the second boiler pass.

**Time constant for third boiler pass, T\_6 (s) — Time constant for third boiler pass**

4 (default) | scalar

Time constant of the third boiler pass, in seconds. This parameter is associated with the block labeled "Low-Pass Filter2" in the diagram.

**HP fraction, K\_5 — Fraction of high-pressure shaft power**

0.3 (default) | scalar

Fraction of the high-pressure shaft power after the third boiler pass.

**LP fraction, K\_6 — Fraction of low-pressure shaft power**

0 (default) | scalar

Fraction of the low-pressure shaft power after the third boiler pass.

**Time constant for fourth boiler pass, T\_7 (s) — Time constant for fourth boiler pass**

0.4 (default) | scalar

Time constant of the fourth boiler pass, in seconds. This parameter is associated with the block labeled "Low-Pass Filter3" in the diagram.

**HP fraction, K\_7 — Fraction of high-pressure shaft power**

0.26 (default) | scalar

Fraction of the high-pressure shaft power after the fourth boiler pass.

**LP fraction, K\_8 – Fraction of low-pressure shaft power**

0 (default) | scalar

Fraction of the low-pressure shaft power after the fourth boiler pass.

**Deadband – Deadband width**

0.02 (default) | scalar

Intentional deadband width.

## Version History

Introduced in R2020a

## References

- [1] *Dynamic Models for Steam and Hydro Turbines in Power System Studies*, IEEE Transactions on Power Apparatus and Systems. Vol. PAS-92, Number 6, 1973, pp. 1904–1915.
- [2] Task Force on Turbine-Governor Modeling, *Dynamic models for turbine-governors in power system studies*, IEEE Power Energy Society, January 2013.
- [3] *IEEE Guide for the Application of Turbine Governor Systems for Hydroelectric Generating Units*, IEEE Std 1207-2011.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

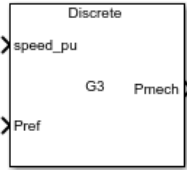
**Blocks**

Governor Type 3 | Controller LCFB1

# Governor Type 3

IEEE type 3 linearized speed-governing hydro turbine model

**Library:** Simscape / Electrical / Control / Turbine-Governors



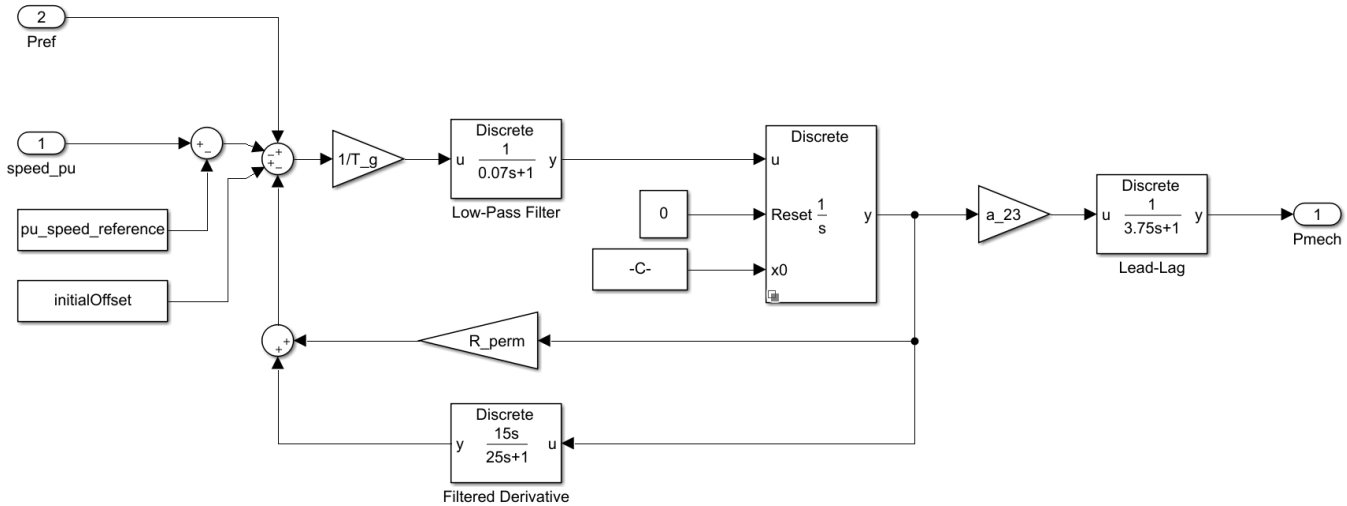
## Description

The Governor Type 3 block models a model IEEEG3 hydro turbine-governor with penstock dynamics.

This block has a more detailed governor model than the Governor Type 1 block and uses a linearized model of the turbine, or water, column model and penstock dynamics.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the governor for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the governor for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the block:



## Ports

### Input

**speed\_pu** — Turbine speed  
scalar



Speed of the turbine, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

### **Pref — Speed reference bias**

scalar

Bias to the speed load reference, specified as a scalar. Connect this port to the **Pref** output port of a turbine load controller block, such as the Controller LCFB1 block.

Data Types: `single` | `double`

## **Output**

### **Pmech — Mechanical power**

scalar

Mechanical power, returned as a scalar.

Data Types: `single` | `double`

## **Parameters**

### **General**

#### **Speed reference (pu) — Speed reference**

1 (default) | scalar

Per-unit speed reference.

#### **Initial torque (pu) — Initial torque**

0.45 (default) | scalar

Per-unit initial torque at the start of the simulation.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### **Governor**

#### **Gate servomotor time constant, T<sub>g</sub> (s) — Time constant for gate servomotor**

0.52 (default) | scalar

Time constant for the gate servomotor, in seconds.

**Pilot value time constant,  $T_p$  (s) – Time constant for pilot value**

0.07 (default) | scalar

Time constant for the pilot value, in seconds. This parameter is associated with the block labeled "Low-Pass Filter" in the diagram.

**Opening gate rate limit,  $U_o$  (pu/s) – Limit of opening gate rate**

0.15 (default) | scalar

Per-unit limit of the opening gate rate. This parameter is associated with the block labeled "Low-Pass Filter" in the diagram.

**Closing gate rate limit,  $U_c$  (pu/s) – Limit of closing gate rate**

-0.15 (default) | scalar

Per-unit limit of the closing gate rate. This parameter is associated with the block labeled "Low-Pass Filter" in the diagram.

**Maximum gate position, on MVA capability,  $P_{max}$  (pu) – Maximum position of gate**

0.75 (default) | scalar

Maximum per-unit position of the gate, on MVA capability.

**Minimum gate position, on MVA capability,  $P_{min}$  (pu) – Minimum position of gate**

0 (default) | scalar

Minimum per-unit position of the gate, on MVA capability.

**Permanent speed droop coefficient,  $R_{perm}$  (pu) – Permanent speed droop coefficient**

0.05 (default) | scalar

Per-unit permanent speed droop coefficient.

**Transient speed droop coefficient,  $R_{temp}$  (pu) – Transient speed droop coefficient**

0.6 (default) | scalar

Per-unit transient speed droop coefficient. This parameter is associated with the block labeled "Filtered Derivative" in the diagram.

**Governor time constant,  $T_r$  (s) – Time constant of governor**

25 (default) | scalar

Time constant of the governor. This parameter is associated with the block labeled "Filtered Derivative" in the diagram.

**Water starting time constant,  $T_w$  (s) – Water starting time constant**

5 (default) | scalar

Water inertia time constant. This parameter is associated with the block labeled "Lead-Lag" in the diagram.

**Penstock coefficient,  $a_{11}$  – Penstock dynamics coefficient**

0.75 (default) | scalar

Coefficient for the penstock dynamics. This parameter is associated with the block labeled "Lead-Lag" in the diagram.

**Penstock coefficient, a<sub>13</sub> – Penstock dynamics coefficient**

0.75 (default) | scalar

Coefficient for the penstock dynamics. This parameter is associated with the block labeled "Lead-Lag" in the diagram.

**Penstock coefficient, a<sub>21</sub> – Penstock dynamics coefficient**

0.75 (default) | scalar

Coefficient for the penstock dynamics. This parameter is associated with the block labeled "Lead-Lag" in the diagram.

**Penstock coefficient, a<sub>23</sub> – Penstock dynamics coefficient**

0.75 (default) | scalar

Coefficient for the penstock dynamics. This parameter is associated with the block labeled "Lead-Lag" in the diagram.

## Version History

Introduced in R2020a

## References

- [1] Overbye, Tom *Power Systems Stability*, Lecture Notes. Texas A&M University.
- [2] *Dynamic Models for Steam and Hydro Turbines in Power System Studies*, IEEE Transactions on Power Apparatus and Systems. Vol. PAS-92, Number 6, 1973, pp. 1904–1915.
- [3] *IEEE Guide for the Application of Turbine Governor Systems for Hydroelectric Generating Units*, IEEE Std 1207-2011.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Governor Type 1 | Controller LCFB1

## Grounded Neutral (Three-Phase)

Connect phases of three-phase system to electrical reference

**Library:** Simscape / Electrical / Connectors & References



### Description

The Grounded Neutral (Three-Phase) block connects the phases of a three-phase system to ground.

---

**Note** If you want to connect the neutral point of the three-phase system to other blocks, use the Neutral Port (Three-Phase) block instead. If you want to create a floating neutral point, use the Floating Neutral (Three-Phase) block.

---

### Ports

#### Conserving

~ — **Three-phase port**

electrical

Expandable composite (**a,b, c**) three-phase port.

### Parameters

**Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Version History

Introduced in R2013b

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

**See Also**

Floating Neutral (Three-Phase) | Open Circuit (Three-Phase) | Neutral Port (Three-Phase)

# GTO

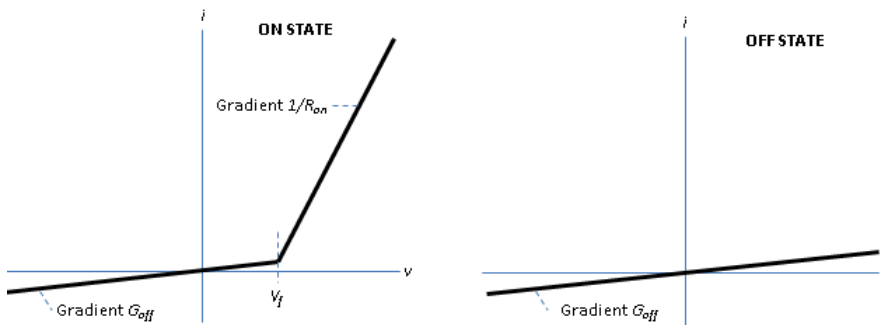
Gate Turn-Off Thyristor

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The GTO block models a gate turn-off thyristor (GTO). The I-V characteristic of a GTO is such that if the gate-cathode voltage exceeds the specified gate trigger voltage, the GTO turns on. If the gate-cathode voltage falls below the specified gate turn-off voltage value, or if the load current falls below the specified holding-current value, the device turns off.



To define the I-V characteristic of the GTO, set the **On-state behaviour and switching losses** parameter to either **Specify constant values** or **Tabulate with temperature and current**. The **Tabulate with temperature and current** option is available only if you expose the thermal port of the block.

In the on state, the anode-cathode path behaves like a linear diode with forward-voltage drop,  $V_f$ , and on-resistance,  $R_{on}$ . However, if you expose the thermal port of the block and parameterize the device using tabulated I-V data, the tabulated resistance is a function of the temperature and current.

In the off state, the anode-cathode path behaves like a linear resistor with a low off-state conductance value,  $G_{off}$ .

The defining Simscape equations for the block are:

```

if ((v > Vf)&&((G>Vgt)|| (i>Ih)))&&(G>Vgt_off)
    i == (v - Vf*(1-Ron*Goff))/Ron;
else
    i == v*Goff;
end

```

where:

- $v$  is the anode-cathode voltage.
- $V_f$  is the forward voltage.
- $G$  is the gate voltage.
- $V_{gt}$  is the gate trigger voltage.
- $i$  is the anode-cathode current.
- $I_h$  is the holding current.
- $V_{gt\_off}$  is the gate turn-off voltage.
- $R_{on}$  is the on-state resistance.
- $G_{off}$  is the off-state conductance.

Using the **Integral Diode** parameters, you can include an integral cathode-anode diode. A GTO that includes an integral cathode-anode diode is known as an asymmetrical GTO (A-GTO) or reverse-conducting GTO (RCGTO). An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

The table shows you how to set the **Integral protection diode** parameter based on your goals.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	Protection diode with no dynamics	The block includes an integral copy of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.
Precisely specify reverse-mode charge dynamics.	Protection diode with charge dynamics	The block includes an integral copy of the dynamic model of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.

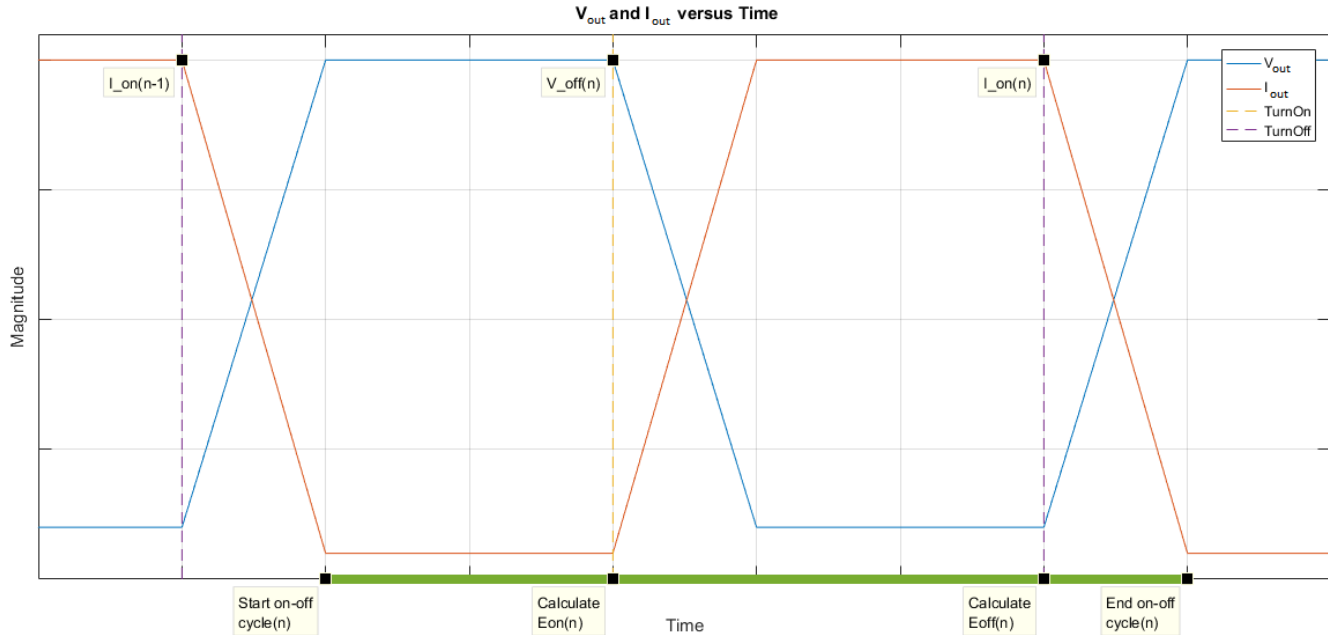
### Model Gate Port and Thermal Effects

You can choose between physical or electrical ports to control the gate terminal and expose the thermal port to model the heat that switching events and conduction losses generate. To choose the gate control port, set the **Gate control port** parameter to PS or Electrical. To expose the thermal port, set the **Modeling option** parameter to either No thermal port or Show thermal port.

For more information about using thermal ports, see “Simulating Thermal Effects in Semiconductors”.

### Thermal Losses

The figure shows an idealized representation of the output voltage,  $V_{out}$ , and the output current,  $I_{out}$ , of the semiconductor device. The interval shown includes the entire  $n^{\text{th}}$  switching cycle, during which the block turns off and then on.



Switching losses are one of the main sources of thermal loss in semiconductors. During each on-off switching transition, the GTO parasitics store and then dissipate energy.

Switching losses depend on the off-state voltage and the on-state current. When the switching device is turned on, the power losses depend on the initial off-state voltage across the device and the final on-state current once the device is fully in its on state. Similarly, when the switching device is force commutated off, the power losses depend on the initial on-state current through the device and the final off-state voltage once the device is fully in its off state. The switch on and force commutated switch off losses are either fixed or dependent on junction temperature and drain-source current, depending on how you specify the **On-state behavior and switching losses** parameter. In both cases, the losses are scaled by the off-state voltage prior to the latest device turn-on event.

When the current falls below the holding current and the device is naturally commutated off, the losses are set by the **Natural commutation rectification loss** parameter. Because it's not possible to know when to measure the starting current or final voltage for the rectification loss, it is not possible to scale it by the off-state voltage or on-state current.

In this block, switching losses are applied by stepping up the junction temperature with a value equal to the switching loss divided by the total thermal mass at the junction.

**Note** As the final current after a switching event is not known during the simulation, if you use the GTO block as a fully-controlled device, the block records the on-state current at the point that the device is commanded off. If you use the GTO block as a partially-controlled device, the block records the on-state current once the current is greater than the holding current for a time longer than the value of the **Wait time before switch-on current measurement** parameter. Similarly, the block records the off-state voltage at the point that the device is commanded on. For this reason, the simlog does not report the switching losses to the thermal network until one switching cycle later

For all ideal switching devices, the switching losses are reported in the simlog as `lastTurnOffLoss` and `lastTurnOnLoss` and recorded as a pulse with amplitude equal to the energy loss. If you use a



script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

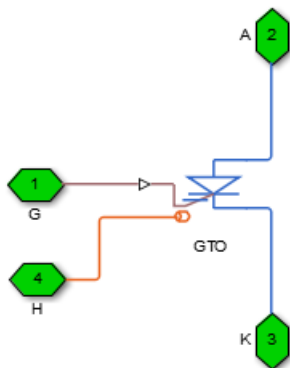
## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

The figure shows the block port names.



## Conserving

### G — Gate terminal

electrical

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

### A — Anode terminal

electrical

Electrical conserving port associated with the anode terminal.

### K — Cathode terminal

electrical

Electrical conserving port associated with the cathode terminal.

**H – Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters**

**Modeling option – Thermal port visibility**

No thermal port (default) | Show thermal port

Whether to enable the thermal port.

**Main**

This table shows how the visibility of **Main** parameters depends on how you configure the **Modeling option** and **On-state behavior and switching losses** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Main Parameter Dependencies**

Parameters and Options		
<b>Modeling option</b>		
<i>No thermal port</i>	<i>Show thermal port</i>	
Gate-control port	Gate-control port	
Gate trigger voltage, Vgt	Gate trigger voltage, Vgt	
Gate turn-off voltage, Vgt_off	Gate turn-off voltage, Vgt_off	
Holding current	Holding current	
Forward voltage, Vf	<b>On-state behaviour and switching losses</b>	
	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>
On-state resistance	Forward voltage, Vf	On-state voltage, Vak(Tj,Iak)
Off-state conductance	On-state resistance	Temperature vector, Tj
	Off-state conductance	Anode-cathode current vector, Iak
		Off-state conductance

**Gate-control port – Whether to specify physical or electrical control port**

PS (default) | Electrical

Whether to specify physical or electrical control port for the switch gate.

**On-state behaviour and switching losses — On-state current for switching loss data**

Specify constant values (default) | Tabulate with temperature and current

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- **Specify constant values** — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data. This is the default parameterization method.
- **Tabulate with temperature and current** — Use vectors to specify the output current, switch-on loss, switch-off loss, and temperature data.

**Dependencies**

See the Main Parameter Dependencies table.

**Gate trigger voltage, Vgt — Gate trigger voltage**

6 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Main Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off — Gate turn-off voltage**

-1 V (default)

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Main Parameter Dependencies table.

**Holding current — Holding current**

1 A (default)

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Main Parameter Dependencies table.

**Forward voltage, Vf — Forward voltage**

0.8 V (default)

Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state resistance — On-state resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the forward voltage. The default value is 0.001.

**Dependencies**

See the Main Parameter Dependencies table.

**Off-state conductance — Off-state conductance**

1e-5 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than 1/R, where R is the value of **On-state resistance**. The default value is 1e-5.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage,  $V_{ak}(T_j, I_{ak})$  — On-state voltage**

[0, .1, .6, .8, 1, 1.3, 1.6, 2, 2.4; 0, .1, .7, 1, 1.2, 1.5, 1.9, 2.4, 2.8] V (default)

Voltage drop across the device while it is in a triggered conductive state. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Temperature vector,  $T_j$  — Temperature vector**

[298.15, 398.15] K (default)

Temperature values at which the on-state voltage is specified. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Anode-cathode current vector,  $I_{ak}$  — Anode-cathode current vector**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Anode-cathode currents for which the on-state voltage is defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Switching Losses**

To enable these parameters, set **Modeling option** to Show thermal port.

**Switch-on loss — Switch-on loss**

22.86e-3 J (default)

Energy dissipated during a single switch-on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Forced commutation switch-off loss — Forced commutation switch-off loss**

17.14e-3 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Natural commutation rectification loss — Natural commutation rectification loss**

10e-3 J (default)

Rectification loss applied at the point that the block switches off when the current falls below the **Holding current**. Specify this parameter using a scalar quantity.

**Off-state voltage for switching loss data — Off-state voltage for losses data**

600 V (default) | 300 V

The output voltage of the device during the off state. This is the blocking voltage at which the switch-on loss and switch-off loss data are defined.

**On-state current for switching loss data — Output current**

600 A (default)

Output currents for which the switch-on loss, switch-off loss, and on-state voltage are defined. The first element must be zero. Specify this parameter using a scalar quantity.

---

**Note** This parameter is measured at the point that the gate voltage falls below the **Gate trigger voltage, Vgt**. The turn-on pulse is longer than the time it takes the current to reach its maximum value.

---

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-on loss,  $E_{on}(T_j, I_{ak})$  — Switch-on loss**

[0, .0024, .024, .12, .2, .48, 1.04, 2.16, 3.24; 0, .003, .03, .15, .25, .6, 1.3, 2.7, 4.05] \* 1e-3 J (default)

Energy dissipated during a single switch on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Switch-off loss,  $E_{\text{off}}(T_j, I_{\text{ak}})$  — Switch-off loss**

[0, .0007, .0066, .033, .066, .17, .33, .83, 1.5;  
0, .001, .01, .05, .1, .25, .5, 1.2, 2.2] \* 1e-3 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Temperature vector for switching losses,  $T_j$  — Temperature vector for switching losses**

[298.15, 398.15] K (default)

Temperature values at which the switch-on loss and switch-off loss are specified. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Anode-cathode current vector for switching losses,  $I_{\text{ak}}$  — Anode-cathode current vector for switching losses**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Anode-cathode currents for which the switch-on loss and switch-off loss are defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Wait time before switch-on current measurement — Wait time before switch-on current measurement**

0 s (default)

Time to wait before recording the on-state current if you use the GTO block as a partially-controlled device. If you use the GTO block as a fully-controlled device, set this parameter to 0.

**Integral Diode****Integral protection diode — Integral protection diode**

None (default) | Protection diode with no dynamics | Protection diode with charge dynamics

Block integral protection diode. The default value is None.

The diodes you can select are:

- Protection diode with no dynamics
- Protection diode with charge dynamics

**Diode model — Diode model**

Piecewise Linear (default) | Tabulated I-V curve

Select one of these diode models:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448. This is the default method.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data plus fixed reverse bias off conductance.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Table type — Tabulated function**

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** is set to Tabulated I-V curve.

**Forward voltage — Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.
- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.

- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

#### **Forward currents, $I_f(T_j, V_f)$ — Vector of forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12]  
A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

#### **Junction temperatures, $T_j$ — Vector of junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures. This parameter must be a vector of at least two elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve.

#### **Forward voltages, $V_f$ — Vector of forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

#### **Forward voltages, $V_f(T_j, I_f)$ — Vector of forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

#### **Forward currents, $I_f$ — Vector of forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

#### **Off conductance — Off conductance**

1e-5 1/Ohm (default)



Conductance of the reverse-biased diode.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Junction capacitance – Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  – Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  – Initial forward current when measuring  $i_{RM}$**

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  – Rate of change of current when measuring  $i_{RM}$**

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Reverse recovery time parameterization – Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select **Specify stretch factor** or **Specify reverse recovery charge**, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics**.

#### Reverse recovery time, $t_{rr}$ — Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify reverse recovery time directly**.

#### Reverse recovery time stretch factor — Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify stretch factor**.

#### Reverse recovery charge, $Q_{rr}$ — Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery charge.

**Diode turn-off voltage when measuring Erec — Diode turn-off voltage when measuring Erec**

-600 V (default)

Voltage between the diode in steady-state.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Stray inductance when measuring Erec — Stray inductance when measuring Erec**

150 nH (default)

Total unintended inductance in the measurement circuit. The block uses this value to calculate **Reverse recovery energy, Erec**.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Reverse recovery energy, Erec — Reverse recovery energy**

0.03 J (default)

Total switching losses due to the diode reverse recovery.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

For more information on these parameters, see Diode.

**Thermal Port**

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

**Version History****Introduced in R2013b****Update on switching losses and thermal modelling options**

*Behavior changed in R2020b*

From R2020b forward, the GTO (Ideal, Switching) block has improved losses and thermal modelling options.

As a result from these changes:

- Electrical and thermal on-state losses are now always identical. The **Thermal loss dependent on** parameter and its options, **Voltage** and **current** and **Voltage, current, and temperature**, have been renamed to **On-state behavior and switching losses**, **Specify constant values**, and **Tabulate with temperature and current**:
  - If you selected **Voltage** and **current** for **Thermal loss dependent on**, then the electrical on-state losses are unchanged and their values are determined using the on-state resistance. However, the thermal on-state losses are now also determined by the on-state resistance. Prior to R2020b, the thermal on-state losses were defined by the product of the **On-state voltage** and **Output current, Iout** parameters.
  - If you selected **Voltage, current, and temperature** for **Thermal loss dependent on**, then the thermal on-state losses are unchanged and the **On-state voltage, Vak(Tj,Iak)** parameter sets their values. However, the electrical on-state losses are now equal to the thermal on-state losses. Prior to R2020b, the electrical on-state losses were defined by the value of the on-state resistance.
- The **On-state voltage** parameter is no longer used.

### **Removal of energy dissipation time constant parameter**

*Behavior changed in R2021a*

From R2021a forward, the **Energy dissipation time constant** parameter of the GTO (Ideal, Switching) block is no longer used. A step in junction temperature now reflects the switching losses. If your model contains a thermal mass directly connected to this block thermal port, remove it and model the thermal mass inside the component itself.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Diode | Ideal Semiconductor Switch | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | N-Channel MOSFET | P-Channel MOSFET | Thyristor (Piecewise Linear)

### **Topics**

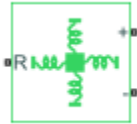
“Simulating Thermal Effects in Semiconductors”

“Switch Between Physical Signal and Electrical Ports”

# Gyro

Behavioral model of MEMS gyro

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Gyro block implements a behavioral model of a MicroElectroMechanical Systems (MEMS) gyro. The gyro provides an output voltage that is proportional to the angular rotation rate presented at the mechanical rotational physical port R. 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 setting the **Dynamics** parameter to `Model sensor bandwidth`. Including dynamics adds a first-order lag between the angular rate presented at port R and the corresponding voltage applied to the electrical + and - ports.

If running your simulation with a fixed-step solver, or generating code for hardware-in-the-loop testing, MathWorks recommends that you set the **Dynamics** parameter to `No dynamics – Suitable for HIL`, because this avoids the need for a small simulation time step if the sensor bandwidth is high.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The **Measured angular rate** variable target specifies the initial output for the sensor.

## Ports

### Conserving

#### R – Translational port

mechanical

Mechanical conserving port associated with the gyro translational port.

#### + – Positive port

electrical

Electrical conserving port associated with the gyro positive terminal.

**- – Negative port**

electrical

Electrical conserving port associated with the gyro negative terminal.

## Parameters

**Sensitivity – Sensitivity**

12.5 s\*mV/deg (default)

The change in output voltage level per unit change in rotation rate when the output is not being limited.

**Output voltage for zero rotation – Output voltage for zero rotation**

2.5 V (default)

The output voltage from the sensor when the rotation rate is zero.

**Maximum output voltage – Maximum output voltage**

4 V (default)

The maximum output voltage from the sensor, which determines the sensor maximum measured rotational rate.

**Minimum output voltage – Minimum output voltage**

1 V (default)

The minimum output voltage from the sensor, which determines the sensor minimum measured rotational rate.

**Dynamics – Dynamics modeling option**

No dynamics – Suitable for HIL (default) | Model sensor bandwidth

Select one of the following options for modeling sensor dynamics:

- **No dynamics – Suitable for HIL** – Do not model sensor dynamics. Use this option when running your simulation fixed step or generating code for hardware-in-the-loop testing, because this avoids the need for a small simulation time step if the sensor bandwidth is high. This is the default option.
- **Model sensor bandwidth** – Model sensor dynamics with a first-order lag approximation, based on the **Bandwidth** parameter value. You can control the initial condition for the lag by specifying the **Measured angular rate** variable target.

**Bandwidth – Bandwidth**

3 kHz (default)

Specifies the 3dB bandwidth for the measured rotational rate assuming a first-order time constant.

**Dependencies**

This parameter is visible only when you select Model sensor bandwidth for the **Dynamics** parameter.

## **Version History**

**Introduced in R2012b**

## **Extended Capabilities**

### **C/C++ Code Generation**

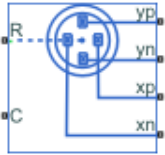
Generate C and C++ code using Simulink® Coder™.

## **See Also**

# Hall-Effect Rotary Encoder

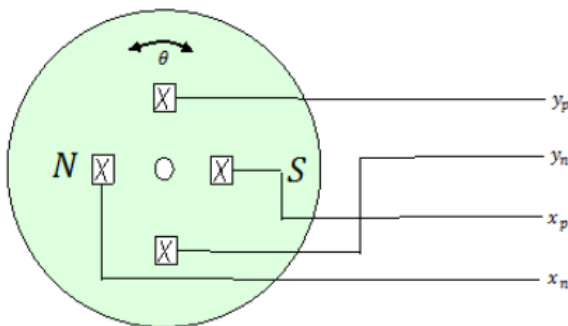
Four-element Hall-effect rotary encoder

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Hall-Effect Rotary Encoder block models a 360° rotary position sensor using four Hall elements equally spaced under a magnetized, rotating magnet. The elements generate four sinusoidal waveforms.



## Equations

The voltages between the elements of the rotary encoder are described by the equations:

$$v_x = V_{x0} + A_x \cos(\theta)$$

$$v_y = V_{y0} + A_y \cos\left(\theta - \frac{\pi}{2} + \beta\right),$$

where:

- $v_x$  is the voltage between elements  $x_p$  and  $x_n$ ;
- $v_y$  is the voltage between elements  $y_p$  and  $y_n$ ;
- $A_x$  and  $A_y$  are the voltage amplitudes for the  $x$  and  $y$  axes that reflect the sensitivity mismatch;
- $V_{x0}$  and  $V_{y0}$  are the voltage offsets for the  $x$  and  $y$  axes;
- $\beta$  is the quadrature error.

The block uses this equation to decode the angle:

$$\theta_{out} = \arctan\left(\frac{v_y}{v_x}\right).$$



## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **R — Rotational velocity for the sensor positive probe**

mechanical rotational

Mechanical rotational conserving port associated with the sensor positive probe.

#### **C — Rotational velocity for the sensor negative probe**

mechanical rotational

Mechanical rotational conserving port associated with the sensor negative (reference) probe.

#### **yp — y-axis positive Hall field**

electrical

Electrical conserving port associated with the y-axis positive Hall field.

### Dependencies

This port is visible only if you select `Electrical` connections for the **Output interface** parameter in the **Parameters** settings.

#### **yn — y-axis negative Hall field**

electrical

Electrical conserving port associated with the y-axis negative Hall field.

### Dependencies

This port is visible only if you select `Electrical` connections for the **Output interface** parameter in the **Parameters** settings.

#### **xp — x-axis positive Hall field**

electrical

Electrical conserving port associated with the x-axis positive Hall field.

### Dependencies

This port is visible only if you select `Electrical` connections for the **Output interface** parameter in the **Parameters** settings.

**xn — x-axis negative Hall field**

electrical

Electrical conserving port associated with the x-axis negative Hall field.

**Dependencies**

This port is visible only if you select `Electrical connections` for the **Output interface** parameter in the **Parameters** settings.

**Output****Angle — Rotational angle**

physical signal

Physical signal output port for the decoded position in terms of the rotational angle of the magnetic field in the x-y plane.

**Dependencies**

This port is visible only if you select `Decoded angular position` for the **Output interface** parameter in the **Parameters** settings.

**Parameters****X-axis voltage amplitude — x-axis voltage**

2.5 V (default)

X-axis voltage amplitude, in V.

**Y-axis voltage amplitude — y-axis voltage**

2.5 V (default)

Y-axis voltage amplitude, in V.

**X-axis voltage offset — x-axis voltage offset**

0 V (default)

Potential difference in the electric current flow through the electric conductor in the absence of external magnetic field on the x-axis, in V.

**Y-axis voltage offset — Y-axis voltage offset**

0 V (default)

Potential difference in the electric current flow through the electric conductor in the absence of external magnetic field on the y-axis, in V.

**Quadrature error — Quadrature error**

0 deg (default)

Amount of quadrature error. Quadrature switching can decrease offset errors.

**Output interface — Interface for decoded information**

Electrical connections (default) | Decoded angular position

Port interface for decoded information.

**Dependencies**

The **yp**, **yn**, **xp**, and **xn** conserving ports are visible only if you select `Electrical connections` for the **Output interface** parameter.

The **Angle** physical signal port is visible only if you select `Decoded angular position` for the **Output interface** parameter.

**Version History**

Introduced in R2019b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Inductive Rotor Position Sensor

# H-Bridge

H-bridge motor driver

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The H-Bridge block represents an H-bridge motor driver. The block has the following two **Simulation mode** options:

- **PWM** — The H-Bridge block output is a controlled voltage that depends on the input signal at the **PWM** port. If the input signal has a value greater than the **Enable threshold voltage** parameter value, the H-Bridge block output is on and has a value equal to the value of the **Output voltage amplitude** parameter. If it has a value less than the **Enable threshold voltage** parameter value, the block maintains the load circuit using one of the following three **Freewheeling mode** options:
  - Via one semiconductor switch and one freewheeling diode
  - Via two freewheeling diodes
  - Via two semiconductor switches and one freewheeling diode

The first and third options are sometimes referred to as synchronous operation.

The signal at the **REV** port determines the polarity of the output. If the value of the signal at the **REV** port is less than the value of the **Reverse threshold voltage** parameter, the output has positive polarity; otherwise, it has negative polarity.

- **Averaged** — This mode has two **Load current characteristics** options:
  - Smoothed
  - Unsmoothed or discontinuous

The Smoothed option assumes that the current is practically continuous due to load inductance. In this case, the H-Bridge block output is:

$$\frac{V_O V_{PWM}}{A_{PWM}} - I_{OUT} R_{ON}$$

where:

- $V_O$  is the value of the **Output voltage amplitude** parameter.
- $V_{PWM}$  is the value of the voltage at the **PWM** port.
- $A_{PWM}$  is the value of the **PWM signal amplitude** parameter.
- $I_{OUT}$  is the value of the output current.

- $R_{ON}$  is the **Bridge on resistance** parameter.

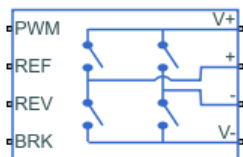
The current will be smooth if the PWM frequency is large enough. Synchronous operation where freewheeling is via a bridge arm back to the supply also helps smooth the current. For cases where the current is not smooth, or possibly discontinuous (that is, it goes to zero between PWM cycles), use the **Unsmoothed** or **discontinuous** option. For this option, you must also provide values for the **Total load series resistance**, **Total load series inductance**, and **PWM frequency**. 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.

Set the **Simulation mode** parameter to **Averaged** to speed up simulations when driving the H-Bridge block with a Controlled PWM Voltage block. You must also set the **Simulation mode** parameter of the Controlled PWM Voltage block to **Averaged** mode. This applies the average of the demanded PWM voltage to the motor. The accuracy of the **Averaged** mode simulation results relies on the validity of your assumption about the load current. If you specify that the current is **Unsmoothed** or **discontinuous**, then the accuracy also depends on the values you provide for load resistance and inductance being representative. This mode also makes some simplifying assumptions about the underlying equations for the case when current is discontinuous. For typical motor and bridge parameters, accuracy should be within a few percent. To verify **Averaged** mode accuracy, run the simulation using the **PWM** mode and compare the results to those obtained from using the **Averaged** mode.

Braking mode is invoked when the voltage presented at the **BRK** port is larger than the **Braking threshold voltage**. Regardless of whether in PWM or **Averaged** mode, when in braking mode the H-bridge is modeled by a series combination of two resistances  $R1$  and  $R2$  where:

- $R1$  is the resistance of a single bridge arm, that is, half the value of the **Total bridge on resistance** parameter.
- $R2$  is the resistance of a single bridge arm in parallel with a diode resistance, that is,  $R1 \cdot Rd / (R1 + Rd)$ , where  $Rd$  is the diode resistance.

### External Power Supply



To model the demands placed on the DC supply, you can choose to expose the power supply ports of the H-Bridge block by setting the **Power supply** parameter to **External**. If the power supply ports are exposed, then only PWM simulation mode is supported.

**Note** If the **Power supply** parameter is set to **External**, the block is able to figure out the load-side voltage offset by referencing to the supply rail voltages. However, if the supply rail connections are not exposed, you must add an Electrical Reference block on the load side.

## Model Thermal Effects

This block has one optional thermal port. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

When you expose the thermal port:

- 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 you can set from the **Thermal Port** parameters.
- The bridge on-resistance and freewheeling diode resistance become functions of temperature. You can define the values for these resistances and the second measurement temperature from the **Temperature Dependence** parameters. 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.

## Assumptions and Limitations

- If you are linearizing your model, set the **Simulation mode** parameter to **Averaged** and ensure that you have specified the operating point correctly. You can only linearize the H-Bridge block for duty cycles that are greater than zero and less than the PWM signal amplitude. If you need to linearize around zero duty cycle, and if your controller always uses regenerative braking mode, set the **Regenerative braking** parameter to **Always enabled** (suitable for linearization).
- In **Averaged** mode, and with the **Unsmoothed** or **discontinuous** choice for **Load current characteristics**, you must provide representative values for load inductance and resistance. If driving a DC Motor, then the resistance is the armature resistance, and the inductance is the sum of the armature inductance plus series smoothing inductor (if present). For a Universal Motor, total resistance is the sum of the armature and field windings, and total inductance is the sum of armature and field inductances plus any series smoothing inductance. For a Shunt Motor, MathWorks recommends that you draw a Thévenin equivalent circuit to determine appropriate values.
- No forward voltage is modeled for the freewheeling diodes. They are approximated as ideal resistances when forward biased, with resistance equal to the **Freewheeling diode on resistance** parameter value.
- If the supply rail connections are exposed, only the PWM simulation mode is supported.

## Ports

### Conserving

+ — **Positive load connection port**  
electrical

Electrical conserving port associated with the positive load connection.

- — **Negative load connection port**  
electrical

Electrical conserving port associated with the negative load connection.

#### **V+ — Positive supply connection**

electrical

Electrical conserving port associated with the positive electrical supply connection.

#### **Dependencies**

This port is visible only when you select External for the **Power supply** parameter.

#### **V- — Negative supply connection**

electrical

Electrical conserving port associated with the negative electrical supply connection.

#### **Dependencies**

This port is visible only when you select External for the **Power supply** parameter.

#### **PWM — Pulse-width modulated signal**

electrical

Electrical conserving port associated with the pulse-width modulated signal. The voltage is defined relative to the **REF** port.

#### **REF — Reference**

electrical

Electrical conserving port associated with the floating zero volt reference.

#### **REV — REV**

electrical

Electrical conserving port associated with the voltage that controls when to reverse the polarity of the H-Bridge block output. The voltage is defined relative to the **REF** port.

#### **BRK — BRK**

electrical

Electrical conserving port associated with the voltage that controls when to short circuit the H-Bridge block output. The voltage is defined relative to the **REF** port.

#### **H — Thermal port**

thermal

Thermal conserving port.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

### **Parameters**

#### **Modeling option — Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model thermal parameters.

### **Simulation Mode & Load Assumptions**

#### **Power supply — Visibility of power supply ports**

Internal (default) | External

Select one of the following options for the type of power supply:

- **Internal** — The power supply ports are internal and not visible. This is the default option.
- **External** — The power supply ports are exposed.

#### **Simulation mode — Simulation mode**

PWM (default) | Averaged

Select one of the following options for the type of output voltage:

- **PWM** — The output voltage is a pulse-width modulated signal. This is the default option.
- **Averaged** — The output voltage is a constant whose value is equal to the average value of the PWM signal.

#### **Freewheeling mode — Freewheeling mode**

Via one semiconductor switch and one freewheeling diode (default) | Via two freewheeling diodes | Via two semiconductor switches and one freewheeling diode

Select one of the following options for the type of H-bridge dissipation circuit:

- **Via one semiconductor switch and one freewheeling diode** — In this mode, the block controls the load by maintaining one high-side bridge arm permanently on and using the PWM signal to modulate the corresponding low-side bridge arm. This means that the block uses only one of the freewheeling diodes in completing the dissipation circuit when the bridge turns off. This option is the default.
- **Via two freewheeling diodes** — In this mode, all bridge arms are off during the bridge off-state. This means that the block dissipates the load current across the power supply by two freewheeling diodes.
- **Via two semiconductor switches and one freewheeling diode** — In this mode, the block 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.

### **Dependencies**

This parameter is visible only when you select PWM for the **Simulation mode** parameter, or when you select Averaged for the **Simulation mode** parameter and Unsmoothed or discontinuous for the **Load current characteristics** parameter.

#### **Regenerative braking — Regenerative braking**

Depends on REV flag and current sign (default) | Always enabled (suitable for linearization)

Select one of the following options for the type of H-bridge dissipation circuit:



- Depends on REV flag and current sign — This option is the default.
- Always enabled (suitable for linearization) — This option can be used when the controller always sets the **REV** flag to ensure regenerative braking. It is useful if you need to linearize the model for control purposes, to ensure that the H-Bridge is linearizable around zero duty cycle demand.

#### Dependencies

This parameter is visible only when you select Averaged for the **Simulation mode** parameter.

#### Load current characteristics — Load current characteristics

Smoothed (default) | Unsmoothed or discontinuous

Select one of the following options for the type of load current:

- Smoothed — Assumes that the current is practically continuous due to load inductance. This option is the default.
- Unsmoothed or discontinuous — Use this option for cases where the current is not smooth, or possibly discontinuous (that is, it goes to zero between PWM cycles). For this option, you must also provide values for the **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.

#### Dependencies

This parameter is only visible when you select Averaged for the **Simulation mode** parameter.

#### Load total series resistance — Load total series resistance

100hm (default)

The total load series resistance seen by the H-bridge.

#### Dependencies

This parameter is only visible when you select Averaged for the **Simulation mode** parameter and Unsmoothed or discontinuous for the **Load current characteristics** parameter.

#### Load total series inductance — Load total series inductance

1e-5H (default)

The total load series inductance seen by the H-bridge. As well as motor inductance, you should include any series inductance added external to the motor to smooth current.

#### Dependencies

This parameter is only visible when you select Averaged for the **Simulation mode** parameter and Unsmoothed or discontinuous for the **Load current characteristics** parameter.

#### PWM frequency — PWM Frequency

10kHz (default)

The PWM frequency at which the H-bridge is driven. For consistency, this should be the same value as the PWM frequency specified by the Controlled PWM Voltage block driving the H-Bridge block.

**Dependencies**

This parameter is only visible when you select Averaged for the **Simulation mode** parameter and Unsmoothed or discontinuous for the **Load current characteristics** parameter.

**Freewheeling diode off-state conductance — Freewheeling diode off-state conductance**

1e-6S (default)

The conductance of the reverse-biased freewheeling diode.

**Dependencies**

This parameter is visible only when you select External for the **Power supply** parameter or Averaged for the **Simulation mode** parameter and Depends on REV flag and current sign for the **Regenerative braking** parameter.

**Input Thresholds****Enable threshold voltage — Enable threshold voltage**

2.5V (default)

Threshold above which the voltage at the **PWM** port must rise to enable the H-Bridge block output.

**Dependencies**

This parameter is used only when **Simulation mode**, a **Simulation Mode & Load Assumptions** parameter, is set to PWM.

**PWM signal amplitude — PWM signal amplitude**

5.0V (default)

The amplitude of the signal at the PWM input.

**Dependencies**

The H-Bridge block uses this parameter only when the **Simulation mode** parameter on the **Simulation Mode & Load Assumptions** tab is set to Averaged.

**Reverse threshold voltage — Reverse threshold voltage**

2.5V (default)

When the voltage at the **REV** port is greater than this threshold, the output polarity becomes negative.

**Braking threshold voltage — Braking threshold voltage**

2.5V (default)

When the voltage at the **BRK** port is greater than this threshold, the H-Bridge block output terminals are short-circuited via the following series of devices:

- One bridge arm
- One bridge arm in parallel with a conducting freewheeling diode

## Bridge Parameters

### Output voltage amplitude — Output voltage amplitude

12V (default)

The amplitude of the voltage across the H-Bridge block output ports when the output is on.

#### Dependencies

This parameter is visible only when you select Internal for the **Power supply** parameter.

### Total bridge on resistance — Total bridge on resistance

0.10hm (default)

The total effective resistance of the two semiconductor switches that connect the load to the two power rails when the voltage at the **PWM** port is greater than the value of the **Enable threshold voltage** parameter on the **Input Thresholds** tab.

### Freewheeling diode on resistance — Freewheeling diode on resistance

0.050hm (default)

The total resistance in the freewheeling diodes that dissipate the current that flows through the motor when the voltage at the **PWM** port is less than the value of the **Enable threshold voltage** parameter on the **Input Thresholds** tab.

### Measurement temperature — Measurement temperature

298.15K (default)

The temperature for which the resistance values on the **Bridge Parameters** tab are specified.

#### Dependencies

To enable this parameter, set **Modeling option** to Show thermal port.

#### Temperature Dependence

To enable these parameters, set **Modeling option** to Show thermal port.

### Total bridge on resistance at second measurement temperature — Total bridge on resistance at second measurement temperature

0.10hm (default)

The total effective resistance of the two semiconductor switches that connect the load to the two power rails (as described in the **Total bridge on resistance** parameter definition), quoted at the **Second measurement temperature**.

### Freewheeling diode on resistance at second measurement temperature — Freewheeling diode on resistance at second measurement temperature

0.050hm (default)

The total resistance in the freewheeling diodes that dissipate the current that flows through the motor (as described in the **Total bridge on resistance** parameter definition), quoted at the **Second measurement temperature**.

### Second measurement temperature — Second measurement temperature

398.15K (default)

The temperature for which for which the resistance values on the **Temperature Dependence** tab are specified.

### **Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Thermal mass – Thermal mass**

100J/K (default)

Thermal mass associated with the thermal port **H**. It represents the energy required to raise the temperature of the thermal port by one degree. The default value is 100 J/K.

#### **Initial temperature – Initial temperature**

298.15K (default)

The temperature of the thermal port at the start of simulation.

## **Version History**

### **Introduced in R2008a**

#### **Freewheeling mode**

*Behavior changed in R2019b*

From R2019b forward, the H-Bridge block has the option to expose the internal supply rail connections.

As a result from this change, inside a model saved in an earlier release, check if the Freewheeling mode parameter of a H-Bridge block in Averaged mode with the assumption of Unsmoothed or discontinuous current is still set to the option you require.

#### **Regenerative braking**

*Behavior changed in R2019b*

Prior to R2019b, in Averaged simulation mode, regardless of the voltage applied at the **REV** port for controlling block output polarity reversal, the H-Bridge block assumed regenerative braking. A model saved from an earlier release that contains an H-Bridge block in Averaged simulation mode will open in later releases with the **Regenerative braking** parameter set to Always enabled (suitable for parameterization) and results will be unchanged. It is recommended that you change this parameter to Depends on REV flag and current sign so that the **REV** flag is correctly interpreted if not set by the controller for regenerative braking.

## **Extended Capabilities**

### **C/C++ Code Generation**

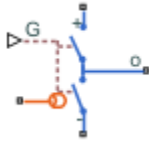
Generate C and C++ code using Simulink® Coder™.

## **See Also**

# Half-Bridge (Ideal, Switching)

Half-bridge with ideal switches and thermal port

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Half-Bridge (Ideal, Switching) block models a half-bridge with ideal switches and a thermal port. To choose the ideal switching device, set the **Switching device** parameter to either MOSFET, IGBT, or GTO.

You can specify an integral protection diode for each switching device. An integral diode protects the semiconductor device by providing a conduction path for a reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

---

**Note** The `lastReverseRecoveryLoss` variable in the simlog includes the reverse recovery losses as a pulse with amplitude equal to the energy loss. If you use a script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

---

## Equations

The protection diodes inside the half-bridge use the Lauritzen and Ma model to capture the charge dynamics effects. The defining equations are:

$$i_{RM} = \frac{q_E - q_M}{T_M} \quad (1-37)$$

$$\frac{dq_M}{dt} + \frac{q_M}{\tau} - \frac{q_E - q_M}{T_M} = 0 \quad (1-38)$$

where:

- $i_{RM}$  is the diode peak reverse current.
- $q_E$  is the junction charge.
- $q_M$  is the total stored charge.

- $T_M$  is the transit time.
- $\tau$  is the carrier lifetime.

The block solves equation 2 at  $t = \theta$  and  $q_M$  in steady-state:

$$q_M = i_{RM}\tau - \tau(i_{RM} - i_F)\exp\left(-\frac{t}{\tau}\right) = \tau i_F. \quad (1-39)$$

At  $t = \theta$  and  $q_E = \theta$ , equation 1 is equal to:

$$i_{RM} = \frac{-q_M}{T_M}. \quad (1-40)$$

The block then substitutes equation 3 into equation 4:

$$i_{RM} = \frac{-\tau}{T_M} i_F, \quad (1-41)$$

where  $i_F$  is the starting forward current when measuring  $i_{RM}$ .

Finally, the block calculates the reverse recovery energy,  $E_{rec}$ , as:

$$E_{rec} = \int_{t_1}^{t_2} i_d v_d dt = \int_{t_1}^{t_2} i_{RM} \exp\left(-\frac{t - t_1}{\tau_{rr}}\right) v_R dt, \quad (1-42)$$

where:

- $i_d$  is the current through the diode.
- $v_d$  is the voltage across the diode.
- $\tau_{rr}$  is the reverse recovery time.

Given  $t_2 = \tau_{rr} \ln(10)$ , the total reverse recovery energy is:

$$E_{rec} = -0.9 \tau_{rr} \frac{\tau}{T_M} i_F v_R = -0.9 \frac{\tau^2}{\tau + T_M} i_F v_R. \quad (1-43)$$

## Assumptions and Limitations

- The current change in the load is negligible. The inductance or the switching frequency are large enough so that the load current is constant between the switches.
- The stray inductance of the circuit is negligible.

## Ports

### Input

#### G – Gate terminals vector

physical signal

Physical signal conserving port associated with the gate terminals of the two switching devices, specified as a vector of two physical signals.

The first element of the vector controls the upper side switch. The second element of the vector controls the lower side switch. If, in the **Diode** settings, you set the **Integral protection diode** parameter to **Yes**, the first and second element of the vector also controls the lower and upper diode, respectively.

### Dependencies

To enable this port, set **Gate-control port** to **PS**.

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the negative terminal.

#### 0 — Output node

electrical

Electrical conserving port associated with the output node.

#### G1 — Gate terminal 1

electrical

Electrical conserving port associated with the gate terminal for the first switching device.

### Dependencies

To enable this port, set **Gate-control port** to **Electrical**.

#### G2 — Gate terminal 2

electrical

Electrical conserving port associated with the gate terminal for the second switching device.

### Dependencies

To enable this port, set **Gate-control port** to **Electrical**.

#### H — Thermal port

thermal

Thermal conserving port.

## Parameters

### Main

The visibility of the **Main** parameters depends on the **Switching device** and **On-state behavior and losses** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Main Parameter Dependencies**

Parameters and Options					
Gate-control port					
Switching Device					
<i>MOSFET</i>		<i>IGBT</i>		<i>GTO</i>	
Threshold voltage, Vth		Threshold voltage, Vth		Gate trigger voltage, Vgt	
				Gate turn-off voltage, Vgt_off	
				Holding current	
On-state behaviour and losses					
<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>
Drain-source on resistance, R_DS(on)	On-state voltage, Vds(Tj,Ids)	Forward voltage, Vf	On-state voltage, Vds(Tj,Ice)	Forward voltage, Vf	On-state voltage, Vak(Tj,Iak)
	Temperature vector, Tj	On-state resistance	Temperature vector, Tj	On-state resistance	Temperature vector, Tj
	Drain-source current vector, Ids		Collector-emitter current vector, Ice		Anode-cathode current vector, Iak
Off-state conductance					

**Gate control port – Gate control port**

PS (default) | Electrical

Whether to use the physical signal input port, **G**, or the electrical conserving ports **G1** and **G2**, as the gate control ports.

**Switching device – Switching device option**

MOSFET (default) | IGBT | GTO

Switching device to use for the half-bridge.

**Threshold voltage, Vth – Threshold voltage**

2 V | 6 V | positive scalar

Threshold voltage at which the device turns on. The default value depends on the **Switching device** setting.

**Dependencies**

See the Main Parameter Dependencies table.

**Gate trigger voltage, Vgt – Gate trigger voltage**

6 V (default) | positive scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.



**Dependencies**

See the Main Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off — Gate turn-off voltage**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Main Parameter Dependencies table.

**Holding current — Current threshold**

1 A (default) | positive scalar

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state behaviour and losses — On-state current for loss data**

Specify constant values (default) | Tabulate with temperature and current

Parameterization method, specified as either:

- **Specify constant values** — Use scalar values to specify the output current, switch-on loss, and switch-off loss data.
- **Tabulate with temperature and current** — Use vectors to specify the output current, switch-on loss, switch-off loss, and temperature data.

**Dependencies**

See the Main Parameter Dependencies table.

**Drain-source on resistance, R\_DS(on) — Drain-source on resistance**

0.01 Ohm (default)

Drain-source resistance when the device is on.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage, Vds(Tj,Ids) — On-state voltage**

[0, 1.1, 1.3, 1.45, 1.75, 2.25, 2.7; 0, 1, 1.15, 1.35, 1.7, 2.35, 3] V (default) | vector of positive scalars

Voltage drop across the device when the device is in a triggered conductive state. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Main Parameter Dependencies table.

**Drain-source current vector,  $I_{ds}$  — Drain-source current vector**

[ 0 10 50 100 200 400 600 ] A (default) | vector of positive scalars

Drain-source currents for which the on-state voltage is defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Forward voltage,  $V_f$  — Forward voltage**

0.8 V (default) | positive scalar

Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of the **On-state resistance** parameter.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state resistance — On-state collector-emitter resistance**

0.001 Ohm (default) | positive scalar

Collector-emitter resistance when the device is on.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage,  $V_{ds}(T_j, I_{ce})$  — On-state voltage**

[0, .1, .6, .8, 1, 1.3, 1.6, 2, 2.4; 0, .1, .7, 1, 1.2, 1.5, 1.9, 2.4, 2.8] V (default) | vector of positive scalars

Voltage drop across the device when the device is in a triggered conductive state. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Main Parameter Dependencies table.

**Collector-emitter current vector,  $I_{ce}$  — Collector-emitter current vector**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default) | vector of positive scalars

Collector-emitter currents for the on-state voltage. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage,  $V_{ak}(T_j, I_{ak})$  — On-state voltage**

[0, .1, .6, .8, 1, 1.3, 1.6, 2, 2.4; 0, .1, .7, 1, 1.2, 1.5, 1.9, 2.4, 2.8] V (default) | vector of positive scalars

Voltage drop across the device when the device is in a triggered conductive state. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Main Parameter Dependencies table.

**Anode-cathode current vector,  $I_{ak}$  — Anode-cathode current vector**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default) | vector of positive scalars

Anode-cathode currents for which the on-state voltage is defined. The first element must be zero.

**Dependencies**

See the Main Parameter Dependencies table.

**Temperature vector,  $T_j$  — Temperature vector**

[298.15, 398.15] K (default) | vector of positive scalars

Temperature values at which the on-state voltage is specified.

**Dependencies**

See the Main Parameter Dependencies table.

**Off-state conductance — Off-state collector-emitter conductance**

1e-6 1/Ohm (default) | positive scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of the **On-state resistance** parameter.

**Dependencies**

See the Main Parameter Dependencies table.

**Losses**

The visibility of the **Losses** parameters depends on the **Switching device** and **On-state behavior and losses** parameters in the **Main** tab. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Losses Parameter Dependencies**

Parameters and Options					
Switching Device					
MOSFET		IGBT		GTO	
On-state behaviour and losses					
Specify constant values	Tabulate with temperature and current	Specify constant values	Tabulate with temperature and current	Specify constant values	Tabulate with temperature and current
Switch-on loss	Switch-on loss, Eon(Tj,Ids)	Switch-on loss	Switch-on loss, Eon(Tj,Ice)	Switch-on loss	Switch-on loss, Eon(Tj,Iak)
Switch-off loss	Switch-off loss, Eoff(Tj,Ids)	Switch-off loss	Switch-off loss, Eoff(Tj,Ice)	Forced commutation switch-off loss	Switch-off loss, Eoff(Tj,Iak)
Diode reverse recovery loss	Diode reverse recovery loss, Erec(Tj,Ids)	Diode reverse recovery loss	Diode reverse recovery loss, Erec(Tj,Ice)	Diode reverse recovery loss	Diode reverse recovery loss, Erec(Tj,Iak)
On-state current for loss data	Temperature vector for losses, Tj	On-state current for loss data	Temperature vector for losses, Tj	On-state current for loss data	Temperature vector for losses, Tj
	Drain-source current vector for losses, Ids		Collector-emitter current vector for losses, Ice	Natural commutation rectification loss	Anode-cathode current vector for losses, Iak
					Natural commutation rectification loss
Off-state voltage for loss data					

**Switch-on loss — Switch-on loss**

0.02286 J (default) | positive scalar

Energy dissipated during a single switch-on event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-off loss — Switch-off loss**

0.01714 J (default) | positive scalar

Energy dissipated during a single switch-off event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Diode reverse recovery loss — Loss from diode-reverse recovery**

5.70e-3 J (default) | positive scalar

Energy dissipated during a diode-reverse recovery event.

**Dependencies**

See the Losses Parameter Dependencies table.

**Off-state voltage for loss data — Off-state voltage**

300 V (default) | positive scalar

Output voltage when the device is off. The loss data define the blocking voltage at this value.

**Dependencies**

See the Losses Parameter Dependencies table.

**On-state current for loss data — Output current**

600 A (default) | positive scalar

Output current for which the switch-on loss, switch-off loss, and on-state voltage are defined.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-on loss,  $E_{on}(T_j, I_{ds})$  — Switch-on loss**

[ 0 2.9e-4 0.00143 0.00286 0.00571 0.01314 0.02286; 0 5.7e-4 0.00263 0.00514 0.01029 0.02057 0.03029 ] J (default) | vector of positive scalars

Energy dissipated during a single switch-on event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-off loss,  $E_{off}(T_j, I_{ds})$  — Switch-off loss**

[0, .21, 1.07, 2.14, 4.29, 9.86, 17.14; 0, .43, 1.97, 3.86, 7.71, 15.43, 22.71] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Diode reverse recovery loss,  $E_{rec}(T_j, I_{ds})$  — Loss from diode-reverse recovery**

[0, .0031, .021, .11, .21, .43, .99, 1.71, 3.078; 0, .0041, .043, .2, .39, .77, 1.54, 2.27, 4.08] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a diode-reverse recovery event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Drain-source current vector for losses,  $I_{ds}$  — Drain-source current for switching losses**

[ 0 10 50 100 200 400 600 ] A (default) | vector of positive scalars

Drain-source currents for which the losses are defined. The first element must be zero.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-on loss,  $E_{on}(T_j, I_{ce})$  — Switch-on loss**

[0, .0024, .024, .12, .2, .48, 1.04, 2.16, 3.24; 0, .003, .03, .15, .25, .6, 1.3, 2.7, 4.05] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a single switch-on event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-off loss,  $E_{off}(T_j, I_{ce})$  — Switch-off loss**

[0, .0007, .0066, .033, .066, .17, .33, .83, 1.5; 0, .001, .01, .05, .1, .25, .5, 1.2, 2.2] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a single switch-off event. This parameter is a function of temperature and final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Diode reverse recovery loss,  $E_{rec}(T_j, I_{ce})$  — Loss from diode-reverse recovery loss**

[0, .0031, .021, .11, .21, .43, .99, 1.71, 3.078; 0, .0041, .043, .2, .39, .77, 1.54, 2.27, 4.08] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a diode-reverse recovery event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Collector-emitter current vector for losses,  $I_{ce}$  — Collector-emitter current for losses**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default) | vector of positive scalars

Collector-emitter currents for which the losses are defined. The first element must be zero.

**Dependencies**

See the Losses Parameter Dependencies table.

**Forced commutation switch-off loss — Forced commutation switch-off loss**

17.14e-3 J (default) | positive scalar

Energy dissipated during a forced commutation switch-off event. This parameter is a function of temperature and final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Natural commutation rectification loss — Natural commutation rectification loss**

10e-3 J (default) | positive scalar

Rectification loss applied when the block switches off because the current falls below the value of the **Holding current** parameter.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-on loss,  $E_{on}(T_j, I_{ak})$  — Switch-on loss**

[0, .0024, .024, .12, .2, .48, 1.04, 2.16, 3.24; 0, .003, .03, .15, .25, .6, 1.3, 2.7, 4.05] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a single switch-on event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Switch-off loss,  $E_{off}(T_j, I_{ak})$  — Switch-off loss**

[0, .0007, .0066, .033, .066, .17, .33, .83, 1.5; 0, .001, .01, .05, .1, .25, .5, 1.2, 2.2] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a single switch-off event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Diode reverse recovery loss,  $E_{rec}(T_j, I_{ak})$  — Loss from diode-reverse recovery loss**

[0, .0031, .021, .11, .21, .43, .99, 1.71, 3.078; 0, .0041, .043, .2, .39, .77, 1.54, 2.27, 4.08] \* 1e-3 J (default) | vector of positive scalars

Energy dissipated during a diode-reverse recovery event. This parameter is a function of temperature and the final on-state output current.

**Dependencies**

See the Losses Parameter Dependencies table.

**Anode-cathode current vector for losses,  $I_{ak}$  — Anode-cathode current for losses**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default) | vector of positive scalars

Anode-cathode currents for which the losses are defined. The first element must be zero.

**Dependencies**

See the Losses Parameter Dependencies table.

**Temperature vector for losses, Tj — Temperature for losses**

[298.15, 398.15] K (default) | vector of positive scalars

Temperature values at which the losses are specified. Specify this parameter using a vector quantity.

**Dependencies**

See the Losses Parameter Dependencies table.

**Temperature vector for losses, Tj — Temperature for losses**

[298.15, 398.15] K (default) | vector of positive scalars

Temperature values at which the losses are specified. Specify this parameter using a vector quantity.

**Dependencies**

See the Losses Parameter Dependencies table.

**Off-state voltage for loss data — Off-state voltage for losses data**

300 V (default) | positive scalar

Output voltage when the device is off. The loss data define the blocking voltage at this value.

**Dependencies**

See the Losses Parameter Dependencies table.

**Integral Diode****Integral protection diode — Protection diode**

No (default) | Yes

Whether to model the block integral protection diode.

**Diode model — Diode model**

Piecewise Linear (default) | Tabulated I-V curve

Diode model, specified as either:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data and fixed reverse bias off conductance.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes.

**Forward voltage — Forward voltage**

0.8 V (default) | positive scalar

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of the **On resistance** parameter.



**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes and **Diode model** to Piecewise linear.

**On resistance — On resistance**

0.001 Ohm (default) | positive scalar

Rate of change of the voltage versus the current above the value of the **Forward voltage** parameter.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes and **Diode model** to Piecewise linear.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes and **Diode model** to Piecewise linear.

**Table type — Tabulated function**

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes and **Diode model** to Tabulated I-V curve.

**Forward currents,  $I_f(T_j, V_f)$  — Forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12]

A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes, **Diode model** to Tabulated I-V curve, and **Table type** to Table in  $I_f(T_j, V_f)$  form.

**Junction temperatures,  $T_j$  — Junction temperatures**

[25, 125] degC (default) | vector of positive scalars

Vector of junction temperatures. This parameter must be a vector of at least two elements.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes, and **Diode model** to Tabulated I-V curve.

**Forward voltages, Vf — Forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default) | nonnegative vector

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes, **Diode model** to Tabulated I-V curve, and **Table type** to Table in If(Tj,Vf) form.

**Forward voltages, Vf(Tj,If) — Forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes, **Diode model** to Tabulated I-V curve, and **Table type** to Table in Vf(Tj,If) form.

**Forward currents, If — Forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default) | nonnegative vector

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Yes, **Diode model** to Tabulated I-V curve, and **Table type** to Table in Vf(Tj,If) form.

**Thermal Port**

Use the thermal port to simulate the effects of generated heat and device temperature. For more information about using thermal ports and the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Version History

Introduced in R2021b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

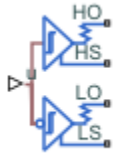
**See Also**

Diode | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Thyristor (Piecewise Linear)

# Half-Bridge Driver

Behavioral model of half-bridge driver integrated circuit

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Half-Bridge Driver block provides an abstracted representation of an integrated circuit for driving MOSFET and IGBT half-bridges. The block models input hysteresis, propagation delay, and turn-on/turn-off dynamics. Unless modeling a gate driver circuit explicitly, always use this block or the Gate Driver block 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.

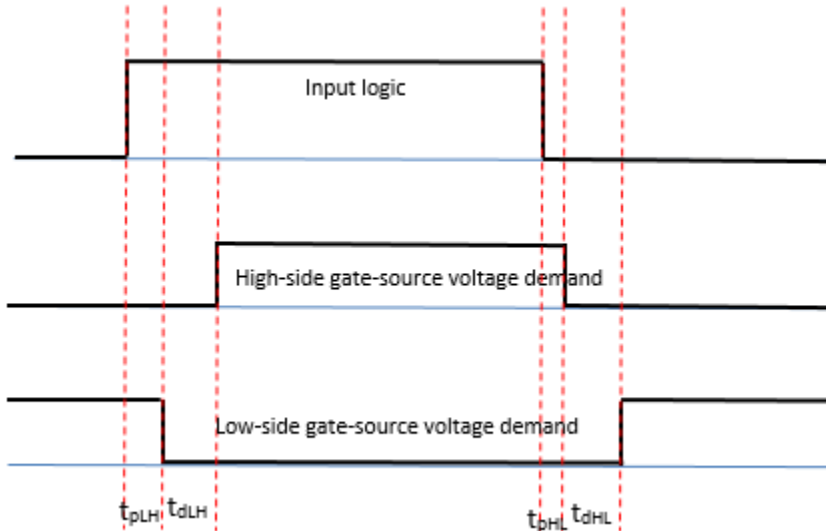
You can model electrical or physical signal input ports by setting the **Modeling option** parameter to either:

- **PS input** — The driver output state is controlled by a physical signal input  $u$ . Use this modeling option if all of your controller, including PWM waveform generation, is determined by Simulink blocks. This modeling option is the default.
- **Electrical input ports** — The driver output state is controlled by two electrical input connections, PWM and REF. Use this option if your model has upstream analog components, such as the Controlled PWM Voltage source.

The first pair of output electrical ports, HO and HS, behave in the same way as the G and S ports of the Gate Driver block. Connect these ports to the high-side MOSFET or IGBT of the half-bridge. The second pair of ports, LO and LS, connect to the low-side MOSFET or IGBT of the half-bridge. They behave in a similar way, except that their logic is inverted with respect to that of the high side.

The diagram shows the timing properties for the half-bridge driver, where:

- $t_{pLH}$  is low-side propagation delay when the input logic goes from 0 to 1.
- $t_{dLH}$  is high-side dead time when the input logic goes from 0 to 1.
- $t_{pHL}$  is high-side propagation delay when the input logic goes from 1 to 0.
- $t_{dHL}$  is low-side dead time when the input logic goes from 1 to 0.



## Faults

You can insert a fault into one or both of the outputs at a specified simulation time. The fault options are:

- Fail input fixed at logic 0
- Fail input fixed at logic 1
- Fail high side off
- Fail high side on
- Fail low side off
- Fail low side on
- Fail high and low sides off
- Fail high and low sides on

## Ports

### Input

**u** — Control signal, unitless

physical signal

Input physical signal that specifies the input control value.

### Dependencies

To enable this port, set **Modeling option** to PS input.

### Conserving

**PWM** — Pulse-width modulated signal

electrical

Electrical conserving port associated with the pulse-width modulated signal.

**Dependencies**

To enable this port, set **Modeling option** to `Electrical input ports`.

**REF — Floating zero volt reference**

`electrical`

Electrical conserving port associated with the floating zero volt reference.

**Dependencies**

To enable this port, set **Modeling option** to `Electrical input ports`.

**H0 — High-side gate connection**

`electrical`

Electrical conserving port associated with the gate on the high side of the half bridge. Connect this port to the gate of a MOSFET or IGBT block.

**HS — High-side source or emitter connection**

`electrical`

Electrical conserving port associated with the source or emitter on the high side of the half bridge. Connect this port to the source of a MOSFET block or the emitter of an IGBT block.

**L0 — Low-side gate connection**

`electrical`

Electrical conserving port associated with the gate on the low side of the half bridge. Connect this port to the gate of a MOSFET or IGBT block.

**LS — Low-side source or emitter connection**

`electrical`

Electrical conserving port associated with the source or emitter on the low side of the half bridge. Connect this port to the source of a MOSFET block or the emitter of an IGBT block.

**Parameters****Modeling option — Whether to model electrical or physical signal input ports**

`PS input (default) | Electrical input ports`

Whether to model physical signal or electrical input ports.

**Input Logic****Logic 1 input value — Signal value for logic 1**

`0.7 (default)`

Value of the input signal corresponding to the logic 1 level.

**Dependencies**

To enable this port, set **Modeling option** to `PS input`.

**Logic 0 input value — Signal value for logic 0**

0.3 (default)

Value of the input signal corresponding to the logic 0 level.

**Dependencies**To enable this port, set **Modeling option** to PS input.**Logic 1 input voltage — Voltage value for logic 1**

2.0 V (default)

Value of the input voltage corresponding to the logic 1 level.

**Dependencies**To enable this port, set **Modeling option** to Electrical input ports.**Logic 0 input voltage — Voltage value for logic 0**

0.8 V (default)

Value of the input voltage corresponding to the logic 0 level.

**Dependencies**To enable this port, set **Modeling option** to Electrical input ports.**Outputs****On-state gate-source voltage — Demanded output voltage for on state**

15 V (default)

Demanded output voltage when the driver is in on state.

**Off-state gate-source voltage — Demanded output voltage for off state**

0 V (default)

Demanded output voltage when the driver is in off state.

**Timing****Low-side propagation delay (logic 0->logic 1) —  $t_{pLH}$  in timing diagram**

50 ns (default)

Low-side propagation delay when the input logic goes from 0 to 1.

**High-side dead time (logic 0->logic 1) —  $t_{dLH}$  in timing diagram**

100 ns (default)

High-side dead time when the input logic goes from 0 to 1.

**High-side propagation delay (logic 1->logic 0) —  $t_{pHL}$  in timing diagram**

50 ns (default)

High-side propagation delay when the input logic goes from 1 to 0.

**Low-side dead time (logic 1->logic 0) —  $t_{dHL}$  in timing diagram**

100 ns (default)

Low-side dead time when the input logic goes from 1 to 0.

## Dynamics

### Parameterization — Select driver parameterization

Output impedance (default) | Rise and fall times

Select the type of driver parameterization:

- Output impedance — Specify on-state and off-state gate drive resistances.
- Rise and fall times — Specify rise time, fall time, and load capacitance.

### On-state gate drive resistance — Demanded output voltage for on state

2  $\Omega$  (default)

Demanded output voltage when the driver is in on state.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Output impedance.

### Off-state gate drive resistance — Demanded output voltage for off state

2  $\Omega$  (default)

Demanded output voltage when the driver is in off state.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Output impedance.

### Rise time — Driver rise time

20 ns (default)

Driver rise time from 10% to 90%.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Rise and fall times.

### Fall time — Driver fall time

20 ns (default)

Driver fall time from 90% to 10%.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Rise and fall times.

### Load capacitance for rise and fall times — Driver load capacitance

10 nF (default)

Driver load capacitance.

#### Dependencies

Enabled when the **Parameterization** parameter is set to Rise and fall times.

## Faults

### Enable faults — Select Yes to enable faults modeling

No (default) | Yes

Select Yes to enable faults modeling. The associated parameters in the **Faults** section become visible to let you specify time to fail and the failure mode.

### Simulation time for fault event — Time before entering faulted state

1 s (default)

Set the simulation time at which you want the block to enter the faulted state.

### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

### Failure mode — State of driver after failure

Fail input fixed at logic 0 (default) | Fail input fixed at logic 1 | Fail high side off | Fail high side on | Fail low side off | Fail low side on | Fail high and low sides off | Fail high and low sides on

Select the driver state after the failure.

### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Controlled PWM Voltage | Gate Driver | N-Channel IGBT | N-Channel MOSFET | P-Channel MOSFET



# Heatsink

Dissipate heat from power semiconductors to ambient temperature

**Library:** Simscape / Electrical / Passive / Thermal



## Description

The Heatsink block models a heatsink that dissipates heat from power semiconductors. The heat from the case conducts through the fins and dissipates to the ambient temperature through convection. The environment and the working fluid are the same.

You can parameterize this block from a datasheet, from tabulated heat transfer properties, or from geometry that assumes an empirical convection correlation. If you set the **Convection** parameter to Forced - specify flow speed, the **v** input port specifies the flow speed.

### Parameterization: Datasheet

To parameterize the Heatsink block from a datasheet, set the **Parameterization** parameter to Datasheet and specify both the **Vector of temperature rises above ambient, T** and **Corresponding heat dissipated to ambient, Q\_TLU1(T)** parameters.

If you enable forced convection in the system by setting **Convection** to Forced - specify flow speed, specify both the **Vector of temperature rises above ambient, T** and **Corresponding heat dissipated to ambient, Q\_TLU2(T,v)** parameters.

### Parameterization: Tabulated Convection and Fin Efficiency

To parameterize the Heatsink block based on the convection coefficient as a function of the coolant flow speed and temperature difference with the ambient temperature and the fin efficiency as a function of the convective coefficient, set the **Parameterization** parameter to Tabulated convection and fin efficiency.

The block uses this equation to calculate the dissipated heat:

$$Q_{dissipated} = h(v_{fluid}, \Delta T) \cdot A_{total} \cdot \frac{Eff(h)}{100} \cdot (T_{heatsink} - T_{ambient}),$$

where:

- $h(v_{fluid}, \Delta T)$  is the convective heat transfer coefficient tabulated against the fluid flow speed (for forced convection) and the temperature rise above the ambient temperature.
- $A_{total}$  is the total heat exchange surface area.
- $Eff(h)$  is the fin efficiency, in percent values, tabulated against different convective heat transfer coefficients. The fin efficiency is the actual heat dissipated by the fin divided by the heat it would

dissipate if all its volume was at the case temperature. This value depends on fin geometry and fin thermal conductivity.

### Parameterization: Assume Rectangular Parallel Fins

If you set the **Parameterization** parameter to Assume rectangular parallel fins, the block uses this equation to calculate the dissipated heat:

$$Q_{dissipated} = h(v_{fluid}, \Delta T) \cdot A_{total} \cdot \frac{Eff(h)}{100} \cdot (T_{heatsink} - T_{ambient}),$$

where:

- $h(v_{fluid}, \Delta T) = h_{natural}(\Delta T) + h_{forced}(v_{fluid})$
- $$h_{natural}(\Delta T) = \frac{k_{fluid}}{H} \left( 0.825 + \frac{0.387 Ra_H^{1/6}}{\left( 1 + \left( \frac{0.492}{Pr} \right)^{9/16} \right)^{8/27}} \right)^2$$
- $$h_{forced}(v_{fluid}) = \frac{k_{fluid}}{b} \left( \frac{1}{\left( \frac{RePr}{2} \right)^3} + \frac{1}{\left( 0.664 \sqrt{RePr}^{0.33} \sqrt{1 + \frac{3.65}{\sqrt{Re}}} \right)^3} \right)^{-0.33}$$
- $Ra_H = \frac{g\beta}{\nu\alpha} H^3 abs(\Delta T)$  is the Rayleigh number.
- $Re = \frac{b^2}{\nu d} abs(v_{fluid})$  is the Reynolds number.
- $g = 9.81 \text{ m/s}^2$  is the acceleration of gravity.
- $\beta$  is the coefficient of volume thermal expansion of the fluid.
- $\nu$  is the fluid kinematic viscosity.
- $\alpha$  is the fluid thermal diffusivity.
- $k_{fluid}$  is the fluid thermal conductivity.
- $H$  is the fin height.
- $d$  is the fin depth.
- $t$  is the fin thickness.
- $b$  is the gap between fins.

If one side of size  $t \times d$  is welded in the base, this equation calculates the total heat exchange area:

$$A_{total} = (2H(d + t) + dt)N.$$

Then this equation computes the efficiency of a rectangular fin:

$$Eff(h) = 100 \frac{\tanh\left(\sqrt{\frac{2h}{kt}} H\right)}{\sqrt{\frac{2h}{kt}} H}$$

where  $k$  is the fin thermal conductivity.

## Ports

### Input

#### **v – Flow speed**

physical

Physical input port associated with the flow speed.

#### **Dependencies**

To enable this port, set **Convection** to Forced - specify flow speed.

### Conserving

#### **A – Ambient temperature**

thermal

Thermal conserving port associated with the ambient temperature.

#### **C – Case**

thermal

Thermal conserving port associated with the case.

## Parameters

### Steady State

#### **Parameterization – Parameterization type**

Assume rectangular parallel fins (default) | Datasheet | Tabulated convection and fin efficiency

Whether to parameterize the block as a function of rectangular parallel fins, datasheet, or tabulated convection and fin efficiency.

#### **Convection – Convection type**

Natural (default) | Forced - specify flow speed

Whether the block models natural convection or uses convection values provided by an input port.

#### **Fin height – Fin height**

38.1e-3 m (default) | positive scalar

Height of the fin.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins.

#### **Fin thickness – Fin thickness**

6.5e-4 m (default) | positive scalar

Thickness of the fin.

**Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins.

**Fin depth — Fin depth**

139.7e-3 m (default) | positive scalar

Depth of the fin.

**Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins.

**Gap between fins — Gap between fins**

9.4e-3 m (default) | positive scalar

Gap between fins.

**Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins and **Convection** to Forced - specify flow speed.

**Number of fins — Number of fins**

11 (default) | positive scalar

Number of fins. The value of this parameter must be equal to or greater than 1.

**Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins.

**Fin thermal conductivity — Fin thermal conductivity**

237 W/(m\*K) (default) | positive scalar

Thermal conductivity of the fins.

**Dependencies**

To enable this parameter, set **Parameterization** to Assume rectangular parallel fins.

**Vector of temperature rises above ambient, T — Vector of temperature rises above ambient temperature**

[10, 30, 50, 70, 90] K (default) | vector of positive scalars

Vector of temperature rises above the ambient temperature. The values in this parameter must be positive and strictly ascending.

**Dependencies**

To enable this parameter, set **Parameterization** to Datasheet or Tabulated convection and fin efficiency.

**Vector of fluid flow speed, v — Vector of fluid flow speed**

[0, 1, 2, 3] m/s (default) | vector of positive scalars

Vector of fluid flow speed. The values in this parameter must be positive and strictly ascending.

**Dependencies**

To enable this parameter, set **Parameterization** to Datasheet or Tabulated convection and fin efficiency, and **Convection** to Forced - specify flow speed.

**Corresponding heat dissipated to ambient,  $Q_{TLU1}(T)$  — Corresponding heat dissipated to ambient against temperature**

[6.1, 23.6, 44.5, 67.5, 92.3] W (default) | vector of positive scalars

Vector of heat values dissipated to the ambient temperature tabulated against the temperature rise above the ambient temperature. The values in this parameter correspond to the values in the **Vector of temperature rises above ambient, T** parameter. The values in this parameter must be positive and ascending.

**Dependencies**

To enable this parameter, set **Parameterization** to Datasheet and **Convection** to Natural.

**Corresponding heat dissipated to ambient,  $Q_{TLU2}(T, v)$  — Corresponding heat dissipated to ambient against temperature and flow speed**

[6.1, 18.8, 22.8, 25.7; 23.6, 61.1, 73, 81.5; 44.5, 106.4, 125.9, 140.1; 67.5, 153.5, 180.7, 200.3; 92.3, 202.1, 236.9, 262] W (default) | vector of positive scalars

Vector of heat values dissipated to the ambient temperature tabulated against the flow speed and temperature rise above the ambient temperature. The values in this parameter correspond to the values in the **Vector of temperature rises above ambient, T** parameter.

**Dependencies**

To enable this parameter, set **Parameterization** to Datasheet and **Convection** to Forced - specify flow speed.

**Corresponding convective heat transfer coefficient,  $h_{TLU1}(T)$  — Corresponding convective heat transfer coefficient against temperature**

[5.4, 7.02, 7.98, 8.68, 9.26] W/(m<sup>2</sup>\*K) (default) | vector of positive scalars

Convective heat transfer coefficient tabulated against the temperature rise above the ambient temperature.

**Dependencies**

To enable this parameter, set **Parameterization** to Tabulated convection and fin efficiency and **Convection** to Natural.

**Corresponding convective heat transfer coefficient,  $h_{TLU2}(T, v)$  — Corresponding convective heat transfer coefficient against temperature and flow speed**

[5.4, 17.86, 22.17, 25.41; 7.02, 19.49, 23.8, 27.03; 7.97, 20.44, 24.75, 27.99; 8.68, 21.15, 25.46, 28.69; 9.26, 21.73, 26.04, 29.27] (default) | W/(m<sup>2</sup>\*K) | vector of positive scalars

Convective heat transfer coefficient tabulated against fluid flow speed caused by forced convection, and temperature rise above the ambient temperature caused by natural convection.

**Dependencies**

To enable this parameter, set **Parameterization** to Tabulated convection and fin efficiency and **Convection** to Forced - specify flow speed.

**Vector of convective heat transfer coefficients, h – Vector of convective heat transfer coefficients**

[5, 10, 15, 20, 25, 30] W/(m<sup>2</sup>\*K) (default) | vector of positive scalars

Convective heat transfer coefficients. This parameter depends on the temperature difference caused by natural convection, and the fluid flow speed caused by forced convection. The values of this parameter must be positive and strictly ascending.

**Dependencies**

To enable this parameter, set **Parameterization** to Tabulated convection and fin efficiency.

**Corresponding fin efficiency (percent), eff\_TLU(h) – Fin efficiency that corresponds to heat transfer coefficients**

[96.97, 94.16, 91.53, 89.08, 86.78, 84.62] (default) | vector of positive scalars

Fin efficiency tabulated against different convective heat transfer coefficients. Fin efficiency is the actual heat dissipated by the fin divided by the heat it would dissipate if all its volume was at case temperature. This parameter depends on fin geometry and fin thermal conductivity.

**Dependencies**

To enable this parameter, set **Parameterization** to Tabulated convection and fin efficiency.

**Total heat exchange surface area – Total heat exchange surface area**

0.1171 m<sup>2</sup> (default) | positive scalar

Total area of the surface of the heat exchange.

**Dependencies**

To enable this parameter, set **Parameterization** to Tabulated convection and fin efficiency.

**Fluid Properties**

To enable these settings, set **Parameterization** to Assume rectangular parallel fins.

**Fluid kinematic viscosity – Fluid kinematic viscosity**

1.51e-5 m<sup>2</sup>/s (default) | positive scalar

Kinematic viscosity of the fluid.

**Fluid thermal diffusivity – Fluid thermal diffusivity**

2.07e-5 m<sup>2</sup>/s (default) | positive scalar

Thermal diffusivity of the fluid.

**Fluid thermal conductivity – Fluid thermal conductivity**

25e-3 W/(m\*K) (default) | positive scalar

Thermal conductivity of the fluid.

**Fluid coefficient of volume thermal expansion — Fluid coefficient of volume thermal expansion**

3.3e-3 1/K (default) | positive scalar

Fluid coefficient of the volume thermal expansion.

**Dynamics**

**Heatsink mass — Heatsink mass**

0.35 kg (default) | positive scalar

Mass of the heatsink.

**Heatsink specific heat — Heatsink specific heat**

437 J/(K\*kg) (default) | positive scalar

Specific heat of the heatsink.

## Version History

Introduced in R2021b

## References

- [1] Churchill, Stuart W.; Chu, Humbert H.S. *Correlating equations for laminar and turbulent free convection from a vertical plate*. International Journal of Heat and Mass Transfer (November 1975): 1323-1329.
- [2] Teertstra, P., Yovanovich, M.M., and Culham, J.R.. *Analytical Forced Convection Modeling of Plate Fin Heat Sinks*. Proceedings of 15th IEEE Semi-Therm Symposium (1999): pp. 34-41.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

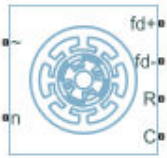
## See Also

IGBT (Ideal, Switching)

# Hybrid Excitation PMSM

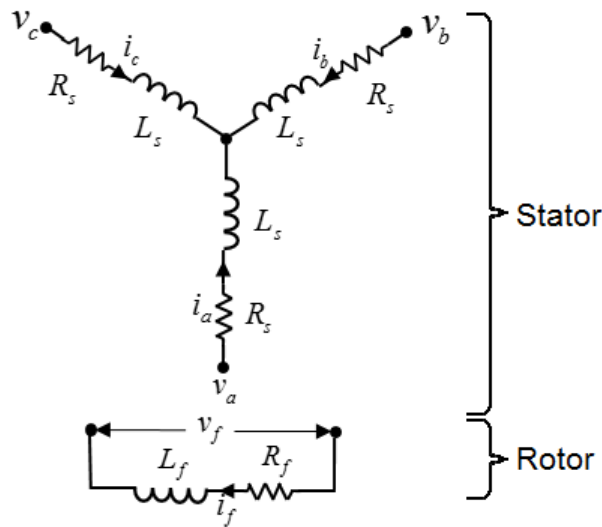
Hybrid excitation synchronous machine with three-phase wye-wound stator

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



## Description

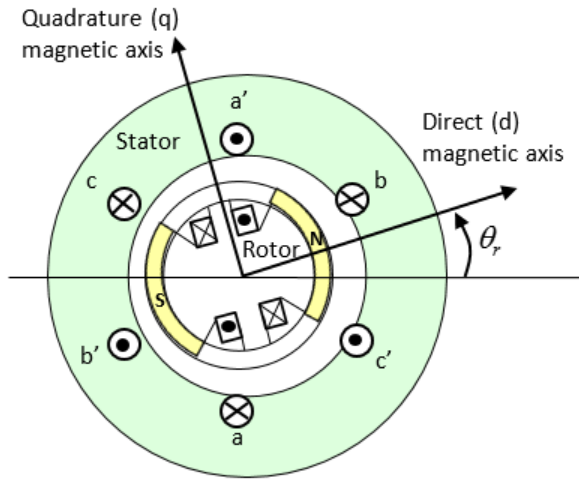
The Hybrid Excitation PMSM block represents a hybrid excitation synchronous machine with a three-phase wye-wound stator. Permanent magnets and excitation windings provide the machine excitation. The figure shows the equivalent electrical circuit for the stator and rotor windings.



## Motor Construction

The diagram shows the motor construction with a single pole-pair on the rotor. For the axes convention, when rotor mechanical angle  $\theta_r$  is zero, the  $a$ -phase and permanent magnet fluxes are aligned. The block supports a second rotor axis definition for which rotor mechanical angle is defined as the angle between the  $a$ -phase magnetic axis and the rotor  $q$ -axis.





### Equations

Voltages across the stator windings are defined by

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ , and  $\frac{d\psi_c}{dt}$  are the rates of change of magnetic flux in each stator winding.

The voltage across the field winding is expressed as

$$v_f = R_f i_f + \frac{d\psi_f}{dt},$$

where:

- $v_f$  is the individual phase voltage across the field winding.
- $R_f$  is the equivalent resistance of the field winding.
- $i_f$  is the current flowing in the field winding.
- $\frac{d\psi_f}{dt}$  is the rate of change of magnetic flux in the field winding.

The permanent magnet, excitation winding, and the three star-wound stator windings contribute to the flux linking each winding. The total flux is defined by

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} + \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix} i_f,$$

where:

- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the total fluxes linking each stator winding.
- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ ,  $L_{bc}$ ,  $L_{ca}$ , and  $L_{cb}$  are the mutual inductances of the stator windings.
- $\psi_{am}$ ,  $\psi_{bm}$ , and  $\psi_{cm}$  are the magnetization fluxes linking the stator windings.
- $L_{amf}$ ,  $L_{bmf}$ , and  $L_{cmf}$  are the mutual inductances of the field winding.

The inductances in the stator windings are functions of rotor electrical angle and are defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e),$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3)),$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_e + \pi/6)),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/6 - 2\pi/3)),$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_e + \pi/6 + 2\pi/3)),$$

where:

- $N$  is the number of rotor pole pairs.
- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the d-axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the q-axis.
- $L_s$  is the stator self-inductance per phase. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is fluctuation in self-inductance and mutual inductance with changing rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The magnetization flux linking winding,  $a-a'$  is a maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore:

$$\psi_m = \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_e - 2\pi/3) \\ \psi_m \cos(\theta_e + 2\pi/3) \end{bmatrix},$$

$$L_{mf} = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix} = \begin{bmatrix} L_{mf} \cos \theta_e \\ L_{mf} \cos(\theta_e - 2\pi/3) \\ L_{mf} \cos(\theta_e + 2\pi/3) \end{bmatrix},$$

and

$$\Psi_f = L_f i_f + L_{mf}^T \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix},$$

where:

- $\psi_m$  is the linked motor flux.
- $L_{mf}$  is the mutual field armature inductance.
- $\psi_f$  is the flux linking the field winding.
- $L_f$  is the field winding inductance.
- $[L_{mf}]^T$  is the transform of the  $L_{mf}$  vector, that is,

$$[L_{mf}]^T = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix}^T = [L_{amf} \ L_{bmf} \ L_{cmf}].$$

### Simplified Equations

Applying the Park transformation to the block electrical defining equations produces an expression for torque that is independent of rotor angle.

The Park transformation is defined by

$$P = 2/3 \begin{bmatrix} \cos \theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin \theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

Applying the Park transformation to the first two electrical defining equations produces equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} + L_{mf} \frac{di_f}{dt} - N \omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N \omega (i_d L_d + \psi_m + i_f L_{mf}),$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

$$v_f = R_f i_f + L_f \frac{di_f}{dt} + \frac{3}{2} L_{mf} \frac{di_d}{dt},$$

$$T = \frac{3}{2} N (i_q (i_d L_d + \psi_m + i_f L_{mf}) - i_d i_q L_q),$$

and

$$J \frac{d\omega}{dt} = T = T_L - B_m \omega.$$

where:

- $v_d$ ,  $v_q$ , and  $v_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence voltages. These voltages are defined by

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

- $L_d$  is the stator  $d$ -axis inductance.  $L_d = L_s + M_s + 3/2 L_m$ .
- $\omega$  is the mechanical rotational speed.
- $L_q$  is the stator  $q$ -axis inductance.  $L_q = L_s + M_s - 3/2 L_m$ .
- $L_0$  is the stator zero-sequence inductance.  $L_0 = L_s - 2M_s$ .
- $T$  is the rotor torque. For the Hybrid Excitation PMSM block, torque flows from the machine case (block conserving port **C**) to the machine rotor (block conserving port **R**).
- $J$  is the rotor inertia.
- $T_L$  is the load torque.
- $B_m$  is the rotor damping.

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and motor temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions

Flux distribution is sinusoidal.

## Ports

### Conserving

#### **R – Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

#### **C – Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

#### **~ – Three-phase composite**

electrical

Expandable three-phase port associated with the stator windings.

### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **n – Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

**Dependencies**

To enable this port, set **Zero sequence** to Include.

**fd+ – Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

**fd- – Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

**HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC – Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR – Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

### Main

#### Electrical connection — Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

#### Modeling fidelity — Modeling fidelity

Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and  $PM$  (default) | Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and  $PM$

Select the modeling fidelity:

- Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and  $PM$  —  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$ , and  $PM$  values are constant and defined by their respective parameters.
- Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and  $PM$  —  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$ , and  $PM$  values are computed online from DQ and field currents look-up tables as follows:

$$L_d = f_1(i_d, i_q, i_f)$$

$$L_q = f_2(i_d, i_q, i_f)$$

$$L_{mf} = f_3(i_d, i_q, i_f)$$

$$\lambda_{PM} = f_4(i_d, i_q, i_f)$$

$$L_f = f_5(i_f)$$

#### Dependencies

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .

#### Number of pole pairs — Rotor pole pairs

6 (default) | integer

Number of permanent magnet pole pairs on the rotor.

#### Permanent magnet flux linkage — Flux linkage

0.09 Wb (default) | positive integer

Peak permanent magnet flux linkage for any of the stator windings.

#### Dependencies

This parameter is visible only when you set the **Modeling fidelity** parameter to Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and  $PM$ .

#### Stator parameterization — Parameterization model

Specify  $L_d$ ,  $L_q$  and  $L_0$  (default) | Specify  $L_s$ ,  $L_m$ , and  $M_s$

Stator parameterization model.

**Dependencies**

The **Stator parameterization** setting affects the visibility of other parameters.

**Stator d-axis inductance, Ld – Inductance**

0.0031 H (default)

Direct-axis inductance of the machine stator.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Constant Ld, Lq, Lmf, Lf and PM.

**Stator q-axis inductance, Lq – Inductance**

0.0045 H (default)

Quadrature-axis inductance of the machine stator.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Constant Ld, Lq, Lmf, Lf and PM.

**Direct-axis current vector, iD – Direct-axis current vector**

[-200, 0, 200] A (default)

Direct-axis current vector, iD.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Quadrature-axis current vector, iQ – Quadrature-axis current vector**

[-200, 0, 200] A (default)

Quadrature-axis current vector, iQ.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Excitation current vector, iF – Excitation current vector**

[-20, 0, 20] A (default)

Excitation current vector, iF.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.



**Ld matrix,  $Ld(id,iq,if)$  – Ld matrix**

0.0031 \* ones(3, 3, 3) H (default)

Ld matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Lq matrix,  $Lq(id,iq,if)$  – Lq matrix**

0.0045 \* ones(3, 3, 3) H (default)

Lq matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Mutual field armature inductance,  $Lmf(id,iq,if)$  – Mutual field armature inductance**

0.0067 \* ones(3, 3, 3) H (default)

Mutual field armature inductance.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Permanent magnet flux linkage,  $PM(id,iq,if)$  – Permanent magnet flux linkage**

0.1 \* ones(3, 3, 3) Wb (default)

Permanent magnet flux linkage.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0, the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Lf vector,  $Lf(if)$  – Lf vector**

0.06 \* ones(1, 3) H (default)

Lf vector.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0, the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf, Lf and PM.

**Stator zero-sequence inductance,  $L0$  – Inductance**

0.0006 H (default)

Zero-axis inductance for the machine stator.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_d$ ,  $L_q$  and  $L_0$  and **Zero sequence** is set to Include.

**Stator self-inductance per phase,  $L_s$  – Inductance**

0.0027 H (default)

Average self-inductance of the three stator windings.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator inductance fluctuation,  $L_m$  – Inductance**

-0.0005 H (default)

Fluctuation in self-inductance and mutual inductance with the rotor angle.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator mutual inductance,  $M_s$  – Inductance**

0.0011 H (default)

Average mutual inductance between the stator windings.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Field winding inductance,  $L_f$  – Inductance**

0.06 H (default)

Inductance of the field winding.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and PM or **Stator parameterization** parameter to Specify  $L_s$ ,  $L_m$ , and  $M_s$

**Mutual field armature inductance,  $L_{mf}$  – Inductance**

0.0067 H (default)

Armature-field mutual inductance.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$  and PM or **Stator parameterization** parameter to Specify  $L_s$ ,  $L_m$ , and  $M_s$

**Stator resistance per phase,  $R_s$  – Resistance**

0.7 Ohm (default)

Resistance of each of the stator windings.

**Field winding resistance,  $R_f$  — Resistance**

2.85 Ohm (default)

Resistance of the field winding.

**Zero sequence — Zero-sequence model**

Include (default) | Exclude

Zero-sequence model:

- **Include** — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Exclude** — Prioritize simulation speed for desktop simulation or real-time deployment.

**Dependencies**

If this parameter is set to:

- **Include** and **Stator parameterization** is set to **Specify  $L_d$ ,  $L_q$ , and  $L_0$**  — The **Stator zero-sequence inductance,  $L_0$**  parameter is visible.
- **Exclude** — The **Stator zero-sequence inductance,  $L_0$**  parameter is not visible.

**Rotor angle definition — Angle**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. If you select the default value, the rotor and  $a$ -phase fluxes are aligned for a zero-rotor angle. Otherwise, an  $a$ -phase current generates the maximum torque value for a zero-rotor angle.

**Mechanical****Rotor inertia — Inertia**0.01 kg\*m<sup>2</sup> (default)Inertia of the rotor attached to mechanical translational port **R**.**Rotor Damping — Damping**

0 N\*m/(rad/s) (default)

Rotary damping.

**Thermal**To enable these parameters, set **Modeling option** to Show thermal port.**Measurement temperature — Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 (default)

The fractional rate of change of permanent magnet flux density with temperature. It is used to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal mass for each stator winding – Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

## Version History

### Introduced in R2017b

### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Mbayed, R. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.
- [3] Anderson, P. M. *Contribution to the Control of the Hybrid Excitation Synchronous Machine for Embedded Applications*. Universite de Cergy Pontoise, 2012.
- [4] Luo, X. and T. A. Lipo. “A Synchronous/Permanent Magnet Hybrid AC Machine.” *IEEE Transactions of Energy Conversion*. Vol. 15, No 2 (2000), pp. 203-210.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

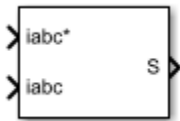
**See Also**

BLDC | Switched Reluctance Machine | Synchronous Reluctance Machine | Synchronous Machine  
Field Circuit | PMSM | Synchronous Machine Measurement

## Hysteresis Current Controller (Three-Phase)

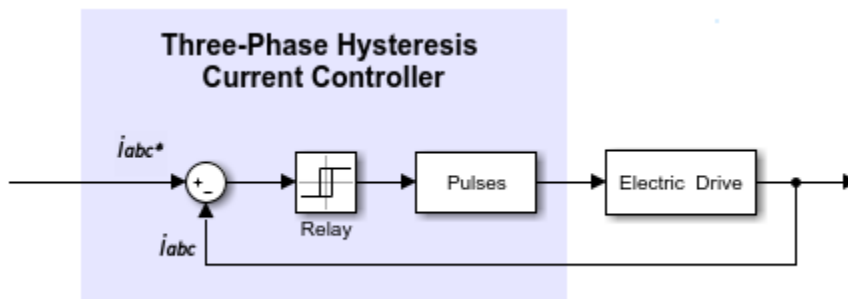
Three-phase hysteresis current control

**Library:** Simscape / Electrical / Control / General Machine Control



### Description

The Hysteresis Current Controller (Three-Phase) block implements three-phase hysteresis current control for power converters.



### Ports

#### Input

##### **$i_{abc}^*$ – Current**

vector

Three-phase reference currents.

Data Types: single | double

##### **$i_{abc}$ – Current**

vector

Measured three-phase currents.

Data Types: single | double

#### Output

##### **S – Controller output**

vector

Six-pulse vector for power converter control.

Data Types: single | double

## Parameters

### Current hysteresis band (A) — Hysteresis band

1 (default) | positive number

Hysteresis band,  $h$ , for the current controller. The switch-on point is  $h/2$  and the switch-off point is  $-h/2$ .

### Sample time (-1 for inherited) — Block sample time

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

Introduced in R2018a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

BLDC | Velocity Controller

# Ideal Semiconductor Switch

Ideal Semiconductor Switch

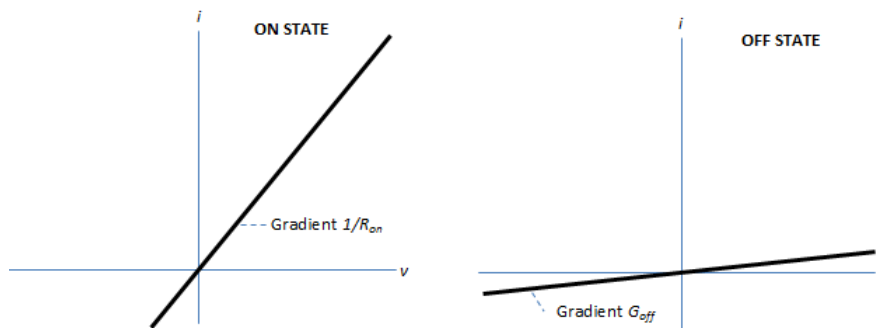
**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Ideal Semiconductor Switch block models an ideal semiconductor switching device.

The figure shows a typical i-v characteristic for an ideal semiconductor switch.



If the gate-cathode voltage exceeds the specified threshold voltage, the ideal semiconductor switch is in the on state. Otherwise the device is in the off state.

In the on state, the anode-cathode path behaves like a linear resistor with on-resistance  $R_{on}$ .

In the off state, the anode-cathode path behaves like a linear resistor with a low off-state conductance  $G_{off}$ .

Using the **Integral Diode** parameters, you can include an integral cathode-anode diode. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

The table shows you how to set the **Integral protection diode** parameter based on your goals.

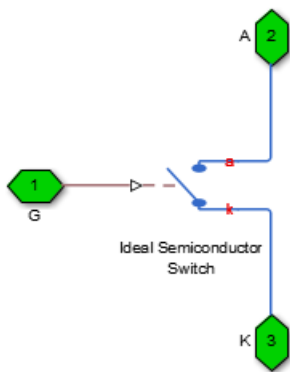
Goal	Value to Select	Block Behavior
Prioritize simulation speed.	Protection diode with no dynamics	The block includes an integral copy of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.



Goal	Value to Select	Block Behavior
Precisely specify reverse-mode charge dynamics.	Protection diode with charge dynamics	The block includes an integral copy of the dynamic model of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.

## Ports

This figure shows the block port names.



## Input

### G – Gate terminal

physical signal

Physical signal port associated with the gate terminal.

### Dependencies

To enable this port, set **Gate-control port** to PS control port.

### Conserving

### G – Gate terminal

electrical

Electrical conserving port associated with the gate terminal.

### Dependencies

To enable this port, set **Gate-control port** to Electrical control port.

### A – Anode terminal

electrical

Electrical conserving port associated with the anode terminal.

**K – Cathode terminal**

electrical

Electrical conserving port associated with the cathode terminal.

**Parameters****Main****Gate-control port – Whether to specify physical or electrical control port**

PS (default) | Electrical

Whether to specify physical or electrical control port for the switch gate.

**On-state resistance – On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Off-state conductance – Off-state conductance**

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Threshold voltage,  $V_{th}$  – Threshold voltage**

0.5 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Integral Diode****Integral protection diode – Integral protection diode**

None (default) | Protection diode with no dynamics | Protection diode with charge dynamics

Specify whether the block includes an integral protection diode. The default value is None.

If you want to include an integral protection diode, there are two options:

- Protection diode with no dynamics
- Protection diode with charge dynamics

**Forward voltage – Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Initial forward current when measuring iRM — Initial forward current when measuring iRM**

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Rate of change of current when measuring iRM — Rate of change of current when measuring iRM**

-50A/μs (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select **Specify stretch factor** or **Specify reverse recovery charge**, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics**.

#### Reverse recovery time, $t_{rr}$ – Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify reverse recovery time directly**.

#### Reverse recovery time stretch factor – Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify stretch factor**.

#### Reverse recovery charge, $Q_{rr}$ – Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify reverse recovery charge**.

For more information on these parameters, see Diode.

## Version History

Introduced in R2013b

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

Diode | GTO | Ideal Semiconductor Switch | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | N-Channel MOSFET | P-Channel MOSFET | Thyristor (Piecewise Linear)

### Topics

“Switch Between Physical Signal and Electrical Ports”

## IGBT (Ideal, Switching)

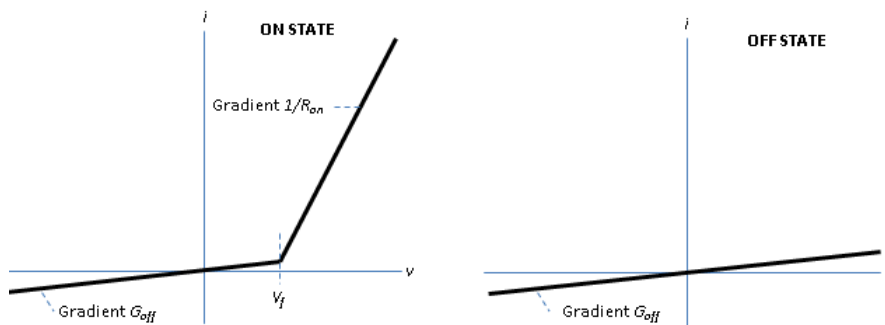
Ideal insulated-gate bipolar transistor for switching applications

**Library:** Simscape / Electrical / Semiconductors & Converters



### Description

The IGBT (Ideal, Switching) block models an ideal insulated-gate bipolar transistor (IGBT) for switching applications. The switching characteristic of an IGBT is such that if the gate-emitter voltage exceeds the specified threshold voltage,  $V_{th}$ , the IGBT is in the on state. Otherwise, the device is in the off state. This figure shows a typical i-v characteristic:



To define the I-V characteristic of the IGBT, set the **On-state behaviour and switching losses** parameter to either **Specify constant values** or **Tabulate with temperature and current**. The **Tabulate with temperature and current** option is available only if you expose the thermal port of the block.

In the on state, the collector-emitter path behaves like a linear diode with forward-voltage drop,  $V_f$ , and on-resistance,  $R_{on}$ . However, if you expose the thermal port of the block and parameterize the device using tabulated I-V data, the tabulated resistance is a function of the temperature and current.

In the off state, the collector-emitter path behaves like a linear resistor with a low off-state conductance value,  $G_{off}$ .

The defining Simscape equations for the block are:

```

if (v>Vf)&&(G>Vth)
    i == (v - Vf*(1-Ron*Goff))/Ron;
else
    i == v*Goff;
end

```

where:

- $v$  is the collector-emitter voltage.
- $V_f$  is the forward voltage.
- $G$  is the gate-emitter voltage.
- $V_{th}$  is the threshold voltage.
- $i$  is the collector-emitter current.
- $R_{on}$  is the on-state resistance.
- $G_{off}$  is the off-state conductance.

### Integral Protection Diode Option

Using the **Integral Diode** parameters, you can include an integral emitter-collector diode. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

Set the **Integral protection diode** parameter based on your goal.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	Protection diode with no dynamics	The block includes an integral copy of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.
Precisely specify reverse-mode charge dynamics.	Protection diode with charge dynamics	The block includes an integral copy of the dynamic model of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.

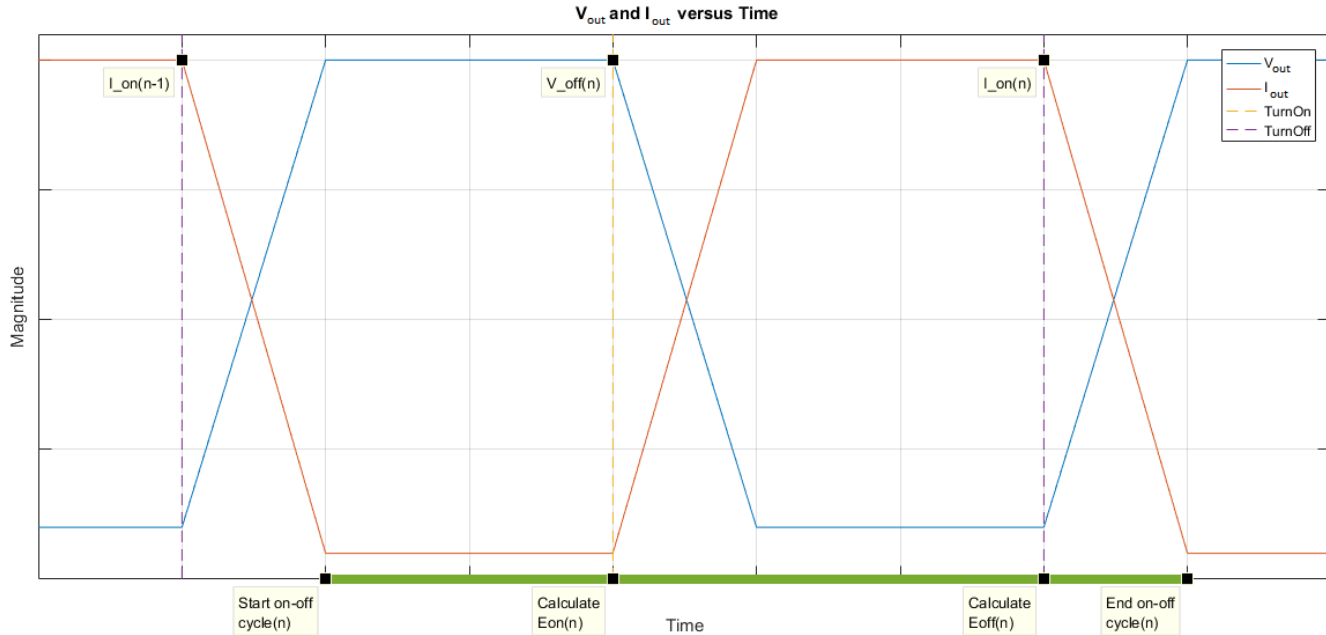
### Model Gate Port and Thermal Effects

You can choose between physical or electrical ports to control the gate terminal and expose the thermal port to model the heat that switching events and conduction losses generate. To choose the gate control port, set the **Gate control port** parameter to PS or Electrical. To expose the thermal port, set the **Modeling option** parameter to either No thermal port or Show thermal port.

For more information about using thermal ports, see “Simulating Thermal Effects in Semiconductors”.

### Thermal Losses

The figure shows an idealized representation of the output voltage,  $V_{out}$ , and the output current,  $I_{out}$ , of the semiconductor device. The interval shown includes the entire  $n^{\text{th}}$  switching cycle, during which the block turns off and then on.



Switching losses are one of the main sources of thermal loss in semiconductors. During each on-off switching transition, the IGBT parasitics store and then dissipate energy.

Switching losses depend on the off-state voltage and the on-state current. When a switching device is turned on, the power losses depend on the initial off-state voltage across the device and the final on-state current once the device is fully in its on state. Similarly, when a switching device is turned off, the power losses depend on the initial on-state current through the device and the final off-state voltage once the device is fully in its off state.

In this block, switching losses are applied by stepping up the junction temperature with a value equal to the switching loss divided by the total thermal mass at the junction. The **Switch-on loss,  $E_{on}(T_j, I_{ce})$**  and **Switch-off loss,  $E_{off}(T_j, I_{ce})$**  parameter values set the sizes of the switching losses and they are either fixed or dependent on junction temperature and drain-source current. In both cases, losses are scaled by the off-state voltage prior to the latest device turn-on event.

---

**Note** As the final current after a switching event is not known during the simulation, the block records the on-state current at the point that the device is commanded off. Similarly, the block records the off-state voltage at the point that the device is commanded on. For this reason, the simlog does not report the switching losses to the thermal network until one switching cycle later.

For all ideal switching devices, the switching losses are reported in the simlog as `lastTurnOffLoss` and `lastTurnOnLoss` and recorded as a pulse with amplitude equal to the energy loss. If you use a script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

---



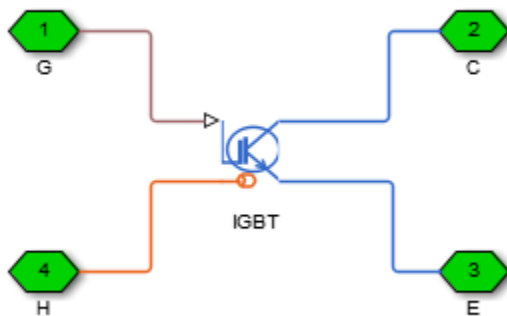
## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

The figure shows the block port names.



## Conserving

### G – Gate terminal

electrical

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port

### C – Collector terminal

electrical

Electrical conserving port associated with the collector terminal

### E – Emitter terminal

electrical

Electrical conserving port associated with the emitter terminal

### H – Thermal port

thermal

Thermal conserving port.

## Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Thermal port visibility

No thermal port (default) | Show thermal port

Whether to enable the thermal port.

### Main

This table shows how the visibility of **Main** parameters depends on how you configure the **Modeling option** and **On-state behavior and switching losses** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

### Main Parameter Dependencies

Parameters and Options		
<b>Modeling option</b>		
<i>No thermal port</i>	<i>Show thermal port</i>	
Gate-control port	Gate-control port	
Forward voltage, Vf	Threshold voltage, Vth	
On-state resistance	<b>On-state behaviour and switching losses</b>	
	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>
Off-state conductance	Forward voltage, Vf	On-state voltage, Vds(Tj,Ice)
Threshold voltage, Vth	On-state resistance	Temperature vector, Tj
	Off-state conductance	Collector-emitter current vector, Ice
		Off-state conductance

### Gate-control port — Whether to specify physical or electrical control port

PS (default) | Electrical

Whether to specify physical or electrical control port for the switch gate.

### On-state behaviour and switching losses — On-state current for switching loss data

*Specify constant values* (default) | *Tabulate with temperature and current*

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- *Specify constant values* — Use scalar values to specify the output current, switch-on loss, and switch-off loss data. This is the default parameterization method.
- *Tabulate with temperature and current* — Use vectors to specify the output current, switch-on loss, switch-off loss, and temperature data.

### Dependencies

See the Main Parameter Dependencies table.

**Forward voltage, Vf — Forward voltage**

0.8 V (default)

Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state resistance — On-state collector-emitter resistance**

0.001 Ohm (default)

Collector-emitter resistance when the device is on.

**Dependencies**

See the Main Parameter Dependencies table.

**Off-state conductance — Off-state collector-emitter conductance**

1e-5 1/Ohm (default)

Collector-emitter conductance when the device is off. The value must be less than  $1/R$ , where R is the value of **On-state resistance**.

**Dependencies**

See the Main Parameter Dependencies table.

**Threshold voltage, Vth — Threshold voltage**

6 V (default)

Gate-emitter voltage at which the device turns on.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage, Vds(Tj,Ice) — On-state voltage**

[0, .1, .6, .8, 1, 1.3, 1.6, 2, 2.4; 0, .1, .7, 1, 1.2, 1.5, 1.9, 2.4, 2.8] V (default)

Voltage drop across the device while it is in a triggered conductive state. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Temperature vector, Tj — Temperature vector**

[298.15, 398.15] K (default)

Temperature values at which the on-state voltage is specified. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Collector-emitter current vector, Ice — Collector-emitter current vector**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Collector-emitter currents for the on-state voltage is defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Switching Losses**

To enable these parameters, set **Modeling option** to Show thermal port.

**Temperature vector for switching losses, Tj — Temperature vector for switching losses**

[298.15, 398.15] K (default)

Temperature values at which the switch-on loss and switch-off loss are specified. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Collector-emitter current vector for switching losses, Ice — Collector-emitter current vector for switching losses**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Collector-emitter currents for which the switch-on loss and switch-off-loss are defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Switch-on loss — Switch-on loss**

22.86e-3 J (default)

Energy dissipated during a single switch-on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-off loss — Switch-off loss**

17.14e-3 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Off-state voltage for switching loss data — Off-state voltage for losses data**

600 V (default) | 300 V

The output voltage of the device during the off state. This is the blocking voltage at which the switch-on loss and switch-off loss data are defined.

**On-state current for switching loss data — Output current**

600 A (default)

Output currents for which the switch-on loss, switch-off loss, and on-state voltage are defined. The first element must be zero. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-on loss,  $E_{on}(T_j, I_{ce})$  — Switch-on loss**

[0, .0024, .024, .12, .2, .48, 1.04, 2.16, 3.24; 0, .003, .03, .15, .25, .6, 1.3, 2.7, 4.05] \* 1e-3 J (default)

Energy dissipated during a single switch on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Switch-off loss,  $E_{off}(T_j, I_{ce})$  — Switch-off loss**

[0, .0007, .0066, .033, .066, .17, .33, .83, 1.5; 0, .001, .01, .05, .1, .25, .5, 1.2, 2.2] \* 1e-3 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Integral Diode****Integral protection diode — Integral protection diode**

None (default) | Protection diode with no dynamics | Protection diode with charge dynamics

Block integral protection diode.

The diodes you can select are:

- None
- Protection diode with no dynamics
- Protection diode with charge dynamics

**Diode model — Diode model**

Piecewise Linear (default) | Tabulated I-V curve

Select one of these diode models:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448. This is the default method.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data plus fixed reverse bias off conductance.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Table type — Tabulated function**Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** is set to Tabulated I-V curve.

**Forward voltage — Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.
- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.
- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

#### **Forward currents, $I_f(T_j, V_f)$ – Vector of forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12]  
A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

#### **Junction temperatures, $T_j$ – Vector of junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures. This parameter must be a vector of at least two elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve.

#### **Forward voltages, $V_f$ – Vector of forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

#### **Forward voltages, $V_f(T_j, I_f)$ – Vector of forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

#### **Forward currents, $I_f$ – Vector of forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

#### **Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** 

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge



Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

#### Reverse recovery time, trr – Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, iRM** parameter divided by the value of the **Rate of change of current when measuring iRM** parameter.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery time directly.

#### Reverse recovery time stretch factor – Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify stretch factor.

#### Reverse recovery charge, Qrr – Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery charge.

**Diode turn-off voltage when measuring Erec — Diode turn-off voltage when measuring Erec**

-600 V (default)

Voltage between the diode in steady-state.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Stray inductance when measuring Erec — Stray inductance when measuring Erec**

150 nH (default)

Total unintended inductance in the measurement circuit. The block uses this value to calculate **Reverse recovery energy, Erec**.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Reverse recovery energy, Erec — Reverse recovery energy**

0.03 J (default)

Total switching losses due to the diode reverse recovery.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Thermal Port**

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Version History

**Introduced in R2013b****Update on switching losses and thermal modelling options**

*Behavior changed in R2020b*

From R2020b forward, the IGBT (Ideal, Switching) block has improved losses and thermal modelling options.

As a result from these changes:

- Electrical and thermal on-state losses are now always identical. The **Thermal loss dependent on** parameter and its options, **Voltage and current** and **Voltage, current, and temperature**, have been renamed to **On-state behavior and switching losses**, **Specify constant values**, and **Tabulate with temperature and current**:
  - If you selected **Voltage and current** for **Thermal loss dependent on**, then the electrical on-state losses are unchanged and their values are determined using the on-state resistance. However, the thermal on-state losses are now also determined by the on-state resistance. Prior to R2020b, the thermal on-state losses were defined by the product of the **On-state voltage** and **Output current, Iout** parameters.
  - If you selected **Voltage, current, and temperature** for **Thermal loss dependent on**, then the thermal on-state losses are unchanged and the **On-state voltage, Vds(Tj,Ice)** parameter sets their values. However, the electrical on-state losses are now equal to the thermal on-state losses. Prior to R2020b, the electrical on-state losses were defined by the value of the on-state resistance.
- The **On-state voltage** parameter is no longer used.

### Removal of energy dissipation time constant parameter

*Behavior changed in R2021a*

From R2021a forward, the **Energy dissipation time constant** parameter of the IGBT (Ideal, Switching) block is no longer used. A step in junction temperature now reflects the switching losses. If your model contains a thermal mass directly connected to this block thermal port, remove it and model the thermal mass inside the component itself.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

Diode | GTO | Ideal Semiconductor Switch | MOSFET (Ideal, Switching) | Thyristor (Piecewise Linear)

### Topics

“Simulating Thermal Effects in Semiconductors”

“Switch Between Physical Signal and Electrical Ports”

# Incandescent Lamp

Incandescent lamp with temperature dependent resistance

**Library:** Simscape / Electrical / Passive



## Description

The Incandescent Lamp block models an incandescent lamp, the key characteristic of which is that the resistance increases as the filament warms up.

Under the simplifying assumption that the rate of heat loss from the filament is proportional to temperature difference to ambient, the temperature of the filament is governed by

$$kt_c \frac{dT}{dt} = i^2 R - kT$$

and the filament resistance is governed by the following equation

$$R = R_0(1 + \alpha T)$$

where:

- $R_0$  is the initial resistance at turn-on (when filament is at ambient temperature).
- $T$  is the filament temperature relative to ambient temperature.
- $\alpha$  is the resistance temperature coefficient.
- $t_c$  is the thermal time constant.
- $k$  is the heat transfer coefficient.
- $R$  is the filament resistance.
- $i$  is the filament current.

There are two parameterization options:

- If you select **Specify resistance values directly**, the block uses values that you provide for filament resistance when on and at turn-on to determine the value for the heat transfer coefficient.
- If you select **Specify currents**, the block uses values that you provide for filament current when on and at turn-on to determine the value for the heat transfer coefficient.

Optionally you can specify a simulation time at which the lamp fails by providing a finite value for the **Simulation time for open circuit fault event** parameter on the **Faults** tab. When in the open-circuit state, the lamp resistance is set to be the value of the **Open-circuit resistance** parameter.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the lamp positive voltage.

#### - – Negative terminal

electrical

Electrical conserving port associated with the lamp negative voltage.

## Parameters

### Resistance

#### Parameterization – Block parameterization

Specify resistance values directly (default) | Specify currents

Select one of the following methods for block parameterization:

- Specify resistance values directly — Provide the values for filament resistance at turn-on and when on in steady state. The block determines the value for the heat transfer coefficient based on these values. This is the default option.
- Specify currents — Provide the values for filament current at turn-on and when on in steady state. The block determines the value for the heat transfer coefficient based on these values.

#### Initial resistance at turn-on – Initial resistance at turn-on

0.15 Ohm (default)

The resistance seen by the external circuit when the lamp is initially turned on.

#### Dependencies

This parameter is visible only when you select Specify resistance values directly for the **Parameterization** parameter.

#### Steady-state resistance when on – Steady-state resistance when on

1 Ohm (default)

The resistance seen by the external circuit when the lamp is on and in steady state. This resistance should be greater than the **Initial resistance at turn-on**.

#### Dependencies

This parameter is visible only when you select Specify resistance values directly for the **Parameterization** parameter.

#### Inrush current at turn-on – Inrush current at turn-on

70 A (default)

The current through the lamp when it is initially turned on.

**Dependencies**

This parameter is visible only when you select **Specify currents** for the **Parameterization** parameter.

**Steady-state current when on — Steady-state current when on**

10 A (default)

The current through the lamp when it is on and in steady state. This current should be less than the **Inrush current at turn-on**.

**Dependencies**

This parameter is visible only when you select **Specify currents** for the **Parameterization** parameter.

**Rated voltage — Lamp rated voltage**

12 V (default)

The rated voltage for the lamp, and the voltage value for which the resistance or current values are provided in the on and turn-on states.

**Resistance temperature coefficient — Resistance temperature coefficient**

0.004 1/K (default)

The fractional increase in resistance per unit increase in temperature.

**Dynamics****Thermal time constant — Thermal time constant**

25 ms (default)

The first-order thermal time constant for filament temperature when the lamp is turned on or off.

**Initial lamp state — Initial lamp state**

Off (default) | On

Select between On and Off.

**Faults****Enable open circuit fault — Open circuit fault option**

Off (default) | On

Whether to enable open circuit fault.

**Simulation time for open circuit fault event — Simulation time for open circuit fault event**

inf s (default)

For simulation times greater than this parameter value the filament resistance becomes equal to the **Open-circuit resistance**. Specifying a finite value for this parameter lets you simulate the fault dynamics when the bulb burns out.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Open-circuit resistance — Open-circuit resistance**

1e6 Ohm (default)

The value of the filament resistance used when the lamp goes open-circuit.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Version History**

Introduced in R2012b

**Extended Capabilities****C/C++ Code Generation**

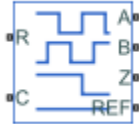
Generate C and C++ code using Simulink® Coder™.

**See Also**

# Incremental Shaft Encoder

Behavioral model that converts angular position to electrical pulses

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Incremental Shaft Encoder block represents a device that converts information about the angular position of a shaft into electrical pulses. The block produces  $N$  pulses on ports **A** and **B** per shaft revolution, where  $N$  is the value you specify for the **Pulses per revolution** parameter. Pulses **A** and **B** are 90 degrees out of phase. If the shaft rotates in a positive direction, then **A** leads **B**. The block produces a single index pulse on port **Z** once per revolution. The **Z**-pulse positive transition always coincides with an **A**- pulse positive transition, and **Z**- pulse length is equal to the length for the **A** and **B** pulses. The voltages at ports **A**, **B**, and **Z** are defined relative to the **Ref** reference port voltage.

Use this block if you need to model the shaft encoder signals, either to support development of a decoding algorithm or to include the quantization effects. Otherwise, use the Ideal Rotational Motion Sensor block from the Simscape Foundation library.

## Assumptions and Limitations

- The Incremental Shaft Encoder block is not linearizable. For control design studies that require model linearization, use the Ideal Rotational Motion Sensor block from the Simscape Foundation library.

## Ports

### Conserving

#### **R — Rotational velocity**

mechanical rotational

Mechanical rotational conserving port associated with the sensor positive probe.

#### **C — Rotational velocity**

mechanical rotational

Mechanical rotational conserving port associated with the sensor negative (reference) probe.

#### **A — Voltage**

electrical

Encoded electrical output.

#### **B — Voltage**

electrical



Encoded electrical output.

**Z – Index or synchronization**

electrical

Index, or synchronization, electrical output.

**Ref – Voltage**

electrical

Floating zero-volt reference.

**Parameters****Pulses per revolution – Pulse count**

2 (default)

Number of pulses produced on each of the *A* and *B* phases per revolution of the shaft.

**Output voltage amplitude – Voltage**

5 V (default)

Amplitude of the shaft encoder output voltage when the output is high.

**Index pulse offset relative to shaft initial angle – Position**

0 deg (default)

Offset of the index pulse *Z* relative to the angle of the shaft at the start of the simulation. This parameter lets you set the initial location of the index pulse.

**Version History**

Introduced in R2017b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

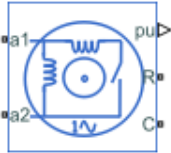
**See Also****Simscape Blocks**

Ideal Rotational Motion Sensor

# Induction Machine (Single-Phase)

Single-phase induction machine with SI or pu fundamental parameterization

**Library:** Simscape / Electrical / Electromechanical / Asynchronous



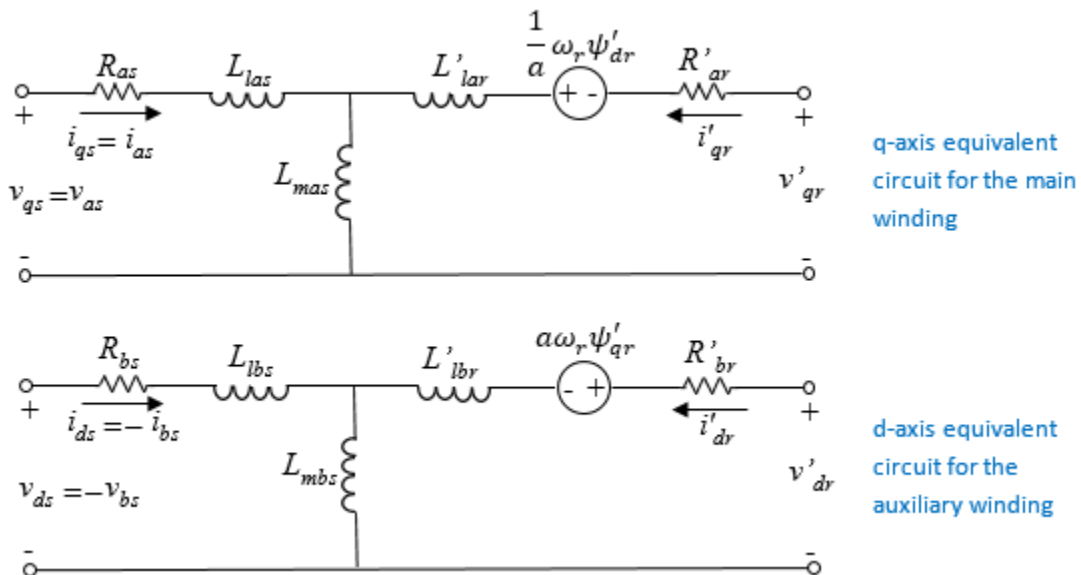
## Description

The Induction Machine (Single-Phase) block represents a single-phase induction machine with a squirrel cage rotor with fundamental parameters expressed in per-unit or in the International System of Units (SI).

This block models four types of single-phase induction machines:

- Split-phase
- Capacitor-start
- Capacitor-start-capacitor-run
- Main and auxiliary windings

The single phase induction machine consists of a squirrel cage rotor and two stator windings, the main and the auxiliary windings. The auxiliary winding is typically only active during startup. However, to improve performance, the auxiliary winding can be active during running for low-power applications. The figure shows the equivalent  $d$ - and  $q$ -axis circuits for the main and auxiliary windings.



The table defines the variables.

Variable	Definition
$v_{qs}v_{ds}$	Stator voltages in the $d$ - $q$ representation
$i_{qs}i_{ds}$	Stator currents in the $d$ - $q$ representation
$\Psi_{qr}\Psi_{dr}$	Rotor fluxes in the $d$ - $q$ representation
$\omega$	Rotor electrical speed
$a$	Auxiliary/main windings turn ratio
$R_{as}$	Main winding stator resistance
$L_{as}$	Main winding stator leakage inductance
$R'_{ar}$	Main winding rotor resistance
$L'_{lar}$	Main winding rotor leakage inductance
$L_{mas}$	Main winding mutual inductance
$R_{bs}$	Auxiliary winding stator resistance
$R_{lbs}$	Auxiliary winding stator leakage inductance
$R'_{br}$	Auxiliary winding rotor resistance
$L'_{lbr}$	Auxiliary winding rotor leakage inductance
$L_{mbs}$	Auxiliary winding mutual inductance

### Equations

The SI model converts the SI values that you enter in the dialog box to per-unit values for simulation. For information on the relationship between SI and per-unit machine parameters, see “Per-Unit Conversion for Machine Parameters”. For information on per-unit parameterization, see “Per-Unit System of Units”.

These transformations reduce the rotor resistance, rotor leakage inductance, and stator mutual inductance to a single set of values by defining the equivalent circuits with respect to the main winding:

$$L_{ms} = L_{mas} = \frac{1}{a^2}L_{mbs}$$

$$R'_{ar} = \frac{1}{a^2}R'_{br}$$

$$L'_{lar} = \frac{1}{a^2}L'_{lbr}$$

The voltage equations for the stator and rotor are:

$$v_{qs} = R_{as}i_{qs} + \frac{d\psi_{qs}}{dt}$$

$$v_{ds} = R_{bs}i_{ds} + \frac{d\psi_{ds}}{dt}$$

$$0 = R'_{ar}i'_{qr} + \frac{d\psi'_{qr}}{dt} - \frac{1}{a}\omega_r\psi'_{dr}$$

$$0 = a^2 R'_{ar} i'_{dr} + \frac{d\psi'_{dr}}{dt} + a\omega_r \psi'_{qr}$$

$$\psi_{qs} = (L_{las} + L_{ms}) i_{qs} + L_{ms} i'_{qr}$$

$$\psi_{ds} = (L_{lbs} + a^2 L_{ms}) i_{ds} + a^2 L_{ms} i'_{dr}$$

$$\psi'_{qr} = (L'_{iar} + L_{ms}) i'_{qr} + L_{ms} i_{qs}$$

$$\psi'_{dr} = a^2 (L'_{iar} + L_{ms}) i'_{dr} + a^2 L_{ms} i_{ds}$$

The expression for the electromagnetic torque,  $T$ , is obtained by applying the principle of virtual displacement [1].

$$T = p \left( a \psi'_{qr} i'_{dr} - \frac{1}{a} \psi'_{dr} i_{qr} \right)$$

The mechanical equation is

$$J \frac{d\omega_m}{dt} = T - T_L - B_m \omega_m,$$

where  $\omega_m$  is the rotor angular velocity.

In split-phase machines, the auxiliary winding is displaced at 90 electrical degrees from the main winding and operates only until the speed reaches the disconnection speed, which is typically 70 to 80 percent of rated speed. In this configuration, the auxiliary winding has high resistance and small reactance compared to the main winding. The resulting phase difference makes the machine behave like a two-phase machine.

The capacitor-start machine is a type of split-phase machine that uses a capacitor in series with the auxiliary winding to start the machine. In this configuration, auxiliary and main windings have the same number of turns. The value of the capacitor ensures that the current in the auxiliary coil leads the current in the main winding by approximately 80 electrical degrees.

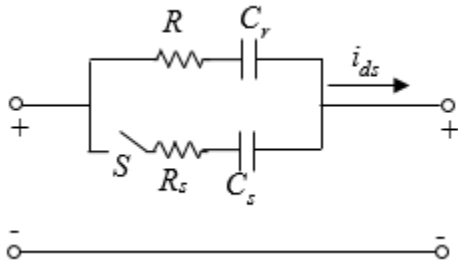
The capacitor-start voltage is:

$$v_c = R_s i_{ds} + \int \frac{1}{C_s} \omega_r i_{ds}.$$

When a capacitor is connected in series with the auxiliary winding, the voltage equation for the  $d$ -axis is

$$v_{ds} = R_{bs} i_{ds} + \frac{d\psi_{ds}}{dt} - R_s i_{ds} - \int \frac{1}{C_s} \omega_r i_{ds}.$$

The extension is obtained immediately for capacitor-start-capacitor-run machines. In this configuration, two capacitors are connected in parallel.



The  $d$ -axis voltage after disconnecting the capacitor-start is

$$v_{ds} = R_{bs}i_{ds} + \frac{d\psi_{ds}}{dt} - R_r i_{ds} - \int \frac{1}{C_r} \omega_r i_{ds}.$$

### Display Option

To display the machine per-unit base values in the MATLAB Command Window, right-click the block and, from the **Electrical** menu, select **Display Base Values**.

### Model Thermal Effects

You can expose thermal ports to simulate the effects of generated heat and motor temperature. To expose the thermal ports and the **Thermal** parameters, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Output

**pu** — Machine per-unit measurements

physical signal | vector

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- pu\_torque
- pu\_velocity
- pu\_vds
- pu\_vqs
- pu\_v0s = 0
- pu\_ids
- pu\_iqs
- pu\_i0s = 0

### Conserving

#### **a1 — Main winding positive terminal**

electrical | scalar

Electrical conserving port associated with the main winding positive terminal.

#### **a2 — Main winding negative terminal**

electrical | scalar

Electrical conserving port associated with the main winding negative terminal.

#### **b1 — Auxiliary winding positive terminal**

electrical | scalar

Electrical conserving port associated with the auxiliary winding positive terminal.

#### **Dependencies**

This port is visible only if you set the **Type** to Main and auxiliary windings.

#### **b2 — Auxiliary winding negative terminal**

electrical | scalar

Electrical conserving port associated with the auxiliary winding negative terminal.

#### **Dependencies**

This port is visible only if you set the **Type** to Main and auxiliary windings.

#### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

#### **HA — Main winding thermal port**

thermal

Thermal conserving port associated with stator main winding.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – Auxiliary winding thermal port**

thermal

Thermal conserving port associated with stator auxiliary winding.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR – Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters**

**Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

**Main**

**Type – Induction machine type**

Split-phase (default) | Capacitor-start | Capacitor-start-capacitor-run | Main and auxiliary windings

Type of induction machine.

**Rated apparent power – Rated apparent power**

180 V\*A (default) | positive scalar

Rated apparent power of the induction machine.

**Rated voltage – Rated voltage**

220 V (default) | positive scalar

RMS voltage.

**Rated electrical frequency – Rated electrical frequency**

60 Hz (default) | positive scalar

Nominal electrical frequency corresponding to the rated apparent power.

**Number of pole pairs – Number of pole pairs**

2 (default) | positive integer

Number of machine pole pairs. The value is used as a back-electromotive force constant.

**Parameterization unit — SI or per-unit parameterization option**

SI (default) | Per unit

Unit system for block parameterization.

**Dependencies**

Selecting:

- SI exposes SI **Electrical** parameters.
- Per unit exposes per-unit **Electrical** parameters.

**Electrical**

For the **Parameterization unit** parameter in the **Main** settings, select SI to expose the SI **Electrical** parameters, or Per unit to expose the per-unit **Electrical** parameters.

**Main winding stator resistance, Ras — Main winding stator resistance**

1.8180 Ohm (default) | positive scalar

SI stator resistance for the main winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Main winding stator leakage inductance, Llas — Main winding stator leakage inductance**

0.0067 H (default) | positive scalar

SI leakage inductance for the main winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Main winding rotor resistance, Rar' — Main winding rotor resistance**

3.7080 Ohm (default) | positive scalar

SI rotor resistance for the main winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Main winding rotor leakage inductance, Llar' — Main winding rotor leakage inductance**

0.0050 H (default) | positive scalar

SI rotor leakage inductance for the main winding.



**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Main winding mutual inductance, Lms — Main winding mutual inductance**

0.1595 H (default) | positive scalar

SI mutual inductance for the main winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Auxiliary winding stator resistance, Rbs — Auxiliary winding stator resistance**

6.4260 Ohm (default) | positive scalar

SI stator resistance for the auxiliary winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Auxiliary winding stator leakage inductance, Llbs — Auxiliary winding stator leakage inductance**

0.0077 H (default) | positive scalar

SI stator leakage inductance for the auxiliary winding.

**Dependencies**

Selecting SI for the **Parameterization unit** parameter in the **Main** settings exposes SI **Electrical** parameters.

**Auxiliary/main windings turn ratio — Turn ratio**

1.1 (default) | positive scalar

Winding ratio between the main and auxiliary winding.

**Main winding stator resistance, Ras (pu) — Main winding stator resistance**

0.0135 (default) | positive scalar

Per-unit stator resistance for the main winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Main winding stator leakage inductance, Llas (pu) — Main winding stator leakage inductance**

0.0188 (default) | positive scalar

Per-unit leakage inductance for the main winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Main winding rotor resistance,  $R_{ar}'$  (pu) — Main winding rotor resistance**

0.0276 (default) | positive scalar

Per-unit rotor resistance for the main winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Main winding rotor leakage inductance,  $L_{lar}'$  (pu) — Main winding rotor leakage inductance**

0.0140 (default) | positive scalar

Per-unit rotor leakage inductance for the main winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Main winding mutual inductance,  $L_{ms}$  (pu) — Main winding mutual inductance**

0.4472 (default) | positive scalar

Per-unit mutual inductance for the main winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Auxiliary winding stator resistance,  $R_{bs}$  (pu) — Auxiliary winding stator resistance**

0.0478 (default) | positive scalar

Per-unit stator resistance for the auxiliary winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Auxiliary winding stator leakage inductance,  $L_{lbs}$  (pu) — Auxiliary winding stator leakage inductance**

0.0216 (default) | positive scalar

Per-unit stator leakage inductance for the auxiliary winding.

**Dependencies**

Selecting Per unit for the **Parameterization unit** parameter in the **Main** settings exposes per-unit **Electrical** parameters.

**Auxiliary/main windings turn ratio – Turn ratio**

1.1 (default) | positive scalar

Winding ratio between the main and auxiliary winding.

**Capacitor-start resistance,  $R_s$  – Capacitor-start resistance**

3 Ohm (default) | positive scalar

SI capacitor-start resistance.

**Dependencies**To enable this parameter, set **Type** to either Capacitor-start-capacitor-run or Capacitor-start and **Parameterization unit** to SI.**Capacitor-start capacitance,  $C_s$  – Capacitor-start capacitance**

0.00018 F (default) | positive scalar

SI capacitor-start capacitance.

**Dependencies**To enable this parameter, set **Type** to either Capacitor-start-capacitor-run or Capacitor-start and **Parameterization unit** to SI.**Capacitor-run resistance,  $R_r$  – Capacitor-run resistance**

9 Ohm (default) | positive scalar

SI capacitor-run resistance.

**Dependencies**To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to SI.**Capacitor-run capacitance,  $C_r$  – Capacitor-run capacitance**

0.000015 F (default) | positive scalar

SI capacitor-run capacitance.

**Dependencies**To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to SI.**Capacitor-start resistance,  $R_s$  (pu) – Capacitor-start resistance**

0.0223 (default) | positive scalar

Per-unit capacitor-start resistance.

**Dependencies**To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to Per unit.**Capacitor-start capacitance,  $C_s$  (pu) – Capacitor-start capacitance**

9.1232 (default) | positive scalar

Per-unit capacitor-start capacitance.

**Dependencies**

To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to Per unit.

**Capacitor-run resistance,  $R_r$  (pu) — Capacitor-run resistance**

0.0669 (default) | positive scalar

Per-unit capacitor-run resistance.

**Dependencies**

To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to Per unit.

**Capacitor-run resistance,  $C_r$  (pu) — Capacitor-run capacitance**

0.000015 (default) | positive scalar

Per-unit capacitor-run capacitance.

**Dependencies**

To enable this parameter, set **Type** to Capacitor-start-capacitor-run and **Parameterization unit** to Per unit.

**Disconnection speed (% of rated speed) — Disconnection speed**

80 (default) | scalar in range [0,100]

Speed when capacitor is disconnected, expressed as a percentage of rated speed. The value must be between 0 and 100, inclusive.

**Dependencies**

To enable this parameter, set **Type** to either Split-phase, Capacitor-start, or Capacitor-start-capacitor-run.

**Mechanical****Rotor inertia — Rotor inertia**

0.01 kg\*m<sup>2</sup> (default) | positive scalar

Inertia of the rotor.

**Rotor damping — Rotor damping**

0 N\*m/(rad/s) (default) | scalar

Damping of the rotor.

**Thermal**

These parameters appear only for blocks with exposed thermal ports.

**Measurement temperature — Rated temperature**

298.15 K (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Cage resistance temperature coefficient – Rotor temperature coefficient**

4.4e-3 (default) | 1/K | scalar

Coefficient relating cage resistance to temperature for the rotor. The default value, 4.4e-3 1/K, is for aluminium.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

**Introduced in R2018b****Induction machine type parameter***Behavior changed in R2021b*

From R2021b forward, to model the different types of induction machine, set the **Type** parameter to either Split-phase, Capacitor-start, Capacitor-start-capacitor-run, or Main and auxiliary windings.

As a result of these changes, inside a model saved in an earlier release, review the **Type** parameter of this block.

## References

[1] Krause, P. C. "Simulation of Unsymmetrical 2-Phase Induction Machines." *IEEE Transactions on Power Apparatus and Systems*. Vol 84, Number 11, 1965, pp. 1025-1037.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Field-Oriented Control (Single-Phase)

# Induction Machine Current Controller

Discrete-time induction machine current PI controller

**Library:** Simscape / Electrical / Control / Induction Machine Control



## Description

The Induction Machine Current Controller implements discrete-time proportional-integral (PI) based induction machine current control in the rotor  $d$ - $q$  reference frame. You typically use the Induction Machine Current Controller in a series of blocks that make up a control structure. For example, to convert the  $dq0$  reference frame output voltage to voltage in an  $abc$  reference frame, connect the Induction Machine Current Controller to an Inverse Clarke Transform in the control structure.

## Equations

The block uses the backward Euler discretization method.

Two PI current controllers that are implemented in the rotor reference frame produce the reference voltage vector:

$$v_d^{ref} = \left( K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) (i_d^{ref} - i_d) + v_{d\_FF},$$

and

$$v_q^{ref} = \left( K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) (i_q^{ref} - i_q) + v_{q\_FF},$$

where

- $v_d^{ref}$ , and  $v_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference voltages, respectively.
- $i_d^{ref}$ , and  $i_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference currents, respectively.
- $i_d$  and  $i_q$  are the  $d$ -axis and  $q$ -axis currents, respectively.
- $K_{p\_id}$ , and  $K_{p\_iq}$  are the proportional gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $K_{i\_id}$  and  $K_{i\_iq}$  are the integral gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $v_{d\_FF}$ , and  $v_{q\_FF}$  are the feedforward voltages for the  $d$ -axis and  $q$ -axis, respectively. The feedforward voltages are obtained from the machine mathematical equations and provided as inputs.
- $T_s$ , is the sample time of the discrete controller.

### Voltage Saturation

Saturation is imposed when the stator voltage vector exceeds the voltage phase limit  $V_{ph\_max}$ :

$$\sqrt{v_d^2 + v_q^2} \leq V_{ph\_max},$$

where  $v_d$ , and  $v_q$  are the  $d$ -axis and  $q$ -axis voltages, respectively.

In the case of axis prioritization, the voltages  $v_1$  and  $v_2$  are introduced, where:

- For  $d$ -axis prioritization —  $v_1 = v_d$  and  $v_2 = v_q$ .
- For  $q$ -axis prioritization —  $v_1 = v_q$  and  $v_2 = v_d$ .

The constrained (saturated) voltages  $v_1^{sat}$  and  $v_2^{sat}$  are obtained as:

$$v_1^{sat} = \min(\max(v_1^{unsat}, -V_{ph\_max}), V_{ph\_max})$$

and

$$v_2^{sat} = \min(\max(v_2^{unsat}, -V_{2\_max}), V_{2\_max}),$$

where:

- $v_1^{unsat}$  and  $v_2^{unsat}$  are the unconstrained (unsaturated) voltages.
- $v_{2\_max}$  is the maximum value of  $v_2$  that does not exceed the voltage phase limit. The equation that define  $v_{2\_max}$  is  $v_{2\_max} = \sqrt{(V_{ph\_max})^2 - (v_1^{sat})^2}$ .

In the case of  $d$ - $q$  equivalence, the direct and quadrature axes have the same priority, and the constrained voltages are:

$$v_d^{sat} = \min(\max(v_d^{unsat}, -V_{d\_max}), V_{d\_max})$$

and

$$v_q^{sat} = \min(\max(v_q^{unsat}, -V_{q\_max}), V_{q\_max}),$$

where:

$$V_{d\_max} = \frac{V_{ph\_max}|v_d^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}|v_q^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

### Integral Anti-Windup

An anti-windup mechanism is employed to avoid the saturation of the integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}(v_d^{sat} - v_d^{unsat})$$

and

$$K_{i\_iq} + K_{aw\_iq}(v_q^{sat} - v_q^{unsat}),$$

where  $K_{aw\_id}$ ,  $K_{aw\_iq}$ , and  $K_{aw\_if}$  are the anti-windup gains for the  $d$ -axis,  $q$ -axis, and field controllers, respectively.

### Assumptions and Limitations

- The plant model for the direct and quadrature axes can be approximated with a first-order system.

## Ports

### Input

#### **idqRef** — Reference currents

vector

Desired  $d$ - and  $q$ -axis currents for control of the induction machine, in A.

Data Types: `single` | `double`

#### **idq** — Measured currents

vector

Actual  $d$ - and  $q$ -axis currents of the controlled induction machine, in A.

Data Types: `single` | `double`

#### **vdqFF** — Feedforward voltages

vector

Feedforward pre-control voltages, in V.

Data Types: `single` | `double`

#### **VphMax** — Maximum phase voltage

scalar

Maximum allowable voltage in each phase, in V.

Data Types: `single` | `double`

#### **Reset** — External reset

scalar

External reset signal (rising edge) for integrators.

Data Types: `Boolean`

### Output

#### **vdqRef** — Reference voltages

vector



Desired  $d$ - and  $q$ -axis voltages for control of the induction machine, in V.

Data Types: `single` | `double`

## Parameters

### Control Parameters

#### **D-axis current proportional gain — $d$ -axis proportional gain**

1 (default)

Proportional gain for direct-axis current control.

#### **D-axis current integral gain — $d$ -axis integral gain**

100 (default)

Integral gain for direct-axis current control.

#### **D-axis current anti-windup gain — $d$ -axis anti-windup gain**

1 (default)

Anti-windup gain for direct-axis current control.

#### **Q-axis current proportional gain — $q$ -axis proportional gain**

1 (default)

Proportional gain for quadrature-axis current control.

#### **Q-axis current integral gain — $q$ -axis integral gain**

100 (default)

Integral gain for quadrature-axis current control.

#### **Q-axis current anti-windup gain — $q$ -axis anti-windup gain**

1 (default)

Anti-windup gain for quadrature-axis current control.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

#### **Axis prioritization — Axis prioritization for voltage limiter**

Q-axis (default) | D-axis | D-Q equivalence

Prioritize or maintain the ratio between the  $d$ - and  $q$ -axes when the block limits voltage.

#### **Enable pre-control voltage — Pre-control voltage**

on (default) | off

Enable or disable pre-control voltage.

## **Version History**

**Introduced in R2017b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

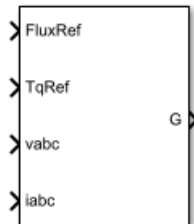
### **Blocks**

Induction Machine Direct Torque Control | Induction Machine Field-Oriented Control | Induction Machine Flux Observer | Induction Machine Scalar Control | Induction Machine Current Controller

# Induction Machine Direct Torque Control

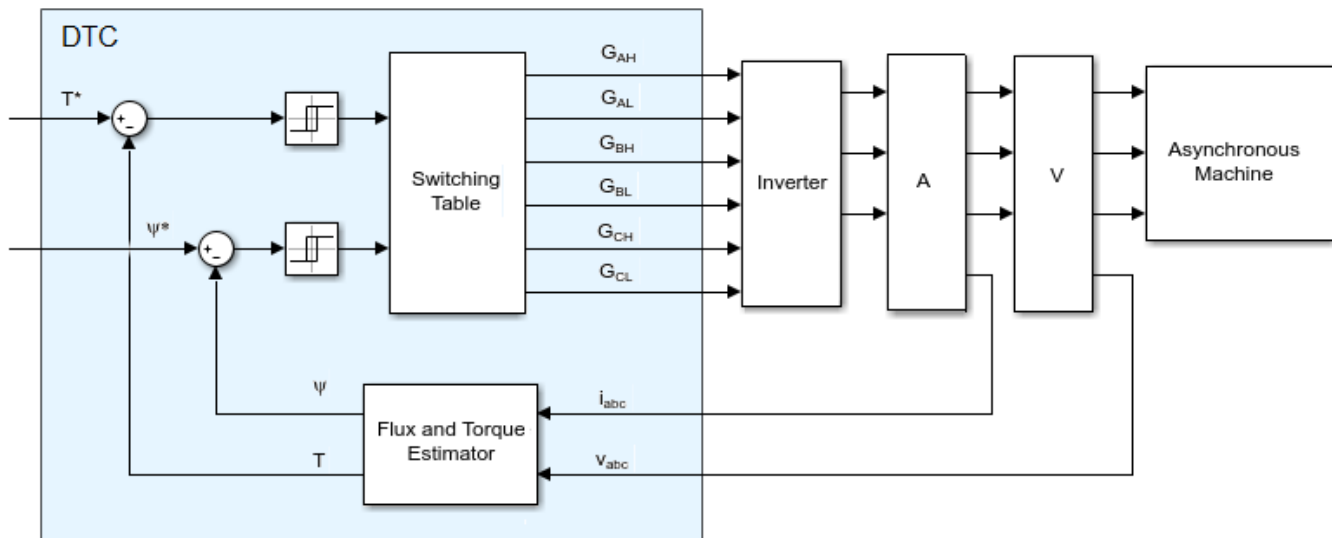
Induction machine DTC

**Library:** Simscape / Electrical / Control / Induction Machine Control



## Description

The Induction Machine Direct Torque Control block implements an induction machine direct torque control (DTC) structure. The figure shows the equivalent circuit for the block.



## Equations

To estimate the torque and flux, the Induction Machine Direct Torque Control block discretizes the machine voltage equations in the stationary  $\alpha\beta$  reference frame using the backward Euler method. The discrete-time equations for stator fluxes in the  $\alpha\beta$  frame are:

$$\psi_{\alpha} = (v_{\alpha} - i_{\alpha}R_s) \frac{T_s z}{z-1}$$

and

$$\psi_{\beta} = (v_{\beta} - i_{\beta}R_s) \frac{T_s z}{z-1}$$

where:

- $v_\alpha$  is  $\alpha$ -axis voltage.
- $i_\alpha$  is  $\alpha$ -axis current.
- $R_s$  is the stator resistance.
- $\Psi_\alpha$  is the  $\alpha$ -axis stator flux.
- $v_\beta$  is  $\beta$ -axis voltage.
- $i_\beta$  is  $\beta$ -axis current.
- $\Psi_\beta$  is the  $\beta$ -axis stator flux.

The block calculates the torque and flux as:

$$T = \frac{3p}{2}(\psi_\alpha i_\beta - \psi_\beta i_\alpha)$$

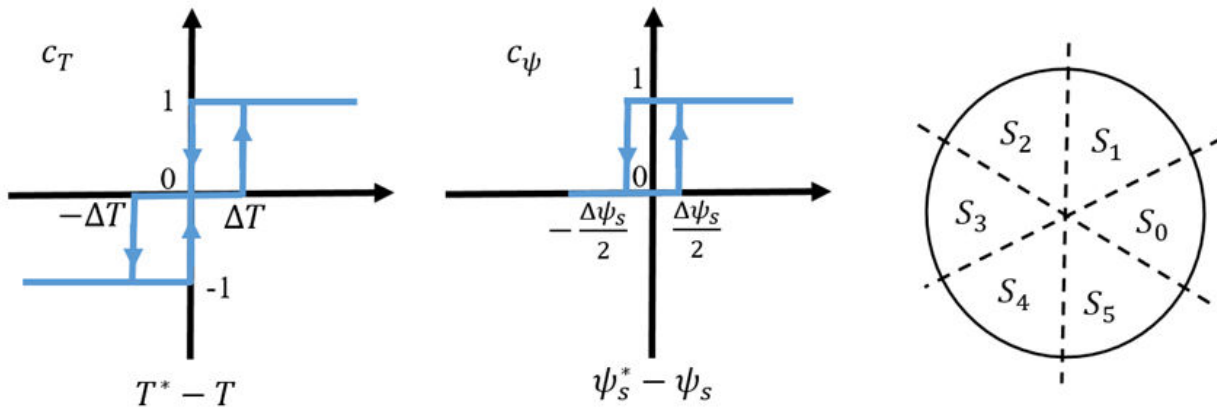
and

$$\psi_s = \sqrt{\psi_\alpha^2 + \psi_\beta^2}$$

where:

- $p$  is the number of pole pairs.
- $\Psi_s$  is the stator flux.

To detect flux and torque estimation errors, the block uses hysteresis comparators. The figure shows hysteresis comparators and the associated switching sectors.



The table shows the optimum switching for an inverter high-side system.

$c_\psi, c_T S(\theta)$		$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$c_\psi = 1$	$c_T = 1$	1, 1, 0	0, 1, 0	0, 1, 1	0, 0, 1	1, 0, 1	1, 0, 0
	$c_T = 0$	1, 1, 1	0, 0, 0	1, 1, 1	0, 0, 0	1, 1, 1	0, 0, 0
	$c_T = -1$	1, 0, 1	1, 0, 0	1, 1, 0	0, 1, 0	0, 1, 1	0, 0, 1
$c_\psi = 0$	$c_T = 1$	0, 1, 0	0, 1, 1	0, 0, 1	1, 0, 1	1, 0, 0	1, 1, 0
	$c_T = 0$	0, 0, 0	1, 1, 1	0, 0, 0	1, 1, 1	0, 0, 0	1, 1, 1

$c_\psi, c_T S(\theta)$	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$c_T = -1$	0, 0, 1	1, 0, 1	1, 0, 0	1, 1, 0	0, 1, 0	0, 1, 1

### Assumptions and Limitations

- The power inverter dead times are not considered. For hardware implementation, add the dead time externally.

## Ports

### Input

#### FluxRef – Flux

scalar

Reference stator flux.

Data Types: single | double

#### TqRef – Torque

scalar

Reference torque.

Data Types: single | double

#### vabc – Voltage

vector

Stator phase voltages.

Data Types: single | double

#### iabc – Current

vector

Stator phase currents.

Data Types: single | double

### Output

#### G – Gate pulses

vector | 0 or 1

Inverter gate pulses. The block does not consider any dead time.

Data Types: single | double

## Parameters

#### Stator resistance (0hm) – Resistance

0.25 (default) | positive scalar

Resistance of the machine stator.

**Number of pole pairs — Pole number**

1 (default) | positive integer

Number of machine pole pairs.

**Flux hysteresis bandwidth (Wb) — Flux**

0.02 (default) | positive scalar

Total bandwidth distributed symmetrically around the flux set point.

**Torque hysteresis bandwidth (N\*m) — Torque**

10 (default) | positive scalar

Total bandwidth distributed symmetrically around the set point.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

**Introduced in R2017b**

## References

- [1] Takahashi, I., and T. Noguchi. "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor." *IEEE Transactions on Industry Applications*. Vol. IA-22, Number 5, 1986, pp. 820 - 827.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

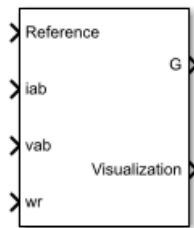
**Blocks**

Induction Machine Current Controller | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Flux Observer | Induction Machine Scalar Control

# Induction Machine Direct Torque Control (Single-Phase)

Single-phase induction machine direct torque control

**Library:** Simscape / Electrical / Control / Induction Machine Control

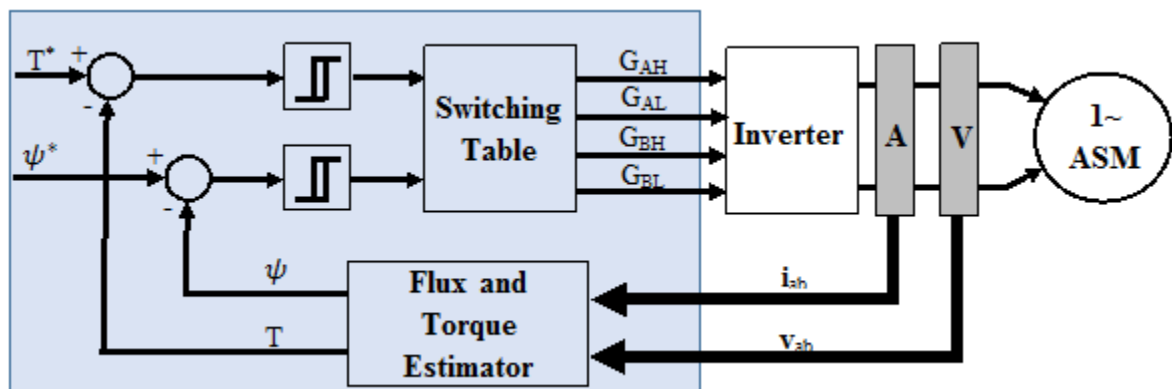


## Description

The Induction Machine Direct Torque Control (Single-Phase) block implements a single-phase induction machine direct torque control structure.

## Equations

This diagram shows the direct torque control architecture for single-phase machines



The torque and flux estimation is based on machine voltage equations. The discrete-time voltage equations using the backward Euler discretization method, are:

$$\psi_a = (v_a - i_a R_{as}) \frac{T_s z}{z - 1}$$

$$\psi_b = (v_b - i_b R_{bs}) \frac{T_s z}{z - 1}$$

where:

- $R_{as}$  and  $R_{bs}$  are the main winding resistance and the auxiliary winding resistance, respectively.

- $i_a$  and  $i_b$  are the main winding current and the auxiliary winding current, respectively.
- $v_a$  and  $v_b$  are the main and auxiliary winding voltage, respectively.
- $\psi_a$  and  $\psi_b$  are the main and auxiliary winding flux, respectively.

The torque and flux are obtained from:

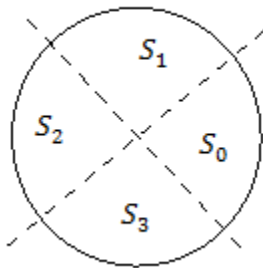
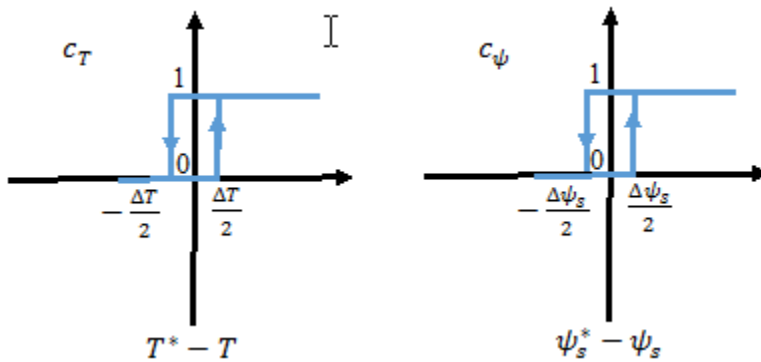
$$T = -p(a\psi_a i_b - \frac{1}{a}\psi_b i_a)$$

$$\psi_s = \sqrt{\psi_a^2 + \psi_b^2}$$

where:

- $p$  is the number of pole pairs.
- $a$  is the auxiliary to main windings turn ratio.

Employing simple hysteresis comparators detects the status of flux and torque errors. The following figures illustrate the hysteresis comparators and the switching sectors.



The table shows the optimum switching table (inverter high side).

$c_\psi, c_T, S(\theta)$		$S_0$	$S_1$	$S_2$	$S_3$
$c_\psi = 1$	$c_T = 1$	1, 0	1, 1	0, 1	0, 0
	$c_T = 0$	1, 1	0, 1	0, 0	1, 0
$c_\psi = 0$	$c_T = 1$	0, 0	1, 0	1, 1	0, 1



$c_\psi, c_T, S(\theta)$	$S_0$	$S_1$	$S_2$	$S_3$
$c_T = 0$	0, 1	0, 0	1, 0	1, 1

The torque reference can be provided as an input, or, in the case of speed control, be generated internally using a PI speed controller.

The flux reference is generated internally using:

$$\psi_s^* = \frac{2\pi f_n \psi_n}{p \min\left(\omega_r, \frac{2\pi f_n}{p}\right)}$$

where,

- $\omega_r$  is the rotor angular mechanical speed in rad/s.
- $f_n$  is the rated frequency.
- $\psi_n$  is the rated flux.

### Limitations

- The power inverter dead-times are not considered in this block. For hardware implementation, add the dead-time externally.

## Ports

### Input

#### Reference – Torque or angular velocity reference

scalar

Specify the angular velocity reference in rad/s or the torque reference in Nm.

Data Types: single | double

#### iab – Stator currents

scalar

Stator currents for main and auxiliary windings.

Data Types: single | double

#### vab – Stator voltages

scalar

Stator voltages for main and auxiliary windings.

Data Types: single | double

#### wr – Rotor angular speed

scalar

Measure rotor angular speed in rad/s.

Data Types: single | double

## Output

### **G — Inverter gate pulses**

0 | 1

Inverter gate pulses, specified as a Boolean. The dead-time is not considered.

Data Types: single | double

### **Visualization — Internal signals for visualization**

Reference | Rotor velocity | TqRef | FluxRef | Torque | Flux

Bus containing internal signals for visualization. The signals are:

- Reference torque or speed
- Rotor velocity, in rad/s
- Torque Reference, in Nm
- Flux Reference, in Wb
- Torque, in Nm
- Flux, in Wb

Data Types: single | double | bus

## Parameters

### General

#### **Control mode — Specify control mode**

Speed control (default) | Torque control

Specify the control mode. The reference input is taken as angular speed, in rad/s, for Speed control and torque, in Nm, for Torque control.

#### **Rated electrical frequency (Hz) — Specify rated electrical frequency**

60 (default) | positive scalar

Rated electrical frequency, in Hz.

#### **Number of pole pairs — Specify pole pairs**

2 (default) | positive scalar

Number of pole pairs.

#### **Main winding stator resistance, Ras (Ohm) — Specify main winding stator resistance**

1.8180 (default) | positive scalar

Resistance of the main winding stator, in Ohms.

#### **Auxiliary winding stator resistance, Rbs (Ohm) — Specify auxiliary winding stator resistance**

6.4260 (default) | positive scalar

Resistance of the auxiliary winding stator, in Ohms.

**Auxiliary/main windings turn ratio – Specify the auxiliary-to-main windings turn ratio**

1.1 (default) | positive scalar

Auxiliary-to-main windings turns ratio.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**Outer Loop**

This table shows how the visibility of some parameters in the **Outer Loop** tab depends on the option that you choose for the **Control mode** parameter in the **General** tab.

Control mode	Available Parameters
Speed control	Speed controller proportional gain
	Speed controller integral gain
	Speed controller integral anti-windup gain
	Maximum torque (Nm)
	Minimum torque (Nm)
	Rated flux (Wb)
Torque control	Rated flux (Wb)

**Speed controller proportional gain – Specify the speed controller proportional gain**

1 (default) | positive scalar

Proportional gain for the PI speed controller.

**Speed controller integral gain – Specify the speed controller integral gain**

10 (default) | positive scalar

Integral gain for the PI speed controller.

**Speed controller integral anti-windup gain – Specify the speed controller integral anti-windup gain**

1000 (default) | positive scalar

Integral anti-windup gain.

**Maximum torque (Nm) – Specify the maximum torque**

5 (default) | positive scalar

Maximum torque used in speed controller saturation, in Nm.

**Minimum torque (Nm) — Specify the minimum torque**

-5 (default) | scalar

Minimum torque used in speed controller saturation, in Nm.

**Rated flux (Wb) — Specify the rated flux**

0.5 (default) | positive scalar

Rated flux, in Wb. The rated flux is used to compute the flux reference.

**Inner Loop****Flux hysteresis band (Wb) — Specify the flux hysteresis band**

0.02 (default) | positive scalar

Flux hysteresis band for the controller, in Wb.

**Torque hysteresis band (Nm) — Specify the torque hysteresis band**

0.1 (default) | positive scalar

Torque hysteresis band for the controller, in Nm.

## Version History

Introduced in R2018b

## References

- [1] Takahashi, I., and T. Noguchi. "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor." *IEEE Transactions on Industry Applications*. Vol. IA-22, Number 5, 1986, pp. 820 - 827.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Flux Observer | Induction Machine Scalar Control

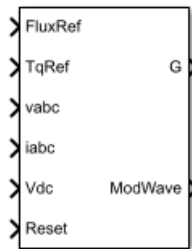
**Topics**

"Single-Phase Asynchronous Machine Direct Torque Control"

# Induction Machine Direct Torque Control with Space Vector Modulator

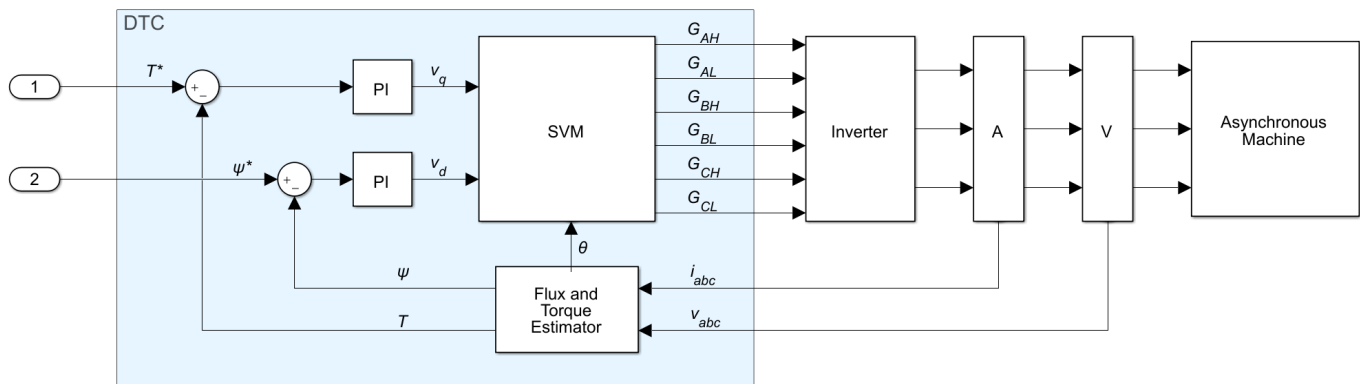
Induction machine DTC structure with SVM

**Library:** Simscape / Electrical / Control / Induction Machine Control



## Description

The Induction Machine Direct Torque Control with Space Vector Modulator implements an induction machine direct torque control structure (DTC) with space vector modulator (SVM). Use this block to generate the gate pulses for an inverter controlling an induction machine. This diagram shows the architecture of the block.



In the diagram:

- You provide the reference torque,  $T^*$ , and flux,  $\psi^*$ .
- The Flux and Torque Estimator estimates the actual torque,  $T$ , and flux,  $\psi$  from the measured phase currents,  $i_{abc}$ , and voltages,  $v_{abc}$ .
- Two PI controllers determine the reference  $d$  and  $q$  voltages,  $v_d$  and  $v_q$ , from the flux and torque errors, respectively.
- The SVM generates the gates pulses,  $G_{ij}$ , required to control an inverter driving the induction machine. Subscript  $i$  corresponds to the phase ( $a$ ,  $b$ , or  $c$ ). Subscript  $j$  corresponds to the high,  $H$ , or low,  $L$ , signal.

### Flux and Torque Estimator

To estimate the torque and flux, the block discretizes the machine voltage equations in the stationary  $\alpha\beta$  reference frame using the backward Euler method. The discrete-time equations for stator fluxes in the  $\alpha\beta$  frame are:

$$\psi_\alpha = (v_\alpha - i_\alpha R_s) \frac{T_s z}{z-1}$$

$$\psi_\beta = (v_\beta - i_\beta R_s) \frac{T_s z}{z-1}$$

Where:

- $v_\alpha$  and  $v_\beta$  are the  $\alpha$ - and  $\beta$ -axis voltages, respectively.
- $i_\alpha$  and  $i_\beta$  are the  $\alpha$ - and  $\beta$ -axis currents, respectively.
- $\Psi_\alpha$  and  $\Psi_\beta$  are the  $\alpha$ - and  $\beta$ -axis stator fluxes, respectively.
- $R_s$  is the stator resistance.

The block calculates the torque and total stator flux as:

$$T = \frac{3p}{2} (\psi_\alpha i_\beta - \psi_\beta i_\alpha)$$

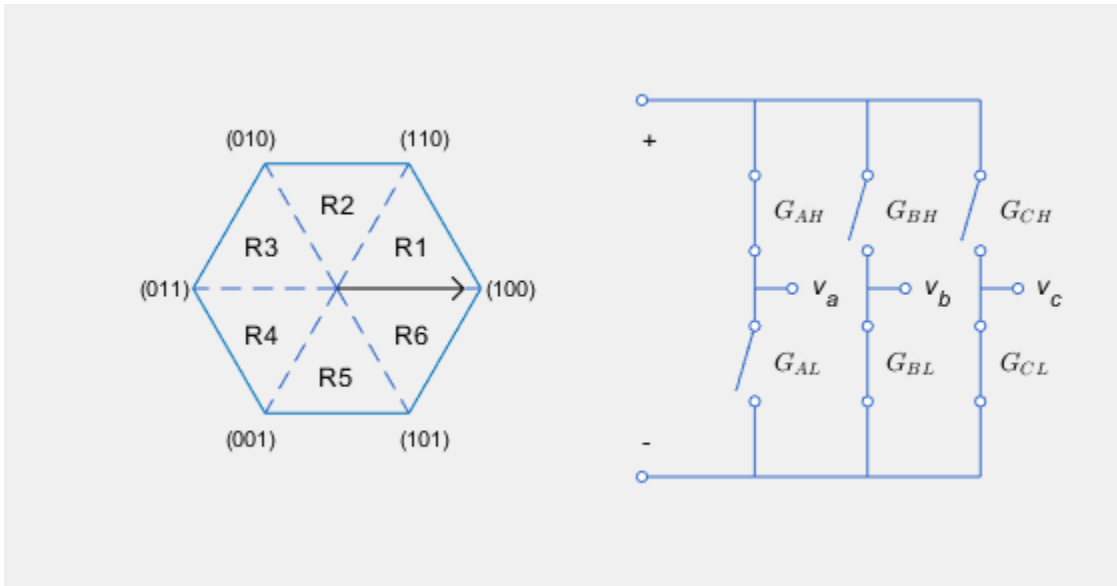
$$\psi_s = \sqrt{\psi_\alpha^2 + \psi_\beta^2}$$

Where:

- $p$  is the number of pole pairs.
- $\Psi_s$  is the total stator flux.

### Space Vector Modulator

The SVM converts the desired voltages into gate pulses, which you use to control an inverter. This figure shows possible switching states of a three-phase inverter.



The hexagon represents the space vector diagram. Each of the six vertices represents a possible switching state ( $G_{AH}, G_{BH}, G_{CH}$ ) of the three-phase inverter. Each low gate takes the opposite state as its corresponding high gate. The inverter diagram illustrates the current state.

The rotating vector in the space vector diagram corresponds to the complex reference voltage vector, which rotates at the desired electrical frequency of the machine. In reality, the switching frequency is much faster than this electrical frequency. As a result, the inverter switches continually between the two states enclosing its current region  $R_i$ , and the zero state corresponding to  $(0,0,0)$ , to generate the desired voltages.

To learn about the implementation of this method, see the PWM Generator (Three-phase, Two-level) block.

## Ports

### Input

#### **FluxRef — Flux**

scalar

Reference stator flux.

Data Types: `single` | `double`

#### **TqRef — Torque**

scalar

Reference torque.

Data Types: `single` | `double`

#### **vabc — Voltage**

vector

Stator phase voltages.

Data Types: `single` | `double`

#### **iabc — Current**

vector

Stator phase currents.

Data Types: `single` | `double`

#### **Vdc — DC-link voltage signal**

scalar

DC-link voltage for the converter.

Data Types: `single` | `double`

#### **Reset — Controller reset**

scalar

Reset for the PI controller integrators.

Data Types: single | double

## Output

### **G – Gate pulses**

vector | 0 or 1

Inverter gate pulses. The block does not consider any dead time.

Data Types: single | double

### **ModWave – Modulation wave**

vector

Modulation wave you deploy to the hardware if you are generating code for a platform with PWM-capable hardware. Otherwise, this data is only for your reference.

## Parameters

### General

#### **Stator resistance (Ohm) – Resistance**

0.25 (default) | positive scalar

Resistance of the machine stator.

#### **Number of pole pairs – Pole number**

1 (default) | positive integer

Number of machine pole pairs.

#### **Inverter dc-link voltage threshold (V) – Voltage**

300 (default) | positive scalar

Voltage threshold to activate the power inverter.

#### **Fundamental sample time (s) – SVM sample time**

5e-6 (default) | positive scalar less than the control sample time

Sample time for the space vector modulator. Fundamental sample time must be less than the control sample time.

#### **Control sample time (s) – PI sample time**

5e-5 (default) | positive scalar greater than the fundamental sample time

Sample time for PI controllers. Control sample time must be greater than the fundamental sample time.

#### **Switching frequency (Hz) – Switching rate**

1000 (default) | positive scalar

Specify the rate at which you want the switches in the power converter to switch.



## Control Parameters

### Flux controller proportional gain — Flux PI proportional gain

150 (default) | positive scalar

Proportional gain for the flux controller.

### Flux controller integral gain — Flux PI integral gain

3000 (default) | positive scalar

Integral gain for the flux controller.

### Flux controller anti-windup gain — Flux PI anti-windup gain

1 (default) | positive scalar

Anti-windup gain for the flux controller.

### Torque controller proportional gain — Torque PI proportional gain

1 (default) | positive scalar

Proportional gain for the torque controller.

### Torque controller integral gain — Torque PI integral gain

50 (default) | positive scalar

Integral gain for the torque controller.

### Torque controller anti-windup gain — Torque PI anti-windup gain

1 (default) | positive scalar

Anti-windup gain for the torque controller.

### Axis prioritization — Axis prioritization for voltage limiter

Q-axis (default) | D-axis | D-Q equivalence

Prioritize or maintain ratio between  $d$ - and  $q$ -axes when the block limits voltage.

## Version History

Introduced in R2018a

## References

- [1] Buja, G. S., and M. P. Kazmierkowski. "Direct Torque Control of PWM Inverter-Fed AC Motors—A Survey." *IEEE Transactions on Industrial Electronics* 51, no. 4, (2004): 744 - 757.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

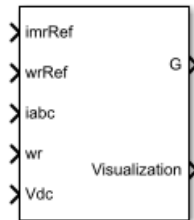
### **Blocks**

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Field-Oriented Control | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Induction Flux Observer | Induction Machine Scalar Control

# Induction Machine Field-Oriented Control

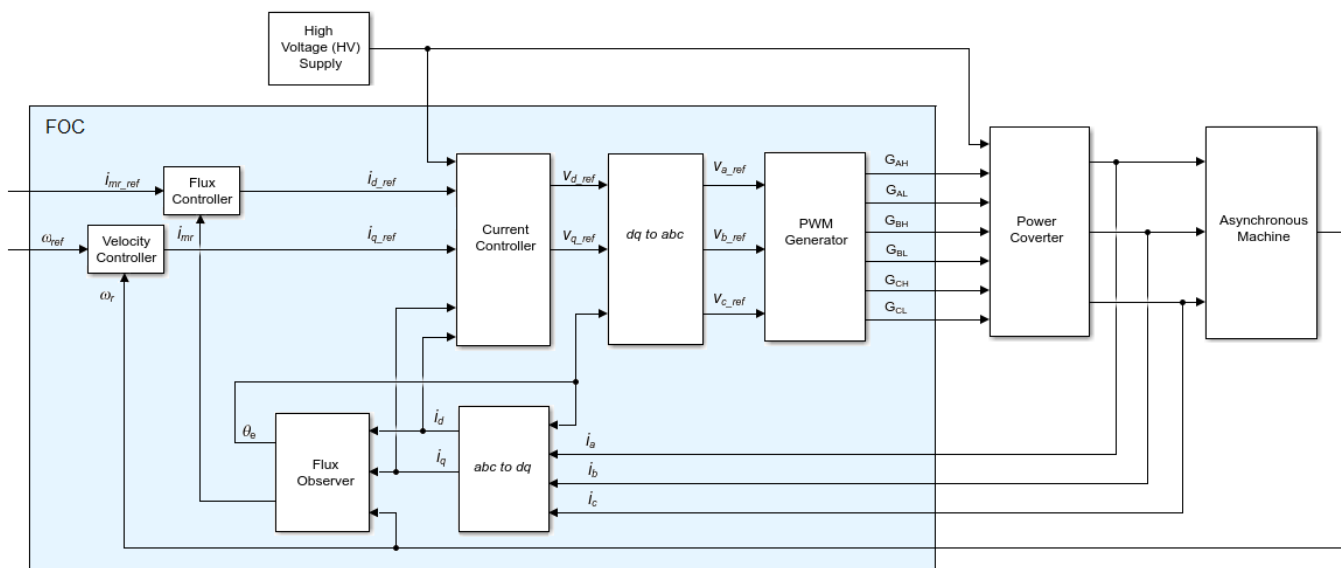
Per-unit discrete-time induction machine FOC

**Library:** Simscape / Electrical / Control / Induction Machine Control



## Description

The Induction Machine Field-Oriented Controller block implements an induction machine field-oriented control (FOC) structure using the per-unit system. To decouple the torque and flux, FOC uses the rotor  $d$ - $q$  reference frame. The figure shows the control structure.



In the diagram:

- $\omega_r$  is the measured angular velocity.
- $\omega_{ref}$  is the reference angular velocity.
- $i_d$  and  $i_q$  are the  $d$ - and  $q$ -axis stator currents.
- $i_a$ ,  $i_b$ , and  $i_c$  are the  $a$ -,  $b$ - and  $c$ -phase stator winding currents.
- $i_{mr\_ref}$  is the reference magnetizing current.
- $i_{mr}$  is the magnetizing current.
- $v_d$  and  $v_q$  are the  $d$ - and  $q$ -axis stator voltages.

- $v_a$ ,  $v_b$ , and  $v_c$  are the  $a$ -,  $b$ - and  $c$ -phase stator winding voltages.
- $\theta_e$  is the rotor electrical angle.
- $G_{AH}$ ,  $G_{AL}$ ,  $G_{BH}$ ,  $G_{BL}$ ,  $G_{CH}$ , and  $G_{CL}$  are the  $a$ -,  $b$ - and  $c$ -phase high (H) and low(L) gate pulses.

### Assumptions and Limitations

- The machine parameters are known.
- The implementation uses the per-unit system.
- The control structure implementation uses a single sample rate.

## Ports

### Input

#### **imRef — Current**

scalar

Magnetizing reference current in the per-unit system.

Data Types: `single` | `double`

#### **wrRef — Velocity**

scalar

Rotor reference velocity in per-unit system.

Data Types: `single` | `double`

#### **iabc — Current**

vector

Measured phase currents in the per-unit system.

Data Types: `single` | `double`

#### **wr — Velocity**

scalar

Measured angular velocity in per-unit system.

Data Types: `single` | `double`

#### **Vdc — Voltage**

scalar

Measured dc-link voltage, in V.

Data Types: `single` | `double`

### Output

#### **G — Gate pulses**

vector | 0 or 1

Inverter gate pulses. The block does not consider any dead time.

Data Types: single | double

### **Visualization — Visualization signals**

vector

Bus containing signals for visualization.

Data Types: single | double | bus

## **Parameters**

### **General**

#### **Rated voltage, rms line-to-line (V) — Voltage**

550 (default)

Nominal voltage.

#### **Rated electrical frequency (Hz) — Frequency**

60 (default)

Nominal electrical frequency.

#### **Rotor resistance, referred to the stator side (pu) — Resistance**

0.01 (default)

Rotor, stator-side resistance in the per-unit system.

#### **Rotor leakage inductance, referred to the stator side (pu) — Inductance**

0.06 (default)

Rotor stator-side leakage inductance, in the pu-unit system.

#### **Magnetizing inductance (pu) — Inductance**

2.7 (default)

Magnetizing inductance in the per-unit system.

#### **Time constant for dq currents filters (s) — Time constant**

1e-4 (default)

Time constant for filtering the  $d$  and  $q$  currents.

#### **Inverter dc-link voltage threshold (V) — Voltage**

500 (default)

Voltage threshold to activate the power inverter.

#### **Fundamental sample time (s) — Time**

5e-6 (default) | positive scalar less than the control sample time

Fundamental sample time must be less than the control sample time.

#### **Control sample time (s) — Time**

5e-5 (default) | positive scalar greater than the fundamental sample time

Control sample time must be greater than the fundamental sample time.

### **Outer Loop**

#### **Magnetizing current controller proportional gain – Gain**

10 (default)

Proportional gain for the magnetizing current controller.

#### **Magnetizing current controller integral gain – Gain**

1000 (default)

Integral gain for the magnetizing current controller.

#### **Magnetizing current controller integral anti-windup gain – Gain**

1000 (default)

Integral anti-windup gain for the magnetizing current controller.

#### **Speed controller proportional gain – Gain**

10 (default)

Proportional gain for the speed controller.

#### **Speed controller integral gain – Gain**

1000 (default)

Integral gain for the speed controller.

#### **Speed controller integral anti-windup gain – Gain**

1000 (default)

Integral anti-windup gain for the speed controller.

#### **Maximum d-axis current [pu] – Current**

2 (default)

Maximum current for the  $d$ -axis.

#### **Maximum q-axis current [pu] – Current**

2 (default)

Maximum current for the  $q$ -axis.

### **Inner Loop**

#### **Phase-a axis alignment – $dq0$ reference frame alignment**

Q-axis (default) | D-axis

Align the  $a$ -phase vector of the  $abc$  reference frame to the  $d$ - or  $q$ -axis of the rotating reference frame.

#### **D-axis current proportional gain – D-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for direct-axis current control.

**D-axis current integral gain — D-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for direct-axis current control.

**D-axis current anti-windup gain — D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for direct-axis current control.

**Q-axis current proportional gain — Q-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for quadrature-axis current control.

**Q-axis current integral gain — Q-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for quadrature-axis current control.

**Q-axis current anti-windup gain — Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for quadrature-axis current control.

**Axis prioritization — Axis prioritization for voltage limiter**

Q-axis (default) | D-axis | D-Q equivalence

Prioritize or maintain ratio between  $d$ - and  $q$ -axes when block limits voltage.

**PWM****PWM method — Pulse width modulation method**

SVM: space vector modulation (default) | SPWM: sinusoidal PWM

Specify the waveform technique.

**Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

**Switching frequency (Hz) — Switching rate**

1000 (default) | positive integer

Specify the rate at which you want the switches in the power converter to switch.

**Version History**

Introduced in R2017b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

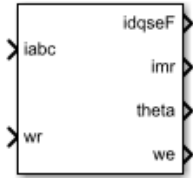
Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Flux Observer | Induction Machine Scalar Control



# Induction Machine Flux Observer

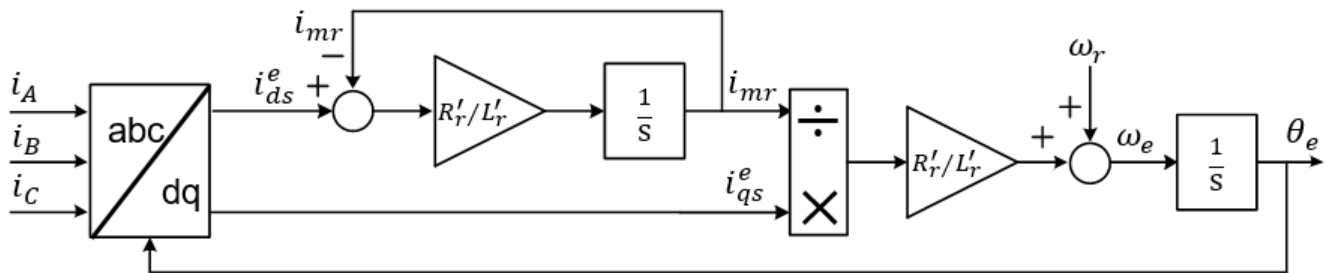
Induction machine flux observer for field-oriented control

**Library:** Simscape / Electrical / Control / Observers



## Description

The Induction Machine Flux Observer block obtains the synchronous speed,  $\omega_e$ , and electrical angle,  $\theta_e$ , that are required for performing rotor field-oriented control (FOC). The figure shows the equivalent circuit for the observer.



## Equations

To determine the synchronous speed and electrical angle, the Induction Machine Flux Observer block uses these relationships:

$$\lambda_{dr}^e = L_r' i_{dr}^e + L_M i_{ds}^e \triangleq L_M i_{mr}^e,$$

$$0 = R_r' i_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e + \frac{d\lambda_{dr}^e}{dt},$$

and

$$i_{qr}^e = -\frac{L_M}{L_r'} i_{qs}^e,$$

in these combined forms:

$$i_{dr}^e = \frac{L_M}{L_r'} (i_{mr} - i_{ds}^e)$$

$$\frac{di_{mr}}{dt} = \frac{R_r'}{L_r'} (i_{ds}^e - i_{mr})$$

and

$$\omega_e = \omega_r + \frac{R_r i_{qs}^e}{L_r i_{mr}}$$

where:

- $\lambda_{dr}^e$  is the  $d$ -axis rotor flux.
- $i_{mr}$  is the magnetizing current.
- $i_{ds}^e$  and  $i_{qs}^e$  are the  $d$ -axis and  $q$ -axis stator currents.
- $i_{dr}^e$  and  $i_{qr}^e$  are the  $d$ -axis and  $q$ -axis rotor currents.
- $\omega_e$  is the synchronous speed.
- $\omega_r$  is the mechanical rotational speed.
- $R_r$  is the rotor resistance, referred to the stator side.
- $L_r$  is the rotor leakage inductance, referred to the stator side.
- $L_M$  the magnetizing inductance.

## Ports

### Input

#### **iabc** — Current

vector

Measured stator currents in the per-unit system.

Data Types: single | double

#### **wr** — Speed

scalar

Measured rotational speed.

Data Types: single | double

### Output

#### **idqseF** — Current

vector

Filtered  $d$ -axis and  $q$ -axis stator currents in the synchronous reference frame.

Data Types: single | double

#### **imr** — Current

scalar

Magnetizing rotor current.

Data Types: single | double

**theta — Electrical angle**

scalar

Rotor electrical angle.

Data Types: single | double

**we — Synchronous speed**

scalar

Rotor synchronous speed.

Data Types: single | double

**Parameters****Rated electrical frequency (Hz) — Frequency**

60 (default) | positive

Machine rated electrical frequency.

**Rotor resistance, referred to the stator side (pu) — Resistance**

0.01 (default) | positive

Rotor resistance, referred to the stator side, in the per-unit system.

**Rotor leakage inductance, referred to the stator side (pu) — Inductance**

0.06 (default) | positive

Rotor leakage inductance, referred to the stator side, in the per-unit system.

**Magnetizing inductance (pu) — Inductance**

2.7 (default) | positive

Magnetizing inductance in the per-unit system.

**Time constant for dq currents filters (s) — Time constant**

1e-4 (default) | 0 or positive

Time constant used to low-pass filter the  $d$ - $q$  currents.**Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**Version History****Introduced in R2017b**

## References

[1] Vas, P. *Electrical Machines and Drives: A Space-vector Theory Approach*. New York: Oxford University Press, 1992.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Scalar Control

# Induction Machine Measurement

Per-unit measurement from induction machine

**Library:** Simscape / Electrical / Electromechanical / Asynchronous



## Description

The Induction Machine Measurement block outputs a per-unit measurement associated with a connected Induction Machine Squirrel Cage or Induction Machine Wound Rotor block. The input of the Induction Machine Measurement block connects to the pu output port of the induction machine block.

You set the **Output** parameter to a per-unit measurement associated with the induction machine. Based on the value you select, the Induction Machine Measurement block:

- Directly outputs the value of an element in the input signal vector
- Calculates the per-unit measurement by using values of elements in the input signal vector in mathematical expressions

The Induction Machine Measurement block outputs a per-unit measurement from the induction machine according to the output value expressions in the table. For example, when you set **Output** to *Stator d-axis voltage*, the block directly outputs the value of the *pu\_vds* element in the input signal vector. However, when you set **Output** to *Slip*, the block calculates the slip value by subtracting the value of the *pu\_velocity* element from 1.

Output Parameter Setting	Output Value Expression
Electrical torque	pu_torque
Rotor velocity	pu_velocity
Stator d-axis voltage	pu_vds
Stator q-axis voltage	pu_vqs
Stator zero-sequence voltage	pu_v0s
Stator d-axis current	pu_ids
Stator q-axis current	pu_iqs
Stator zero-sequence current	pu_i0s
Slip	1-pu_velocity
Apparent power	$\sqrt{pu_{Pt}^2 + pu_{Qt}^2}$
Real power	$pu_{Pt} = (pu_{vds} * pu_{ids}) + (pu_{vqs} * pu_{iqs}) + 2(pu_{v0s} * pu_{i0s})$
Reactive power	$pu_{Qt} = (pu_{vqs} * pu_{ids}) - (pu_{vds} * pu_{iqs})$
Terminal voltage	$\sqrt{pu_{vds}^2 + pu_{vqs}^2}$

Output Parameter Setting	Output Value Expression
Terminal current	$\sqrt{pu\_ids^2 + pu\_iqs^2}$
Power factor angle (rad)	<code>power_factor_angle = atan2(pu_Qt, pu_Pt)</code>
Power factor	<code>cos(power_factor_angle)</code>

## Ports

### Input

#### pu — Input port

physical

Physical signal vector associated with per-unit measurements from a connected induction machine. The vector elements are:

- *pu\_torque*
- *pu\_velocity*
- *pu\_vds*
- *pu\_vqs*
- *pu\_v0s*
- *pu\_ids*
- *pu\_iqs*
- *pu\_i0s*

### Output

#### o — Per-unit measurement output

physical

Physical signal associated with the per-unit measurement output.

## Parameters

#### Output — Per-unit measurement

Electrical torque (default) | Rotor velocity | Stator d-axis voltage | Stator q-axis voltage | Stator zero-sequence voltage | Stator d-axis current | Stator q-axis current | Stator zero-sequence current | Slip | Apparent power | Real power | Reactive power | Terminal voltage | Power factor angle | Power factor angle

Per-unit measurement from induction machine. Options are:

- Electrical torque
- Rotor velocity
- Stator d-axis voltage
- Stator q-axis voltage
- Stator zero-sequence voltage
- Stator d-axis current

- Stator q-axis current
- Stator zero-sequence current
- Slip
- Apparent power
- Real power
- Reactive power
- Terminal voltage
- Power factor angle
- Power factor angle

## Version History

Introduced in R2013b

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

#### Simscape Blocks

Induction Machine (Single-Phase) | Induction Machine Squirrel Cage | Induction Machine Wound Rotor

#### Simscape Blocks

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Flux Observer | Induction Machine Scalar Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Field-Oriented Control (Single-Phase)

# Induction Machine Scalar Control

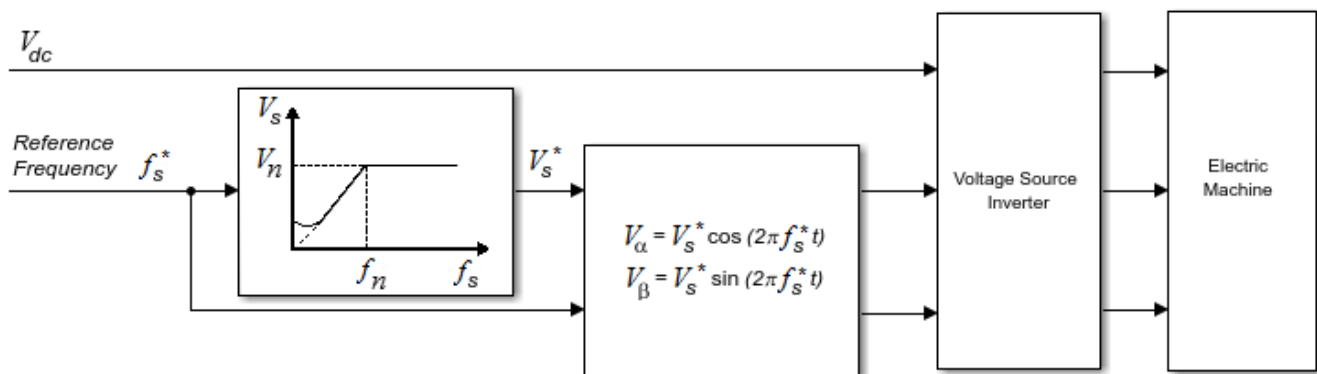
Induction machine V/f control

**Library:** Simscape / Electrical / Control / Induction Machine Control



## Description

The Induction Machine Scalar Control block implements an induction machine scalar, that is V/f or V/Hz, control structure. The diagram shows the open-loop V/f control structure that the block implements.



## Equations

The Induction Machine Scalar Control block computes the magnitude of the stator voltage based on the reference frequency,  $f_s^*$ , as:

$$V_s^* = \left( \frac{V_n - V_{min}}{f_n - f_{min}} \right) f_s^*,$$

where:

- $V_n$  is the rated voltage.
- $V_{min}$  is the minimum voltage.
- $f_n$  is the rated electrical frequency.
- $f_{min}$  is the minimum frequency.

The voltage components in the stationary reference frame are:

$$V_\alpha = V_s^* \cos(2\pi f_s^* t)$$

and



$$V_{\beta} = V_s^* \sin(2\pi f_s^* t).$$

The block obtains  $V_{abc}$  from  $V_{\alpha}$  and  $V_{\beta}$  by using an inverse Clarke transformation.

## Ports

### Input

#### **fRef — Frequency**

scalar

Reference electrical frequency.

Example: Example

Data Types: single | double

### Output

#### **Vabc — Voltage**

vector

Reference phase voltages.

Example: Example

Data Types: single | double

## Parameters

#### **Rated electrical frequency (Hz) — Frequency**

60 (default) | positive scalar

Nominal frequency.

#### **Rated voltage (V) — Voltage**

550 (default) | positive and greater than the value of the **Minimum voltage (V)** parameter

Nominal voltage.

#### **Minimum voltage (V) — Voltage**

10 (default) | zero or positive

Lower bound for the voltage.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## **Version History**

**Introduced in R2017b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

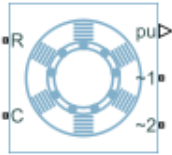
### **Blocks**

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control (Single-Phase) | Induction Machine Flux Observer

# Induction Machine Squirrel Cage

Squirrel-cage-rotor induction machine with per-unit or SI parameterization

**Library:** Simscape / Electrical / Electromechanical / Asynchronous



## Description

The Induction Machine Squirrel Cage block models a squirrel-cage-rotor induction machine with fundamental parameters expressed in per-unit or in the International System of Units (SI). A squirrel-cage-rotor induction machine is a type of induction machine. All stator connections are accessible on the block. Therefore, you can model soft-start regimes using a switch between wye and delta configurations. If you need access to the rotor windings, use the Induction Machine Wound Rotor block instead.

Connect port ~1 to a three-phase circuit. To connect the stator in delta configuration, connect a Phase Permute block between ports ~1 and ~2. To connect the stator in wye configuration, connect port ~2 to a Grounded Neutral (Three-Phase) or a Floating Neutral (Three-Phase) block.

## Induction Machine Initialization Using Load-Flow Target Values

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize the machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”. For an example that shows how initialize an induction machine using data from a load flow analysis, see “Induction Motor Initialization with Loadflow”.

## Equations

For the SI implementation, the block converts the SI values that you enter to per-unit values for simulation. The converted values are based on the machine being connected in a delta-winding configuration.

For the per-unit implementation, you must specify the resistances and inductances in the impedances tab based on the machine being connected in a delta-winding configuration.

For information on the relationship between SI and per-unit machine parameters, see “Per-Unit Conversion for Machine Parameters”. For information on per-unit parameterization, see “Per-Unit System of Units”.

The induction machine equations are expressed with respect to a synchronous reference frame, defined by

$$\theta_e(t) = \int_0^t 2\pi f_{rated} dt,$$

where  $f_{rated}$  is the value of the **Rated electrical frequency** parameter.

The Park transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. The Park transformation is defined by

$$P_s = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

where  $\theta_e$  is the electrical angle.

The Park transformation is used to define the per-unit induction machine equations. The stator voltage equations are defined by

$$v_{ds} = \frac{1}{\omega_{base}} \frac{d\psi_{ds}}{dt} - \omega\psi_{qs} + R_s i_{ds},$$

$$v_{qs} = \frac{1}{\omega_{base}} \frac{d\psi_{qs}}{dt} + \omega\psi_{ds} + R_s i_{qs},$$

and

$$v_{0s} = \frac{1}{\omega_{base}} \frac{d\psi_{0s}}{dt} + R_s i_{0s},$$

where:

- $v_{ds}$ ,  $v_{qs}$ , and  $v_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator voltages, defined by

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages across ports ~1 and ~2.

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_{ds}$ ,  $\psi_{qs}$ , and  $\psi_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages.
- $R_s$  is the stator resistance.
- $i_{ds}$ ,  $i_{qs}$ , and  $i_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents defined by

$$\begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{0s} \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing from port ~1 to port ~2.

The rotor voltage equations are defined by

$$v_{dr} = \frac{1}{\omega_{base}} \frac{d\psi_{dr}}{dt} - (\omega - \omega_r)\psi_{qr} + R_{rd}i_{dr} = 0$$

and

$$v_{qr} = \frac{1}{\omega_{base}} \frac{d\psi_{qr}}{dt} + (\omega - \omega_r)\psi_{dr} + R_{rd}i_{qr} = 0,$$

where:

- $v_{dr}$  and  $v_{qr}$  are the  $d$ -axis and  $q$ -axis rotor voltages.
- $\psi_{dr}$  and  $\psi_{qr}$  are the  $d$ -axis and  $q$ -axis rotor flux linkages.
- $\omega$  is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- $\omega_r$  is the per-unit mechanical rotational speed.
- $R_{rd}$  is the rotor resistance referred to the stator.
- $i_{dr}$  and  $i_{qr}$  are the  $d$ -axis and  $q$ -axis rotor currents.

The stator flux linkage equations are defined by

$$\psi_{ds} = L_{ss}i_{ds} + L_m i_{dr},$$

$$\psi_{qs} = L_{ss}i_{qs} + L_m i_{qr},$$

and

$$\psi_{0s} = L_{ss}i_{0s},$$

where  $L_{ss}$  is the stator self-inductance and  $L_m$  is the magnetizing inductance.

The rotor flux linkage equations are defined by

$$\psi_{dr} = L_{rrd}i_{dr} + L_m i_{ds}$$

and

$$\psi_{qr} = L_{rrd}i_{qr} + L_m i_{qs},$$

where  $L_{rrd}$  is the rotor self-inductance referred to the stator.

The rotor torque is defined by

$$T = \psi_{ds}i_{qs} - \psi_{qs}i_{ds}.$$

The stator self-inductance  $L_{ss}$ , stator leakage inductance  $L_{ls}$ , and magnetizing inductance  $L_m$  are related by

$$L_{ss} = L_{ls} + L_m.$$

The rotor self-inductance  $L_{rrd}$ , rotor leakage inductance  $L_{lrd}$ , and magnetizing inductance  $L_m$  are related by

$$L_{rrd} = L_{lrd} + L_m.$$

When a saturation curve is provided, the equations to determine the saturated magnetizing inductance as a function of magnetizing flux are:

$$L_{m\_sat} = f(\psi_m)$$

$$\psi_m = \sqrt{\psi_{dm}^2 + \psi_{qm}^2}$$

For no saturation, the equation reduces to

$$L_{m\_sat} = L_m$$

### Plotting and Display Options

You can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select an option:

- **Display Base Values** — Displays the machine per-unit base values in the MATLAB Command Window.
- **Plot Torque Speed (SI)** — Plots torque versus speed, both measured in SI units, in a MATLAB figure window using the current machine parameters.
- **Plot Torque Speed (pu)** — Plots torque versus speed, both measured in per-unit, in a MATLAB figure window using the current machine parameters.
- **Plot Open-Circuit Saturation** — Plots terminal voltage versus no-load line current, both in per-unit, in a MATLAB figure window. The plot contains three traces:
  - Unsaturated — Stator magnetizing inductance (unsaturated).
  - Saturated — Open-circuit lookup table ( $v$  versus  $i$ ) you specify.
  - Derived — Open-circuit lookup table derived from the per-unit open-circuit lookup table ( $v$  versus  $i$ ) you specify. This data is used to calculate the saturated magnetizing inductance,  $L_{m\_sat}$ , and the saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_m$ , characteristics.
- **Plot Saturation Factor** — Plots saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_m$ , in a MATLAB figure window using the machine parameters. This parameter is derived from other parameters that you specify:
  - No-load line current saturation data,  $i$
  - Terminal voltage saturation data,  $v$
  - Leakage inductance,  $L_{ls}$
- **Plot Saturated Inductance** — Plots magnetizing inductance,  $L_{m\_sat}$ , versus magnetic flux linkage,  $\psi_m$ , in a MATLAB figure window using the machine parameters. This parameter is derived from other parameters that you specify:
  - No-load line current saturation data,  $i$
  - Terminal voltage saturation data,  $v$
  - Leakage inductance,  $L_{ls}$

For the SI implementation,  $v$  is in V (phase-phase RMS) and  $i$  is in A (rms).

## Model Thermal Effects

You can expose thermal ports to simulate the effects of generated heat and motor temperature. To expose the thermal ports and the **Thermal** parameters, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The type of variables that are visible in the **Initial Targets** section depends on the initialization method that you select, in the **Main** section, for the **Initialization option** parameter. To specify target values using:

- **Flux variables** — Set the **Initialization option** parameter to **Set targets for flux variables**.
- **Data from a load-flow analysis** — Set the **Initialization option** parameter to **Set targets for load flow variables**.

If you select **Set targets for load flow variables**, to fully specify the initial condition, you must include an initialization constraint in the form of a high-priority target value. For example, if your induction machine is connected to an Inertia block, the initial condition for the induction machine is completely specified if, in the **Initial Targets** section of the Inertia block, the **Priority** for **Rotational velocity** is set to High. Alternatively, you could set the **Priority** to None for the Inertia block **Rotational velocity**, and instead set the **Priority** for the induction machine block **Slip**, **Real power generated**, or **Mechanical power consumed** to High.

## Ports

### Input

#### **pu — Per-unit measurements output port**

physical

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- *pu\_torque*

- *pu\_velocity*
- *pu\_vds*
- *pu\_vqs*
- *pu\_v0s*
- *pu\_ids*
- *pu\_iqs*
- *pu\_i0s*

## **Conserving**

### **R — Machine rotor**

mechanical

Mechanical rotational conserving port associated with the machine rotor.

### **C — Machine case**

mechanical

Mechanical rotational conserving port associated with the machine case.

### **~1 — Stator positive-end connections**

electrical

Expandable three-phase port associated with the stator positive-end connections.

#### **Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

### **~2 — Stator negative-end connections**

electrical

Expandable three-phase port associated with the stator negative-end connections.

#### **Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

### **a1 — a positive-end connection**

electrical

Electrical conserving port associated with *a* positive-end connection.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

### **b1 — b positive-end connection**

electrical

Electrical conserving port associated with *b* positive-end connection.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.



**c1 – c positive-end connection**

electrical

Electrical conserving port associated with *c* positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**a2 – a negative-end connection**

electrical

Electrical conserving port associated with *a* negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b2 – b negative-end connection**

electrical

Electrical conserving port associated with *b* negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c2 – c negative-end connection**

electrical

Electrical conserving port associated with *c* negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA – a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC – c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters**

All default parameter values are based on a machine delta-winding configuration.

**Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power — Rated apparent power**

15e3 V\*A (default)

Rated apparent power of the induction machine at the electrical side.

**Rated voltage — RMS voltage**

220 V (default)

RMS line-line voltage.

**Rated electrical frequency — Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency corresponding to the rated apparent power.

**Number of pole pairs — Machine pole pairs**

1 (default)

Number of machine pole pairs.

**Parameterization unit — Unit system for block parameterization**

SI (default) | Per unit

Unit system for block parameterization. Choose between SI, the international system of units, and Per unit, the per-unit system.

**Dependencies**

Selecting:

- SI exposes SI parameters in the **Impedances** and **Saturation** settings.
- Per unit exposes per-unit parameters in the **Impedances** and **Saturation** settings.

**Squirrel cage — Squirrel cage specification option**

Single squirrel cage (default) | Double squirrel cage

Option to specify a single or double squirrel cage for the machine.

Setting this parameter to Double squirrel cage exposes **Impedances** parameters for the second cage.

**Zero sequence — Zero sequence**

Include (default) | Exclude

Zero-sequence model:

- Include — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- Exclude — Prioritize simulation speed for desktop simulation or real-time deployment.

**Dependencies**

If this parameter is set to:

- Include and **Parameterization unit** is set to SI — The **Stator zero-sequence reactance, X0** parameter in the **Impedances** settings is visible.
- Include and **Parameterization unit** is set to Per unit — The **Stator zero-sequence inductance, pu\_L0 (pu)** parameter in the **Impedances** settings is visible.
- Exclude — The stator zero-sequence parameter in the **Impedances** settings is not visible.

**Initialization option — Initialization method**

Set targets for flux variables (default) | Set targets for load flow variables

Initialization method. You can initialize a machine for steady-state simulation using either flux data or data from a load-flow analysis.

If you select Set targets for load flow variables, to fully specify the initial condition, you must include an initialization constraint in the form of a high-priority target value. For example, if your induction machine is connected to an Inertia block, the initial condition for the induction machine is completely specified if, in the **Initial Targets** section of the Inertia block, the **Priority** for **Rotational velocity** is set to High. Alternatively, you could set the **Priority** to None for the Inertia block **Rotational velocity**, and instead set the **Priority** for the induction machine block **Slip, Real power generated**, or **Mechanical power consumed** to High.

**Dependencies**

The type of variables that are visible in the **Initial Targets** section depends on the initialization method that you select, in the **Main** section, for the **Initialization option** parameter. To specify target values using:

- Flux variables — Set the **Initialization option** parameter to `Set targets for flux variables`.
- Data from a load-flow analysis — Set the **Initialization option** parameter to `Set targets for load flow variables`.

### **Impedances**

For the **Parameterization unit** parameter in the **Main** settings, select `SI` to expose SI parameters or `Per unit` to expose per-unit parameters.

#### **Stator resistance, $R_s$ — Stator resistance**

0.25 Ohm (default)

Stator resistance.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to `SI`.

#### **Stator leakage reactance, $X_{ls}$ — Stator leakage reactance**

0.4 Ohm (default)

Stator leakage reactance.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to `SI`.

#### **Referred rotor resistance, $R_r'$ — Referred rotor resistance**

0.14 Ohm (default)

Rotor resistance referred to the stator.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to `SI` and the **Squirrel cage** parameter in the **Main** setting to `Single squirrel cage`.

#### **Referred rotor leakage reactance, $X_{lr}'$ — Referred rotor leakage reactance**

0.41 Ohm (default)

Rotor leakage reactance referred to the stator.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to `SI` and the **Squirrel cage** parameter in the **Main** setting to `Single squirrel cage`.

#### **Referred rotor resistance in cage 1, $R_{r1}'$ — Referred rotor resistance in the first cage**

0.28 Ohm (default)

Rotor resistance in cage 1.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor leakage reactance in cage 1,  $X_{lr1}'$  — Referred rotor leakage reactance in the first cage**

0.82 Ohm (default)

Rotor leakage reactance in cage 1.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor resistance in cage 2,  $R_{r2}'$  — Referred rotor resistance in the second cage**

0.28 Ohm (default)

Rotor resistance in cage 2.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor leakage reactance in cage 2,  $X_{lr2}'$  — Referred rotor leakage reactance in the first cage**

0.82 Ohm (default)

Rotor leakage reactance in cage 2.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Magnetizing reactance,  $X_m$  — Magnetizing reactance**

17 Ohm (default)

Magnetizing reactance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI.

**Stator zero-sequence reactance,  $X_0$  — Stator zero-sequence reactance**

0.4 Ohm (default)

Stator zero-sequence reactance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to SI and the **Zero sequence** parameter in the **Main** setting to Include.

**Stator resistance,  $R_s$  (pu) — Per-unit stator resistance**

0.0258 (default)

Per-unit stator resistance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit.

**Stator leakage inductance,  $L_{ls}$  (pu) — Per-unit stator leakage inductance**

0.0413 (default)

Per-unit stator leakage inductance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit.

**Referred rotor resistance,  $R_r'$  (pu) — Per-unit referred rotor resistance**

0.0145 (default)

Per-unit rotor resistance referred to the stator.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Single squirrel cage.

**Referred rotor leakage inductance,  $L_{lr}'$  (pu) — Per-unit referred rotor leakage inductance**

0.0424 (default)

Per-unit rotor leakage inductance referred to the stator.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Single squirrel cage.

**Referred rotor resistance in cage 1,  $R_{r1}'$  (pu) — Per-unit referred rotor resistance in cage 1**

0.0290 (default)

Per-unit rotor resistance in cage 1.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor leakage inductance in cage 1,  $L_{lr1}'$  (pu) — Per-unit referred rotor leakage inductance**

0.0848 (default)

Per-unit rotor leakage inductance in cage 1.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor resistance in cage 2,  $R_{r2'}$  (pu) — Per-unit referred rotor resistance in cage 2**

0.0290 (default)

Per-unit rotor resistance in cage 2.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Referred rotor leakage inductance in cage 2,  $L_{lr2'}$  (pu) — Per-unit referred rotor leakage inductance**

0.0848 (default)

Per-unit rotor leakage inductance in cage 2.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Squirrel cage** parameter in the **Main** setting to Double squirrel cage.

**Magnetizing inductance,  $L_m$  (pu) — Per-unit magnetizing inductance**

1.7562 (default)

Per-unit magnetizing inductance, that is, the peak value of stator-rotor mutual inductance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit.

**Stator zero-sequence inductance,  $L_0$  (pu) — Per-unit stator zero-sequence inductance**

0.0413 (default)

Per-unit stator zero-sequence inductance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit and the **Zero sequence** parameter in the **Main** setting to Include.

**Saturation**

For the **Parameterization unit** parameter in the **Main** settings, select SI to expose SI parameters or Per unit to expose per-unit parameters.

The saturation characteristic is based on the machine being connected in a delta-winding configuration.

**Magnetic saturation representation — Magnetic saturation representation**

None (default) | Open-circuit lookup table (v versus i)

Block magnetic saturation representation.

**Dependencies**

If this parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ), related parameters are visible.

**No-load line current saturation data,  $i$  (rms) — No-load line RMS current saturation data**

[0, 4, 9, 18, 25, 34, 50, 68, 95, 120] A (default)

Current  $i$  data populates the voltage  $v$  versus field current  $i$  lookup table. This parameter must contain a vector with at least 10 elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to SI.

**Terminal voltage saturation data,  $v$  (phase-phase, rms) — Terminal RMS voltage saturation data**

[0, 88, 154, 198, 220, 242, 264, 286, 308, 327] V (default)

Terminal voltage  $v$  data populates the voltage  $v$  versus current  $i$  lookup table. This parameter must contain a vector with at least 10 elements. The number of elements must match the number of elements in the vector for the **No-load line current saturation data,  $i$  (rms)** parameter.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to SI.

**Per-unit no-load line current saturation data,  $i$  — Per-unit no-load line current saturation data**

[0, .176, .396, .792, 1.1, 1.496, 2.2, 2.992, 4.18, 5.28] (default)

Current  $i$  data populates the voltage  $v$  versus field current  $i$  lookup table. This parameter must contain a vector with at least 10 elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to Per unit.

**Per-unit terminal voltage saturation data,  $v$  — Per-unit terminal voltage saturation data**

[0, .4, .7, .9, 1, 1.1, 1.2, 1.3, 1.4, 1.4864] (default)

Terminal voltage  $v$  data populates the voltage  $v$  versus current  $i$  lookup table. This parameter must contain a vector with at least 10 elements. The number of elements must match the number of elements in the vector for the **Per-unit no-load line current saturation data,  $i$**  parameter.



**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i) and, in the **Main** settings, the **Parameterization unit** parameter is set to Per unit.

**Thermal**

These parameters appear only for blocks with exposed thermal ports.

**Measurement temperature — Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Temperature coefficient**

$3.93e-3$  1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value,  $3.93e-3$  1/K, is for copper.

**Cage resistance temperature coefficient — Rotor temperature coefficient**

$4.4e-3$  (default) | 1/K | scalar

Coefficient relating cage resistance to temperature for the rotor. The default value,  $4.4e-3$  1/K, is for aluminium.

**Thermal mass for each stator winding — Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass — Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

**Version History****Introduced in R2013b****Electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports. Also, the simlog format is identical for Composite three-phase ports and Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## Saturation characteristic update

*Behavior changed in R2020a*

From R2020a forward, for the saturation characteristic, the Induction Machine Squirrel Cage block takes into account the magnetic flux due to the leakage inductance.

The default values for these parameters have changed:

- **Stator leakage reactance,  $X_{ls}$**
- **Stator zero-sequence reactance,  $X_0$**
- **Stator leakage inductance,  $L_{ls}$  (pu)**
- **Stator zero-sequence inductance,  $L_0$  (pu)**
- **Terminal voltage saturation data,  $v$  (phase-phase, rms)**
- **Per-unit terminal voltage saturation data,  $v$**

As a result of these changes, inside a model saved in an earlier release, if you have used the Induction Machine Squirrel Cage block with saturation enabled, review the saturation data.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York: McGraw Hill, 1993.
- [2] Lyshevski, S. E. *Electromechanical Systems, Electric Machines and Applied Mechatronics*. Boca Raton, FL: CRC Press, 1999.
- [3] Ojo, J. O., Consoli, A., and Lipo, T. A., "An improved model of saturated induction machines", *IEEE Transactions on Industry Applications*. Vol. 26, no. 2, pp. 212-221, 1990.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Induction Machine (Single-Phase) | Induction Machine Measurement | Induction Machine Wound Rotor

### Blocks

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Flux Observer | Induction Machine Scalar Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Field-Oriented Control (Single-Phase)

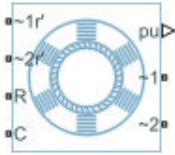
### Topics

"Perform a Load-Flow Analysis Using Simscape Electrical"  
"Expand and Collapse Three-Phase Ports on a Block"

# Induction Machine Wound Rotor

Wound-rotor induction machine with per-unit or SI parameterization

**Library:** Simscape / Electrical / Electromechanical / Asynchronous



## Description

The Induction Machine Wound Rotor block models a wound-rotor asynchronous machine with fundamental parameters expressed in per-unit or in the International System of Units (SI). A wound-rotor asynchronous machine is a type of induction machine. All stator and rotor connections are accessible on the block. Therefore, you can model soft-start regimes using a switch between wye and delta configurations or by increasing rotor resistance. If you do not need access to the rotor windings, use the Induction Machine Squirrel Cage block instead.

Connect port **~1** to a three-phase circuit. To connect the stator in delta configuration, connect a Phase Permute block between ports **~1** and **~2**. To connect the stator in wye configuration, connect port **~2** to a Grounded Neutral or a Floating Neutral block. If you do not need to vary rotor resistance, connect rotor port **~1r'** to a Floating Neutral block and rotor port **~2r'** to a Grounded Neutral block.

The rotor circuit is referred to the stator. Therefore, when you use the block in a circuit, refer any additional circuit parameters to the stator.

## Induction Machine Initialization Using Load-Flow Target Values

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize the machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”. For an example that shows how initialize an induction machine using data from a load flow analysis, see “Induction Motor Initialization with Loadflow”.

## Equations

For the SI implementation, the block converts the SI values that you enter to per-unit values for simulation. The converted values are based on the machine being connected in a delta-winding configuration.

For the per-unit implementation, you must specify the resistances and inductances in the impedances tab based on the machine being connected in a delta-winding configuration.

For information on the relationship between SI and per-unit machine parameters, see “Per-Unit Conversion for Machine Parameters”. For information on per-unit parameterization, see “Per-Unit System of Units”.

The asynchronous machine equations are expressed with respect to a synchronous reference frame, defined by

$$\theta_e(t) = \int_0^t 2\pi f_{rated} dt,$$

where  $f_{rated}$  is the value of the **Rated electrical frequency** parameter.

The Park transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. The Park transformation is defined by

$$P_s = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

where  $\theta_e$  is the electrical angle.

The rotor equations are mapped to another reference frame, defined by the difference between the electrical angle and the product of rotor angle  $\theta_r$  and number of pole pairs  $N$ :

$$P_r = \frac{2}{3} \begin{bmatrix} \cos(\theta_e - N\theta_r) & \cos(\theta_e - N\theta_r - \frac{2\pi}{3}) & \cos(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ -\sin(\theta_e - N\theta_r) & -\sin(\theta_e - N\theta_r - \frac{2\pi}{3}) & -\sin(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformation is used to define the per-unit asynchronous machine equations. The stator voltage equations are defined by

$$v_{ds} = \frac{1}{\omega_{base}} \frac{d\psi_{ds}}{dt} - \omega\psi_{qs} + R_s i_{ds},$$

$$v_{qs} = \frac{1}{\omega_{base}} \frac{d\psi_{qs}}{dt} + \omega\psi_{ds} + R_s i_{qs},$$

and

$$v_{0s} = \frac{1}{\omega_{base}} \frac{d\psi_{0s}}{dt} + R_s i_{0s},$$

where:

- $v_{ds}$ ,  $v_{qs}$ , and  $v_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator voltages, defined by

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{0s} \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages across ports ~1 and ~2.

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_{ds}$ ,  $\psi_{qs}$ , and  $\psi_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages.
- $R_s$  is the stator resistance.
- $i_{ds}$ ,  $i_{qs}$ , and  $i_{0s}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents, defined by

$$\begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{0s} \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing from port  $\sim 1$  to port  $\sim 2$ .

The rotor voltage equations are defined by

$$v_{dr} = \frac{1}{\omega_{base}} \frac{d\psi_{dr}}{dt} - (\omega - \omega_r)\psi_{qr} + R_{rd}i_{dr},$$

$$v_{qr} = \frac{1}{\omega_{base}} \frac{d\psi_{qr}}{dt} + (\omega - \omega_r)\psi_{dr} + R_{rd}i_{qr},$$

and

$$v_{0r} = \frac{1}{\omega_{base}} \frac{d\psi_{0r}}{dt} + R_{rd}i_{0s},$$

where:

- $v_{dr}$ ,  $v_{qr}$ , and  $v_{0r}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence rotor voltages, defined by

$$\begin{bmatrix} v_{dr} \\ v_{qr} \\ v_{0r} \end{bmatrix} = P_r \begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix}.$$

$v_{ar}$ ,  $v_{br}$ , and  $v_{cr}$  are the rotor voltages across ports  $\sim 1r'$  and  $\sim 2r'$ .

- $\psi_{dr}$ ,  $\psi_{qr}$ , and  $\psi_{0r}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence rotor flux linkages.
- $\omega$  is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- $\omega_r$  is the per-unit mechanical rotational speed.
- $R_{rd}$  is the rotor resistance referred to the stator.
- $i_{dr}$ ,  $i_{qr}$ , and  $i_{0r}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence rotor currents, defined by

$$\begin{bmatrix} i_{dr} \\ i_{qr} \\ i_{0r} \end{bmatrix} = P_r \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix}.$$

$i_{ar}$ ,  $i_{br}$ , and  $i_{cr}$  are the rotor currents flowing from port  $\sim 1r'$  to port  $\sim 2r'$ .

The stator flux linkage equations are defined by

$$\psi_{ds} = L_{ss}i_{ds} + L_m i_{dr},$$

$$\psi_{qs} = L_{ss}i_{qs} + L_m i_{qr},$$

and

$$\psi_{0s} = L_{ss}i_{0s},$$

where  $L_{ss}$  is the stator self-inductance and  $L_m$  is the magnetizing inductance.

The rotor flux linkage equations are defined by

$$\psi_{dr} = L_{rrd}i_{dr} + L_m i_{ds}$$

$$\psi_{qr} = L_{rrd}i_{qr} + L_m i_{qs},$$

and

$$\psi_{0r} = L_{rrd}i_{0r},$$

where  $L_{rrd}$  is the rotor self-inductance referred to the stator.

The rotor torque is defined by

$$T = \psi_{ds}i_{qs} - \psi_{qs}i_{ds}.$$

The stator self-inductance  $L_{ss}$ , stator leakage inductance  $L_{ls}$ , and magnetizing inductance  $L_m$  are related by

$$L_{ss} = L_{ls} + L_m.$$

The rotor self-inductance  $L_{rrd}$ , rotor leakage inductance  $L_{lr}$ , and magnetizing inductance  $L_m$  are related by

$$L_{rrd} = L_{lr} + L_m.$$

When a saturation curve is provided, the equations to determine the saturated magnetizing inductance as a function of magnetizing flux are:

$$L_{m\_sat} = f(\psi_m)$$

$$\psi_m = \sqrt{\psi_{dm}^2 + \psi_{qm}^2}$$

For no saturation, the equation reduces to

$$L_{m\_sat} = L_m$$

### Plotting and Display Options

You can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select an option:

- **Display Base Values** — Displays the machine per-unit base values in the MATLAB Command Window.
- **Plot Torque Speed (SI)** — Plots torque versus speed, both measured in SI units, in a MATLAB figure window using the current machine parameters.
- **Plot Torque Speed (pu)** — Plots torque versus speed, both measured in per-unit, in a MATLAB figure window using the current machine parameters.

- **Plot Open-Circuit Saturation** — Plots terminal voltage versus no-load line current, both in per-unit, in a MATLAB figure window. The plot contains three traces:
  - Unsaturated — Stator magnetizing inductance (unsaturated).
  - Saturated — Open-circuit lookup table ( $v$  versus  $i$ ) you specify.
  - Derived — Open-circuit lookup table derived from the per-unit open-circuit lookup table ( $v$  versus  $i$ ) you specify. This data is used to calculate the saturated magnetizing inductance,  $L_{m\ sat}$ , and the saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_m$ , characteristics.
- **Plot Saturation Factor** — Plots saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_m$ , in a MATLAB figure window using the machine parameters. This parameter is derived from other parameters that you specify:
  - No-load line current saturation data,  $i$
  - Terminal voltage saturation data,  $v$
  - Leakage inductance,  $L_{ls}$
- **Plot Saturated Inductance** — Plots magnetizing inductance,  $L_{m\ sat}$ , versus magnetic flux linkage,  $\psi_m$ , in a MATLAB figure window using the machine parameters. This parameter is derived from other parameters that you specify:
  - No-load line current saturation data,  $i$
  - Terminal voltage saturation data,  $v$
  - Leakage inductance,  $L_{ls}$

For the SI implementation,  $v$  is in V (phase-phase RMS) and  $i$  is in A (rms).

### Model Thermal Effects

You can expose thermal ports to simulate the effects of generated heat and motor temperature. To expose the thermal ports and the **Thermal** parameters, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The type of variables that are visible in the **Initial Targets** section depends on the initialization method that you select, in the **Main** section, for the **Initialization option** parameter. To specify target values using:

- Flux variables — Set the **Initialization option** parameter to `Set targets for flux variables`.
- Data from a load-flow analysis — Set the **Initialization option** parameter to `Set targets for load flow variables`.

If you select `Set targets for load flow variables`, to fully specify the initial condition, you must include an initialization constraint in the form of a high-priority target value. For example, if your induction machine is connected to an Inertia block, the initial condition for the induction machine is completely specified if, in the **Initial Targets** section of the Inertia block, the **Priority** for **Rotational velocity** is set to `High`. Alternatively, you could set the **Priority** to `None` for the Inertia block **Rotational velocity**, and instead set the **Priority** for the induction machine block **Slip, Real power generated**, or **Mechanical power consumed** to `High`.

## Ports

### Output

#### **pu — Per-unit measurements output port**

physical

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- *pu\_torque*
- *pu\_velocity*
- *pu\_vds*
- *pu\_vqs*
- *pu\_v0s*
- *pu\_ids*
- *pu\_iqs*
- *pu\_i0s*

### Conserving

#### **R — Machine rotor**

mechanical

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical

Mechanical rotational conserving port associated with the machine case.

#### **~1 — Stator positive-end connections**

electrical

Expandable three-phase port associated with the stator positive-end connections.



**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**~2 — Stator negative-end connections**

electrical

Expandable three-phase port associated with the stator negative-end connections.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**~1r' — Rotor positive-end connections**

electrical

Expandable three-phase port associated with the rotor positive-end connections.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**~2r' — Rotor negative-end connections**

electrical

Expandable three-phase port associated with the rotor negative-end connections.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**a1 — a stator positive-end connection**

electrical

Electrical conserving port associated with *a* stator positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b1 — b stator positive-end connection**

electrical

Electrical conserving port associated with *b* stator positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c1 — c stator positive-end connection**

electrical

Electrical conserving port associated with *c* stator positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**a2 — a stator negative-end connection**

electrical

Electrical conserving port associated with *a* stator negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b2 — b stator negative-end connection**

electrical

Electrical conserving port associated with *b* stator negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c2 — c negative-end connection**

electrical

Electrical conserving port associated with *c* stator negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**a1r' — a rotor positive-end connection**

electrical

Electrical conserving port associated with *a* rotor positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b1r' — b rotor positive-end connection**

electrical

Electrical conserving port associated with *b* rotor positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c1r' — c rotor positive-end connection**

electrical

Electrical conserving port associated with *c* rotor positive-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**a2r' — a rotor negative-end connection**

electrical

Electrical conserving port associated with *a* rotor negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b2r' — b rotor negative-end connection**

electrical

Electrical conserving port associated with *b* rotor negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c2r' — c rotor-end connection**

electrical

Electrical conserving port associated with *c* rotor negative-end connection.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HAr — a-phase terminal thermal port**

thermal

Thermal conserving port associated with rotor winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HBr – b-phase terminal thermal port**

thermal

Thermal conserving port associated with rotor winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HCr – c-phase terminal thermal port**

thermal

Thermal conserving port associated with rotor winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters**

All default parameter values are based on a machine delta-winding configuration.

**Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

**Main****Electrical connection – Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power – Rated apparent power**

15e3 V\*A (default)

Rated apparent power of the induction machine.

**Rated voltage – RMS voltage**

220 V (default)

RMS line-line voltage.

**Rated electrical frequency – Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency corresponding to the rated apparent power.

**Number of pole pairs – Machine pole pairs**

1 (default)

Number of machine pole pairs.

**Parameterization unit — Unit system for block parameterization**

SI (default) | Per unit

Unit system for block parameterization. Choose between SI, the international system of units, and Per unit, the per-unit system.

**Dependencies**

Selecting:

- SI exposes SI parameters in the **Impedances** and **Saturation** settings.
- Per unit exposes per-unit parameters in the **Impedances** and **Saturation** settings.

**Zero sequence — Zero sequence**

Include (default) | Exclude

Zero-sequence model:

- Include — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- Exclude — Prioritize simulation speed for desktop simulation or real-time deployment.

**Dependencies**

If this parameter is set to:

- Include and **Parameterization unit** is set to SI — The **Stator zero-sequence reactance, X0** parameter in the **Impedances** settings is visible.
- Include and **Parameterization unit** is set to Per unit — The **Stator zero-sequence inductance, L0 (pu)** parameter in the **Impedances** settings is visible.
- Exclude — The stator zero-sequence parameter in the **Impedances** settings is not visible.

**Initialization option — Initialization method**

Set targets for flux variables (default) | Set targets for load flow variables

Initialization method. You can initialize a machine for steady-state simulation using either flux data or data from a load-flow analysis.

If you select Set targets for load flow variables, to fully specify the initial condition, you must include an initialization constraint in the form of a high-priority target value. For example, if your induction machine is connected to an Inertia block, the initial condition for the induction machine is completely specified if, in the **Initial Targets** section of the Inertia block, the **Priority** for **Rotational velocity** is set to High. Alternatively, you could set the **Priority** to None for the Inertia block **Rotational velocity**, and instead set the **Priority** for the induction machine block **Slip, Real power generated**, or **Mechanical power consumed** to High.

**Dependencies**

The type of variables that are visible in the **Initial Targets** section depends on the initialization method that you select, in the **Main** section, for the **Initialization option** parameter. To specify target values using:

- Flux variables — Set the **Initialization option** parameter to Set targets for flux variables.

- Data from a load-flow analysis — Set the **Initialization option** parameter to **Set targets** for load flow variables.

### **Impedances**

For the **Parameterization unit** parameter in the **Main** settings, select **SI** to expose SI parameters or **Per unit** to expose per-unit parameters.

#### **Stator resistance, $R_s$ — Stator resistance**

0.25 Ohm (default)

Stator resistance.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

#### **Stator leakage reactance, $X_l$ — Stator leakage reactance**

0.4 Ohm (default)

Stator leakage reactance.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

#### **Referred rotor resistance, $R_r'$ — Referred rotor resistance**

0.14 Ohm (default)

Rotor resistance referred to the stator.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

#### **Referred rotor leakage reactance, $X_{lr}'$ — Referred rotor leakage reactance**

0.41 Ohm (default)

Rotor leakage reactance referred to the stator.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

#### **Magnetizing reactance, $X_m$ — Magnetizing reactance**

17 Ohm (default)

Magnetizing reactance.

#### **Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

**Stator zero-sequence reactance,  $X_0$  — Stator zero-sequence reactance**

0.4 Ohm (default)

Stator zero-sequence reactance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **SI**.

**Stator resistance,  $R_s$  (pu) — Per-unit stator resistance**

0.0258 (default)

Per-unit stator resistance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **Per unit**.

**Stator leakage inductance,  $L_{ls}$  (pu) — Per-unit stator leakage inductance**

0.0413 (default)

Per-unit stator leakage inductance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **Per unit**.

**Referred rotor resistance,  $R_r'$  (pu) — Per-unit referred rotor resistance**

0.0145 (default)

Per-unit rotor resistance referred to the stator.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **Per unit**.

**Referred rotor leakage inductance,  $L_{lr}'$  (pu) — Per-unit referred rotor leakage inductance**

0.0424 (default)

Per-unit rotor leakage inductance referred to the stator.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to **Per unit**.

**Magnetizing inductance,  $L_m$  (pu) — Per-unit magnetizing inductance**

1.7562 (default)

Per-unit magnetizing inductance, that is, the peak value of stator-rotor mutual inductance.

**Dependencies**

This parameter is visible only if you set the **Parameterization unit** parameter in the **Main** setting to Per unit.

**Stator zero-sequence inductance,  $L_0$  (pu) — Per-unit stator zero-sequence inductance**  
0.0413 (default)

Per-unit stator zero-sequence inductance.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Parameterization unit** parameter is set to Per unit and the **Zero sequence** parameter is set to Include.

**Saturation**

For the **Parameterization unit** parameter in the **Main** settings, select SI to expose SI parameters or Per unit to expose per-unit parameters.

The saturation characteristic is based on the machine being connected in a delta-winding configuration.

**Magnetic saturation representation — Magnetic saturation representation**  
None (default) | Open-circuit lookup table (v versus i)

Block magnetic saturation representation.

**Dependencies**

If this parameter is set to Open-circuit lookup table (v versus i), related parameters are visible.

**No-load line current saturation data,  $i$  (rms) — No-load line RMS current saturation data**  
[0, 4, 9, 18, 25, 34, 50, 68, 95, 120] A (default)

Current  $i$  data populates the voltage  $v$  versus field current  $i$  lookup table. This parameter must contain a vector with at least 10 elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i) and, in the **Main** settings, the **Parameterization unit** parameter is set to SI.

**Terminal voltage saturation data,  $v$  (phase-phase, rms) — Terminal RMS voltage saturation data**  
[0, 88, 154, 198, 220, 242, 264, 286, 308, 327] V (default)

Terminal voltage  $v$  data populates the voltage  $v$  versus current  $i$  lookup table. This parameter must contain a vector with at least 10 elements. The number of elements must match the number of elements in the vector for the **No-load line current saturation data,  $i$  (rms)** parameter.



**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to SI.

**Per-unit no-load line current saturation data,  $i$  — Per-unit no-load line current saturation data**

[0, .176, .396, .792, 1.1, 1.496, 2.2, 2.992, 4.18, 5.28] (default)

Current  $i$  data populates the voltage  $v$  versus field current  $i$  lookup table. This parameter must contain a vector with at least 10 elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to Per unit.

**Per-unit terminal voltage saturation data,  $v$  — Per-unit terminal voltage saturation data**

[0, .4, .7, .9, 1, 1.1, 1.2, 1.3, 1.4, 1.4864] (default)

Terminal voltage  $v$  data populates the voltage  $v$  versus current  $i$  lookup table. This parameter must contain a vector with at least 10 elements. The number of elements must match the number of elements in the vector for the **Per-unit no-load line current saturation data,  $i$**  parameter.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table ( $v$  versus  $i$ ) and, in the **Main** settings, the **Parameterization unit** parameter is set to Per unit.

**Thermal**

These parameters appear only for blocks with exposed thermal ports.

**Measurement temperature — Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Temperature coefficient**

$3.93e-3$  1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value,  $3.93e-3$  1/K, is for copper.

**Thermal mass for each stator winding — Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass — Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

### Introduced in R2013b

#### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports. Also, the simlog format is identical for Composite three-phase ports and Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

#### Saturation characteristic update

*Behavior changed in R2020a*

From R2020a forward, for the saturation characteristic, the Induction Machine Wound Rotor block takes into account the magnetic flux due to the leakage inductance.

The default values for these parameters have changed:

- **Stator leakage reactance,  $X_{ls}$**
- **Stator zero-sequence reactance,  $X_0$**
- **Stator leakage inductance,  $L_{ls}$  (pu)**
- **Stator zero-sequence inductance,  $L_0$  (pu)**
- **Terminal voltage saturation data,  $v$  (phase-phase, rms)**
- **Per-unit terminal voltage saturation data,  $v$**

As a result of these changes, inside a model saved in an earlier release, if you have used the Induction Machine Wound Rotor block with saturation enabled, review the saturation data.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Lyshevski, S. E. *Electromechanical Systems, Electric Machines and Applied Mechatronics*. Boca Raton, FL: CRC Press, 1999.
- [3] Ojo, J. O., Consoli, A., and Lipo, T. A., "An improved model of saturated induction machines", *IEEE Transactions on Industry Applications*. Vol. 26, no. 2, pp. 212-221, 1990.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Induction Machine (Single-Phase) | Induction Machine Measurement | Induction Machine Squirrel Cage

### Blocks

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Flux Observer | Induction Machine Scalar Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Field-Oriented Control (Single-Phase)

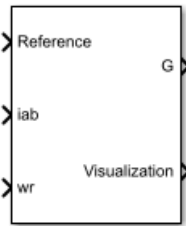
### Topics

“Perform a Load-Flow Analysis Using Simscape Electrical”  
“Expand and Collapse Three-Phase Ports on a Block”

# Induction Machine Field-Oriented Control (Single-Phase)

Per-unit discrete-time single-phase induction machine field-oriented control

**Library:** Simscape / Electrical / Control / Induction Machine Control

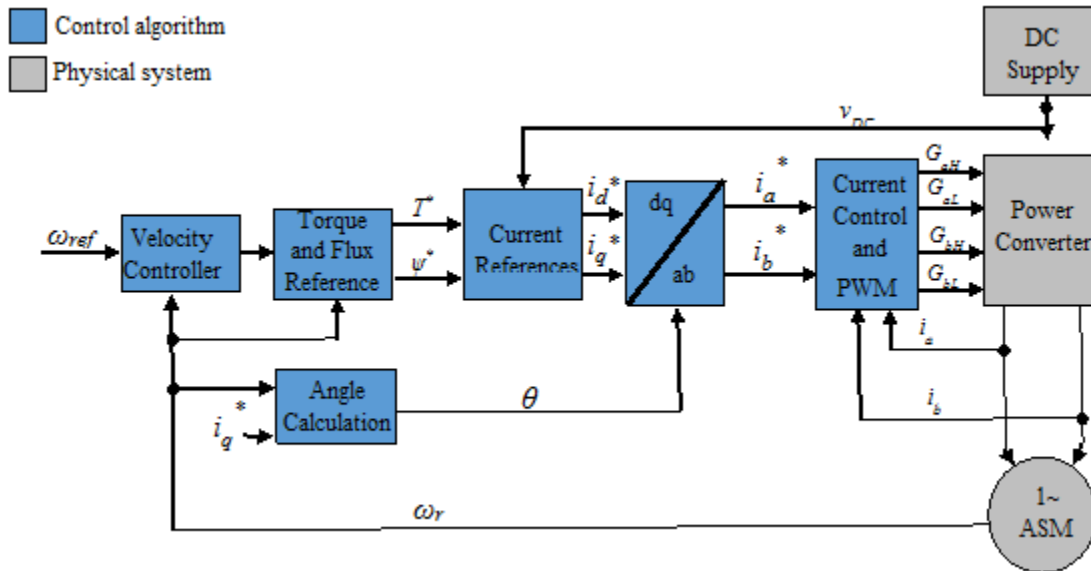


## Description

The Induction Machine Field-Oriented Control (Single-Phase) block implements a single-phase induction machine field-oriented control structure.

## Equations

The field-oriented control architecture for a single-phase induction machine is:



You can provide the torque reference as an input, or, in the case of speed control, generate the reference internally using a PI speed controller. The torque reference as derived from a PI speed controller is:

$$T^* = (K_{p_\omega} + K_{i_\omega} \frac{T_s z}{z-1})(\omega_{ref} - \omega_r)$$

where  $\omega_r$  is the rotor angular mechanical speed in rad/s.

The block generates the flux reference as

$$\psi^* = \frac{2\pi f_n \psi_n}{p \min\left(\left|\omega_r\right|, \frac{2\pi f_n}{p}\right)}$$

where:

- $p$  is the number of pole pairs.
- $f_n$  is the rated frequency.
- $\psi_n$  is the rated flux.

Current references are obtained from machine parameters:

$$i_q^* = \frac{(L_{ms} + L_{lar})T^*}{paL_{ms}\psi^*}$$

$$i_d^* = \frac{\psi^*}{a^2L_{ms}}$$

where:

- $L_{ms}$  is the main winding mutual inductance.
- $L_{lar}$  is the main winding rotor leakage inductance.
- $a$  is the auxiliary-to-main windings turn ratio.

The angle is calculated by solving:

$$\frac{d\theta}{dt} = p\omega_r + \frac{a^3L_{ms}R_{ar}}{\psi^*(L_{ms} + L_{lar})}$$

The transformation to the stationary reference frame is done by using:

$$\begin{pmatrix} i_a \\ i_b \end{pmatrix} = \begin{pmatrix} a^2 \sin(\theta) & a^2 \cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{pmatrix} \begin{pmatrix} i_d \\ i_q \end{pmatrix}$$

### Limitations

- The control structure is implemented in a single sample rate.

## Ports

### Input

#### Reference — Torque or angular velocity reference

scalar

Specify the angular velocity reference, in rad/s, or the torque reference, in Nm.

Data Types: `single` | `double`

#### **iab** — Stator currents

scalar

Stator currents for main and auxiliary windings.

Data Types: `single` | `double`

### **wr — Rotor angular speed**

scalar

Measured rotor angular speed, in rad/s.

Data Types: `single` | `double`

## **Output**

### **G — Inverter gate pulses**

`0` | `1`

Inverter gate pulses, specified as a Boolean. The dead-time is not considered.

Data Types: `single` | `double`

### **Visualization — Internal signals for visualization**

`Reference` | `Rotor velocity` |  `$\theta$`  | `iab` | `iabRef` | `TqRef` | `FluxRef`

Bus containing internal signals for visualization. The signals are:

- Reference torque or speed
- Rotor velocity, in rad/s
- $\theta$ , in radians
- Stator currents (*iab*), in A
- Stator current reference (*iabRef*), in A
- Torque Reference (*TqRef*), in Nm
- Flux Reference (*FluxRef*), in Wb

Data Types: `single` | `double` | `bus`

## **Parameters**

### **General**

#### **Control mode — Specify control mode**

`Speed control` (default) | `Torque control`

Specify the control mode. The reference input is taken as angular speed, in rad/s, for `Speed control` and torque, in Nm for `Torque control`.

#### **Rated electrical frequency (Hz) — Specify rated electrical frequency**

`60` (default) | positive scalar

Rated electrical frequency, in Hz.

#### **Number of pole pairs — Specify pole pairs**

`2` (default) | positive scalar

Number of pole pairs.

**Main winding rotor resistance,  $R_{ar}'$  (Ohm) – Specify main winding rotor resistance**

3.7080 (default) | positive scalar

Main winding rotor resistance, in Ohms.

**Main winding rotor leakage inductance,  $L_{lar}'$  (H) – Specify main winding rotor leakage inductance**

0.0050 (default) | positive scalar

Main winding rotor leakage inductance, in H.

**Main winding mutual inductance,  $L_{ms}$  (H) – Specify main winding mutual inductance**

0.1595 (default) | positive scalar

Main winding mutual inductance, in H.

**Auxiliary/main windings turn ratio – Specify the auxiliary-to-main windings turn ratio**

1.1 (default) | positive scalar

Auxiliary-to-main windings turns ratio.

**Fundamental sample time (s) – Specify the block sample time**

5e-6 (default) | positive scalar

Sample time for the block, in s.

**Control sample time (s) – Specify the control sample time**

5e-5 (default) | positive scalar

Control sample time, in s.

**Outer Loop**

This table shows how the visibility of some parameters in the **Outer Loop** tab depend on the option that you choose for the **Control mode** parameter in the **General** tab.

Control mode	Available Parameters
Speed control	Speed controller proportional gain
	Speed controller integral gain
	Speed controller integral anti-windup gain
	Maximum torque (Nm)
	Minimum torque (Nm)
Torque control	Rated flux (Wb)
	Rated flux (Wb)

**Speed controller proportional gain – Specify the speed controller proportional gain**

1 (default) | positive scalar

Proportional gain for the PI speed controller.

**Speed controller integral gain – Specify the speed controller integral gain**

10 (default) | positive scalar

Integral gain for the PI speed controller.

**Speed controller integral anti-windup gain — Specify the speed controller integral anti-windup gain**

1000 (default) | positive scalar

Integral anti-windup gain.

**Maximum torque (Nm) — Specify the maximum torque**

5 (default) | positive scalar

Maximum torque used in speed controller saturation, in Nm.

**Minimum torque (Nm) — Specify the minimum torque**

-5 (default) | scalar

Minimum torque used in speed controller saturation, in Nm.

**Rated flux (Wb) — Specify the rated flux**

0.5 (default) | positive scalar

Rated flux, in Wb. The rated flux is used to compute the flux reference.

**Inner Loop**

This table shows how the visibility of some parameters in the **Inner Loop** tab depend on the option that you choose for the **Control type** parameter.

Control type	Available Parameters
Hysteresis control	Current hysteresis band (A)
PI control	A-phase current proportional gain
	A-phase current integral gain
	A-phase current anti-windup gain
	B-phase current proportional gain
	B-phase current integral gain
	B-phase current anti-windup gain

**Control type — Specify the control type**

Hysteresis control (default) | PI control

Control type for current control loop.

**Current hysteresis band (A) — Specify the current hysteresis band**

0.5 (default) | positive scalar

Current hysteresis band for the controller, in A.

**A-phase current proportional gain — Main winding current proportional gain**

1 (default) | positive scalar

Proportional gain for the A-phase (main winding) controller current.



**A-phase current integral gain – Main winding current integral gain**

500 (default)

Integral gain for the A-phase (main winding) controller current.

**A-phase current anti-windup gain – Main winding current anti-windup gain**

1000 (default)

Anti-windup gain for the A-phase (main winding) controller current.

**B-phase current proportional gain – Auxiliary winding current proportional gain**

1 (default)

Proportional gain for the B-phase (auxiliary winding) controller current.

**B-phase current integral gain – Auxiliary winding current integral gain**

500 (default)

Integral gain for the B-phase (auxiliary winding) controller current.

**B-phase current anti-windup gain – Auxiliary winding current anti-windup gain**

1000 (default)

Anti-windup gain for the B-phase (auxiliary winding) controller current.

## Version History

Introduced in R2018b

## References

- [1] Correa, M. B. R., Jacobina, C. B., Lima, A. M. N., Da Silva, E. R. C. "Field Oriented Control of a Single-Phase Induction Motor Drive." *PESC 98 Record. 29th Annual IEEE Power Electronics Specialists Conference*. Vol. 2, 1998, pp. 990 - 996.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Induction Machine Current Controller | Induction Machine Direct Torque Control | Induction Machine Direct Torque Control (Single-Phase) | Induction Machine Direct Torque Control with Space Vector Modulator | Induction Machine Field-Oriented Control | Induction Machine Flux Observer | Induction Machine Scalar Control

### Topics

“Single-Phase Asynchronous Machine Field-Oriented Control”

# Inductor

Inductor including optional tolerance, operational limits and fault behavior

**Library:** Simscape / Electrical / Passive



## Description

The Inductor block lets you model linear inductors, including the following effects:

- “Tolerances” on page 1-868
- “Operating Limits” on page 1-869
- “Faults” on page 1-869

You can turn these modeling options on and off independently of each other. When all the additional options are turned off, the component behavior is identical to the Simscape Foundation library Inductor block.

In its simplest form, the Inductor block models a linear inductor, described with the following equation:

$$V = L \frac{dI}{dt}$$

where:

- $V$  is voltage.
- $L$  is inductance.
- $I$  is current.
- $t$  is time.

To model a nonlinear inductor, use the Nonlinear Inductor block.

## Tolerances

You can apply tolerances to the nominal value you provide for the **Inductance** parameter. Datasheets typically provide a tolerance percentage for a given inductor type. The table shows how the block applies tolerances and calculates inductance based on the selected **Tolerance application** option.

Option	Inductance Value
None – use nominal value	$L$
Random tolerance	Uniform distribution: $L \cdot (1 - tol + 2 \cdot tol \cdot rand)$ Gaussian distribution: $L \cdot (1 + tol \cdot randn / nSigma)$
Apply maximum tolerance value	$L \cdot (1 + tol)$

Option	Inductance Value
Apply minimum tolerance value	$L \cdot (1 - tol)$

In the table,

- $L$  is the **Inductance** parameter value, nominal inductance.
- $tol$  is fractional tolerance, **Inductance tolerance (%)** /100.
- $nSigma$  is the value you provide for the **Number of standard deviations for quoted tolerance** parameter.
- `rand` and `randn` are standard MATLAB functions for generating uniform and normal distribution random numbers.

---

**Note** If you choose the `Random tolerance` option and you are in "Fast Restart" mode, the random tolerance value is updated on every simulation if at least one between the fractional tolerance,  $tol$ , or the **Number of standard deviations for quoted tolerance**,  $nSigma$ , is set to Run-time and is defined with a variable (even if you do not modify that variable).

---

### Operating Limits

Inductors are typically rated with a particular saturation current, and possibly with a maximum allowable power dissipation. You can specify operating limits in terms of these values, to generate warnings or errors if the inductor is driven outside its specification.

When an operating limit is exceeded, the block can either generate a warning or stop the simulation with an error. For more information, see the "Operating Limits" on page 1-872 parameters section.

### Faults

Instantaneous changes in inductor parameters are unphysical. Therefore, when the Inductor block enters the faulted state, short-circuit and open-circuit voltages transition to their faulted values over a period of time based on this formula:

$$CurrentValue = FaultedValue - (FaultedValue - UnfaultedValue) \cdot \operatorname{sech}(\Delta t / \tau)$$

where:

- $\Delta t$  is time since the onset of the fault condition.
- $\tau$  is user-defined time constant associated with the fault transition.

For short-circuit faults, the conductance of the short-circuit path also changes according to the  $\operatorname{sech}(\Delta t / \tau)$  function from a small value (representing an open-circuit path) to a large value.

The block can trigger the start of fault transition:

- At a specific time
- After voltage exceeds the maximum permissible value a certain number of times
- When current exceeds the maximum permissible value for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first

mechanism to trigger the fault transition takes precedence. In other words, a component fails no more than once per simulation.

You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

Faultable inductors often require that you use the fixed-step local solver rather than the variable-step solver. In particular, if you model transitions to a faulted state that include short circuits, MathWorks recommends that you use the fixed-step local solver. For more information, see “Making Optimal Solver Choices for Physical Simulation”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The **Inductor current** variable lets you specify a high-priority target for the initial inductor current at the start of simulation.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the inductor positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the inductor negative terminal.

## Parameters

### Main

#### Inductance – Nominal inductance value

1e-6 H (default)

The nominal inductance value. Inductance value must be greater than zero.

#### Tolerance (%) – Inductor tolerance, in percent

20 (default)

The inductor tolerance as defined on the manufacturer datasheet.

**Tolerance application — Select how to apply tolerance during simulation**

None — use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation:

- **None — use nominal value** — The block does not apply tolerance, it uses the nominal inductance value.
- **Random tolerance** — The block applies random offset to the inductance value, within the tolerance value limit. You can choose Uniform or Gaussian distribution for calculating the random number by using the **Tolerance distribution** parameter.
- **Apply maximum tolerance value** — The inductance is increased by the specified tolerance percent value.
- **Apply minimum tolerance value** — The inductance is decreased by the specified tolerance percent value.

**Tolerance distribution — Select the distribution type**

Uniform (default) | Gaussian

Select the distribution type for random tolerance:

- **Uniform** — Uniform distribution
- **Gaussian** — Gaussian distribution

**Dependencies**

Enabled when the **Tolerance application** parameter is set to Random tolerance.

**Number of standard deviations for quoted tolerance — Used for calculating the Gaussian random number**

4 (default)

Number of standard deviations for calculating the Gaussian random number.

**Dependencies**

Enabled when the **Tolerance distribution** parameter is set to Gaussian.

**Series resistance — Equivalent series resistance of the inductor**

0 Ohm (default)

Equivalent series resistance (ESR) of the inductor, as sometimes specified on manufacturer datasheets. The default value is consistent with the Simscape Foundation library Inductor block. If you model faults, specify a positive value for this parameter.

**Parallel conductance — Parallel leakage path associated with the inductor**

1e-9 1/Ohm (default)

Parallel leakage path associated with the inductor. Simulation of some circuits may require the presence of a small parallel conductance. You can also use this parameter to model the inductor core losses.

## Operating Limits

### Enable operating limits — Select Yes to enable reporting when the operational limits are exceeded

No (default) | Yes

Select Yes to enable reporting when the operational limits are exceeded. The associated parameters in the **Operating Limits** section become visible to let you select the reporting method and specify the operating limits in terms of power and current.

### Reporting when operating limits exceeded — Select the reporting method

Warn (default) | Error

Select what happens when an operating limit is exceeded:

- Warn — The block issues a warning.
- Error — Simulation stops with an error.

## Dependencies

Enabled when the **Enable operating limits** parameter is set to Yes.

### Saturation current — Inductor saturation current

1 A (default)

Inductor saturation current, as defined in the manufacturer datasheets. If the current exceeds this value, the core material enters saturation.

## Dependencies

Enabled when the **Enable operating limits** parameter is set to Yes.

### Power rating — Maximum power dissipation in the inductor

1 W (default)

Maximum instantaneous power dissipation in the resistance and conductance elements associated with the inductor.

## Dependencies

Enabled when the **Enable operating limits** parameter is set to Yes.

## Faults

### Enable faults — Select Yes to enable faults modeling

No (default) | Yes

Select Yes to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

### Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Location of fault node (% of total turns from - terminal) — Percentage of turns in the subinductor that is in contact with the - port of the block

50 (default)

In practice, faults are enabled by segmenting the inductor into two coupled subinductors, connected in a series. The inductance is proportional to the square of the number of turns in the respective segment, and the series resistance of each subinductor is proportional to the number of turns in each segment. The parallel conductance spans both segments.

This parameter indicates the percentage of turns that are assigned to the subinductor that is in contact with the - port of the block. The remaining turns are assigned to the other subinductor. The default value is 50, which means that the overall inductance is divided into two equal, coupled subinductors.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Faulted coupling factor — Mutual coupling between the two subinductors

0.9999 (default)

The faulted value for the mutual coupling between the two subinductors. The differential equations governing such a construction break down in the limit of perfect coupling, so the coupling should be less than unity. A value of 0 corresponds to no coupling at all between the subinductors. Physically, this corresponds to a fault that affects the flux within the inductor core. This could be a crack in the core material, or windings coming away from the core.

The default value of this parameter is also the internal value the block uses when computing a faultable inductor in the unfaulted state. For an unfaultable inductor, there is only a single equation being solved, and this corresponds to the ideal case of perfect mutual coupling.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Short-circuit turns — Select whether fault results in one of the segments being short-circuited

No (default) | To negative terminal | To positive terminal

Select whether the fault results in one of the subinductor segments being short-circuited:

- No — The fault does not produce a short circuit.
- To negative terminal — The fault short-circuits the subinductor that is in contact with the - port of the block.
- To positive terminal — The fault short-circuits the subinductor that is in contact with the + port of the block.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Open-circuit at fault node — Select whether to apply an open-circuit fault between the segments**

No (default) | Yes

Select whether to apply an open-circuit fault between the two subinductor segments. The default is No. Even with an open-circuit fault, the characteristics of the subinductors may still be related, depending on the value of the **Faulted coupling factor** parameter:

- If the coupling factor is not zero, the subinductors are galvanically isolated from each other, but they are still magnetically coupled. Physically, this corresponds to a break in the winding.
- With zero coupling factor, the subinductors are galvanically and magnetically isolated.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Ground fault — Select whether fault results in one of the segments being short-circuited**

No (default) | Negative terminal side of fault node | Positive terminal side of fault node

Select whether, in case of fault, there is a path for current to flow towards the ground node:

- No — The fault does not result in a connection to ground.
- Negative terminal side of fault node — The side that is in contact with the - port of the block is connected to ground.
- Positive terminal side of fault node — The side that is in contact with the + port of the block is connected to ground.

If the **Open-circuit at fault node** parameter is set to Yes, you need to specify which side (negative or positive) is connected to ground. If there is no open circuit, the two options behave similarly. Physically, this corresponds to a breakdown in the insulation between the windings and the grounded core or chassis.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Conductance of faulted ground path — Mutual coupling between the two subinductors**

1 1/0hm (default)

If there is a ground fault, this parameter represents the conductance of the current path to ground. For example, if the path to ground is through the core material, then specify a small conductance value depending on the core material being used. For highly conductive core material or for chassis-shorts, specify a higher conductance value.

**Dependencies**

Enabled when the **Ground fault** parameter is set to Negative terminal side of fault node or Positive terminal side of fault node.

**Fault transition time constant — Time constant for the transition to faulted state**

1e-3 s (default)



Time constant associated with the transition to the faulted state, as described in “Faults” on page 1-872.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Enable temporal fault trigger – Select Yes to enable time-based fault triggering

No (default) | Yes

Select Yes to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Simulation time for a fault event – Time before entering faulted state

1 s (default)

Set the simulation time at which you want the block to start entering the fault state.

#### Dependencies

Enabled when the **Enable temporal fault trigger** parameter is set to Yes.

#### Enable behavioral fault trigger – Select Yes to enable behavioral fault triggering

No (default) | Yes

Select Yes to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

#### Dependencies

Enabled when the **Enable faults** parameter is set to Yes.

#### Maximum permissible voltage – Voltage threshold to fault transition

100 V (default)

Define the voltage threshold to a fault transition. If the voltage value exceeds this threshold a certain number of times, specified by the **Number of events to fail when exceeding voltage** parameter value, then the block starts entering the fault state.

#### Dependencies

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

#### Number of events to fail when exceeding voltage – Maximum number of times the voltage exceeds the threshold

1 (default)

Since the physical mechanism underlying voltage-based failures depends on one or more partial discharge events occurring, this parameter allows you to set the number of voltage overshoots that the inductor can withstand before the fault transition begins. Note that the block does not check the time spent in the overvoltage condition, only the number of transitions.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Maximum permissible current – Current threshold to fault transition**

1 A (default)

Define the current threshold to a fault transition. If the current value exceeds this threshold for longer than the **Time to fail when exceeding current** parameter value, then the block starts entering the fault state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Time to fail when exceeding current – Maximum length of time the current exceeds the threshold**

1 s (default)

Set the maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

## Version History

Introduced in R2016a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Capacitor | Fault | Mutual Inductor | Nonlinear Inductor | Resistor

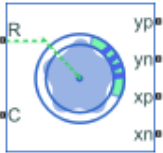
**Topics**

“Buck Converter with Faults”

# Inductive Rotor Position Sensor

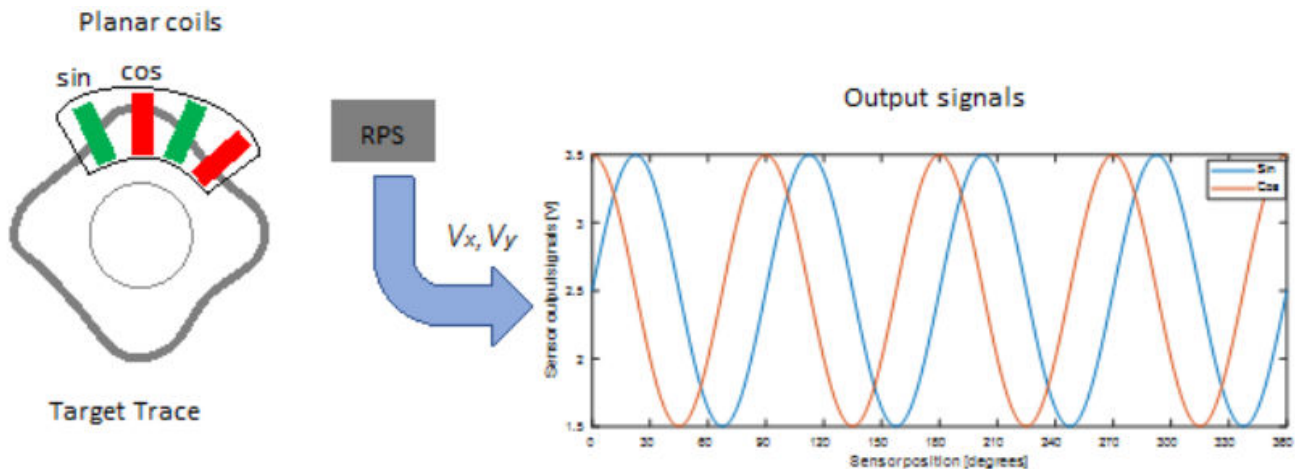
Inductive rotor position sensor with four inductive coils

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Inductive Rotor Position Sensor block uses the theory of eddy current losses to obtain the rotor position. It consists of four planar coils and a target trace that detect the rotor position. The target trace is shaped in a sinusoidal way and made from conductive material. The distance between the four planar coils is equivalent to 90 degrees of one cycle.



## Equations

The voltages between the elements of the sensor are described by the equations:

$$v_x = V_{x0} + A_x \cos(N\theta) + K \sin(\theta)$$

$$v_y = V_{y0} + A_y \cos(N\theta - \frac{\pi}{2} + \beta) + K \sin(\theta)$$

where:

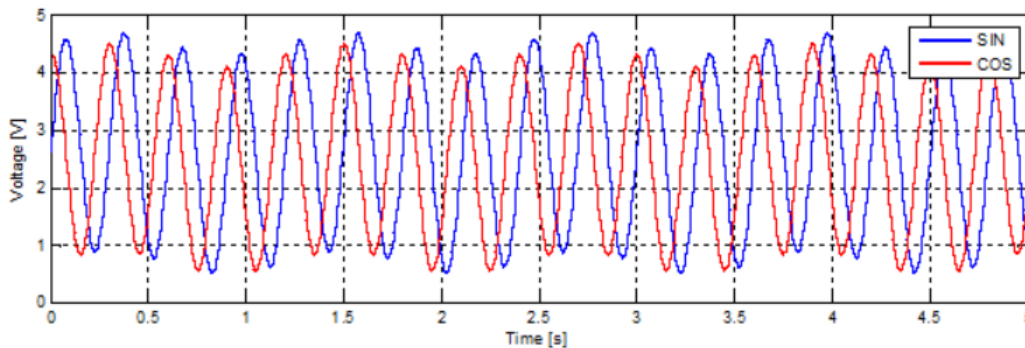
- $v_x$  is the cosine voltage.
- $v_y$  is the sine voltage.
- $A_x$  and  $A_y$  are the voltage amplitudes for the x and y axes that reflect the sensitivity mismatch.
- $V_{x0}$  and  $V_{y0}$  are the voltage offsets for the x and y axes.

- $N$  is the number of pole pairs.
- $\theta$  is the mechanical position.
- $\beta$  is the quadrature error.
- $K$  is the tumbling factor.

The block uses this equation to decode the angle:

$$\theta_{out} = \arctan\left(\frac{v_y}{v_x}\right).$$

This figure shows the effects of the tumbling factor:



## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **R — Rotational velocity for sensor positive probe**

mechanical rotational

Mechanical rotational conserving port associated with the sensor positive probe.

#### **C — Rotational velocity for the sensor negative probe**

mechanical rotational

Mechanical rotational conserving port associated with the sensor negative (reference) probe.

#### **yp — y-axis positive terminal**

electrical

Electrical conserving port associated with the y-axis positive terminal.

**Dependencies**

To enable this port, set **Output interface** to `Electrical connections`.

**yn — y-axis negative terminal**

`electrical`

Electrical conserving port associated with the y-axis negative terminal.

**Dependencies**

To enable this port, set **Output interface** to `Electrical connections`.

**xp — x-axis positive terminal**

`electrical`

Electrical conserving port associated with the x-axis positive terminal.

**Dependencies**

To enable this port, set **Output interface** to `Electrical connections`.

**xn — x-axis negative terminal**

`electrical`

Electrical conserving port associated with the x-axis negative terminal.

**Dependencies**

To enable this port, set **Output interface** to `Electrical connections`.

**Output****Angle — Rotational angle**

`physical signal`

Rotational angle of the magnetic field in the x-y plane, returned as a physical signal.

**Dependencies**

To enable this port, set **Output interface** to `Decoded angular position`.

**Parameters****Number of pole pairs — Number of pole pairs**

2 (default) | scalar

Number of pole pairs.

**X-axis voltage amplitude — x-axis voltage**

2.5 V (default) | scalar

x-axis voltage amplitude.

**Y-axis voltage amplitude — y-axis voltage**

2.5 V (default) | scalar

y-axis voltage amplitude.

**X-axis voltage offset — x-axis voltage offset**

0 V (default) | scalar

Potential difference in the electric current flow through the electric conductor in the absence of external magnetic field on the x-axis.

**Y-axis voltage offset — y-axis voltage offset**

0 V (default) | scalar

Potential difference in the electric current flow through the electric conductor in the absence of external magnetic field on the y-axis.

**Quadrature error — Quadrature error**

0 deg (default) | scalar

Amount of quadrature error. Quadrature switching can decrease offset errors.

**Tumbling coefficient — Tumbling coefficient**

0 (default) | scalar

Tumbling coefficient, in degrees.

**Output interface — Interface for decoded information**

Electrical connections (default) | Decoded angular position

Port interface for the decoded information, specified as either `Electrical connections` or `Decoded angular position`.

## Version History

Introduced in R2020a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

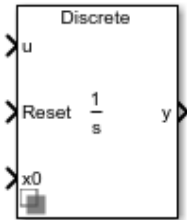
**See Also**

Hall-Effect Rotary Encoder

# Integrator (Discrete or Continuous)

Discrete-time or continuous-time integrator

**Library:** Simscape / Electrical / Control / General Control



## Description

The Integrator (Discrete or Continuous) block implements a simple integrator in conformance with IEEE 421.5-2016<sup>[1]</sup>.

You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

## Equations

### Continuous

To configure the integrator for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{1}{s}.$$

From the preceding transfer function, the integrator defining equations are:

$$\begin{cases} \dot{x}(t) = u(t) \\ y(t) = x(t) \end{cases} \quad x(0) = x_0,$$

where:

- $u$  is the integrator input.
- $x$  is the integrator state.
- $y$  is the integrator output.
- $t$  is the simulation time.
- $x_0$  is the initial state of the integrator.

### Discrete

To configure the integrator for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$G(z) = \frac{T_s}{z - 1},$$

where  $T_s$  is the sample time. From the discrete transfer function, the integrator equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = x(n) + T_s u(n) \\ y(n) = x(n) \end{cases} \quad x(0) = x_0,$$

where:

- $u$  is the integrator input.
- $x$  is the integrator state.
- $y$  is the integrator output.
- $n$  is the simulation time step.
- $x_0$  is the initial state of the integrator.

### Defining Initial Conditions

You can define the state initial conditions using the input port **x0**. The integrator state reverts to the initial condition any time it is reset.

### Limiting the Integral

You can limit the integral output using one of two methods:

- Set **Limit type** to **Anti-windup** to use the anti-windup saturation method.

The anti-windup method limits the integrator state  $x$  between the lower saturation limit  $A$  and upper saturation limit  $B$ :

$$A \leq x \leq B.$$

Because the state is limited, the output can respond immediately to a reversal of the input sign when the integral is saturated.

- Set **Limit type** to **Windup** to use the windup saturation method.

The windup method limits the integrator output  $y$  between the lower saturation limit  $A$  and upper saturation limit  $B$ :

$$A \leq y \leq B.$$

Because the output is limited, the state can continue to grow when the integrator is saturated. As a result, the output cannot respond to a reversal of the input sign until the state has reached the limiting saturation point.

### Resetting the State

You can reset the state of the integrator by passing a nonzero signal to the **Reset** port of the block.



## Ports

### Input

#### **u — Integrator input**

vector

Integrator input.

Data Types: `single` | `double`

#### **Reset — Revert to initial state**

scalar

Integrator reset. To reset the integrator state to the value of the **x0** port, pass a nonzero value to this port. Alternatively, attach a zero-valued Constant block to this port to override the external reset.

Data Types: `single` | `double`

#### **x0 — Initial state**

vector

Integrator initial state. To specify the value of the state after a reset, pass a signal to this port.

Data Types: `single` | `double`

### Output

#### **y — Integrator output**

vector

Integrator output.

Data Types: `single` | `double`

## Parameters

#### **External reset — Reset strategy**

`level` (default) | `rising` | `falling` | `either`

Select the external reset strategy for the integrator:

- Select `rising` to reset the state when the reset signal rises from a negative or zero value to a positive value.
- Select `falling` to reset the state when the reset signal falls from a positive value to a zero or negative value.
- Select `either` to reset the state when the reset signal changes from zero to a nonzero value, from a nonzero value to zero, or changes sign.
- Select `level` to reset the state when the reset signal is nonzero at the current time step or changes from nonzero at the previous time step to zero at the current time step.

#### **Limit type — Saturation strategy**

`Anti-windup` (default) | `Windup`

Select the limit type of the integrator:

- Select Anti-windup to limit the state of the integrator, preventing windup.
- Select windup to limit the output of the integrator, allowing windup of the integrator state.

**Upper saturation limit – State upper limit**

`inf` (default) | real number

Integrator upper saturation limit. Set this to `inf` for an unsaturated upper limit, or to a finite value to saturate the integrator using the strategy set by **Limit type**.

**Lower saturation limit – State lower limit**

`-inf` (default) | real number

Integrator lower saturation limit. Set this to `-inf` for an unsaturated lower limit, or to a finite value to saturate the integrator using the strategy set by **Limit type**.

**Sample time (-1 for inherited) – Block sample time**

`-1` (default) | `0` | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify `-1`. For discrete-time operation, specify a positive integer. For continuous-time operation, specify `0`.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

**Introduced in R2017b**

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

Filtered Derivative (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Washout (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

# Integrator with Wrapped State (Discrete or Continuous)

Discrete-time or continuous-time integrator with wrapped state

**Library:** Simscape / Electrical / Control / General Control



## Description

The Integrator with Wrapped State (Discrete or Continuous) block implements a wrapped state integrator in conformance with IEEE 421.5-2016<sup>[1]</sup>.

Use this block to generate periodic signals such as angles or to represent a voltage-controlled oscillator. You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

## Equations

### Continuous

To configure the integrator for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{1}{s}.$$

From the preceding transfer function, the integrator defining equations are:

$$\begin{cases} \dot{x}(t) = u(t) \\ y(t) = x(t) \end{cases} \quad x(0) = x_0,$$

where:

- $u$  is the integrator input.
- $x$  is the integrator state.
- $y$  is the integrator output.
- $t$  is the simulation time.
- $x_0$  is the initial state of the integrator.

### Discrete

To configure the integrator for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$G(z) = \frac{T_s}{z-1},$$

where  $T_s$  is the sample time. From the discrete transfer function, the integrator equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = x(n) + T_s u(n) \\ y(n) = x(n) \end{cases} \quad x(0) = x_0,$$

where:

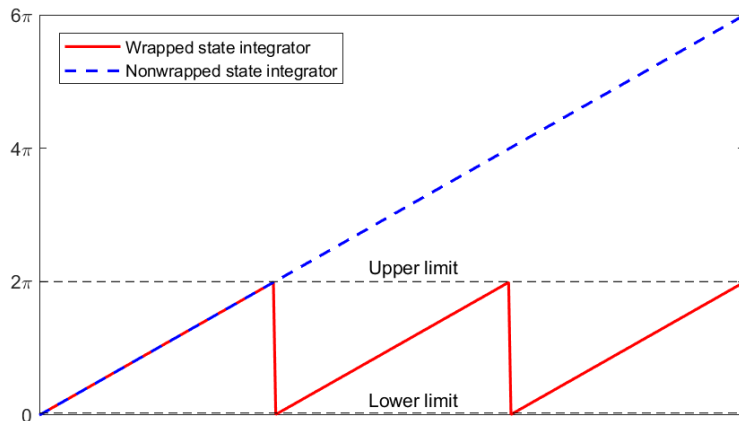
- $u$  is the integrator input.
- $x$  is the integrator state.
- $y$  is the integrator output.
- $n$  is the simulation time step.
- $x_0$  is the initial state of the integrator.

### Defining Initial Conditions

You can define the state initial conditions using **Initial condition** parameter.

### Wrapping Cyclic States

The integrator wraps its state between the specified lower and upper values. This diagram shows the outputs of a wrapped and nonwrapped state integrator for a constant input.



In the diagram, the lower and upper limits are  $0$  and  $2\pi$ , respectively.

## Ports

### Input

#### **u** — Integrator input

vector

Integrator input.

Data Types: single | double

## Output

### **y** – Integrator output

vector

Integrator output.

Data Types: `single` | `double`

## Parameters

### **Wrapped state upper limit** – State upper limit

$2\pi$  (default) | real number

Integrator upper limit.

### **Wrapped state lower limit** – State lower limit

0 (default) | real number

Integrator lower limit.

### **Initial condition** – State initial value

0 (default) | real number

Integrator initial state.

### **Sample time (-1 for inherited)** – Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Filtered Derivative (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Washout (Discrete or Continuous) | Integrator (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

# Inverse Clarke Transform

Implement  $\alpha\beta 0$  to  $abc$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

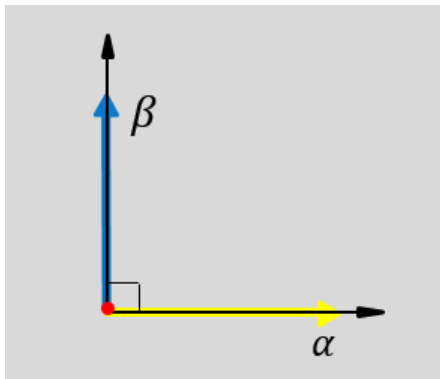


## Description

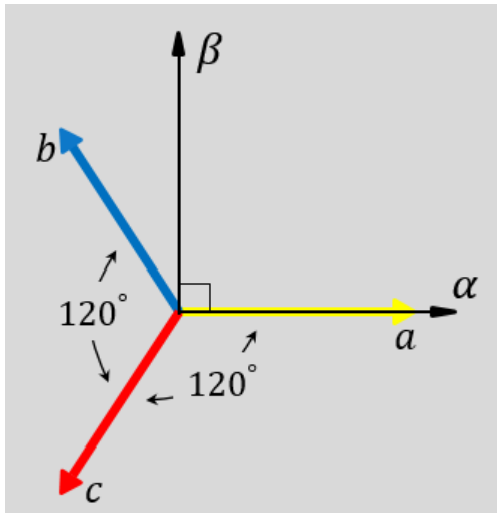
The Inverse Clarke Transform block converts the time-domain alpha, beta, and zero components in a stationary reference frame to three-phase components in an  $abc$  reference frame. The block can preserve the active and reactive powers with the powers of the system in the stationary reference frame by implementing an invariant power version of the inverse Clarke transform. If the zero component is zero, the components in the three-phase system are balanced.

The figures show:

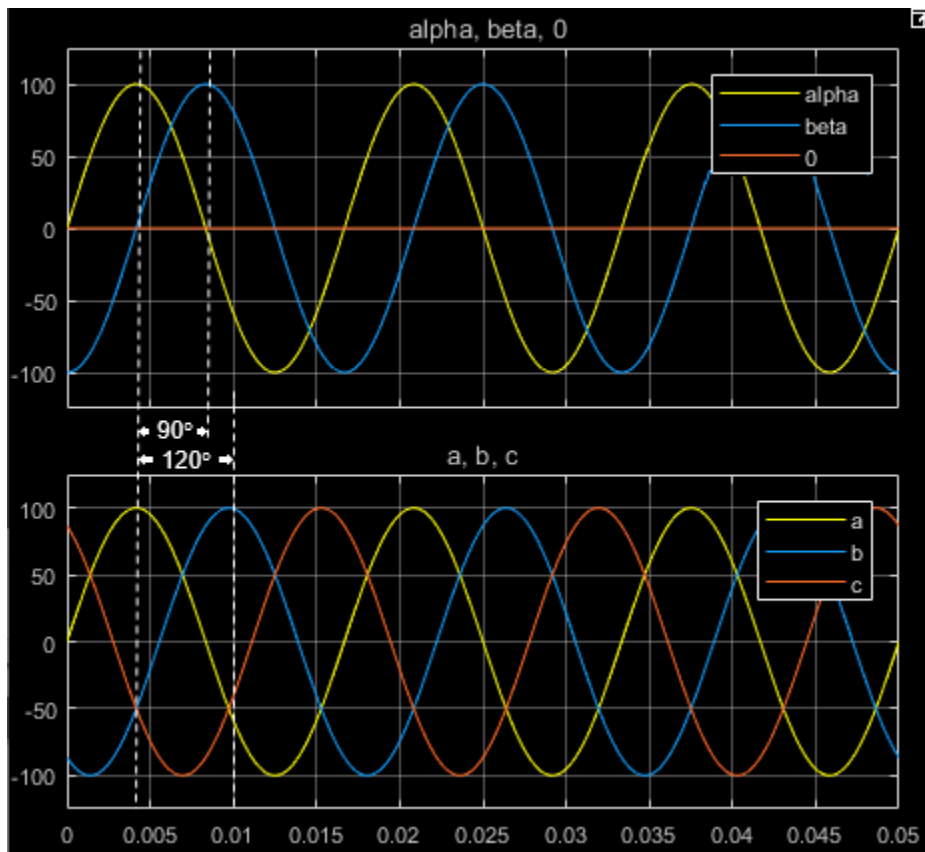
- Balanced  $\alpha$ ,  $\beta$ , and zero components in a stationary reference frame



- The direction of the magnetic axes of the stator windings in the stationary  $\alpha\beta 0$  reference frame and the  $abc$  reference frame



- The time-response of the individual components of equivalent balanced  $\alpha\beta 0$  and  $abc$  systems



### Equations

The block implements the inverse Clarke transform as



$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}$$

where:

- $\alpha$  and  $\beta$  are the components in the stationary reference frame.
- $0$  is the zero component in the stationary reference frame.
- $a$ ,  $b$ , and  $c$  are the components of the three-phase system in the  $abc$  reference frame.

The block implements this power invariant version of the inverse Clarke transform as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}$$

## Ports

### Input

#### $\alpha\beta 0$ — $\alpha$ - $\beta$ axis and zero components

vector

Alpha-axis component  $\alpha$ , beta-axis component  $\beta$ , and zero component in the stationary reference frame.

Data Types: single | double

### Output

#### $abc$ — $a$ -, $b$ -, and $c$ -phase components

vector

Components of the three-phase system in the  $abc$  reference frame.

Data Types: single | double

## Parameters

#### Power Invariant — Power invariant transform

off (default) | on

Preserve the active and reactive power of the system in the rotating reference frame.

## Version History

Introduced in R2017b

## References

[1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscatawy, NJ: Wiley-IEEE Press, 2013.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Clarke Transform | Clarke to Park Angle Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform

# Inverse Coordinate Transform (Five-Phase)

Implement  $dqxy0$  to  $abcde$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



## Description

The Inverse Coordinate Transform (Five-Phase) block converts the time-domain direct, quadrature,  $x$ ,  $y$ , and zero components in a rotating reference frame to the components of a five-phase system in an  $abcde$  reference frame.

You can configure the block to align the phase  $a$ -axis of the five-phase system to either the  $d$ -axis or  $q$ -axis of the rotating reference frame at time  $t = 0$ .

$\theta = \omega t + \text{Offset}$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment, where:

- $\omega$  is the rotational speed of the rotational reference frame.
- $t$  is the time, in s, from the initial alignment.
- $\text{Offset} = \frac{\pi}{2}$ ,  $d$  - axis alignment
- $0$ ,  $q$  - axis alignment

## Equations

The Inverse Coordinate Transform (Five-Phase) block implements the transform as

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta & \sin\theta & \cos\theta & \frac{1}{\sqrt{2}} \\ \sin\left(\theta - \frac{2\pi}{5}\right) & \cos\left(\theta - \frac{2\pi}{5}\right) & \sin\left(\theta + \frac{4\pi}{5}\right) & \cos\left(\theta + \frac{4\pi}{5}\right) & \frac{1}{\sqrt{2}} \\ \sin\left(\theta - \frac{4\pi}{5}\right) & \cos\left(\theta - \frac{4\pi}{5}\right) & \sin\left(\theta - \frac{2\pi}{5}\right) & \cos\left(\theta - \frac{2\pi}{5}\right) & \frac{1}{\sqrt{2}} \\ \sin\left(\theta + \frac{4\pi}{5}\right) & \cos\left(\theta + \frac{4\pi}{5}\right) & \sin\left(\theta + \frac{2\pi}{5}\right) & \cos\left(\theta + \frac{2\pi}{5}\right) & \frac{1}{\sqrt{2}} \\ \sin\left(\theta + \frac{2\pi}{5}\right) & \cos\left(\theta + \frac{2\pi}{5}\right) & \sin\left(\theta - \frac{4\pi}{5}\right) & \cos\left(\theta - \frac{4\pi}{5}\right) & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} d \\ q \\ x \\ y \\ 0 \end{bmatrix}$$

## Ports

### Input

**dqxy0** — Direct, quadrature,  $x$ ,  $y$ , and zero components  
vector

Direct, quadrature, x-axis, y-axis, and zero components of the system in the rotating reference frame, specified as a vector.

Data Types: `single` | `double`

### **$\theta_{elec}$ — Rotational angle**

scalar

Angular position of the rotating reference frame, specified as a scalar.

Data Types: `single` | `double`

### **Output**

#### **$abcde$ — $a$ -, $b$ -, $c$ -, $d$ -, and $e$ -phase components**

vector

Components of the five-phase system in the  $abcde$  reference frame, returned as a vector.

Data Types: `single` | `double`

### **Parameters**

#### **Phase-a axis alignment — $dqxy0$ reference frame alignment**

Q-axis (default) | D-axis

Whether to align the  $a$ -phase vector of the  $abcde$  reference frame to the  $d$ -axis or  $q$ -axis of the rotating reference frame.

## **Version History**

Introduced in R2021a

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

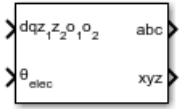
### **See Also**

Coordinate Transform (Five-Phase) | Inverse Decoupled Transform (Six-Phase) | Decoupled Transform (Six-Phase)

## Inverse Decoupled Transform (Six-Phase)

Implement  $dqz_1z_2o_1o_2$  to  $abc$  and  $xyz$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



### Description

The Inverse Decoupled Transform (Six-Phase) block converts the direct, quadrature,  $z_1$ ,  $z_2$ ,  $o_1$ , and  $o_2$  components in a rotating reference frame to the components of a six-phase system in the  $abc$  and  $xyz$  reference frame. The three-phase systems  $abc$  and  $xyz$  are spatially shifted by 30 electrical degrees.

You can configure the block to align the phase  $a$ -axis of the five-phase system to either the  $d$ -axis or  $q$ -axis of the rotating reference frame at time  $t = 0$ .

$\theta = \omega t + \text{Offset}$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment, where:

- $\omega$  is the rotational speed of the rotational reference frame.
- $t$  is the time, in s, from the initial alignment.
- $\text{Offset} = 0$ ,  $d$  - axis alignment
- $\text{Offset} = -\frac{\pi}{2}$ ,  $q$  - axis alignment

### Equations

The Inverse Decoupled Transform (Six-Phase) block implements the transform as

$$\begin{bmatrix} a \\ b \\ c \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 1 & 0 & 1 & 0 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 & 0 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 & 0 \\ \cos\left(\theta - \frac{\pi}{6}\right) & -\sin\left(\theta - \frac{\pi}{6}\right) & -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 1 \\ \cos\left(\theta - \frac{5\pi}{6}\right) & -\sin\left(\theta - \frac{5\pi}{6}\right) & \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 1 \\ \cos\left(\theta + \frac{\pi}{2}\right) & -\sin\left(\theta + \frac{\pi}{2}\right) & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ z_1 \\ z_2 \\ o_1 \\ o_2 \end{bmatrix}$$

The  $z_1$ - $z_2$  and  $o_1$ - $o_2$  pairs do not form a rotating magnetic potential. They are generalized zero-sequence components.

## Ports

### Input

#### **dqz1z2o1o2 — Direct, Quadrature, $z_1$ , $z_2$ , $o_1$ , and $o_2$ components**

vector

Direct, quadrature,  $z_1$ -axis,  $z_2$ -axis,  $o_1$ -axis, and  $o_2$ -axis components of the system in the rotating reference frame, specified as a vector.

Data Types: single | double

#### **$\theta_{elec}$ — Rotational angle**

scalar

Angular position of the rotating reference frame, specified as a scalar.

Data Types: single | double

### Output

#### **abc — a-, b-, and c-phase components**

vector

Components of the three-phase system in an *abc* reference frame, returned as a vector.

Data Types: single | double

#### **xyz — x-, y-, and z-phase components**

vector

Components of the three-phase system in an *xyz* reference frame, returned as a vector.

Data Types: single | double

## Parameters

#### **Phase-a axis alignment — $dqz_1z_2o_1o_2$ reference frame alignment**

Q-axis (default) | D-axis

Whether to align the *a*-phase vector of the *abc* reference frame to the *d*-axis or *q*-axis of the rotating reference frame.

## Version History

Introduced in R2021a

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Inverse Coordinate Transform (Five-Phase) | Coordinate Transform (Five-Phase) | Decoupled Transform (Six-Phase)

## Inverse Park Transform

Implement  $dq0$  to  $abc$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

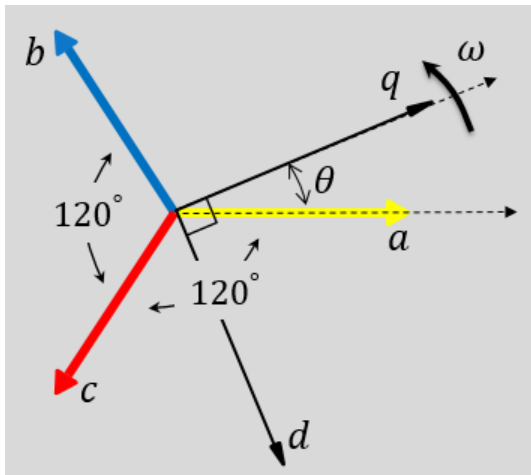


### Description

The Inverse Park Transform block converts the time-domain direct, quadrature, and zero components in a rotating reference frame to the components of a three-phase system in an  $abc$  reference frame. The block can preserve the active and reactive powers with the powers of the system in the rotating reference frame by implementing an invariant version of the Park transform. For a balanced system, the zero component is equal to zero.

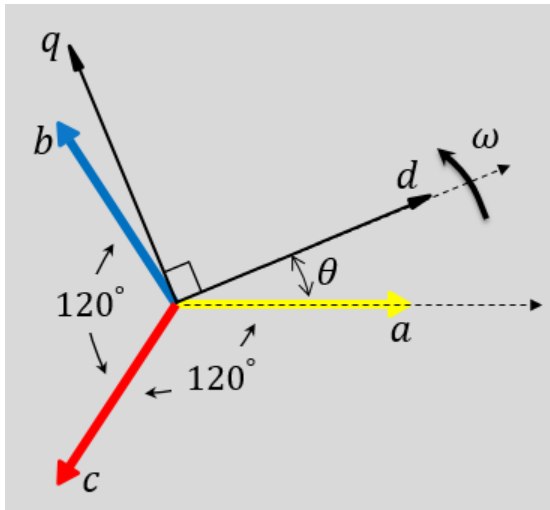
You can configure the block to align the  $a$ -axis of the three-phase system to either the  $d$ - or  $q$ -axis of the rotating reference frame at time,  $t = 0$ . The figures show the direction of the magnetic axes of the stator windings in an  $abc$  reference frame and a rotating  $d$ - $q$  reference frame where:

- The  $a$ -axis and the  $q$ -axis are initially aligned.



- The  $a$ -axis and the  $d$ -axis are initially aligned.



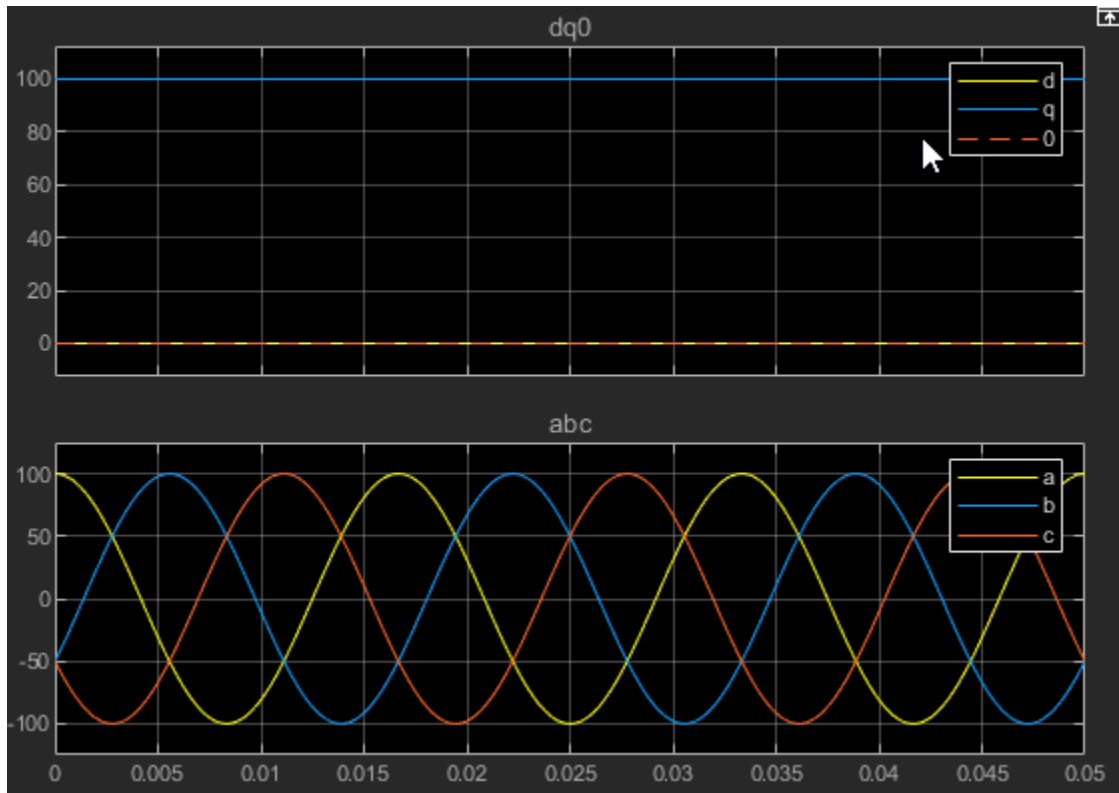


In both cases, the angle  $\theta = \omega t$ , where

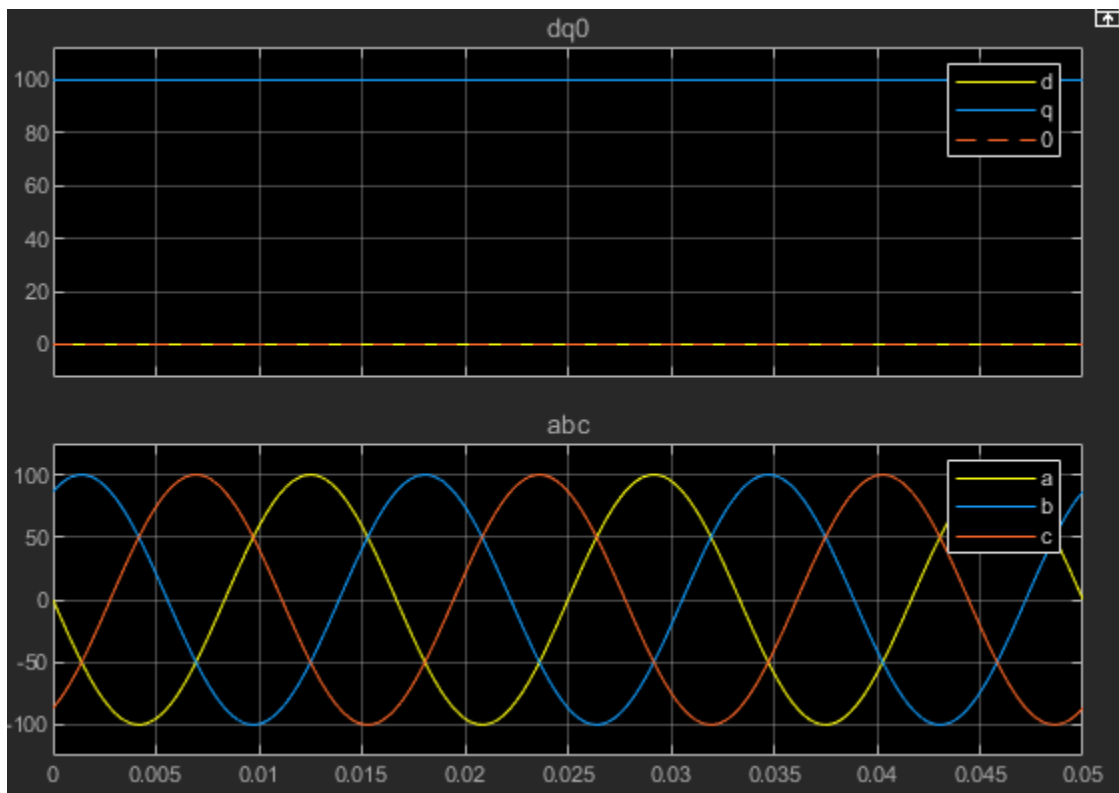
- $\theta$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment.
- $\omega$  is the rotational speed of the  $d$ - $q$  reference frame.
- $t$  is the time, in s, from the initial alignment.

The figures show the time-response of the individual components of equivalent balanced  $dq0$  and  $abc$  for an:

- Alignment of the  $a$ -phase vector to the  $q$ -axis



- Alignment of the  $a$ -phase vector to the  $d$ -axis



## Defining Equations

The Inverse Park Transform block implements the transform for an  $a$ -phase to  $q$ -axis alignment as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 1 \\ \sin(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & 1 \\ \sin(\theta + \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix},$$

where:

- $d$  and  $q$  are the components of the two-axis system in the rotating reference frame.
- $a$ ,  $b$ , and  $c$  are the components of the three-phase system in the  $abc$  reference frame.
- $0$  is the zero component of the two-axis system in the stationary reference frame.

For a power invariant  $a$ -phase to  $q$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin(\theta) & \cos(\theta) & \sqrt{\frac{1}{2}} \\ \sin(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \sqrt{\frac{1}{2}} \\ \sin(\theta + \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix}.$$

For an  $a$ -phase to  $d$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix}.$$

The block implements a power invariant  $a$ -phase to  $d$ -axis alignment as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \sqrt{\frac{1}{2}} \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) & \sqrt{\frac{1}{2}} \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix}.$$

## Ports

### Input

#### **dq0 — $d$ - $q$ axis and zero components**

vector

Direct-axis and quadrature-axis components and the zero component of the system in the rotating reference frame.

Data Types: `single` | `double`

**$\theta_{abc}$  — Rotational angle**

scalar | in radians

Angular position of the rotating reference frame. The value of this parameter is equal to the polar distance from the vector of the  $a$ -phase in the  $abc$  reference frame to the initially aligned axis of the  $dq0$  reference frame.

Data Types: single | double

**Output****abc — a-, b-, and c-phase components**

vector

Components of the three-phase system in the  $abc$  reference frame.

Data Types: single | double

**Parameters****Power Invariant — Power invariant transform**

off (default) | on

Option to preserve the active and reactive power of the  $abc$  reference frame.

**Phase-a axis alignment —  $dq0$  reference frame alignment**

Q-axis (default) | D-axis

Align the  $a$ -phase vector of the  $abc$  reference frame to the  $d$ - or  $q$ -axis of the rotating reference frame.

**Version History**

Introduced in R2017b

**References**

[1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscatawy, NJ: Wiley-IEEE Press, 2013.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

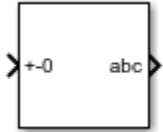
**See Also****Blocks**

Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Park Transform | Park to Clarke Angle Transform

# Inverse Symmetrical-Components Transform

Implement  $+0$  to  $abc$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



## Description

The Inverse Symmetrical-Components Transform block implements an inverse symmetrical transform of a positive, negative, and zero phasor. The transform splits a symmetrical set of three phasors into the equivalent unbalanced set of  $a$ ,  $b$ , and  $c$  phasors.

Use this transform to regenerate a three-phase signal from a system that was decoupled using the Symmetrical-Components Transform block.

Use the **Power invariant** property to choose between the Fortescue transform, and the alternative, power-invariant version.

## Equations

The inverse symmetrical-components transform regenerates an unbalanced three-phase signal  $[V_a, V_b, V_c]$  from the  $a$  components of a balanced set of phasors  $[V_{a+}, V_{a-}, V_{a0}]$ , given in the  $+0$  domain:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{K} \begin{bmatrix} 1 & 1 & 1 \\ a^2 & a & 1 \\ a & a^2 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}$$

where,  $a$  is the complex rotation operator

$$a = e^{2\pi i/3},$$

and  $K$  is the constant that determines the type of transform:

$$\begin{cases} K = 1 & \text{Fortescue transform} \\ K = \sqrt{3} & \text{Power-invariant transform} \end{cases}$$

If the transform was performed using the power-invariant option, enable the **Power invariant** property to select the power-invariant inverse transform and regenerate the correct  $abc$  signal.

## Symmetrical-Components Transform

The symmetrical-components transform separates an unbalanced three-phase signal given in phasor quantities into three balanced sets of phasors:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} v_{a+} \\ v_{b+} \\ v_{c+} \end{bmatrix} + \begin{bmatrix} v_{a-} \\ v_{b-} \\ v_{c-} \end{bmatrix} + \begin{bmatrix} v_{a0} \\ v_{b0} \\ v_{c0} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  make up the original, unbalanced set of phasors.
- $v_{a+}$ ,  $v_{b+}$ , and  $v_{c+}$  make up the balanced, positive set of phasors.
- $v_{a-}$ ,  $v_{b-}$ , and  $v_{c-}$  make up the balanced, negative set of phasors.
- $v_{a0}$ ,  $v_{b0}$ , and  $v_{c0}$  make up the balanced, zero set of phasors.

The symmetrical-components transform calculates the symmetric  $a$ -phase as:

$$\begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix} = \frac{K}{3} \begin{bmatrix} 1 & a & a^2 \\ 1 & a^2 & a \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}.$$

Because the remaining two sets of symmetrical phasors are not often used in calculation, the transformation only generates the first set. However, you can calculate the  $b$ - and  $c$ -sets in terms of simple rotations of the first:

$$\begin{bmatrix} V_{b+} \\ V_{b-} \\ V_{b0} \end{bmatrix} = \begin{bmatrix} a^2 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix},$$

and

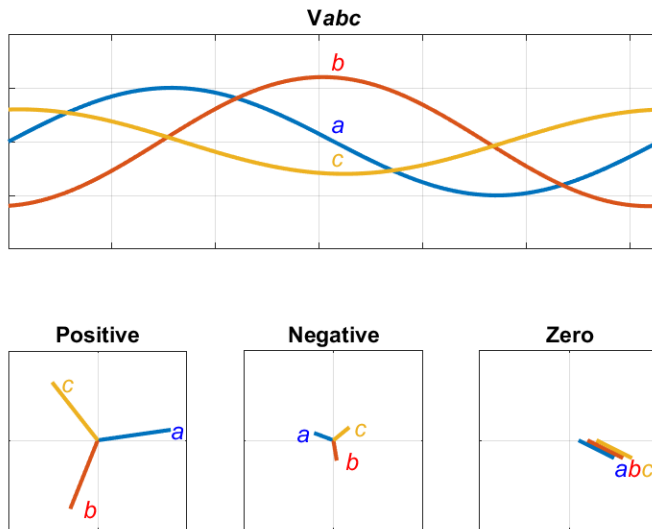
$$\begin{bmatrix} V_{c+} \\ V_{c-} \\ V_{c0} \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}.$$

### Operating Principle

The three sets of balanced phasors generated by the symmetrical-components transform have the following properties:

- The positive set has the same order as the unbalanced set of phasors  $a$ - $b$ - $c$ .
- The negative set has the opposite order as the unbalanced set of phasors  $a$ - $c$ - $b$ .
- The zero set has no order because all three phasor angles are equal.

This diagram visualizes the separation performed by the transform.



In the diagram, the top axis shows an unbalanced three-phase signal with components  $a$ ,  $b$ , and  $c$ . The bottom set of axes separates the three-phase signal into symmetrical positive, negative, and zero phasors.

Observe that in each case, the  $a$ ,  $b$ , and  $c$  components are symmetrical and are separated by:

- +120 degrees for the positive set.
- -120 degrees for the negative set.
- 0 degrees for the zero set.

## Ports

### Input

#### + - 0 — Balanced $a$ phasor components

vector

Positive, negative, and zero  $a$  phasors given as a complex signal. Use the rotations given in the Symmetrical-Components Transform section to compute the  $b$  and  $c$  phasor sets.

Data Types: single | double

### Output

#### abc — $a$ , $b$ , and $c$ phasors

vector

Regenerated three-phase set of unbalanced phasors, output as a complex signal.

Data Types: single | double

## Parameters

### Power invariant – Transform type

off (default) | on

Power invariant toggle. Select this parameter to use the power-invariant alternative of the original Fortescue transform.

## Version History

Introduced in R2017b

## References

[1] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

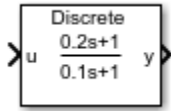
Symmetrical-Components Transform | Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform



## Lead-Lag (Discrete or Continuous)

Discrete-time or continuous-time lead-lag compensator

**Library:** Simscape / Electrical / Control / General Control



### Description

The Lead-Lag (Discrete or Continuous) block implements a lead-lag compensator in conformance with IEEE 421.5-2016<sup>[1]</sup>.

You can switch between continuous and discrete implementations of the block using the **Sample time** parameter.

### Equations

#### Continuous

To configure the compensator for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{T_1s + 1}{T_2s + 1},$$

where:

- $T_1$  is the lead time constant.
- $T_2$  is the lag time constant.

From the preceding transfer function, the compensator defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T_2}(u(t) - x(t)) \\ y(t) = \frac{T_1}{T_2}u(t) + \left(1 - \frac{T_1}{T_2}\right)x(t) \end{cases} \quad y(0) = x(0) = u_0,$$

where:

- $u$  is the block input.
- $x$  is the block state.
- $y$  is the block output.
- $t$  is the simulation time.
- $u_0$  is the initial input to the block.

### Discrete

To configure the compensator for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$\frac{T_1 z + (T_s - T_1)}{T_2 z + (T_s - T_2)},$$

where:

- $T_1$  is the lead time constant.
- $T_2$  is the lag time constant.
- $T_s$  is the compensator sample time.

From the discrete transfer function, the compensator equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = \left(1 - \frac{T_s}{T_2}\right)x(n) + \left(\frac{T_s}{T_2}\right)u(n) \\ y(n) = \left(1 - \frac{T_1}{T_2}\right)x(n) + \left(\frac{T_1}{T_2}\right)u(n) \end{cases} \quad y(0) = x(0) = u_0,$$

where:

- $u$  is the block input.
- $x$  is the state.
- $y$  is the block output.
- $n$  is the simulation time step.
- $u_0$  is the initial input to the block.

### Initial Conditions

To specify the initial conditions of this block, set **Initialization** to:

- **Inherited from block input** — The block sets the state and output initial conditions to the initial input.
- **Specify as parameter** — The block sets the state initial condition to the value of **Initial state**.

### Limiting the Integral

Set the **Upper saturation limit** and **Lower saturation limit** parameters to use the anti-windup saturation method.

The anti-windup method limits the compensator state between the lower saturation limit  $A$  and upper saturation limit  $B$ :

$$A < = x < = B .$$

Because the state is limited, the output can respond immediately to a reversal of the input sign when the integral is saturated.

This block does not provide a windup saturation method. To use the windup saturation method, set the **Upper saturation limit** parameter to `inf`, the **Lower saturation limit** parameter to `-inf`, and attach a Saturation block to the output.

### Bypass Compensator Dynamics

Set the lag time constant to zero or to a value equal to that of the lead time constant to ignore the dynamics of the compensator. When bypassed, the block feeds the input directly to the output:

$$\left. \begin{array}{l} T_1 = 0 \\ T_2 = 0 \\ T_1 = T_2 \end{array} \right\} y = u .$$

In the continuous case, both the sample time and at least one time constant must be zero.

## Ports

### Input

#### **u — Compensator input**

vector

Lead-lag compensator input signal. The block uses the input initial value to determine the state initial value.

Data Types: `single` | `double`

### Output

#### **y — Compensator output**

vector

Lead-lag compensator output.

Data Types: `single` | `double`

## Parameters

#### **Lead time constant, T1 — Lead time constant**

0.2 (default) | positive number

Compensator lead time constant. To bypass the dynamics of the compensator, set this value to 0 or to the value of the **Lag time constant, T2** parameter.

#### **Lag time constant, T2 — Lag time constant**

0.1 (default) | positive number

Compensator lag time constant. To bypass the dynamics of the compensator, set this value to 0 or to the value of the **Lead time constant, T1** parameter.

#### **Upper saturation limit — State upper limit**

`inf` (default) | real number

Compensator upper state limit. Set this to `inf` for an unsaturated upper limit, or to a finite value to prevent upper windup of the system's integrator.

**Lower saturation limit — State lower limit**

`-inf` (default) | real number

Compensator lower state limit. Set this to `-inf` for an unsaturated lower limit, or to a finite value to prevent lower windup of the system's integrator.

**Initialization — Initial state specification**

Inherited from block input (default) | Specify as parameter

Specify the initial state condition for this block. For more information, see “Initial Conditions” on page 1-908.

**Initial state — Initial state**

`0` (default) | real number

Block initial state.

**Dependencies**

To enable this parameter, set **Initialization** to `Specify as parameter`.

**Sample time (-1 for inherited) — Block sample time**

`-1` (default) | `0` | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify `-1`. For discrete-time operation, specify a positive integer. For continuous-time operation, specify `0`.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Filtered Derivative (Discrete or Continuous) | Washout (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

# Light-Emitting Diode

Exponential light-emitting diode with optical power output port

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Light-Emitting Diode block represents a light-emitting diode as an exponential diode in series with a current sensor. The optical power presented at the signal port W is equal to the product of the current flowing through the diode and the **Optical power per unit current** parameter value.

The exponential diode model provides the following relationship between the diode current  $I$  and the diode voltage  $V$ :

$$I = IS \cdot \left( e^{\frac{qV}{NkT_{m1}}} - 1 \right)$$

where:

- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $N$  is the emission coefficient.
- $IS$  is the saturation current.
- $T_{m1}$  is the temperature at which the diode parameters are specified, as defined by the **Measurement temperature** parameter value.

When  $(qV / NkT_{m1}) > 80$ , the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} - 79)e^{80}$ , which matches the gradient of the diode current at  $(qV / NkT_{m1}) = 80$  and extrapolates linearly. When  $(qV / NkT_{m1}) < -79$ ,

the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} + 80)e^{-79}$ , which also matches the gradient and extrapolates linearly. Typical electrical circuits do not reach these extreme values. The block provides this linear extrapolation to help convergence when solving for the constraints during simulation.

When you select **Use parameters IS and N** for the **Parameterization** parameter, you specify the diode in terms of the **Saturation current IS** and **Emission coefficient N** parameters. When you select **Use I-V curve data points** for the **Parameterization** parameter, you specify two voltage and current measurement points on the diode I-V curve and the block derives the  $IS$  and  $N$  values. When you specify current and voltage measurements, the block calculates  $IS$  and  $N$  as follows:

- $N = ((V_1 - V_2)/V_t) / (\log(I_1) - \log(I_2))$
- $IS = (I_1 / (\exp(V_1 / (NV_t)) - 1) + I_2 / (\exp(V_2 / (NV_t)) - 1)) / 2$

where:

- $V_t = kT_{m1} / q$ .

- $V_1$  and  $V_2$  are the values in the **Voltages [V1 V2]** vector.
- $I_1$  and  $I_2$  are the values in the **Currents [I1 I2]** vector.

The exponential diode model provides the option to include a junction capacitance:

- When you select **Fixed** or **zero junction capacitance** for the **Parameterization** parameter, the capacitance is fixed.
- When you select **Use parameters CJO, VJ, M & FC** for the **Parameterization** parameter, the block uses the coefficients  $CJO$ ,  $VJ$ ,  $M$ , and  $FC$  to calculate a junction capacitance that depends on the junction voltage.
- When you select **Use C-V curve data points** for the **Parameterization** parameter, the block uses three capacitance values on the C-V capacitance curve to estimate  $CJO$ ,  $VJ$ , and  $M$  and uses these values with the specified value of  $FC$  to calculate a junction capacitance that depends on the junction voltage. The block calculates  $CJO$ ,  $VJ$ , and  $M$  as follows:
  - $CJO = C_1((V_{R2} - V_{R1})/(V_{R2} - V_{R1}(C_2/C_1)^{-1/M}))^M$
  - $VJ = -(-V_{R2}(C_1/C_2)^{-1/M} + V_{R1})/(1 - (C_1/C_2)^{-1/M})$
  - $M = \log(C_3/C_2)/\log(V_{R2}/V_{R3})$

where:

- $V_{R1}$ ,  $V_{R2}$ , and  $V_{R3}$  are the values in the **Reverse bias voltages [VR1 VR2 VR3]** vector.
- $C_1$ ,  $C_2$ , and  $C_3$  are the values in the **Corresponding capacitances [C1 C2 C3]** vector.

It is not possible to estimate  $FC$  reliably from tabulated data, so you must specify its value using the **Capacitance coefficient FC** parameter. In the absence of suitable data for this parameter, use a typical value of 0.5.

The reverse bias voltages (defined as positive values) should satisfy  $V_{R3} > V_{R2} > V_{R1}$ . This means that the capacitances should satisfy  $C_1 > C_2 > C_3$  as reverse bias widens the depletion region and hence reduces capacitance. Violating these inequalities results in an error. Voltages  $V_{R2}$  and  $V_{R3}$  should be well away from the Junction potential  $VJ$ . Voltage  $V_{R1}$  should be less than the Junction potential  $VJ$ , with a typical value for  $V_{R1}$  being 0.1 V.

The voltage-dependent junction is defined in terms of the capacitor charge storage  $Q_j$  as:

- For  $V < FC \cdot VJ$ :

$$Q_j = CJO \cdot (VJ/(M - 1)) \cdot ((1 - V/VJ)^{1-M} - 1)$$

- For  $V \geq FC \cdot VJ$ :

$$Q_j = CJO \cdot F_1 + (CJO/F_2) \cdot (F_3 \cdot (V - FC \cdot VJ) + 0.5(M/VJ) \cdot (V^2 - (FC \cdot VJ)^2))$$

where:

- $F_1 = (VJ/(1 - M)) \cdot (1 - (1 - FC)^{1-M})$
- $F_2 = (1 - FC)^{1+M}$
- $F_3 = 1 - FC \cdot (1 + M)$

These equations are the same as used in [2], except that the temperature dependence of  $VJ$  and  $FC$  is not modeled. This model does not include the diffusion capacitance term that affects performance for high frequency switching applications.

The Light-Emitting Diode block contains several options for modeling the dependence of the diode current-voltage relationship on the temperature during simulation. Temperature dependence of the junction capacitance is not modeled, this being a much smaller effect. For details, see the Diode reference page.

### **Thermal Port**

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### **Variables**

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## **Assumptions and Limitations**

- When you select **Use I-V curve data points** for the **Parameterization** parameter, choose a pair of voltages near the diode turn-on voltage. Typically this is in the range from 0.05 to 1 Volt. Using values outside of this region may lead to numerical issues and poor estimates for  $IS$  and  $N$ .
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.

## **Ports**

### **Output**

#### **W — Optical output power**

physical signal

Physical signal port associated with the optical output power.



**Conserving****+ — Positive terminal**

electrical

Electrical conserving port associated with the anode.

**- — Negative terminal**

electrical

Electrical conserving port associated with the cathode.

**H — Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

**Main****Optical power per unit current — Optical power per unit current**

0.005 W/A (default)

Amount of optical power the light-emitting diode generates per unit of current flowing through the diode.

**Parameterization — Model parameterization**

Use I-V curve data points (default) | Use parameters IS and N

Select one of the following methods for model parameterization:

- **Use I-V curve data points** — Specify measured data at two points on the diode I-V curve. This is the default method.
- **Use parameters IS and N** — Specify saturation current and emission coefficient.

**Currents [I1 I2] — Vector of current values at two points**

[ 0.0017 0.003 ] A (default)

A vector of the current values at the two points on the diode I-V curve that the block uses to calculate IS and N.

**Dependencies**

This parameter is visible only when you select Use I-V curve data points for the **Parameterization** parameter.

**Voltages [V1 V2] — Vector of voltage values at two points**

[ 0.9 1.05 ] V (default)

A vector of the voltage values at the two points on the diode I-V curve that the block uses to calculate  $I_S$  and  $N$ .

**Dependencies**

This parameter is visible only when you select Use I-V curve data points for the **Parameterization** parameter.

**Saturation current,  $I_S$  — Saturation current**

5e-5 A (default)

The magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels.

**Dependencies**

This parameter is visible only when you select Use parameters  $I_S$  and  $N$  for the **Parameterization** parameter.

**Emission coefficient,  $N$  — Emission coefficient**

10 (default)

The diode emission coefficient or ideality factor.

**Dependencies**

This parameter is visible only when you select Use parameters  $I_S$  and  $N$  for the **Parameterization** parameter.

**Ohmic resistance,  $R_S$  — Series diode connection resistance**

0.1 Ohm (default)

The series diode connection resistance.

**Measurement temperature — Measurement temperature**

25 degC (default)

The temperature at which  $I_S$  or the I-V curve was measured.

**Junction Capacitance****Parameterization — Junction capacitance parameterization**

Fixed or zero junction capacitance (default) | Use C-V curve data points | Use parameters  $C_{J0}$ ,  $V_J$ ,  $M$  &  $FC$

Select one of the following options for modeling the junction capacitance:

- Fixed or zero junction capacitance — Model the junction capacitance as a fixed value.
- Use C-V curve data points — Specify measured data at three points on the diode C-V curve.
- Use parameters  $C_{J0}$ ,  $V_J$ ,  $M$  &  $FC$  — Specify zero-bias junction capacitance, junction potential, grading coefficient, and forward-bias depletion capacitance coefficient.

**Junction capacitance — Junction capacitance**

20 pF (default)

Fixed value of the junction capacitance.

**Dependencies**

This parameter is visible only when you select Fixed or zero junction capacitance for the **Parameterization** parameter.

**Zero-bias junction capacitance, CJ0 — Zero-bias junction capacitance**

20 pF (default)

Value of the capacitance placed in parallel with the exponential diode term.

**Dependencies**

This parameter is visible only when you select Use parameters CJ0, VJ, M & FC for the **Parameterization** parameter.

**Reverse bias voltages [VR1 VR2 VR3] — Reverse bias voltages**

[ 0.1 10 100 ] V (default)

A vector of the reverse bias voltage values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select Use C-V curve data points for the **Parameterization** parameter.

**Corresponding capacitances [C1 C2 C3] — Corresponding capacitances**

[ 15 10 2 ] pF (default)

A vector of the capacitance values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select Use C-V curve data points for the **Parameterization** parameter.

**Junction potential, VJ — Junction potential**

1 V (default)

Junction potential.

**Dependencies**

This parameter is visible only when you select Use parameters CJ0, VJ, M & FC for the **Parameterization** parameter.

**Grading coefficient, M — Grading coefficient**

0.5 (default)

The grading coefficient.

**Dependencies**

This parameter is visible only when you select Use parameters  $CJ0$ ,  $VJ$ ,  $M$  &  $FC$  for the **Parameterization** parameter.

**Capacitance coefficient, FC — Capacitance coefficient**

0.5 (default)

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage.

**Dependencies**

This parameter is visible only when you select Use C-V curve data points or Use parameters  $CJ0$ ,  $VJ$ ,  $M$  &  $FC$  for the **Parameterization** parameter.

**Temperature Dependence****Parameterization — Temperature dependence parameterization**

None — Simulate at parameter measurement temperature (default) | Use an I-V data point at second measurement temperature  $T2$  | Specify saturation current at second measurement temperature  $T2$  | Specify the energy gap  $EG$

Select one of the following methods for temperature dependence parameterization:

- None — Simulate at parameter measurement temperature — Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Use an I-V data point at second measurement temperature  $T2$  — If you select this option, you specify a second measurement temperature  $T_{m2}$ , and the current and voltage values at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify saturation current at second measurement temperature  $T2$  — If you select this option, you specify a second measurement temperature  $T_{m2}$ , and saturation current value at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify the energy gap  $EG$  — Specify the energy gap value directly.

**Current  $I1$  at second measurement temperature — Current  $I1$  at second measurement temperature**

0.0034 A (default)

Specify the diode current  $I1$  value when the voltage is  $V1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature  $T2$  for the **Parameterization** parameter.

**Voltage  $V1$  at second measurement temperature — Voltage  $V1$  at second measurement temperature**

1.05 V (default)

Specify the diode voltage  $V1$  value when the current is  $I1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 for the **Parameterization** parameter.

**Saturation current, IS, at second measurement temperature – Saturation current, IS, at second measurement temperature**

1.8e-4 A (default)

Specify the saturation current *IS* value at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Specify the value for the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 or Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Energy gap parameterization – Energy gap parameterization**

Use nominal value for silicon (EG=1.11eV) (default) | Use nominal value for 4H-SiC silicon carbide (EG=3.23eV) | Use nominal value for 6H-SiC silicon carbide (EG=3.00eV) | Use nominal value for germanium (EG=0.67eV) | Use nominal value for gallium arsenide (EG=1.43eV) | Use nominal value for selenium (EG=1.74eV) | Use nominal value for Schottky barrier diodes (EG=0.69eV) | Specify a custom value

Select a value for the energy gap from a list of predetermined options, or specify a custom value:

- Use nominal value for silicon (EG=1.11eV) – This is the default.
- Use nominal value for 4H-SiC silicon carbide (EG=3.23eV)
- Use nominal value for 6H-SiC silicon carbide (EG=3.00eV)
- Use nominal value for germanium (EG=0.67eV)
- Use nominal value for gallium arsenide (EG=1.43eV)
- Use nominal value for selenium (EG=1.74eV)
- Use nominal value for Schottky barrier diodes (EG=0.69eV)
- Specify a custom value – If you select this option, the **Energy gap, EG** parameter appears in the dialog box, to let you specify a custom value for *EG*.

**Dependencies**

This parameter is visible only when you select Specify the energy gap EG for the **Parameterization** parameter.

**Energy gap, EG – Energy gap**

1.11 eV (default)

Specify a custom value for the energy gap, *EG*.

**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Energy gap parameterization** parameter.

**Saturation current temperature exponent parameterization – Saturation current temperature exponent parameterization**

Use nominal value for pn-junction diode (XTI=3) (default) | Use nominal value for Schottky barrier diode (XTI=2) | Specify a custom value

Select one of the following options to specify the saturation current temperature exponent value:

- Use nominal value for pn-junction diode (XTI=3) — This is the default.
- Use nominal value for Schottky barrier diode (XTI=2)
- Specify a custom value — If you select this option, the **Saturation current temperature exponent, XTI** parameter appears in the dialog box, to let you specify a custom value for *XTI*.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Specify a custom value for the saturation current temperature exponent, *XTI*.

**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Saturation current temperature exponent parameterization** parameter.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Specify the value for the temperature  $T_s$ , at which the device is to be simulated.

## Version History

Introduced in R2008a

## References

- [1] H. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition, Cambridge University Press, 1984.
- [2] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

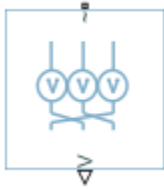
**See Also**

Diode | Optocoupler | Photodiode

## Line Voltage Sensor (Three-Phase)

Ideal three-phase line voltage measurement

**Library:** Simscape / Electrical / Sensors & Transducers



### Description

The Line Voltage Sensor (Three-Phase) block represents an ideal three-phase line voltage sensor. The block measures the line-line voltages of a three-phase system and outputs a three-element physical signal vector. Each element of the physical signal output vector is proportional to the voltage between the phases as follows:

- Element 1:  $V_{ab} = V_a - V_b$
- Element 2:  $V_{bc} = V_b - V_c$
- Element 3:  $V_{ca} = V_c - V_a$

where  $V_a$ ,  $V_b$ , and  $V_c$  are the absolute  $a$ -,  $b$ -, and  $c$ -phase voltages.

---

**Note** You must connect the Line Voltage Sensor (Three-Phase) block in shunt with the three-phase element. For more information on how to use this block, see “Build and Simulate Composite and Expanded Three-Phase Models”.

---

### Ports

#### Output

##### **V** — Three-elements vector

physical signal

Three-element physical signal vector output port associated with the voltages between the phases

#### Conserving

##### **~1** — Three-phase port

electrical

Expandable three-phase port.



## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Version History

Introduced in R2013b

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Phase Voltage Sensor (Three-Phase)

## Topics

“Build and Simulate Composite and Expanded Three-Phase Models”

“Expand and Collapse Three-Phase Ports on a Block”

## Load Flow Source

Three-phase voltage source for performing load flow analysis

**Library:** Simscape / Electrical / Sources



### Description

The Load Flow Source block is a three-phase voltage source that you use for a load-flow analysis. This block is only for three-phase Simscape networks that are compatible with frequency-time simulation mode.

The block can be used with the Busbar block to implement a swing bus, PV bus, or PQ bus. The type of bus is selected from the **Source type** drop-down menu parameter. See “Perform a Load-Flow Analysis Using Simscape Electrical” for more information about load flow in Simscape Electrical.

When you select **None** for the **Source impedance** parameter, the Load Flow Source block models an ideal three-phase voltage source that maintains sinusoidal voltage of the specified magnitude across its terminals, independently of the current flowing through the source.

The source has a wye configuration, and port **n** provides a connection to the center of the wye. Port **~** is a three-phase port that represents the phases *a*, *b*, and *c*. The current is positive if it flows from positive to the center of the wye, and the voltage across each phase is equal to the difference between the voltage at the positive terminal and the center of the wye,  $V(+)$  -  $V_n$ .

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Limitations

- If you do not initialize or simulate by using frequency-time equations, the **Source type** parameter must be set to Voltage source with series impedance.
- When initializing the block when **Source type** is set to swing bus, PV bus or PQ bus, including the internal impedance can result in initialization convergence issues. Resolve the convergence issues by using the **Internal source phase search range** parameter, omitting the impedance, or modelling the impedance external to the Load Flow Source block.

## Ports

### Conserving

#### ~ — Three-phase voltage

electrical

Three-phase port associated with the voltage phases *a*, *b*, and *c*.

#### n — Wye center

electrical

Electrical conserving port associated with the center of the wye.

## Parameters

### Main

#### Source type — Voltage source model

Swing bus (default) | Voltage source with series impedance | PV bus | PQ bus

Type of voltage source that the block models.

#### Dependencies

The visibility of **Source voltage (pu)**, **Terminal voltage (pu)**, **Active power generated**, **Active power consumed**, **Reactive power consumed**, **Source phase**, **Minimum voltage (pu)** and **Internal source phase search range** depend on the value that you choose for this parameter.

#### Source voltage (pu) — Source voltage

1 (default) | scalar

Per-unit voltage that the source generates.

#### Dependencies

This parameter is visible only when you set the **Source type** parameter to **Voltage source with series impedance**.

#### Terminal voltage (pu) — Terminal voltage

1 (default) | scalar

Per-unit terminal voltage that the source generates.

#### Dependencies

This parameter is visible only if you set the **Source type** parameter to **Swing bus** or **PV bus**.

#### Active power generated — Active power generated

0 W (default) | scalar

Active power that the source generates.

#### Dependencies

This parameter is visible only if you set the **Source type** parameter to **PV bus**.

**Active power consumed — Active power consumed**

0 W (default)

Active power that the source consumes.

**Dependencies**

This parameter is visible only if you set the **Source type** parameter to PQ bus.

**Reactive power consumed — Reactive power consumed**

0 V\*A (default)

Reactive power that the source consumes.

**Dependencies**

This parameter is visible only if you set the **Source type** parameter to PQ bus.

**Rated voltage (phase-to-phase RMS) — Rated phase-to-phase RMS voltage**

$\sqrt{3} * 100 / \sqrt{2}$  V (default) | scalar

Rated phase-to-phase root mean square (RMS) voltage.

**Rated frequency — Rated frequency**

60 Hz (default) | scalar

Rated frequency, in Hz.

**Source phase — Source phase**

0 deg (default) | scalar

Phase shift of the source.

**Dependencies**

This parameter is visible only if you set the **Source type** parameter to Voltage source with series impedance.

**Impedance****Source impedance — Source impedance method**

None (default) | X/R Ratio | Series R | Series L | Series RL

Type of source impedance. Selecting any other options enables other parameters. The options are:

- None
- X/R Ratio
- Series R
- Series L
- Series RL

**Dependencies**

The visibility of **Short-circuit power level**, **Source X/R ratio**, **Source impedance parasitic parallel conductance**, **Source resistance**, and **Source inductance** depend on the value that you choose for this parameter.

**Short-circuit power level – Short-circuit power level**

1e6 V\*A (default) | scalar

Short-circuit power level.

**Dependencies**This parameter is visible only if you set the **Source impedance** parameter to X/R ratio.**Source X/R ratio – Reactance-to-resistance ratio**

15 (default) | scalar

Reactance-to-resistance ratio.

**Dependencies**This parameter is visible only if you set the **Source impedance** parameter to X/R ratio.**Source impedance parasitic parallel conductance – Source impedance parasitic parallel conductance**

0 1/0hm (default) | scalar

Source impedance parasitic parallel conductance.

**Dependencies**This parameter is visible only if you set the **Source impedance** parameter to X/R ratio or Series L or Series RL.**Source resistance – Source resistance**

0.01 Ohm (default) | scalar

Source resistance.

**Dependencies**This parameter is visible only if you set the **Source impedance** parameter to Series R or Series RL.**Source inductance – Source inductance**

3.97e-4 H (default) | scalar

Source inductance.

**Dependencies**This parameter is visible only if you set the **Source impedance** parameter to Series L or Series RL.**Expected Ranges****Minimum voltage (pu) – Minimum voltage**

0.95 (default) | scalar

Per-unit minimum voltage. This parameter assists with the initial solve in models with non-physical, very low voltages.

**Dependencies**

This parameter is visible only if you set the **Source type** parameter in the **Main** setting to **Swing bus**, **PV bus** or **PQ bus**.

**Internal source phase search range — Phase search range**

`[-inf, inf] deg (default) | vector`

Vector of the lower and upper boundaries for the internal source phase search. Use this range to help the initial solve converge to the desired load flow solution.

**Dependencies**

This parameter is visible only if, in the **Main** setting, you set the **Source type** parameter to **PV bus** or **PQ bus**.

## Version History

**Introduced in R2019b**

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Busbar | **Load-Flow Analysis**

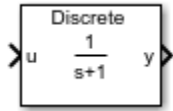
**Topics**

“Perform a Load-Flow Analysis Using Simscape Electrical”

## Low-Pass Filter (Discrete or Continuous)

Discrete-time or continuous-time low-pass filter

**Library:** Simscape / Electrical / Control / General Control



### Description

The Low-Pass Filter (Discrete or Continuous) block implements a low-pass filter in conformance with IEEE 421.5-2016<sup>[1]</sup>. In the standard, the filter is referred to as a Simple Time Constant.

You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

### Equations

#### Continuous

To configure the filter for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{K}{Ts + 1},$$

where:

- $K$  is the filter gain.
- $T$  is the filter time constant.

From the preceding transfer function, the filter defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}(Ku(t) - x(t)) \\ y(t) = x(t) \end{cases} \quad y(0) = x(0) = Ku_0,$$

where:

- $u$  is filter input.
- $x$  is filter state.
- $y$  is filter output.
- $t$  is simulation time.
- $u_0$  is the initial input to the block.

### Discrete

To configure the filter for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$G(z) = K \frac{(T_s/T)z^{-1}}{1 + (T_s/T - 1)z^{-1}},$$

where:

- $K$  is the filter gain.
- $T$  is the filter time constant.
- $T_s$  is the filter sample time.

From the discrete transfer function, the filter equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = \left(1 - \frac{T_s}{T}\right)x(n) + K\left(\frac{T_s}{T}\right)u(n) \\ y(n) = x(n) \end{cases} \quad y(0) = x(0) = Ku_0,$$

where:

- $u$  is the filter input.
- $x$  is the filter state.
- $y$  is the filter output.
- $n$  is the simulation time step.
- $u_0$  is the initial input to the block.

### Initial Conditions

To specify the initial conditions of this block, set **Initialization** to:

- **Inherited from block input** — The block sets the state and output initial conditions to the initial input.
- **Specify as parameter** — The block sets the state initial condition to the value of **Initial state**.

### Limiting the Integral

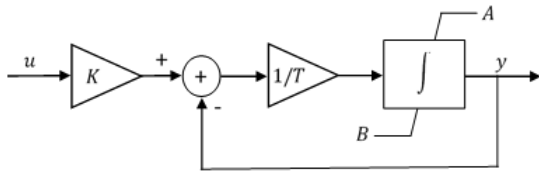
Set the **Upper saturation limit** and **Lower saturation limit** parameters to use the anti-windup saturation method.

The anti-windup method limits the integrator state between the lower saturation limit  $A$  and upper saturation limit  $B$ :

$$A < = x < = B .$$

Because the state is limited, the output can respond immediately to a reversal of the input sign when the integral is saturated. This block diagram depicts the implementation of the anti-windup saturation method in the filter.





This block does not provide a windup saturation method. To use the windup saturation method, set the **Upper saturation limit** parameter to `inf`, the **Lower saturation limit** parameter to `-inf`, and attach a saturation block to the output.

### Bypass Filter Dynamics

Set the time constant to a value smaller than or equal to the sample time to ignore the dynamics of the filter. When bypassed, the block feeds the gain-scaled input directly to the output:

$$T \leq T_s \rightarrow y = Ku$$

In the continuous case, the sample time and time constant must both be zero.

## Ports

### Input

#### **u** – Filter input

vector

Low-pass filter input signal. The block uses the input initial value to determine the state initial value.

Data Types: `single` | `double`

### Output

#### **y** – Filter output

vector

Low-pass filter output.

Data Types: `single` | `double`

## Parameters

#### **Gain** – Filter gain

1 (default) | positive number

Low-pass filter gain.

#### **Time constant** – Filter time constant

1 (default) | positive number

Low-pass filter time constant. In the discrete implementation, set this value to less than the **Sample time** to bypass the dynamics of the filter.

#### **Upper saturation limit** – State upper limit

`inf` (default) | real number

Low-pass filter upper state limit. Set this to `inf` for an unsaturated upper limit, or to a finite value to prevent upper windup of the filter's integrator.

**Lower saturation limit — State lower limit**

`-inf` (default) | real number

Low-pass filter lower state limit. Set this to `-inf` for an unsaturated lower limit, or to a finite value to prevent lower windup of the filter's integrator.

**Initialization — Initial state specification**

Inherited from block input (default) | Specify as parameter

Specify the initial state condition for this block. For more information, see “Initial Conditions” on page 1-930.

**Initial state — Initial state**

`0` (default) | real number

Block initial state.

**Dependencies**

To enable this parameter, set **Initialization** to `Specify as parameter`.

**Sample time (-1 for inherited) — Block sample time**

`-1` (default) | `0` | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify `-1`. For discrete-time operation, specify a positive integer. For continuous-time operation, specify `0`.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Filtered Derivative (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Washout (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous)

# Luenberger Observer

Discrete-time Luenberger observer

**Library:** Simscape / Electrical / Control / Observers



## Description

The Luenberger Observer block implements a discrete time Luenberger observer. Use this block to estimate the states of an observable system using:

- The discrete inputs and outputs of the system.
- A discrete state-space representation of the system.

The Luenberger Observer is also sometimes referred to as a state observer or simply an observer.

You can control multi-input, multi-output systems by passing the output state vector of this block to a State Feedback Controller block.

## Defining Equations

The block implements a discrete time Luenberger Observer using the backward Euler method due to its simplicity and stability.

The estimator is given by this difference equation:

$$\hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + L_d (y(k) - \hat{y}(k)),$$

where:

- $\hat{x}(k)$  is the  $k^{\text{th}}$  estimated state vector.
- $\hat{y}(k)$  is the  $k^{\text{th}}$  estimated output vector.
- $u(k)$  is the  $k^{\text{th}}$  input vector.
- $y(k)$  is the  $k^{\text{th}}$  measured output vector.
- $A_d$  is the discretized state matrix.
- $B_d$  is the discretized input matrix.
- $L_d$  is the discretized observer gain matrix.

The dynamics of the estimation error are described by:

$$e(k+1) = (A_d - L_d C_d) e(k),$$

where:

- $e(k)$  is the  $k^{\text{th}}$  error vector.

- $C_d$  is the output matrix.

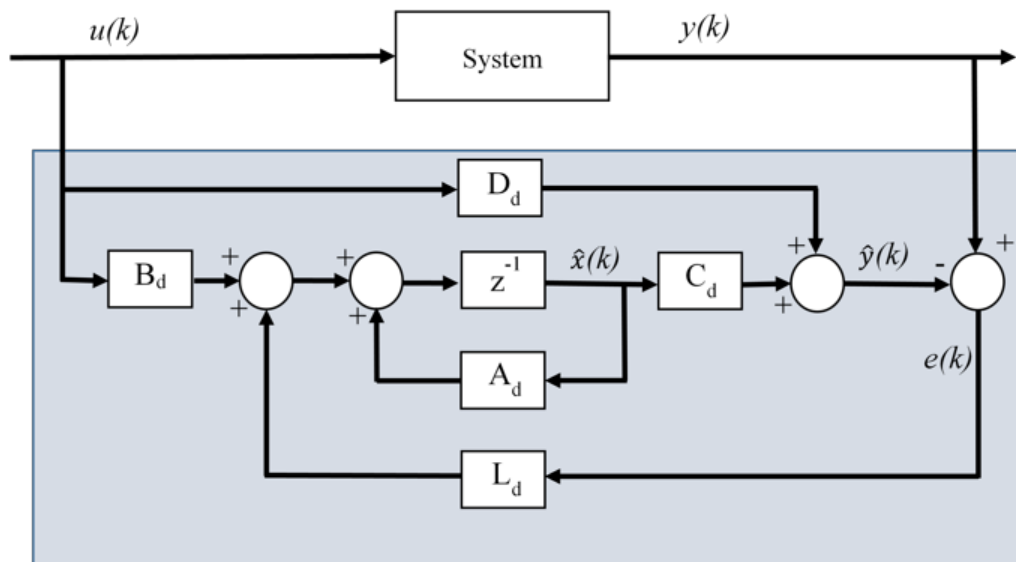
The estimation error converges to zero when  $A_d - L_d C_d$  has its eigenvalues inside the unit circle. Therefore, the value of  $L_d$  should be such that this goal is achieved. The block computes the observer gain by solving

$$L_d^T = GX^{-1},$$

where  $G$  is an arbitrary matrix and  $X$  is obtained by solving the Sylvester equation:

$$A_d^T X - X \Lambda = C_d^T G.$$

Here,  $\Lambda$  is a matrix with the desired eigenvalues, which are not the same as the eigenvalues of  $A_d$ . This diagram shows the basic structure of a discrete time Luenberger Observer.



### Assumptions

The system is observable, which is true if the state of the system can be determined from the input and output in a finite time. Mathematically, this means that the system observability matrix has full rank.

### Limitations

The desired eigenvalues are not the same as the eigenvalues of the open-loop model.

### Ports

#### Input

##### **u** – Control input

vector

Input signal to the system whose state we want to estimate, specified as a vector.

Data Types: `single` | `double`

### **y — System output**

vector

Measured output of the system whose state we want to estimate, specified as a vector.

Data Types: `single` | `double`

### **Output**

#### **xhat — State estimate**

vector

Estimate of the state of the system, specified as a vector.

Data Types: `single` | `double`

## **Parameters**

### **State-space parameterization — State-space parameterization**

`Discrete-time` (default) | `Continuous-time`

Select the strategy for parameterizing the state-space matrices and desired poles for the observer. The block implementation is discrete regardless of this parameterization.

#### **Discrete A matrix — A matrix in discrete time**

1 (default) | real scalar or matrix

State matrix of the discrete-time state-space model. The A matrix must be square, with the number of rows and columns equal to the order of the system.

#### **Dependencies**

To enable this parameter, set **State-space parameterization** to `Discrete-time`.

#### **Discrete B matrix — B matrix in discrete time**

1 (default) | real scalar or matrix

Input matrix of the discrete-time state-space model. The B matrix must have the number of rows equal to the order of the system, and the number of columns equal to the number of system inputs.

#### **Dependencies**

To enable this parameter, set **State-space parameterization** to `Discrete-time`.

#### **Discrete C matrix — C matrix in discrete time**

1 (default) | real scalar or matrix

Output matrix of the discrete-time state-space model. The C matrix must have the number of rows equal to the number of outputs of the system, and the number of columns equal to the order of the system.

#### **Dependencies**

To enable this parameter, set **State-space parameterization** to `Discrete-time`.

**Discrete D matrix — D matrix in discrete time**

1 (default) | real scalar or matrix

Feedthrough matrix of the discrete-time state-space model. The D matrix must have the number of rows equal to the number of system outputs, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Discrete-time.

**Continuous A matrix — A matrix in continuous time**

1 (default) | real scalar or matrix

State matrix of the continuous-time state-space model. The A matrix must be square, with the number of rows and columns equal to the order of the system.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Continuous-time.

**Continuous B matrix — B matrix in continuous time**

1 (default) | real scalar or matrix

Input matrix of the continuous-time state-space model. The B matrix must have the number of rows equal to the order of the system, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Continuous-time.

**Continuous C matrix — C matrix in continuous time**

1 (default) | real scalar or matrix

Output matrix of the continuous-time state-space model. The C matrix must have the number of rows equal the number of outputs of the system, and the number of columns equal to the order of the system.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Continuous-time.

**Continuous D matrix — D matrix in continuous time**

1 (default) | real scalar or matrix

Feedthrough matrix of the continuous-time state-space model. The D matrix must have the number of rows equal to the number of system outputs, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Continuous-time.

**Observer design — State-space parameterization**

Desired eigenvalues (default) | Observer gain

Select the strategy for parameterizing observer gain.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Discrete-time.

**Observer gain — Observer gain**

1 (default) | real scalar or matrix

Specify the observer gain that puts all eigenvalues of the matrix  $A_d - L_d C_d$  inside the unit circle. The gain matrix must have the number of rows equal to number of system inputs and the number of columns equal to the order of the system.

**Dependencies**

To enable this parameter, set:

- **State-space parameterization** to Discrete-time.
- **Observer design** to Observer gain.

**Desired eigenvalues — Observer eigenvalues**

0 (default) | real vector

Specify the location of the eigenvalues:

- To have negative real part if **State-space parameterization** is set to Continuous-time. In this case, the eigenvalues of the continuous-time system are approximated to the discrete ones based on the **Discretization sample time**.
- To lie within the unit circle if **State-space parameterization** is set to Discrete-time.

The Observer gain is then calculated based on these eigenvalues. The size of the vector should be the same as the system order.

**Initial conditions — Initial conditions**

0 (default) | real vector with length equal to system order

Select the initial condition of each state.

**Discretization sample time — Discretization sample time**

0.1 (default) | positive real number

Value used to discretize the state space matrices and also approximate the discrete-time eigenvalues.

**Dependencies**

To enable this parameter, set **State-space parameterization** to Continuous-time.

**Sample time — Sample time**

0.1 (default) | -1 or positive real number

Value used to simulate the dynamics of the model. Choose the same value as **Discretization sample time**, unless the block is placed within a triggered subsystem, in which case you must set it to -1.

## Version History

Introduced in R2017b



## References

- [1] Luenberger, D. G. "An Introduction to Observers." *IEEE Transactions on Automatic Control*. Vol. 16, Number 6, 1971, pp. 596-602.
- [2] Alessandri, A., and P. Coletta. "Design of Luenberger observers for a class of hybrid linear systems." *In International Workshop on Hybrid Systems: Computation and Control*, Berlin, March 2001.
- [3] Varga, A. "Robust pole assignment via Sylvester equation based state feedback parametrization." *In Computer-Aided Control System Design*, pp. 13-18., Anchorage, Alaska, 2000.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

State Feedback Controller | Induction Machine Flux Observer

# Machine Inertia

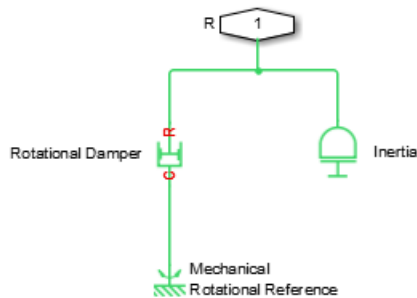
Machine inertia parameterized using machine inertia constant or anchor inertia

**Library:** Simscape / Electrical / Electromechanical / Mechanical



## Description

The Machine Inertia block models inertia and damping that you connect to the mechanical rotational R port of a three-phase machine. The block has an internal connection to a mechanical rotational reference. The figure shows an equivalent configuration to the Machine Inertia block using Simscape mechanical rotational components.



Based on the value you select for the **Specify inertia parameterization** by parameter, you specify inertia  $J$  directly or using the machine inertia constant  $H$ .

If you specify the inertia constant, the block calculates inertia by

$$J = \frac{2HS_{rated}}{(2\pi F_{rated}/N)^2},$$

where:

- $J$  is inertia in  $\text{kg}\cdot\text{m}^2$ .
- $H$  is the inertia constant in  $\text{sW}/\text{VA}$ .
- $S_{rated}$  is the machine rated apparent power in VA.
- $F_{rated}$  is the machine rated electrical frequency in Hz.
- $N$  is the number of machine pole pairs.

You specify damping that represents viscous friction between the machine rotor and mechanical rotational reference. Based on the value you select for the **Specify damper parameterization** by parameter, you specify a damping coefficient in SI units or in per-unit. If you specify the damping coefficient in per-unit, the block calculates the damping coefficient in SI units by

$$\omega_{base} = \frac{2\pi F_{rated}}{N},$$

$$T_{base} = \frac{S_{rated}}{\omega_{base}},$$

$$D_{base} = \frac{T_{base}}{\omega_{base}},$$

and

$$D = D_{pu}D_{base},$$

where:

- $\omega_{base}$  is the base mechanical angular speed in rad/s.
- $T_{base}$  is the base damping torque in Nm.
- $D_{base}$  is the base damping coefficient in Nm/(rad/s).
- $D_{pu}$  is the damping coefficient in per-unit.
- $D$  is the damping coefficient in SI units of Nm/(rad/s).

### Display Option

You can display machine parameters using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select **Display Parameters** to display the machine per-unit base values and inertia parameters in the MATLAB Command Window.

## Ports

### Conserving

#### R — Machine rotor

mechanical

Mechanical rotational conserving port associated with the machine rotor.

## Parameters

### Main

#### Rated apparent power — Rated apparent power

555e6 V\*A (default)

Machine rated apparent power.

#### Rated electrical frequency — Nominal electrical frequency

60 Hz (default)

Nominal electrical frequency corresponding to the machine rated apparent power.

#### Number of pole pairs — Number of pole pairs

1 (default)

Number of pole pairs of the connected machine.

### **Inertia**

#### **Specify inertia parameterization by — Inertia parameterization**

Inertia constant, H (default) | Actual Inertia, J

Inertia specification.

#### **Inertia constant, H — Inertia constant**

3.525 s\*W/VA (default)

Inertia constant.

#### **Dependencies**

To enable this parameter, set **Specify inertia parameterization by** to Inertia constant, H.

#### **Actual inertia, J — Actual inertia**

27548 kg\*m<sup>2</sup> (default)

Total rotational mechanical inertia of machine rotor. This parameter is visible only if you set **Specify inertia parameterization by** to Actual Inertia, J.

#### **Dependencies**

To enable this parameter, set **Specify inertia parameterization by** to Actual Inertia, J.

#### **Specify damper parameterization by — Damping parameterization**

Per-unit damping coefficient, pu\_D (default) | SI damping coefficient, D

Damping specification.

#### **Per-unit damping coefficient — Per-unit damping coefficient**

0.01 (default)

Damping coefficient in per-unit.

#### **Dependencies**

To enable this parameter, set **Specify damper parameterization by** to Per-unit damping coefficient, pu\_D.

#### **SI damping coefficient — Damping coefficient**

39.0509 Nm/(rad/s) (default)

Damping coefficient in SI units.

#### **Dependencies**

To enable this parameter, set **Specify damper parameterization by** to SI damping coefficient, D.

### **Initial Conditions**

#### **Specify initialization by — Frequency initialization**

Initial electrical frequency (default) | Initial mechanical frequency

Frequency initialization.

### **Initial electrical frequency – Initial electrical frequency**

60 Hz (default)

Initial electrical frequency, specified in SI frequency units, such as Hz, kHz, MHz, or GHz.

For more information about the unit of this parameter, see “Units for Angular Velocity and Frequency”.

#### **Dependencies**

To enable this parameter, set **Specify initialization by** to Initial electrical frequency.

### **Initial mechanical frequency – Initial mechanical frequency**

60 Hz (default)

Initial mechanical frequency, specified in SI frequency units, such as Hz, kHz, MHz, or GHz.

For more information about the unit of this parameter, see “Units for Angular Velocity and Frequency”.

#### **Dependencies**

To enable this parameter, set **Specify initialization by** to Initial mechanical frequency.

## **Version History**

**Introduced in R2013b**

## **References**

[1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

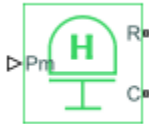
## **See Also**

Machine Mechanical Power

## Machine Mechanical Power

Machine mechanical power defined in the SI or per-unit system

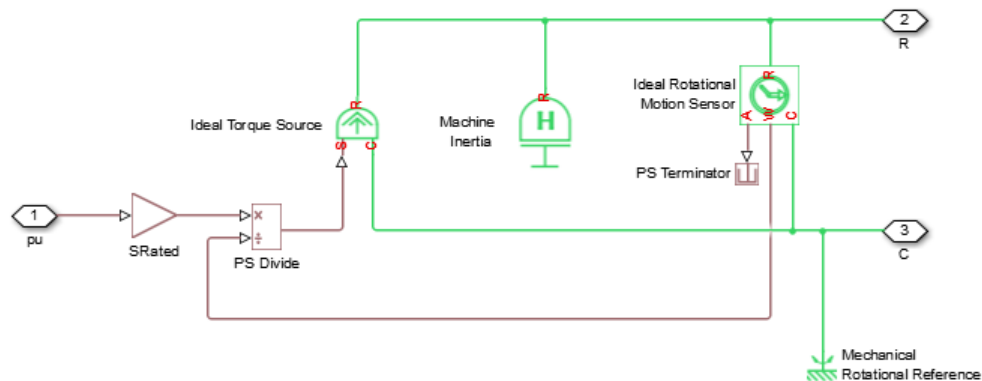
**Library:** Simscape / Electrical / Electromechanical / Mechanical



### Description

The Machine Mechanical Power block supplies specified power to, or draws specified power from, the machine that it connects to. It includes a representation of machine inertia and a mechanical rotational reference. In generator mode, the physical signal input defines the mechanical power that is input to the machine. In motor mode, it defines the mechanical power output from the machine.

The figure shows an equivalent configuration to the per-unit variant model of the Machine Mechanical Power block using Simscape mechanical rotational components.



### Electrical Defining Equations

The SI model converts the SI values that you enter in the dialog box to per-unit values for simulation. For information on the relationship between SI and per-unit machine parameters, see “Per-Unit Conversion for Machine Parameters”. For information on per-unit parameterization, see “Per-Unit System of Units”.

To calculate the torque that it applies to the inertia, the block divides the power demand by the present speed. To set the peak torque limit, specify a value for the **Peak torque to rated torque ratio** parameter. Use the **Specify inertia parameterization by** parameter to specify inertia,  $J$ , directly or indirectly, with the inertia constant for the machine,  $H$ .

If you specify the inertia constant for the machine, the block calculates inertia as

$$J = \frac{2HS_{rated}}{(2\pi F_{rated}/N)^2},$$

where:

- $J$  is inertia in kg·m<sup>2</sup>.
- $H$  is the inertia constant in sW/VA.
- $S_{rated}$  is the rated apparent power of the connected machine in VA.
- $F_{rated}$  is the rated electrical frequency of the connected machine in Hz.
- $N$  is the number of machine pole pairs.

Damping represents viscous friction between the machine rotor and mechanical rotational reference. Based on the value you select for the **Specify damper parameterization by** parameter, you specify a damping coefficient in per-unit or in SI units. If you specify the damping coefficient in per-unit, the block calculates the damping coefficient in SI units using these equations:

$$\omega_{base} = \frac{2\pi F_{rated}}{N},$$

$$T_{base} = \frac{S_{rated}}{\omega_{base}},$$

$$D_{base} = \frac{T_{base}}{\omega_{base}},$$

and

$$D = D_{pu}D_{base},$$

where:

- $\omega_{base}$  is the base mechanical speed in rad/s.
- $T_{base}$  is the base damping torque in Nm.
- $D_{base}$  is the base damping coefficient in Nm/(rad/s).
- $D_{pu}$  is the damping coefficient in per-unit.
- $D$  is the damping coefficient in SI units of Nm/(rad/s).

## Ports

### Input

#### **Pm — Mechanical power**

physical | positive

Physical signal input port associated with mechanical power, in W.

#### **Dependencies**

This port is visible only if you set the **PS input unit** parameter in the **Main** setting to SI.

#### **pu — Per-unit mechanical power**

physical | positive

Physical signal input port associated with mechanical power, in per-unit.

## Dependencies

This port is visible only if you set the **PS input unit** parameter in the **Main** setting to Per unit.

## Conserving

### R – Machine rotor

mechanical

Mechanical rotational conserving port associated with the machine rotor.

### C – Machine case

mechanical

Mechanical rotational conserving port associated with the machine case.

## Parameters

### Main

#### Input power sign convention – Type specification

Generator (default) | Motor

Machine type specification.

#### Rated apparent power – Rated apparent power

555e6 V\*A (default)

Rated apparent power of the connected machine.

#### Rated electrical frequency – Nominal electrical frequency

60 Hz (default)

Nominal electrical frequency corresponding to the rated apparent power of the connected machine.

#### Number of pole pairs – Number of pole pairs

1 (default)

Number of pole pairs of the connected machine.

#### PS input unit – Physical signal unit system

SI (default) | Per unit

Unit system for physical signal input.

Select SI to expose the **Pm** port or Per unit to expose the **pu** port.

#### Peak torque to rated torque ratio – Peak torque to rated torque ratio

2 (default)

Ratio that the block multiplies by the base torque to provide the upper limit for the torque that accelerates the inertia.



**Inertia****Specify inertia parameterization by – Inertia parameterization**

Inertia constant, H (default) | Actual Inertia, J

Inertia specification.

**Inertia constant, H – Inertia constant**

3.525 s\*W/VA (default)

Inertia constant.

**Dependencies**

This parameter is visible only if you set the **Specify inertia parameterization by** parameter to Inertia constant, H.

**Actual inertia, J – Actual inertia**27548 kg\*m<sup>2</sup> (default)

Total rotational mechanical inertia of machine rotor. This parameter is visible only if you set **Specify inertia parameterization by** to Actual Inertia, J.

**Dependencies**

This parameter is visible only if you set the **Specify inertia parameterization by** parameter to Actual Inertia, J.

**Specify damper parameterization by – Damping parameterization**

Per-unit damping coefficient, pu\_D (default) | SI damping coefficient, D

Damping specification.

**Per-unit damping coefficient – Per-unit damping coefficient**

0.01 (default)

Damping coefficient in per-unit.

**Dependencies**

This parameter is visible only if you set the **Specify damper parameterization by** parameter to Per-unit damping coefficient, pu\_D.

**SI damping coefficient – Damping coefficient**

39.0509 Nm/(rad/s) (default)

Damping coefficient in SI units.

**Dependencies**

This parameter is visible only if you set the **Specify damper parameterization by** parameter to SI damping coefficient, D.

**Initial Conditions****Specify initialization by – Frequency initialization**

Initial electrical frequency (default) | Initial mechanical frequency

Frequency initialization.

**Initial electrical frequency – Initial electrical frequency**

60 Hz (default)

Initial electrical frequency.

**Dependencies**

This parameter is visible only if you set the **Specify initialization by** parameter to **Initial electrical frequency**.

**Initial mechanical frequency – Initial mechanical frequency**

60 Hz (default)

Initial mechanical frequency.

**Dependencies**

This parameter is visible only if you set the **Specify initialization by** parameter to **Initial mechanical frequency**.

## **Version History**

**Introduced in R2014b**

## **Extended Capabilities**

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

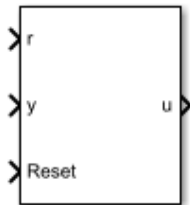
## **See Also**

Machine Inertia

# Model Reference Adaptive Controller

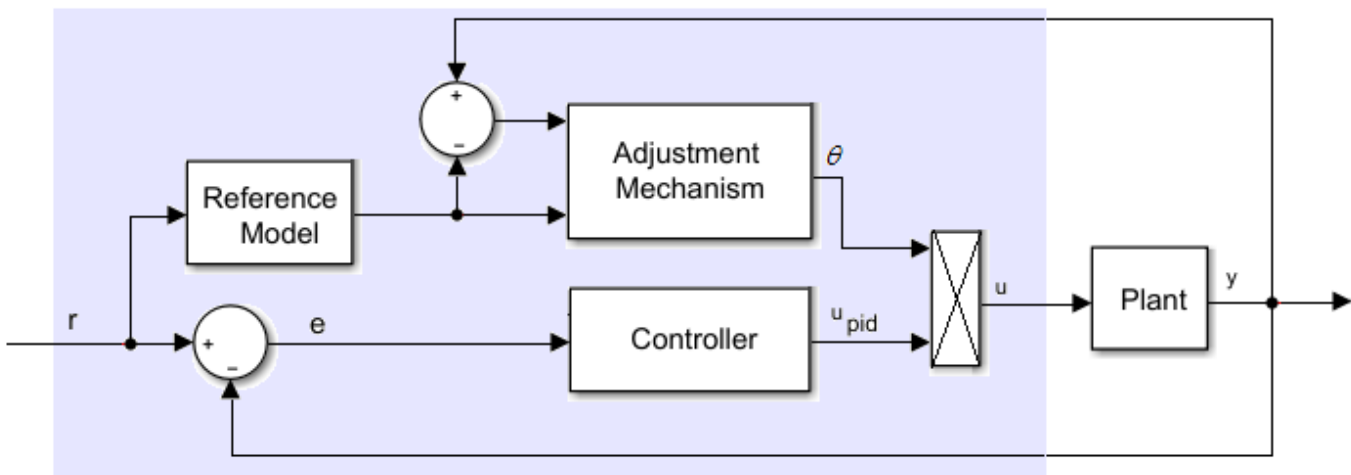
Discrete-time PID-based model reference adaptive control

**Library:** Simscape / Electrical / Control / General Control



## Description

The Model Reference Adaptive Controller block implements discrete-time proportional-integral-derivative (PID) model reference adaptive control (MRAC). The three main components of an MRAC system are the reference model, the adjustment mechanism, and the controller.



## Equations

The control equation is

$$u_{pid}(k) = \left[ K_p + K_i \frac{T_s z}{z-1} + K_d \frac{z-1}{T_s z} \right] e(k),$$

where:

- $u_{pid}$  is the controller output.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $K_d$  is the differential gain.

- $T_s$  is the sample time.
- $e$  is the error.

The reference model is the transfer function for the closed-loop system. This model captures the desired behavior of the closed-loop system. It is implemented as the discrete-time transfer function

$$G_m(z) = \frac{B(z)}{A(z)}.$$

The adaptation mechanism adjusts the control action based on the error between the plant output and the reference model output as

$$\theta = (y - y_m)y_m \frac{-\gamma T_s z}{z-1},$$

where:

- $\theta$  is the adaptation parameter.
- $y$  is the plant output.
- $y_m$  is the reference model output.
- $\gamma$  is the learning rate.

Increasing the value of  $\gamma$  results in faster adaptation to plant changes.

The adjusted control signal,  $u$ , is

$$u(k) = u_{pid}(k)\theta(k).$$

## Ports

### Input

#### **r — Plant reference, r**

scalar

Plant system reference signal.

Data Types: `single` | `double`

#### **y — Plant output, y**

scalar

Plant system output signal.

Data Types: `single` | `double`

#### **Reset — Integrator reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `Boolean`

**Output****u — Controller adjusted output,  $u$** 

scalar

Adjusted control signal.

Data Types: single | double

**Parameters****Controller Parameters****Proportional gain — Controller proportional gain,  $K_p$** 

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.**Integral gain — Controller integral gain,  $K_i$** 

1 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.**Derivative gain — Controller derivative gain,  $K_d$** 

0 (default)

Derivative gain,  $K_d$ , of the controller.**Anti-windup gain — Controller anti-windup gain**

1 (default) | positive scalar

Anti-windup gain of the controller.

**Use filtered derivative — Filter option**

on (default) | off

Choose whether to use a filter coefficient for the reference signal.

**Dependencies**The **Filter coefficient** parameter is only visible when the **Use filtered derivative** check box is selected.**Filter coefficient — Reference signal filter coefficient**

100 (default)

Filter coefficient for the reference signal.

**Dependencies**This parameter is only visible when the **Use filtered derivative** check box is selected.**Control action upper limit — Control signal upper limit,  $u_{pid\_max}$** 

5 (default)

Upper bound for the control signal.

**Control action lower limit – Control signal lower limit,  $u_{pid\_min}$**   
-5 (default)

Lower bound for the control signal.

**Sample time (-1 for inherited) – Block sample time**  
-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

#### Dependencies

If you set **Sample time (-1 for inherited)** to -1 and, in the **Reference Model** settings, set **Model parameterization** to Continuous-time, the **Discretization sample time** parameter becomes visible in the **Reference Model** settings.

#### Reference Model

**Model parameterization – Discrete or continuous model option**  
Discrete-time (default) | Continuous-time

Mathematical model for the controller.

#### Dependencies

Choosing:

- Discrete-time makes the discrete-time parameters visible.
- Continuous-time makes the continuous-time parameters visible. Also, in the **Controller Parameter** settings, if **Sample time (-1 for inherited)** is set to -1, choosing this option makes the **Discretization sample time** parameter visible in the **Reference Model** settings.

**Discrete-time numerator – Discrete-time transfer function numerator**  
[0.01 -0.0099] (default)

Numerator for the discrete-time transfer function.

#### Dependencies

Choosing Discrete-time for **Model parameterization** makes this parameter visible.

**Discrete-time denominator – Discrete-time transfer function denominator**  
[1 -1.9801 0.9802] (default) | vector

Denominator for the discrete-time transfer function.

#### Dependencies

Choosing Discrete-time for **Model parameterization** makes this parameter visible.

**Discretization sample time — Sample time for discretization**

0.01 (default) | -1 or positive number

Time, in seconds, between consecutive discretizations. If block sample time is inherited, specify the discretization sample time explicitly.

**Dependencies**

This parameter is only visible when both of these conditions are met:

- In the **Control Parameters** settings, **Sample time (-1 for inherited)** is set to -1.
- In the **Reference Model** settings, **Model parameterization** is set to Continuous-time.

**Continuous-time numerator — Continuous-time transfer function numerator**

[1 1] (default) | vector

Numerator for the continuous-time transfer function.

**Dependencies**

Choosing Continuous-time for **Model parameterization** makes this parameter visible.

**Continuous-time denominator — Continuous-time transfer function denominator**

[1 2 1] (default) | vector

Denominator for the continuous-time transfer function.

**Dependencies**

Choosing Continuous-time for **Model parameterization** makes this parameter visible.

**Adjustment Mechanism****Learning rate —  $\gamma$** 

0.5 (default)

Rate of adjustment to plant changes.

**Version History**

Introduced in R2018a

**References**

- [1] Butler, H. *Model-Reference Adaptive Control-From Theory to Practice*. Upper Saddle River, NJ: Prentice Hall, 1992.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

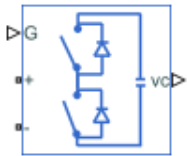
RST Controller



# Modular Multilevel Converter Arm

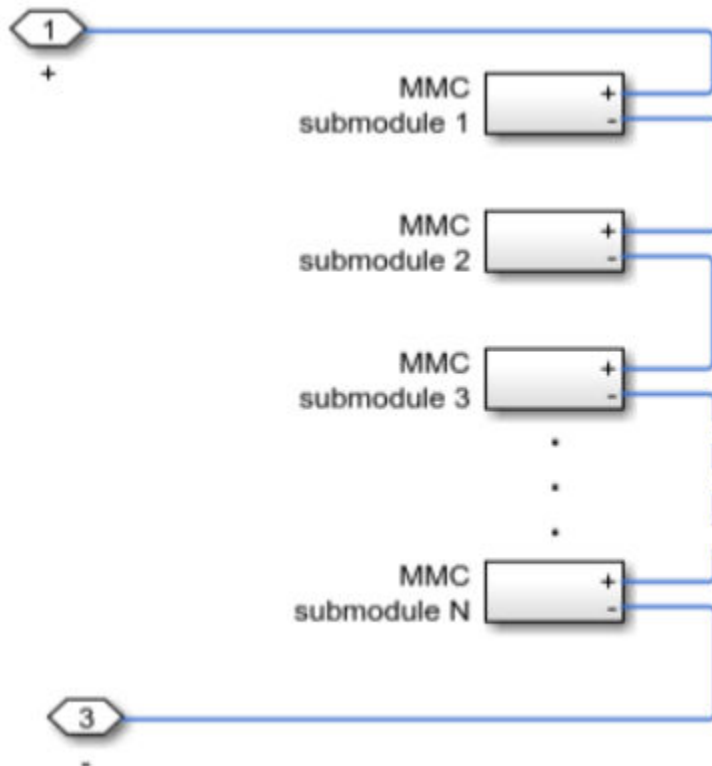
Modular multilevel converter arm with series-connected power submodules

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters

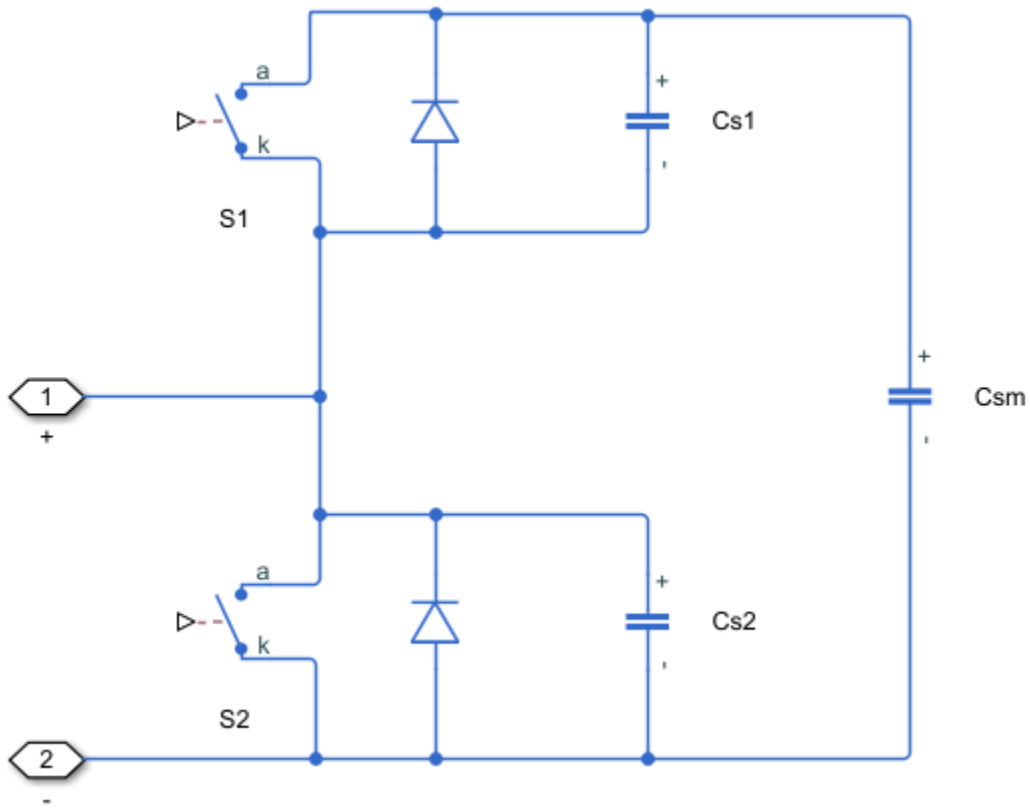


## Description

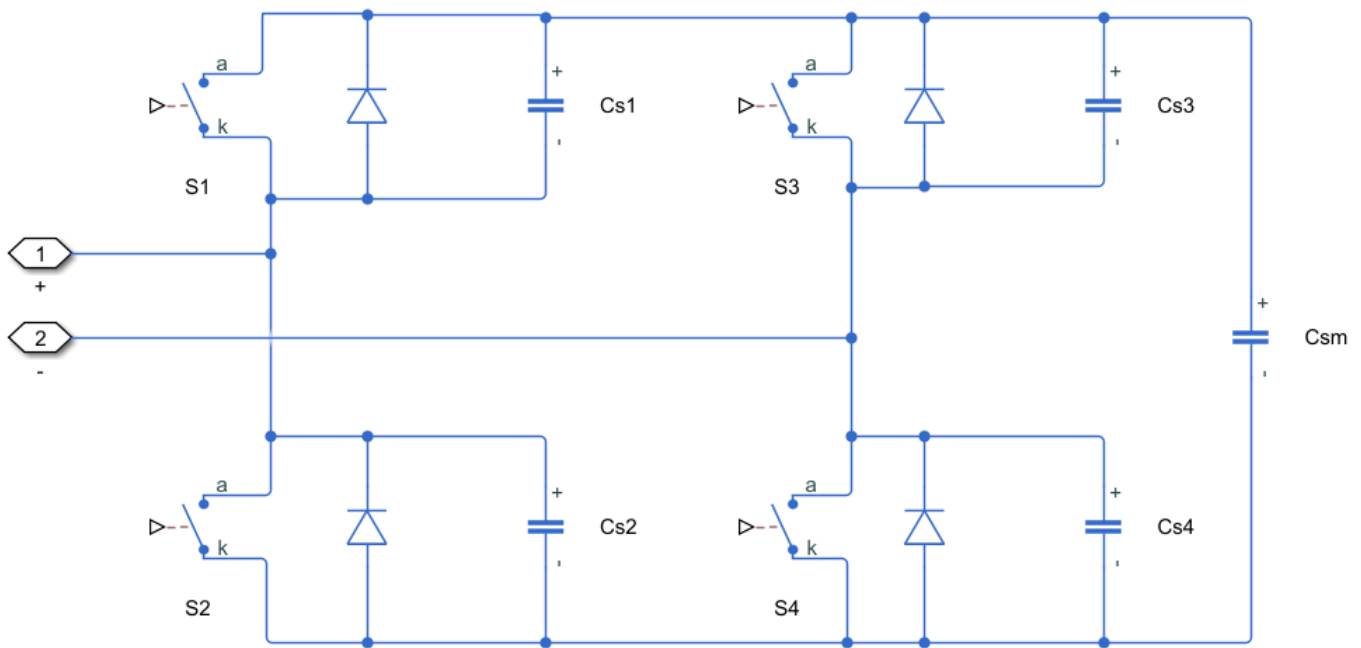
The Modular Multilevel Converter Arm block models a modular multilevel converter arm as a number of series-connected power submodules.



**Half-Bridge Topology**



**Full-Bridge Topology**



This blocks allows you to select the level of model fidelity by choosing between a detailed model with switching devices or an equivalent model. You can choose from these switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Ports

### Input

#### **G — Submodule gate control**

vector of physical signals

Physical signal port associated with the gate signal for all submodules of the modular multilevel converter arm, specified as a vector of physical signals.

If you set the **Converter topology** parameter to *Half-bridge*, the output is a vector of length  $2 * Nsm$ , where  $Nsm$  is the **Number of power submodules**.

If you set the **Converter topology** parameter to Full-bridge, the output is a vector of length  $4 \cdot N_{sm}$ .

#### **Dependencies**

To enable this port, set **Fidelity level** to Detailed model - switching devices or Equivalent model - PWM-controlled.

#### **ref — Submodules reference waveform**

physical signal

Physical signal port associated with the reference waveforms, specified as a physical signal.

#### **Dependencies**

To enable this port, set **Fidelity level** to Equivalent model - waveform-controlled.

#### **Output**

#### **vc — Capacitor voltages**

vector of physical signals

Physical signal port associated with the capacitor voltages for each submodule in the modular multilevel converter arm, specified as a vector of physical signals.

#### **Dependencies**

To enable this port, set **Capacitor voltages** to Instrumented.

#### **Conserving**

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the positive terminal of the modular multilevel converter arm.

#### **- — Negative terminal**

electrical

Electrical conserving port associated with the negative terminal of the modular multilevel converter arm.

## **Parameters**

### **Main**

#### **Converter topology — Converter topology**

Half-bridge (default) | Full-bridge

Topology of the modular multilevel converter.

#### **Fidelity level — Model fidelity**

Detailed model - switching devices (default) | Equivalent model - PWM-controlled |  
Equivalent model - waveform-controlled

Level of fidelity of the model.

**Capacitor voltages — Capacitor voltage instrumentation**

Instrumented (default) | Uninstrumented

Whether to instrument the capacitor voltages.

**Number of power submodules — Number of submodules**

1 (default) | positive scalar

Number of power submodules of the modular multilevel converter.

**Capacitance — Submodule capacitance**

1e-7 F (default) | scalar | vector

Capacitance of a submodule. If you enter a vector, the vector must be of length  $N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.**Capacitor effective series resistance — Capacitor effective series resistance**

1e-6 Ohm (default) | scalar | vector

Capacitor effective series resistance. If you enter a vector, the vector must be of length  $N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.**Dependencies**To enable this parameter, set **Fidelity level** to Detailed model - switching devices.**Capacitor initial voltage — Capacitor initial voltage**

500 V (default) | scalar | vector

Initial voltage of the capacitor. If you enter a vector, the vector must be of length  $N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.**Switching Devices**

This table shows how enabled parameters in the **Switching Devices** settings depend on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2. To enable these settings, set **Fidelity level** to Detailed model - switching devices.

## Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
<b>On-state resistance</b>	<b>Forward voltage</b>	<b>Forward voltage</b>	<b>Drain-source on resistance</b>	<b>Forward voltage</b>	<b>On-state resistance</b>
<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>On-state resistance</b>	<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>Integer for piecewise constant approximation of gate input (0 for disabled)</b>
<b>Threshold voltage</b>	<b>Off-state conductance</b>	<b>Off-state conductance</b>	<b>Threshold voltage</b>	<b>Off-state conductance</b>	
	<b>Gate trigger voltage, Vgt</b>	<b>Threshold voltage</b>		<b>Gate trigger voltage, Vgt</b>	
	<b>Gate turn-off voltage, Vgt_off</b>			<b>Holding current</b>	
	<b>Holding current</b>				

### Switching device – Switch type

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the converter.

### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage – Voltage

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is the:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

### Dependencies

See the Switching Devices Parameter Dependencies table.

### On-state resistance – Resistance

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is the:

- GTO — Rate of change of the voltage versus the current above the forward voltage

- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Drain-source on resistance — Resistance

0.001 Ohm (default) | scalar

Resistance between the drain and the source. The gate-to-source voltage also affects the **Drain-source on resistance** parameter.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Off-state conductance — Conductance when device is off

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is the:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Threshold voltage — Voltage threshold

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. The voltage threshold applies to different devices depending on the switching device used:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate trigger voltage, Vgt — Gate-cathode activation voltage threshold

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off – Gate-cathode deactivation voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Holding current – Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Integer for piecewise constant approximation of gate input (0 for disabled) – Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Protection Diodes**

For more information on these parameters, see Diode.

**Model dynamics – Diode model**

Diode with no dynamics (default) | Diode with charge dynamics | None

Diode type. The options are:

- **None** — The block does not model diode dynamics.
- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you set **Switching device** to Averaged Switch in the **Switching Device** settings, the **Diode with no dynamics** setting is automatically selected.

---



**Dependencies**

To enable this parameter, set **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

**Forward voltage — Forward voltage**

0.8 V (default) | scalar

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**On resistance — On resistance**

0.001 Ohm (default) | scalar

Rate of change of voltage versus the current above the **Forward voltage**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**Off conductance — Off conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Rate of change of current when measuring iRM – Rate of change of current when measuring iRM**

-50 A/ $\mu$ s (default) | negative scalar

Rate of change of the current when measuring the peak reverse current. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Reverse recovery time parameterization – Reverse recovery time option**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Determines how you specify reverse recovery time.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Reverse recovery time, trr – Reverse recovery time**

15  $\mu$ s (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, iRM** parameter divided by the value of the **Rate of change of current when measuring iRM** parameter.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery time directly.

**Reverse recovery time stretch factor – Reverse recovery time stretch factor**

3 (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify stretch factor.

**Reverse recovery charge,  $Q_{rr}$  — Reverse recovery charge**1500 s\* $\mu$ A (default) | scalar

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ , where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery charge.

**Snubbers**

To enable the **Snubbers** settings, set **Fidelity level** to Detailed model - switching devices and **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

**Snubber — Snubber**

None (default) | RC snubber

Snubber for each switching device:

- None
- RC snubber

**Snubber resistance — Snubber resistance**

0.1 Ohm (default) | scalar

Snubber resistance.

**Dependencies**

To enable this parameter, set **Snubber** to RC snubber.

**Snubber capacitance — Snubber capacitance**

1e-7 F (default) | scalar

Snubber capacitance.

**Dependencies**

To enable this parameter, set **Snubber** to RC snubber.

**Version History**

Introduced in R2020b

## References

- [1] Saad, Hani, Sebastien Denetiere, and Jean Mahseredjian. "On Modelling of MMC in EMT-Type Program." 2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL), 1-7. Trondheim, Norway: IEEE, 2016. <https://doi.org/10.1109/COMPEL.2016.7556717>.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

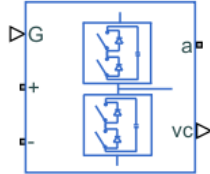
### See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear) | Boost Converter | PWM Generator (Multilevel) | Modular Multilevel Converter Leg | Modular Multilevel Converter (Three-Phase)

# Modular Multilevel Converter Leg

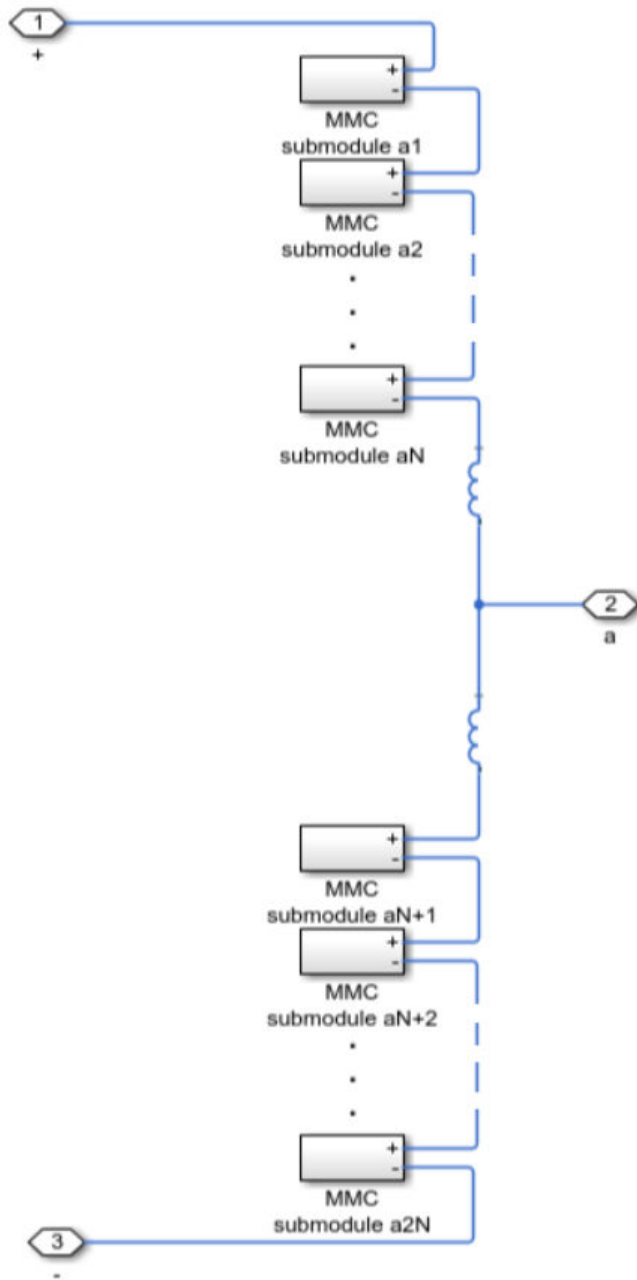
Modular multilevel converter leg with series-connected power submodules

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters

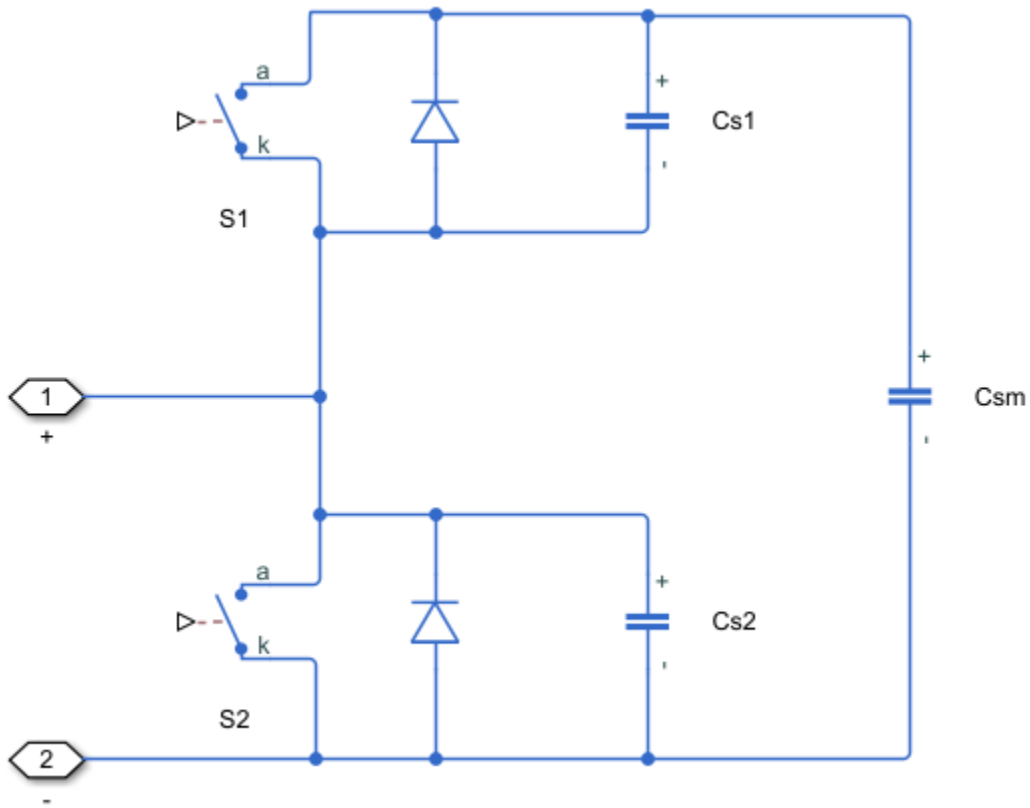


## Description

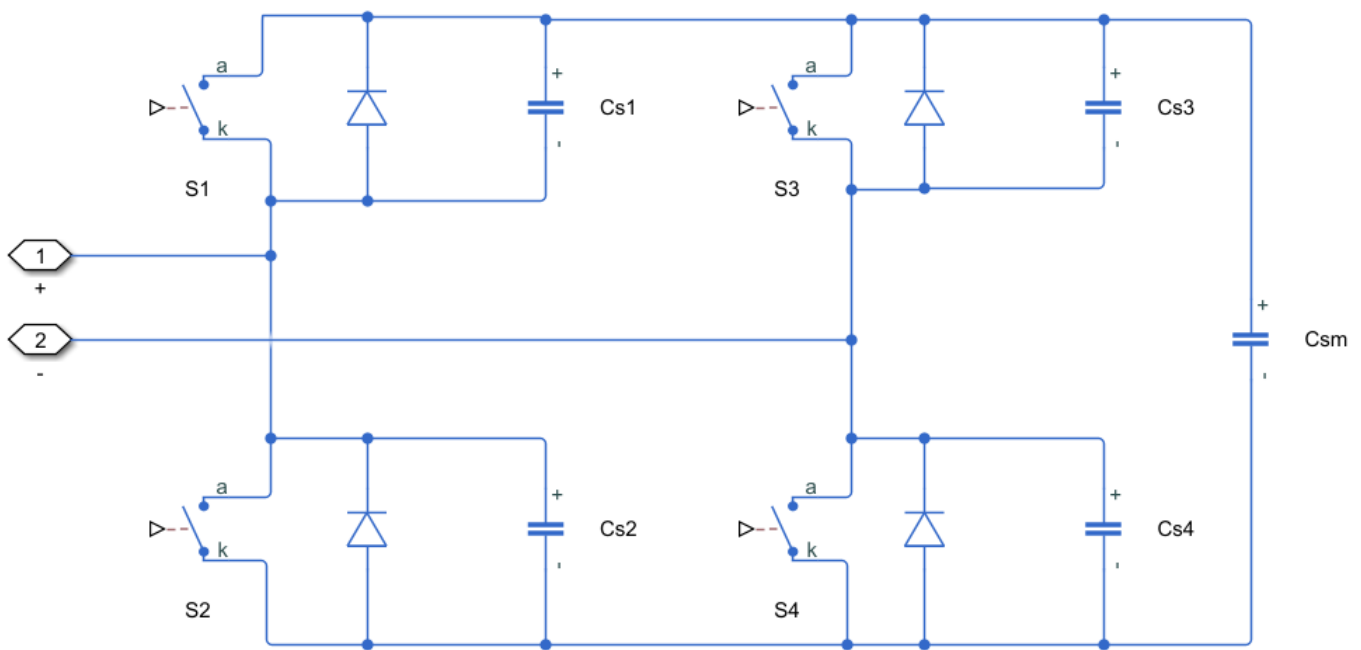
The Modular Multilevel Converter Leg block models a modular multilevel converter leg as a two arms that are implemented with a number of series-connected power submodules.



**Half-Bridge Topology**



**Full-Bridge Topology**



This blocks allows you to select the level of model fidelity by choosing between a detailed model with switching devices or an equivalent model. You can choose from these switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to **Averaged switch** and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.



## Ports

### Input

#### **G — Submodule gate control**

vector of physical signals

Physical signal port associated with the gate signal for all submodules of the modular multilevel converter leg, specified as a vector of physical signals.

If you set the **Converter topology** parameter to `Half-bridge`, the output is a vector of length  $4 * N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.

If you set the **Converter topology** parameter to `Full-bridge`, the output is a vector of length  $8 * N_{sm}$ .

#### **Dependencies**

To enable this port, set **Fidelity level** to `Detailed model - switching devices` or `Equivalent model - PWM-controlled`.

#### **ref — Submodules reference waveform**

vector of physical signals

Physical signal port associated with the reference waveforms, specified as a vector of physical signals of length 2.

#### **Dependencies**

To enable this port, set **Fidelity level** to `Equivalent model - waveform-controlled`.

### Output

#### **vc — Capacitor voltages**

vector of physical signals

Physical signal port associated with the capacitor voltages for each submodule in the modular multilevel converter leg, specified as a vector of physical signals.

#### **Dependencies**

To enable this port, set **Capacitor voltages** to `Instrumented`.

### Conserving

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the positive terminal of the modular multilevel converter leg.

#### **- — Negative terminal**

electrical

Electrical conserving port associated with the negative terminal of the modular multilevel converter leg.

**a – Phase a terminal**

electrical

Electrical conserving port associated with the single-phase terminal of the modular multilevel converter leg.

**Parameters****Main****Converter topology – Converter topology**

Half-bridge (default) | Full-bridge

Topology of the modular multilevel converter.

**Fidelity level – Model fidelity**

Detailed model - switching devices (default) | Equivalent model - PWM-controlled | Equivalent model - waveform-controlled

Level of fidelity of the model.

**Capacitor voltages – Capacitor voltage instrumentation**

Instrumented (default) | Uninstrumented

Whether to instrument the capacitor voltages.

**Number of power submodules – Number of submodules**

1 (default) | positive scalar

Number of power submodules of the modular multilevel converter.

**Capacitance – Submodule capacitance**

1e-7 F (default) | scalar | vector

Capacitance of a submodule. If you enter a vector, the vector must be of length  $2 * N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.

**Capacitor effective series resistance – Capacitor effective series resistance**

1e-6 Ohm (default) | scalar | vector

Capacitor effective series resistance. If you enter a vector, the vector must be of length  $2 * N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.

**Dependencies**

To enable this parameter, set **Fidelity level** to Detailed model - switching devices.

**Arm inductance – Arm inductance**

1e-6 H (default) | scalar

Inductance of the arm.

**Arm inductance series resistance – Arm inductance series resistance**

0 Ohm (default) | scalar

Arm inductance series resistance.

**Capacitor initial voltage – Capacitor initial voltage**

500 V (default) | scalar | vector

Initial voltage of the capacitor. If you enter a vector, the vector must be of length  $2 * N_{sm}$ , where  $N_{sm}$  is the **Number of power submodules**.

**Switching Devices**

This table shows how enabled parameters in the **Switching Devices** settings depend on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2. To enable these settings, set **Fidelity level** to Detailed model - switching devices.

**Switching Devices Parameter Dependencies**

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
<b>On-state resistance</b>	<b>Forward voltage</b>	<b>Forward voltage</b>	<b>Drain-source on resistance</b>	<b>Forward voltage</b>	<b>On-state resistance</b>
<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>On-state resistance</b>	<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>Integer for piecewise constant approximation of gate input (0 for disabled)</b>
<b>Threshold voltage</b>	<b>Off-state conductance</b>	<b>Off-state conductance</b>	<b>Threshold voltage</b>	<b>Off-state conductance</b>	
	<b>Gate trigger voltage, Vgt</b>	<b>Threshold voltage</b>		<b>Gate trigger voltage, Vgt</b>	
	<b>Gate turn-off voltage, Vgt_off</b>			<b>Holding current</b>	
	<b>Holding current</b>				

**Switching device – Switch type**

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the converter.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Forward voltage – Voltage**

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is the:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**

- Thyristor — Minimum voltage required for the device to turn on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**On-state resistance — Resistance**

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is the:

- GTO — Rate of change of the voltage versus the current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source. The gate-to-source voltage also affects the **Drain-source on resistance** parameter.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance when device is off**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is the:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. The voltage threshold applies to different devices depending on the switching device used:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate trigger voltage, $V_{gt}$ — Gate-cathode activation voltage threshold

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate turn-off voltage, $V_{gt\_off}$ — Gate-cathode deactivation voltage threshold

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Holding current — Current threshold

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

#### Dependencies

To enable this parameter, set **Switching device** to Averaged Switch.

#### Protection Diodes

For more information on these parameters, see Diode.

#### Model dynamics — Diode model

Diode with no dynamics (default) | Diode with charge dynamics | None

Diode type. The options are:

- **None** — The block does not model diode dynamics.
- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you set **Switching device** to Averaged Switch in the **Switching Device** settings, the Diode with no dynamics setting is automatically selected.

---

### **Dependencies**

To enable this parameter, set **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

### **Forward voltage — Forward voltage**

0.8 V (default) | scalar

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### **Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

### **On resistance — On resistance**

0.001 Ohm (default) | scalar

Rate of change of voltage versus the current above the **Forward voltage**.

### **Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

### **Off conductance — Off conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

### **Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

### **Junction capacitance — Junction capacitance**

50 nF (default) | scalar

Diode junction capacitance.

### **Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** -50 A/ $\mu$ s (default) | negative scalar

Rate of change of the current when measuring the peak reverse current. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time option**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Determines how you specify reverse recovery time.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Reverse recovery time,  $t_{rr}$  — Reverse recovery time**15  $\mu$ s (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery time directly.

**Reverse recovery time stretch factor – Reverse recovery time stretch factor**

3 (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify stretch factor.

**Reverse recovery charge, Qrr – Reverse recovery charge**

1500 s\*μA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ , where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery charge.

**Snubbers**

To enable the **Snubbers** settings, set **Fidelity level** to Detailed model - switching devices and **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

**Snubber – Snubber**

None (default) | RC snubber

Snubber for each switching device:

- None
- RC snubber

**Snubber resistance – Snubber resistance**

0.1 Ohm (default) | scalar

Snubber resistance.

**Dependencies**

To enable this parameter, set **Snubber** to RC snubber.

**Snubber capacitance – Snubber capacitance**

1e-7 F (default) | scalar



Snubber capacitance.

### Dependencies

To enable this parameter, set **Snubber** to RC snubber.

## Version History

Introduced in R2020b

### References

- [1] Saad, Hani, Sebastien Denetiere, and Jean Mahseredjian. "On Modelling of MMC in EMT-Type Program." 2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL), 1-7. Trondheim, Norway: IEEE, 2016. <https://doi.org/10.1109/COMPEL.2016.7556717>.

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

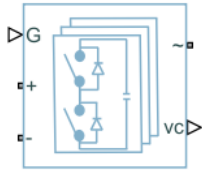
### See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear) | Boost Converter | PWM Generator (Multilevel) | Modular Multilevel Converter Arm | Modular Multilevel Converter (Three-Phase)

## Modular Multilevel Converter (Three-Phase)

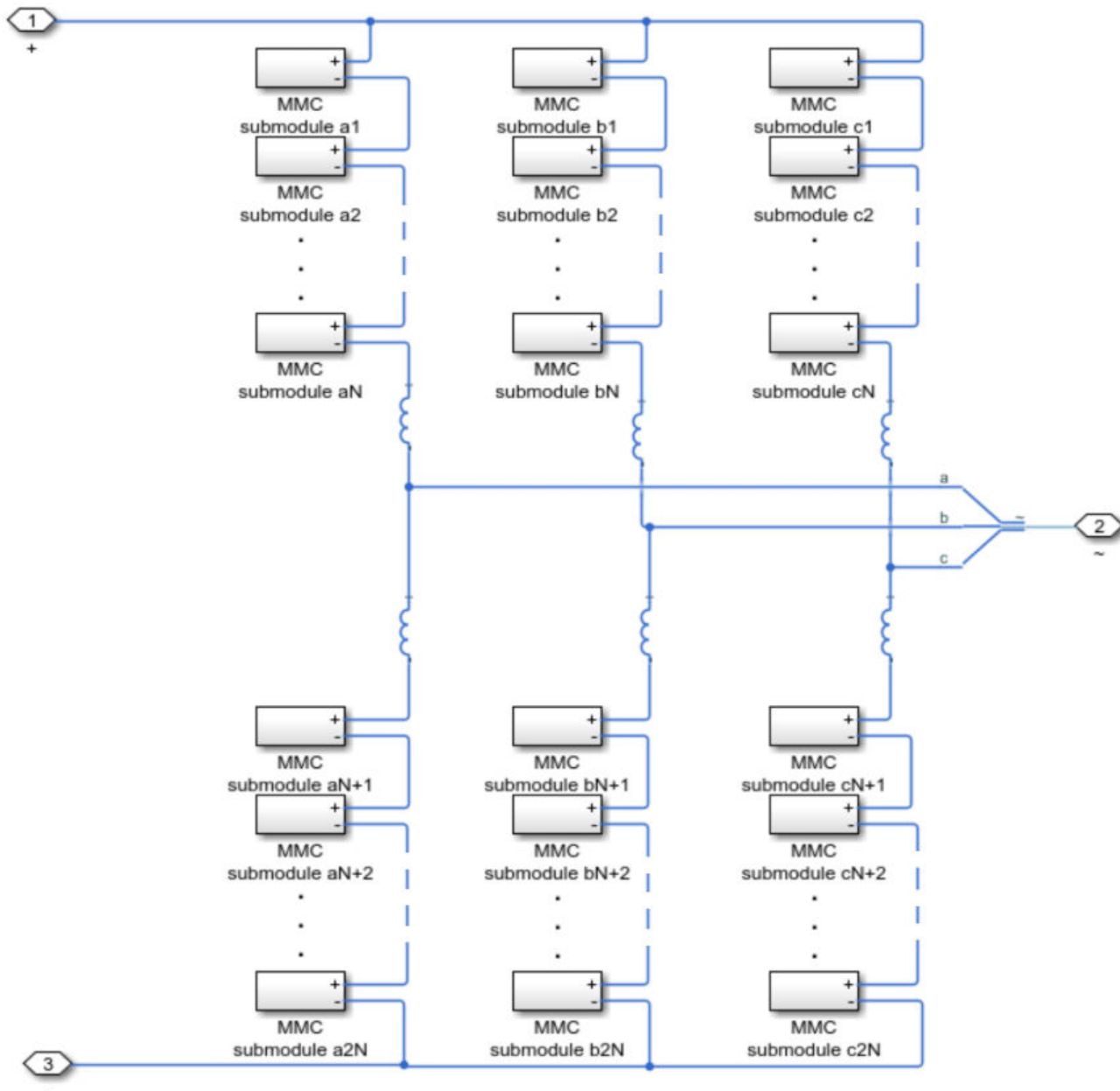
Three-phase modular multilevel converter with series-connected power submodules

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



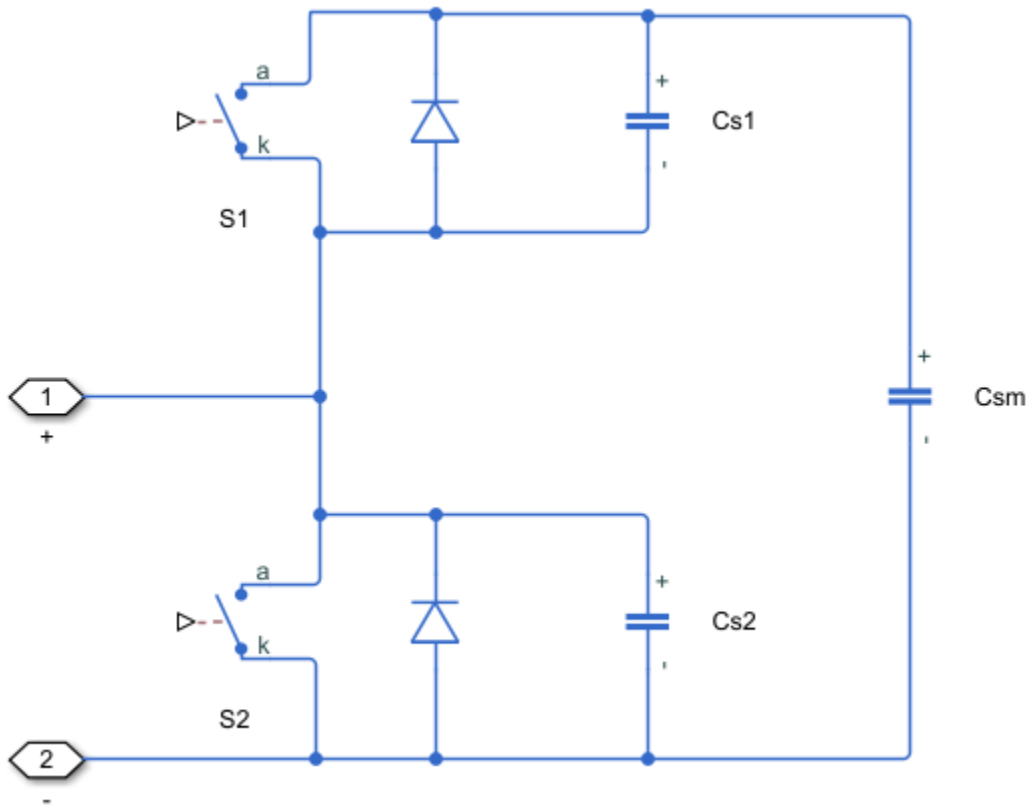
### Description

The Modular Multilevel Converter (Three-Phase) block models a three-phase modular multilevel converter. Each phase consists of two arms that are implemented with a number of series-connected power submodules.

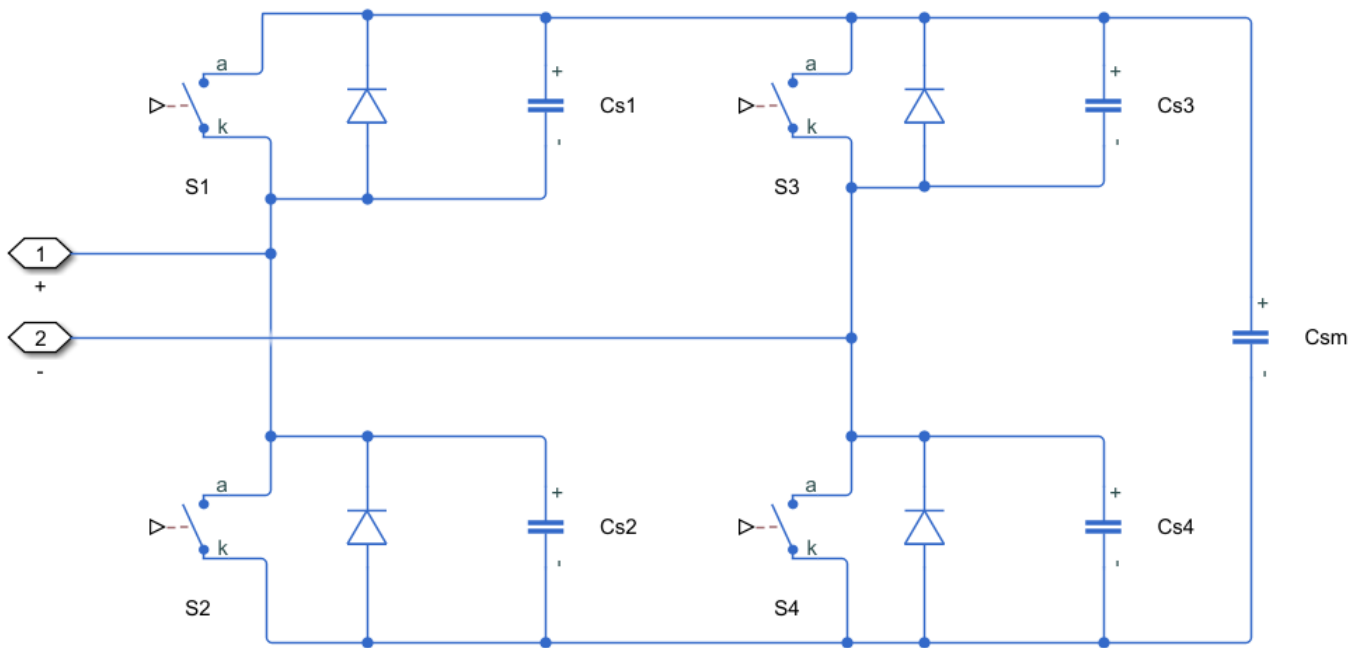


Each submodule consists of a half-bridge or full-bridge converter and a capacitor.

### Half-Bridge Topology



**Full-Bridge Topology**



This blocks allows you to select the level of model fidelity by choosing between a detailed model with switching devices or an equivalent model. You can choose from these switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to **Averaged switch** and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Input

#### **G — Submodule gate control**

matrix of physical signals

Physical signal port associated with the gate signal for all submodules of the modular multilevel converter, specified as a matrix.

If you set the **Converter topology** parameter to `Half-bridge`, the signal is a vector of length  $12 * Nsm$ , where  $Nsm$  is the **Number of power submodules**.

If you set the **Converter topology** parameter to `Full-bridge`, the signal is a vector of length  $24 * Nsm$ .

#### **Dependencies**

To enable this port, set **Fidelity level** to `Detailed model - switching devices` or `Equivalent model - PWM-controlled`.

#### **ref — Submodule reference waveform**

vector of physical signals

Physical signal port associated with the reference waveforms of the submodules, specified as a vector of length 6.

#### **Dependencies**

To enable this port, set **Fidelity level** to `Equivalent model - waveform-controlled`.

### Output

#### **vc — Capacitor voltages**

vector of physical signals

Physical signal port associated with the capacitor voltages for each submodule in the modular multilevel converter, returned as a vector.

#### **Dependencies**

To enable this port, set **Capacitor voltages** to `Instrumented`.

### Conserving

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the positive terminal of the modular multilevel converter.

#### **- — Negative terminal**

electrical

Electrical conserving port associated with the negative terminal of the modular multilevel converter.

#### **~ — Three-phase port**

electrical

Expandable three-phase electrical port associated with the three-phase terminal of the modular multilevel converter.

#### **Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

#### **Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

## **Parameters**

### **Main**

#### **Electrical connection – Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

#### **Converter topology – Converter topology**

Half-bridge (default) | Full-bridge

Topology of the modular multilevel converter.

#### **Fidelity level – Model fidelity**

Detailed model - switching devices (default) | Equivalent model - PWM-controlled |  
Equivalent model - waveform-controlled

Level of fidelity of the model.

#### **Capacitor voltages – Capacitor voltage instrumentation**

Instrumented (default) | Uninstrumented

Whether to instrument the capacitor voltages.

**Number of power submodules – Number of submodules**

1 (default) | positive scalar

Number of power submodules of the modular multilevel converter.

**Capacitance – Submodule capacitance**

1e-7 F (default) | scalar | vector

Capacitance of a submodule. If you enter a vector, the vector must be of length  $6 * Nsm$ , where  $Nsm$  is the **Number of power submodules**.

**Capacitor effective series resistance – Capacitor effective series resistance**

1e-6 Ohm (default) | scalar | vector

Capacitor effective series resistance. If you enter a vector, the vector must be of length  $6 * Nsm$ , where  $Nsm$  is the **Number of power submodules**.

**Dependencies**

To enable this parameter, set **Fidelity level** to Detailed model - switching devices.

**Arm inductance – Arm inductance**

1e-6 H (default) | scalar

Inductance of the arm.

**Arm inductance series resistance – Arm inductance series resistance**

0 Ohm (default) | scalar

Arm inductance series resistance.

**Capacitor initial voltage – Capacitor initial voltage**

500 V (default) | scalar | vector

Initial voltage of the capacitor. If you enter a vector, the vector must be of length  $6 * Nsm$ , where  $Nsm$  is the **Number of power submodules**.

**Switching Devices**

This table shows how enabled parameters in the **Switching Devices** settings depend on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2. To enable these settings, set **Fidelity level** to Detailed model - switching devices.



## Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
<b>On-state resistance</b>	<b>Forward voltage</b>	<b>Forward voltage</b>	<b>Drain-source on resistance</b>	<b>Forward voltage</b>	<b>On-state resistance</b>
<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>On-state resistance</b>	<b>Off-state conductance</b>	<b>On-state resistance</b>	<b>Integer for piecewise constant approximation of gate input (0 for disabled)</b>
<b>Threshold voltage</b>	<b>Off-state conductance</b>	<b>Off-state conductance</b>	<b>Threshold voltage</b>	<b>Off-state conductance</b>	
	<b>Gate trigger voltage, Vgt</b>	<b>Threshold voltage</b>		<b>Gate trigger voltage, Vgt</b>	
	<b>Gate turn-off voltage, Vgt_off</b>			<b>Holding current</b>	
	<b>Holding current</b>				

### Switching device – Switch type

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the converter.

### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage – Voltage

0.8 V (default) | scalar

For the different switching device types, the **Forward voltage** is the:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

### Dependencies

See the Switching Devices Parameter Dependencies table.

### On-state resistance – Resistance

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is the:

- GTO — Rate of change of the voltage versus the current above the forward voltage

- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Drain-source on resistance — Resistance**

0.001 Ohm (default) | scalar

Resistance between the drain and the source. The gate-to-source voltage also affects the **Drain-source on resistance** parameter.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Off-state conductance — Conductance when device is off**

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is the:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. The voltage threshold applies to different devices depending on the switching device used:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate trigger voltage, Vgt — Gate-cathode activation voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate turn-off voltage, `Vgt_off` – Gate-cathode deactivation voltage threshold

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Holding current – Current threshold

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Integer for piecewise constant approximation of gate input (0 for disabled) – Integer mode for FPGA deployment

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

#### Dependencies

To enable this parameter, set **Switching device** to Averaged Switch.

#### Protection Diodes

For more information on these parameters, see Diode.

#### Model dynamics – Diode model

Diode with no dynamics (default) | Diode with charge dynamics | None

Diode type. The options are:

- **None** — The block does not model diode dynamics.
- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you set **Switching device** to Averaged Switch in the **Switching Device** settings, the **Diode with no dynamics** setting is automatically selected.

---

**Dependencies**

To enable this parameter, set **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

**Forward voltage — Forward voltage**

0.8 V (default) | scalar

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**On resistance — On resistance**

0.001 Ohm (default) | scalar

Rate of change of voltage versus the current above the **Forward voltage**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**Off conductance — Off conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with no dynamics or Diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

#### Dependencies

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

#### Rate of change of current when measuring **i<sub>RM</sub>** — Rate of change of current when measuring **i<sub>RM</sub>**

-50 A/μs (default) | negative scalar

Rate of change of the current when measuring the peak reverse current. This value must be less than zero.

#### Dependencies

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

#### Reverse recovery time parameterization — Reverse recovery time option

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Determines how you specify reverse recovery time.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

#### Dependencies

To enable this parameter, set **Model dynamics** to Diode with charge dynamics.

#### Reverse recovery time, **trr** — Reverse recovery time

15 μs (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, i<sub>RM</sub>** parameter divided by the value of the **Rate of change of current when measuring i<sub>RM</sub>** parameter.

#### Dependencies

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery time directly.

#### Reverse recovery time stretch factor — Reverse recovery time stretch factor

3 (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify stretch factor.

**Reverse recovery charge, Q<sub>rr</sub> — Reverse recovery charge**

1500 s\*μA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ , where:

- $i_{RM}$  is the value specified for **Peak reverse current, i<sub>RM</sub>**.
- $a$  is the value specified for **Rate of change of current when measuring i<sub>RM</sub>**.

**Dependencies**

To enable this parameter, set **Model dynamics** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery charge.

**Snubbers**

To enable the **Snubbers** settings, set **Fidelity level** to Detailed model - switching devices and **Switching device** to GTO, Ideal Semiconductor Switch, IGBT, MOFSET, or Thyristor.

**Snubber — Snubber**

None (default) | RC snubber

Snubber for each switching device:

- None
- RC snubber

**Snubber resistance — Snubber resistance**

0.1 Ohm (default) | scalar

Snubber resistance.

**Dependencies**

To enable this parameter, set **Snubber** to RC snubber.

**Snubber capacitance — Snubber capacitance**

1e-7 F (default) | scalar

Snubber capacitance.

**Dependencies**

To enable this parameter, set **Snubber** to RC snubber.

**Version History****Introduced in R2020b**

## Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Saad, Hani, Sebastien Denetiere, and Jean Mahseredjian. "On Modelling of MMC in EMT-Type Program." 2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL), 1-7. Trondheim, Norway: IEEE, 2016. <https://doi.org/10.1109/COMPEL.2016.7556717>.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

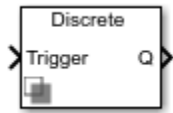
## See Also

Average-Value DC-DC Converter | Bidirectional DC-DC Converter | Buck Converter | Buck-Boost Converter | Converter (Three-Phase) | GTO | IGBT (Ideal, Switching) | MOSFET (Ideal, Switching) | Ideal Semiconductor Switch | PWM Generator | PWM Generator (Three-phase, Two-level) | Three-Level Converter (Three-Phase) | Thyristor (Piecewise Linear) | Boost Converter | PWM Generator (Multilevel) | Modular Multilevel Converter Arm | Modular Multilevel Converter Leg

## Monostable Flip-Flop

Rising edge, falling edge, either edge monostable flip-flop

**Library:** Simscape / Electrical / Control / General Control



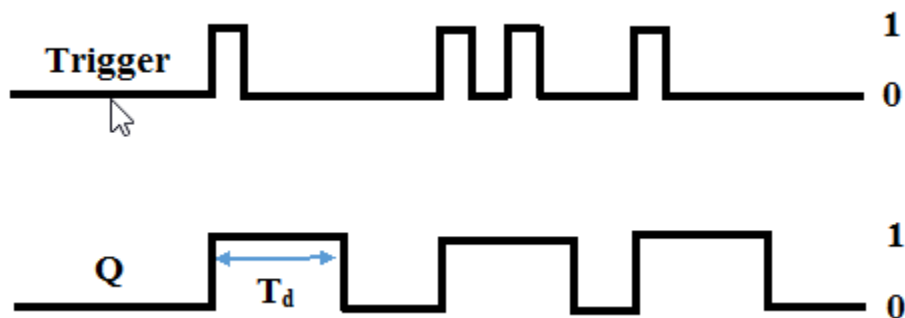
### Description

The Monostable Flip-Flop (or monostable multivibrator) block generates a single output pulse of a specified duration when it is triggered externally. The external trigger is a Boolean signal. Pulse generation is triggered when a change is detected in the external trigger signal. The change detection can be:

- Rising edge
- Falling edge
- Either edge

When the output is true, a change in the external trigger signal has no effect.

The operation of a monostable flip-flop is represented in the following figure:



### Ports

#### Input

**Trigger – Flip-flop trigger**

0 | 1

Input trigger signal to the flip-flop.

Data Types: Boolean



## Output

### Q — Flip-flop output

0 | 1

Flip-flop output signal.

Data Types: Boolean

## Parameters

### Change detection — Change in trigger signal

Rising edge (default) | Falling edge | Either edge

Change detection strategy for output pulse generation.

### Initial output — Flip-flop initial output

0 (default) | scalar

The initial output of the flip-flop.

### Pulse duration (s) — Output pulse duration

0.01 (default) | positive scalar

Duration of the output pulse, in s.

### Sample time — Block sample time

0.001 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For discrete-time operation, set the sample time to a positive value. For continuous-time operation, set the sample time to 0.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2018b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Set-Reset Flip-Flop | Change Detector | Signal Sample and Hold

# MOSFET (Ideal, Switching)

Ideal N-channel MOSFET for switching applications

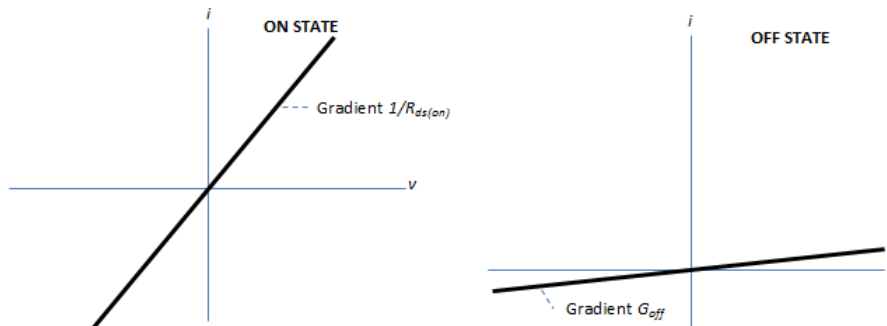
**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The MOSFET (Ideal, Switching) block models the ideal switching behavior of an n-channel metal-oxide-semiconductor field-effect transistor (MOSFET).

The switching characteristic of an n-channel MOSFET is such that if the gate-source voltage exceeds the specified threshold voltage, the MOSFET is in the on state. Otherwise, the device is in the off state. This figure shows a typical i-v characteristic:



To define the I-V characteristic of the MOSFET, set the **On-state behaviour and switching losses** parameter to either **Specify constant values** or **Tabulate with temperature and current**. The **Tabulate with temperature and current** option is available only if you expose the thermal port of the block.

In the on state, the drain-source path behaves like a linear resistor with resistance,  $R_{ds\ on}$ . However, if you expose the thermal port of the block and parameterize the device using tabulated  $\bar{I}$ - $\bar{V}$  data, the tabulated resistance is a function of the temperature and current.

In the off state, the drain-source path behaves like a linear resistor with low off-state conductance,  $G_{off}$ .

Then, the defining Simscape equations for the block are:

```
if G > Vth      v == i*Rds_on;      else      v == i/Goff;      end
```

where:

- $G$  depends on the value of the **Gate control port** parameter.
- If you set the **Gate control port** parameter to PS, you control the gate terminal through a physical signal.  $G$  is the value at the input port **G**.

- If you set the **Gate control port** parameter to `Electrical`, you control the gate terminal through an electrical signal.  $G$  is equal to:

```
if v >= 0      G = G.v - S.v;      else      G = G.v - D.v;      end
```

where  $G.v$  is the gate voltage,  $S.v$  is the source voltage, and  $D.v$  is the drain voltage.

- $V_{th}$  is the threshold voltage.
- $v$  is the drain-source voltage.
- $i$  is the drain-source current.
- $R_{ds\_on}$  is the on-state resistance.
- $G_{off}$  is the off-state conductance.

Using the **Integral Diode** settings, you can include the body diode or an integral protection diode. The integral diode provides a conduction path for reverse current. For example, to provide a path for a high reverse-voltage spike that is generated when a semiconductor device suddenly switches off the voltage supply to an inductive load.

Set the **Integral protection diode** parameter based on your goal.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	Protection diode with no dynamics	The block includes an integral copy of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.
Precisely specify reverse-mode charge dynamics.	Protection diode with charge dynamics	The block includes an integral copy of the dynamic model of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.

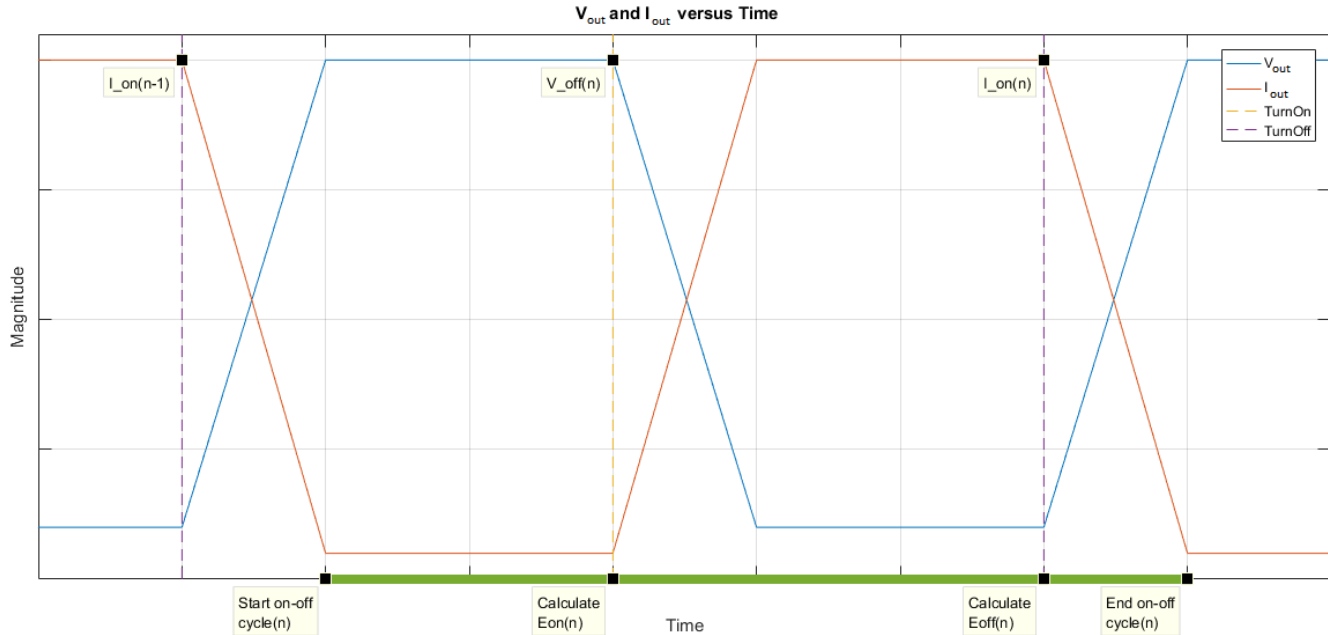
### Model Gate Port and Thermal Effects

You can choose between physical or electrical ports to control the gate terminal and expose the thermal port to model the heat that switching events and conduction losses generate. To choose the gate control port, set the **Gate control port** parameter to `PS` or `Electrical`. To expose the thermal port, set the **Modeling option** parameter to either `No thermal port` or `Show thermal port`.

For more information about using thermal ports, see “Simulating Thermal Effects in Semiconductors”.

### Thermal Losses

The figure shows an idealized representation of the output voltage,  $V_{out}$ , and the output current,  $I_{out}$ , of the semiconductor device. The interval shown includes the entire  $n^{th}$  switching cycle, during which the block turns off and then on.



Switching losses are one of the main sources of thermal loss in semiconductors. During each on-off switching transition, the MOSFET parasitics store and then dissipate energy.

Switching losses depend on the off-state voltage and the on-state current. When a switching device is turned on, the power losses depend on the initial off-state voltage across the device and the final on-state current once the device is fully in its on state. Similarly, when a switching device is turned off, the power losses depend on the initial on-state current through the device and the final off-state voltage across the device when in the fully off state.

In this block, switching losses are applied by stepping up the junction temperature with a value equal to the switching loss divided by the total thermal mass at the junction. The **Switch-on loss,  $E_{on}(T_j, I_{ds})$**  and **Switch-off loss,  $E_{off}(T_j, I_{ds})$**  parameter values set the sizes of the switching losses and they are either fixed or dependent on junction temperature and drain-source current. In both cases, losses are scaled by the off-state voltage prior to the latest device turn-on event.

**Note** As the final current after a switching event is not known during the simulation, the block records the on-state current at the point that the device is commanded off. Similarly, the block records the off-state voltage at the point that the device is commanded on. For this reason, the simlog does not report the switching losses to the thermal network until one switching cycle later.

For all ideal switching devices, the switching losses are reported in the simlog as `lastTurnOffLoss` and `lastTurnOnLoss` and recorded as a pulse with amplitude equal to the energy loss. If you use a script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

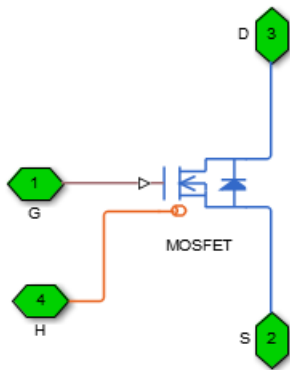
## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

The figure shows the block port names.



## Conserving

### G – Gate terminal

electrical

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

### S – Source terminal

electrical

Electrical conserving port associated with the source terminal.

### D – Drain terminal

electrical

Electrical conserving port associated with the drain terminal.

### H – Thermal port

thermal

Thermal conserving port.

## Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Thermal port visibility

No thermal port (default) | Show thermal port

Whether to enable the thermal port.

### Main

This table shows how the visibility of **Main** parameters depends on how you configure the **Modeling option** and **On-state behaviour and switching losses** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

### Main Parameter Dependencies

Parameters and Options		
<b>Modeling option</b>		
<i>No thermal port</i>	<i>Show thermal port</i>	
Gate-control port	Gate-control port	
Drain-source on resistance, R_DS(on)	Threshold voltage, Vth	
Off-state conductance	<b>On-state behaviour and switching losses</b>	
	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>
Threshold voltage, Vth	Drain-source on resistance, R_DS(on)	On-state voltage, Vds(Tj,Ids)
	Off-state conductance	Temperature vector, Tj
		Drain-source current vector, Ids
		Off-state conductance

### Gate-control port – Whether to specify physical or electrical control port

PS (default) | Electrical

Whether to specify physical or electrical control port for the switch gate.

**Note** If you set this parameter to `Electrical`, use an internal or external reverse diode along with this block. For numerical considerations, the forward voltage of the diode must be smaller than the value of the **Threshold voltage, Vth** parameter of this block.

### On-state behaviour and switching losses – On-state current for switching loss data

Specify constant values (default) | Tabulate with temperature and current

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- **Specify constant values** — Use scalar values to specify the output current, switch-on loss, and switch-off loss data. This is the default parameterization method.
- **Tabulate with temperature and current** — Use vectors to specify the output current, switch-on loss, switch-off loss, and temperature data.

**Dependencies**

See the Main Parameter Dependencies table.

**Drain-source on resistance,  $R_{DS(on)}$  — Drain-source on resistance**

0.01 Ohm (default)

Drain-source resistance when the device is on.

**Dependencies**

See the Main Parameter Dependencies table.

**Off-state conductance — Off-state conductance**

1e-6 1/Ohm (default)

Drain-source conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Dependencies**

See the Main Parameter Dependencies table.

**Threshold voltage,  $V_{th}$  — Threshold voltage**

2 V (default)

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage,  $V_{ds}(T_j, I_{ds})$  — On-state voltage**

[0, 1.1, 1.3, 1.45, 1.75, 2.25, 2.7; 0, 1, 1.15, 1.35, 1.7, 2.35, 3] V (default)

Voltage drop across the device while it is in a triggered conductive state. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Temperature vector,  $T_j$  — Temperature vector**

[298.15, 398.15] K (default)

Temperature values at which the on-state voltage is specified. Specify this parameter using a vector quantity.



**Dependencies**

See the Main Parameter Dependencies table.

**Drain-source current vector, Ids — Drain-source current vector**

[ 0 10 50 100 200 400 600 ] A (default)

Drain-source currents for which the on-state voltage is defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Switching Losses**

To enable these parameters, set **Modeling option** to Show thermal port.

**Switch-on loss — Switch-on loss**

0.02286 J (default)

Energy dissipated during a single switch-on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-off loss — Switch-off loss**

0.01714 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Off-state voltage for switching loss data — Off-state voltage for losses data**

300 V (default)

The output voltage of the device during the off state. This is the blocking voltage at which the switch-on loss and switch-off loss data are defined.

**On-state current for switching loss data — Output current**

600 A (default)

Output currents for which the switch-on loss, switch-off loss, and on-state voltage are defined. The first element must be zero. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-on loss,  $E_{on}(T_j, I_{ds})$  — Switch-on loss**

[ 0 2.9e-4 0.00143 0.00286 0.00571 0.01314 0.02286; 0 5.7e-4 0.00263 0.00514 0.01029 0.02057 0.03029 ] J (default)

Energy dissipated during a single switch on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Switch-off loss,  $E_{off}(T_j, I_{ds})$  — Switch-off loss**

[0, .21, 1.07, 2.14, 4.29, 9.86, 17.14; 0, .43, 1.97, 3.86, 7.71, 15.43, 22.71] \* 1e-3 J (default)

Energy dissipated during a single switch-off event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Temperature vector for switching losses,  $T_j$  — Temperature vector for switching losses**

[298.15, 398.15] K (default)

Temperature values at which the switch-on loss and switch-off loss are specified. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Drain-source current vector for switching losses,  $I_{ds}$  — Drain-source current vector for switching losses**

[ 0 10 50 100 200 400 600 ] A (default)

Drain-source currents for which the switch-on loss and switch-off-loss are defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Tabulate with temperature and current.

**Integral Diode****Integral protection diode — Protection diode**

Protection diode with no dynamics (default) | None | Protection diode with charge dynamics

Block integral protection diode. The default value is Protection diode with no dynamics.

The diodes you can select are:

- Protection diode with no dynamics
- Protection diode with charge dynamics

### Diode model — Diode model

Piecewise Linear (default) | Tabulated I-V curve

Select one of these diode models:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448. This is the default method.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data plus fixed reverse bias off conductance.

### Dependencies

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

### Table type — Tabulated function

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

### Dependencies

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** is set to Tabulated I-V curve.

### Forward voltage — Forward voltage

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### Dependencies

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.
- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

### On resistance — On resistance

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

### Dependencies

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics.
- If the thermal port is exposed, set **Integral protection diode** to Protection diode with no dynamics or Protection diode with charge dynamics and **Diode model** to Piecewise linear.

**Forward currents,  $I_f(T_j, V_f)$  – Vector of forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12]  
A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

**Junction temperatures,  $T_j$  – Vector of junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures. This parameter must be a vector of at least two elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve.

**Forward voltages,  $V_f$  – Vector of forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

**Forward voltages,  $V_f(T_j, I_f)$  – Vector of forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

**Forward currents,  $I_f$  – Vector of forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** -50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

#### Reverse recovery time, $t_{rr}$ – Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery time directly.

#### Reverse recovery time stretch factor – Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify stretch factor.

#### Reverse recovery charge, $Q_{rr}$ – Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery charge.

**Diode turn-off voltage when measuring Erec — Diode turn-off voltage when measuring Erec**

-600 V (default)

Voltage between the diode in steady-state.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Stray inductance when measuring Erec — Stray inductance when measuring Erec**

150 nH (default)

Total unintended inductance in the measurement circuit. The block uses this value to calculate **Reverse recovery energy, Erec**.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Reverse recovery energy, Erec — Reverse recovery energy**

0.03 J (default)

Total switching losses due to the diode reverse recovery.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

For more information on these parameters, see Diode.

**Thermal Port**

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

**Version History****Introduced in R2013b****Update on switching losses and thermal modelling options**

*Behavior changed in R2020b*

From R2020b forward, the MOSFET (Ideal, Switching) block has improved losses and thermal modelling options.

As a result of these changes:

- Electrical and thermal on-state losses are now always identical. The **Thermal loss dependent on** parameter and its options, **Voltage** and **current** and **Voltage, current, and temperature**, have been renamed to **On-state behavior and switching losses**, **Specify constant values**, and **Tabulate with temperature and current**:
  - If you selected **Voltage** and **current** for **Thermal loss dependent on**, then the electrical on-state losses are unchanged and their values are determined using the on-state resistance. However, the thermal on-state losses are now also determined by the on-state resistance. Prior to R2020b, the thermal on-state losses were defined by the product of the **On-state voltage** and **Output current, Iout** parameters.
  - If you selected **Voltage, current, and temperature** for **Thermal loss dependent on**, then the thermal on-state losses are unchanged and the **On-state voltage, Vds(Tj,Ids)** parameter sets their values. However, the electrical on-state losses are now equal to the thermal on-state losses. Prior to R2020b, the electrical on-state losses were defined by the value of the on-state resistance.
- The **On-state voltage** parameter is no longer used.

### **Removal of energy dissipation time constant parameter**

*Behavior changed in R2021a*

From R2021a forward, the **Energy dissipation time constant** parameter of the MOSFET (Ideal, Switching) block is no longer used. A step in junction temperature now reflects the switching losses. If your model contains a thermal mass directly connected to this block thermal port, remove it and model the thermal mass inside the component itself.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Diode | GTO | Ideal Semiconductor Switch | IGBT (Ideal, Switching) | N-Channel MOSFET | P-Channel MOSFET | Thyristor (Piecewise Linear)

### **Topics**

“Simulating Thermal Effects in Semiconductors”

“Switch Between Physical Signal and Electrical Ports”



# Moving Average

Moving average-value computation

**Library:** Simscape / Electrical / Control / General Control



## Description

The Moving Average block computes the moving average value of the input signal. Use this block to filter higher frequency signal components and to smooth noisy signals.

## Equations

The moving average is computed based on a moving time window. The moving average for continuous-time is calculated as

$$\bar{u} = \frac{1}{T_0} \int_{t_0}^{t_0 + T_0} u(t) dt,$$

where:

- $u(t)$  is the input signal,
- $\frac{1}{f}$
- $f$  is the fundamental frequency of the signal.

The moving average for discrete-time is calculated as:

$$\bar{u}(k) = \frac{1}{T_0} \sum_{i=0}^{n-1} u(k-i).$$

---

**Note** If you use this block for continuous-time operations and set the **Sample time (-1 for inherited)** parameter to  $\emptyset$ , you should also specify the value of the **Buffer size** parameter to ensure it covers the moving time window.

---

## Assumptions

- The output is initialized with the initial condition in the time interval  $[\emptyset, T_\emptyset]$

## Ports

### Input

#### **u — Moving average input**

scalar | vector

Input signal.

Data Types: single | double

### Output

#### **Mean — Moving average output**

scalar | vector

Moving average of the input signal.

Data Types: single | double

## Parameters

#### **Fundamental frequency (Hz) — Signal fundamental frequency**

60 (default) | positive scalar | vector with positive values

Fundamental frequency of the signal, in Hz. If you specify the fundamental frequency using a vector, it must match the input vector dimension.

#### **Initial value — Initial input signal value**

0 (default) | scalar

The initial value of the input signal.

#### **Buffer size — Buffer size**

8192 (default) | scalar

Size of the buffer.

### Dependencies

To enable this parameter, set **Sample time (-1 for inherited)** to 0.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2018b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

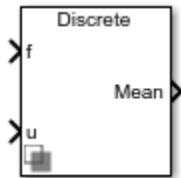
### **Blocks**

Fourier Analysis | Signal Sample and Hold

## Moving Average (Variable Frequency)

Variable frequency moving average-value computation

**Library:** Simscape / Electrical / Control / General Control



### Description

The Moving Average (Variable Frequency) block computes the moving average value of an input signal of variable frequency. Use this block to filter higher frequency signal components and to smooth noisy signals.

### Equations

The moving average is computed based on a moving time window. The moving average for continuous-time is calculated as

$$\bar{u} = \frac{1}{T_0} \int_{t_0}^{t_0 + T_0} u(t) dt,$$

where:

- $u(t)$  is the input signal.
- $T_0$  is equal to  $\frac{1}{f}$
- $f$  is the fundamental frequency of the signal.

The moving average for discrete-time is calculated as:

$$\bar{u}(k) = \frac{1}{T_0} \sum_{i=0}^{n-1} u(k-i).$$

---

**Note** If you use this block for continuous-time operations and set the **Sample time (-1 for inherited)** parameter to  $\emptyset$ , you should also specify the value of the **Buffer size** parameter to ensure it covers the moving time window.

---

### Assumptions and Limitations

The output is initialized with an initial condition in the time interval  $[\emptyset, T_\emptyset]$ .

## Ports

### Input

#### **u — Moving average input**

scalar | vector

Moving average input, specified as a scalar or vector.

Data Types: single | double

#### **f — Frequency, Hz**

scalar | vector

Frequency, specified as a scalar or vector in Hz.

Data Types: single | double

### Output

#### **Mean — Moving average output**

scalar | vector

Moving average of the input signal, returned as a scalar or vector.

Data Types: single | double

## Parameters

#### **Initial frequency (Hz) — Initial signal frequency**

60 (default) | positive scalar | vector with positive values

Initial frequency of the signal, in Hz. If you specify the initial frequency using a vector, it must match the dimensions of the input vector.

#### **Minimum frequency (Hz) — Minimum signal frequency**

30 (default) | positive scalar | vector with positive values

Minimum frequency of the signal, in Hz. If you specify the minimum frequency using a vector, it must match the dimensions of the input vector.

#### **Initial value — Initial input signal value**

0 (default) | scalar

Initial value of the input signal.

#### **Buffer size — Buffer size**

8192 (default) | scalar

Size of the buffer.

### Dependencies

To enable this parameter, set **Sample time (-1 for inherited)** to 0.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2020a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

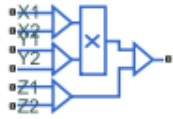
### **Blocks**

Fourier Analysis | Signal Sample and Hold | Moving Average

# Multiplier

Integrated circuit multiplier

**Library:** Simscape / Electrical / Integrated Circuits



## Description

The Multiplier block models an integrated circuit multiplier. The block implements the following equation, which defines the voltage applied to the output port:

$$V_{out} = A \left( \frac{(X_1 - X_2)(Y_1 - Y_2)}{K} - (Z_1 - Z_2) \right)$$

where  $X_1$ ,  $X_2$ ,  $Y_1$ ,  $Y_2$ ,  $Z_1$ ,  $Z_2$  are the voltages presented at the input ports,  $A$  is the gain, and  $K$  is the scale factor.

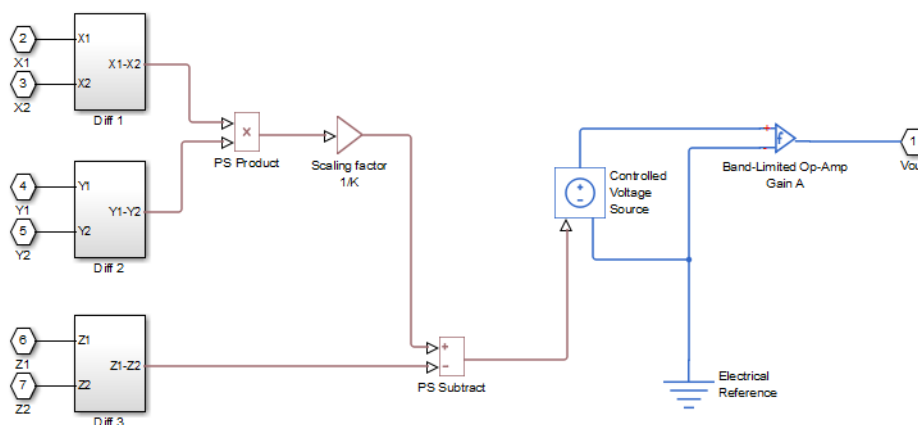
In a typical multiplication circuit, the output is fed back into input  $Z_1$ , which results in the following gain (assuming that  $A$  is large):

$$V_{out} = \left( \frac{(X_1 - X_2)(Y_1 - Y_2)}{K} + Z_2 \right)$$

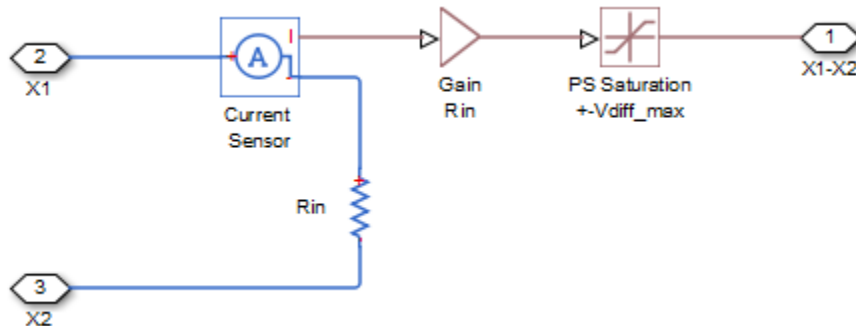
The value of the scale factor  $K$  is usually altered by an external resistor bias network. The Multiplier block implements  $K$  as an internal gain, and the external bias network is not necessary for system simulation. A typical value for  $K$  is 10, with a typical adjustment down to 3.

You can use the Multiplier block to implement a number of other functions, as well as multiplication. Examples include division, squares, and square roots. For example circuits, consult manufacturer datasheets.

The following figure shows the internal model structure of the Multiplier block. It includes the Band-Limited Op-Amp block to model finite bandwidth and slew-rate limiting.



The next figure shows one of the differential subsystem blocks. All three differential subsystem blocks are identical in structure.



## Assumptions and Limitations

- Only differential limiting of the inputs is implemented. You must ensure that the absolute values of the inputs you use keep the actual device operating in its linear region.
- Output current is such that the integrated circuit is operating in the linear I-V region, which can be approximated by a voltage source plus a series output resistance.
- Input offset voltage is not modeled, and the input voltage-current relationship is treated as linear within the differential signal voltage range.

## Ports

### Conserving

#### X1 – X-input non-inverting input

electrical

Electrical conserving port associated with the multiplier X1 non-inverting input.

#### X2 – X-input inverting input

electrical

Electrical conserving port associated with the multiplier X2 inverting input.

#### Y1 – Y-input non-inverting input

electrical

Electrical conserving port associated with the multiplier Y1 non-inverting input.

#### Y2 – Y-input inverting input

electrical

Electrical conserving port associated with the multiplier Y2 inverting input.

#### Z1 – Z-input non-inverting input

electrical

Electrical conserving port associated with the multiplier Z1 non-inverting input.



**Z2 – Z-input inverting input**

electrical

Electrical conserving port associated with the multiplier Z2 inverting input.

**~ – Multiplied output signal**

electrical

Electrical conserving port associated with the multiplier output port.

**Parameters****Main****Scaling factor, K – Scaling factor**

10 V (default)

The scaling factor  $K$  in the equation that defines output voltage. Datasheets sometimes refer to it as the scale factor, or SF.

**Gain, A – Gain**

3000 (default)

The gain of the internal operational amplifier, corresponding to the gain  $A$  in the equation that defines output voltage.

**Inputs****Differential resistance, Rin – Differential resistance**

10e6 Ohm (default)

Each of the differential inputs is approximated as a linear resistor with value  $R_{in}$ . Set this value to the datasheet value for differential resistance.

**Differential signal voltage range – Differential signal voltage range**

10 V (default)

This value,  $V_{diff\_max}$ , is used to limit the magnitude of each of the three differential input voltages. Set this value to the datasheet value for differential signal voltage range.

**Outputs****Output resistance, Rout – Output resistance**

0.1 Ohm (default)

The multiplier output stage is modeled as a voltage source plus series resistor inside the Band-Limited Op-Amp block. This parameter specifies the value of this series resistor.

**Minimum output, Vmin – Minimum output**

-11 V (default)

The lower limit of the output voltage.

**Maximum output, Vmax – Maximum output**

11 V (default)

The upper limit of the output voltage.

**Maximum slew rate, Vdot – Maximum slew rate**

20 V/us (default)

The maximum positive or negative rate of change of output voltage magnitude.

**Bandwidth, f – Bandwidth**

1 MHz (default)

The bandwidth of the Band-Limited Op-Amp block.

**Initial output voltage, V0 – Initial output voltage**

0 V (default)

The value of the initial Multiplier block output if the **Start simulation from steady-state** option is not selected in the Solver block.

## Version History

Introduced in R2010b

## Extended Capabilities

**C/C++ Code Generation**

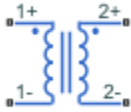
Generate C and C++ code using Simulink® Coder™.

## See Also

# Mutual Inductor

Mutual inductor model with nominal inductance optional tolerances for each winding, operating limits and faults

**Library:** Simscape / Electrical / Passive / Transformers



## Description

The Mutual Inductor block lets you model a mutual inductor (two-winding transformer) with nominal inductance tolerances for each winding. The model includes the following effects:

- “Tolerances” on page 1-1021
- “Operating Limits” on page 1-1022
- “Faults” on page 1-1022

You can turn these modeling options on and off independently of each other.

In the unfaulted state, the following equations describe the Mutual Inductor block behavior:

$$v_1 = L_1 \frac{di_{L1}}{dt} + M \frac{di_{L2}}{dt} + i_{L1} R_1$$

$$v_2 = L_2 \frac{di_{L2}}{dt} + M \frac{di_{L1}}{dt} + i_{L2} R_2$$

$$M = k \sqrt{L_1 L_2}$$

where:

- $v_1$  and  $v_2$  are voltages across the primary and secondary winding, respectively.
- $L_1$  and  $L_2$  are inductances of the primary and secondary winding.
- $R_1$  and  $R_2$  are series resistances of the primary and secondary winding.
- $M$  is mutual inductance.
- $k$  is coefficient of coupling. To reverse one of the winding directions, use a negative value.
- $t$  is time.

A parallel conductance is placed across the + and - terminals of the primary and secondary windings, so that  $i_{L1} = i_1 - G_1 v_1$ , where  $G_1$  is the parallel conductance of the primary winding, and  $i_1$  is the terminal current into the primary. Similar definitions and equation apply to  $i_{L2}$ .

## Tolerances

You can apply tolerances separately for each winding. Datasheets typically provide a tolerance percentage for a given inductor type. Therefore, this value is the same for both windings. The table

shows how the block applies tolerances to the nominal inductance value and calculates inductance based on the selected tolerance application option for the winding, **L1 tolerance application** or **L2 tolerance application**.

Option	Inductance Value
None – use nominal value	$L$
Random tolerance	Uniform distribution: $L \cdot (1 - tol + 2 \cdot tol \cdot rand)$ Gaussian distribution: $L \cdot (1 + tol \cdot randn / nSigma)$
Apply maximum tolerance value	$L \cdot (1 + tol)$
Apply minimum tolerance value	$L \cdot (1 - tol)$

In the table:

- $L$  is nominal inductance for the primary or secondary winding, **Inductance L1** or **Inductance L2** parameter value.
- $tol$  is fractional tolerance, **Tolerance (%)** /100.
- $nSigma$  is the value you provide for the **Number of standard deviations for quoted tolerance** parameter.
- $rand$  and  $randn$  are standard MATLAB functions for generating uniform and normal distribution random numbers.

**Note** If you choose the Random tolerance option and you are in "Fast Restart" mode, the random tolerance value is updated on every simulation if at least one between the fractional tolerance,  $tol$ , or the **Number of standard deviations for quoted tolerance**,  $nSigma$ , is set to Run-time and is defined with a variable (even if you do not modify that variable).

### Operating Limits

Inductors are typically rated with a particular saturation current, and possibly with a maximum allowable power dissipation. You can specify operating limits in terms of these values, to generate warnings or errors if the inductor is driven outside its specification.

When an operating limit is exceeded, the block can either generate a warning or stop the simulation with an error. For more information, see the "Operating Limits" on page 1-1026 parameters section.

### Faults

Instantaneous changes in inductor parameters are unphysical. Therefore, when the Mutual Inductor block enters the faulted state, short-circuit and open-circuit voltages transition to their faulted values over a period of time based on this formula:

$$CurrentValue = FaultedValue - (FaultedValue - UnfaultedValue) \cdot \operatorname{sech}(\Delta t / \tau)$$

where:

- $\Delta t$  is time since the onset of the fault condition.
- $\tau$  is user-defined time constant associated with the fault transition.

For short-circuit faults, the conductance of the short-circuit path also changes according to the  $\text{sech}(\Delta t / \tau)$  function from a small value (representing an open-circuit path) to a large value.

The Mutual Inductor block lets you select whether the faults occur in the primary or secondary winding. The block models the faulted winding as a faulted inductor. The unfaulted winding is coupled to the faulted winding. As a result, the actual equations involve a total of three coupled windings: two for the faulted winding and one for the unfaulted winding. The coupling between the primary and secondary windings is defined by the **Coefficient of coupling** parameter.

The block can trigger the start of fault transition:

- At a specific time
- After voltage exceeds the maximum permissible value a certain number of times
- When current exceeds the maximum permissible value for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault transition takes precedence. In other words, a component fails no more than once per simulation.

You can also choose whether to issue an assertion when a fault occurs by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

Faultable inductors often require that you use the fixed-step local solver, rather than the variable-step solver. In particular, if you model transitions to a faulted state that include short circuits, MathWorks recommends that you use the fixed-step local solver. For more information, see “Making Optimal Solver Choices for Physical Simulation”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The **Primary current** and **Secondary current** variables let you specify a high-priority target for the initial inductor current in the respective winding at the start of simulation.

## Ports

### Conserving

#### 1+ — Positive terminal of the primary winding

electrical

Electrical conserving port associated with the primary winding positive terminal.

#### 1- — Negative terminal of the primary winding

electrical

Electrical conserving port associated with the primary winding negative terminal.

**2+ — Positive terminal of the secondary winding**

electrical

Electrical conserving port associated with the secondary winding positive terminal.

**2- — Negative terminal of the secondary winding**

electrical

Electrical conserving port associated with the secondary winding negative terminal.

## Parameters

### Main

**Inductance L1 — Nominal inductance value in the primary winding**

10 H (default)

The nominal inductance value in the primary winding. Inductance value must be greater than zero.

**Inductance L2 — Nominal inductance value in the secondary winding**

0.1 H (default)

The nominal inductance value in the secondary winding. Inductance value must be greater than zero.

**Coefficient of coupling — Mutual inductance coupling between windings**

0.9 (default)

The coupling between the primary and secondary windings. This coefficient defines the mutual inductance. To reverse one of the winding directions, use a negative value.

**Tolerance (%) — Inductor tolerance, in percent**

20 (default)

The inductor tolerance as defined on the manufacturer datasheet. Datasheets typically provide a tolerance percentage for a given inductor type. Therefore, this value is the same for both windings.

**L1 tolerance application — Select how to apply tolerance to primary winding**

None — use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation to the primary winding:

- **None — use nominal value** — The block does not apply tolerance, it uses the nominal inductance value.
- **Random tolerance** — The block applies random offset to the inductance value, within the tolerance value limit. You can choose Uniform or Gaussian distribution for calculating the random number by using the **Tolerance distribution** parameter.
- **Apply maximum tolerance value** — The inductance is increased by the specified tolerance percent value.
- **Apply minimum tolerance value** — The inductance is decreased by the specified tolerance percent value.

**L1 tolerance distribution — Select the distribution type for primary winding**

Uniform (default) | Gaussian

Select the distribution type for random tolerance:

- Uniform — Uniform distribution
- Gaussian — Gaussian distribution

**Dependencies**

Enabled when the **L1 tolerance application** parameter is set to Random tolerance.

**L1 number of standard deviations for quoted tolerance — Used for calculating the Gaussian random number for primary winding**

4 (default)

Number of standard deviations for calculating the Gaussian random number.

**Dependencies**

Enabled when the **L1 tolerance distribution** parameter is set to Gaussian.

**L2 tolerance application — Select how to apply tolerance to secondary winding**

None — use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation to the secondary winding:

- None — use nominal value — The block does not apply tolerance, it uses the nominal inductance value.
- Random tolerance — The block applies random offset to the inductance value, within the tolerance value limit. You can choose Uniform or Gaussian distribution for calculating the random number by using the **Tolerance distribution** parameter.
- Apply maximum tolerance value — The inductance is increased by the specified tolerance percent value.
- Apply minimum tolerance value — The inductance is decreased by the specified tolerance percent value.

**L2 tolerance distribution — Select the distribution type for secondary winding**

Uniform (default) | Gaussian

Select the distribution type for random tolerance:

- Uniform — Uniform distribution
- Gaussian — Gaussian distribution

**Dependencies**

Enabled when the **L2 tolerance application** parameter is set to Random tolerance.

**L2 number of standard deviations for quoted tolerance — Used for calculating the Gaussian random number for secondary winding**

4 (default)

Number of standard deviations for calculating the Gaussian random number.

**Dependencies**

Enabled when the **L2 tolerance distribution** parameter is set to Gaussian.

**Resistance****Series resistance, [R\_primary R\_secondary] – Equivalent series resistance of the primary and secondary winding**

[0.001, 0.001] Ohm (default)

Equivalent series resistance of the primary and secondary winding, specified as a two-element vector. The first number corresponds to the primary winding, the second number to the secondary winding. For a faulted winding, the block allocates the resistance to each segment in proportion to the number of turns in that segment.

**Parallel conductance, [G\_primary G\_secondary] – Parallel leakage path associated with the primary and secondary winding**

[1e-9, 1e-9] 1/Ohm (default)

Parallel leakage path associated with the primary and secondary winding, specified as a two-element vector. The first number corresponds to the primary winding, the second number to the secondary winding. The parallel conductances are placed directly across the + and - terminals of the primary and secondary winding, respectively.

**Operating Limits****Enable operating limits – Select Yes to enable reporting when the operational limits are exceeded**

No (default) | Yes

Select Yes to enable reporting when the operational limits are exceeded. The associated parameters in the **Operating Limits** section become visible to let you select the reporting method and specify the operating limits in terms of power and current.

**Reporting when operating limits exceeded – Select the reporting method**

Warn (default) | Error

Select what happens when an operating limit is exceeded:

- Warn – The block issues a warning.
- Error – Simulation stops with an error.

**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Saturation current – Inductor saturation current**

1 A (default)

Inductor saturation current, as defined in the manufacturer datasheets. If the net current into the primary and secondary windings exceeds this value, the core material enters saturation, and the block reports an operating limits violation. That is, the block compares the limit against  $|i_1 + i_2|$ , where currents are defined as being positive when they are into the + nodes.



**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Power rating — Maximum power dissipation in the inductor**

1 A (default)

Maximum instantaneous (total) power dissipation in the resistance and conductance elements associated with the mutual inductor. If the total power (including both primary and secondary winding) exceeds this number, the block reports an operating limits violation.

**Dependencies**

Enabled when the **Enable operating limits** parameter is set to Yes.

**Faults****Enable faults — Select On to enable faults modeling**

Off (default) | On

Select On to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

**Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs**

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

**Dependencies**

Enabled when the **Enable faults** parameter is set to On.

**Faulted winding — Select winding to use for fault modeling**

Primary (default) | Secondary

Select whether the faults occur in the primary or secondary winding.

**Dependencies**

Enabled when the **Enable faults** parameter is set to On.

**Location of fault node (% of total turns from - terminal) — Percentage of turns in the subinductor that is in contact with the - port of the faulted winding**

50 (default)

In practice, faults are enabled by segmenting the faulted winding into two coupled subinductors, connected in a series. The inductance is proportional to the square of the number of turns in the respective segment, and the series resistance of each subinductor is proportional to the number of turns in each segment. The parallel conductance spans both segments.

This parameter indicates the percentage of turns that are assigned to the subinductor that is in contact with the - port of the faulted winding. The remaining turns are assigned to the other subinductor. The default value is 50, which means that the overall inductance of the faulted winding is divided into two equal, coupled subinductors.

**Dependencies**

Enabled when the **Enable faults** parameter is set to On.

**Short-circuit turns — Select whether fault results in one of the segments being short-circuited**

No (default) | To negative terminal | To positive terminal

Select whether the fault results in one of the subinductor segments being short-circuited:

- No — The fault does not produce a short circuit.
- To negative terminal — The fault short-circuits the subinductor that is in contact with the - port of the block.
- To positive terminal — The fault short-circuits the subinductor that is in contact with the + port of the block.

**Dependencies**

Enabled when the **Enable faults** parameter is set to On.

**Open-circuit at fault node — Select whether to apply an open-circuit fault between the segments**

No (default) | Yes

Select whether to apply an open-circuit fault between the two subinductor segments. The default is No. Even with an open-circuit fault, the characteristics of the subinductors are still related because they are magnetically coupled even in the faulted state.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Ground fault — Select whether fault results in one of the segments being short-circuited**

No (default) | Negative terminal side of fault node | Positive terminal side of fault node

Select whether, in case of fault, there is a path for current to flow towards the ground node:

- No — The fault does not result in a connection to ground.
- Negative terminal side of fault node — The side that is in contact with the - port of the block is connected to ground.
- Positive terminal side of fault node — The side that is in contact with the + port of the block is connected to ground.

If the **Open-circuit at fault node** parameter is set to Yes, you need to specify which side (negative or positive) is connected to ground. If there is no open circuit, the two options behave similarly. Physically, this corresponds to a breakdown in the insulation between the windings and the grounded core or chassis.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Conductance of faulted ground path — Mutual coupling between the two subinductors**

0 1/0hm (default)

If there is a ground fault, this parameter represents conductance of the current path to ground. For example, if the path to ground is through the core material, then specify a small conductance value depending on the core material being used. For highly conductive core material or for chassis-shorts, specify a higher conductance value.

**Dependencies**

Enabled when the **Ground fault** parameter is set to Negative terminal side of fault node or Positive terminal side of fault node.

**Fault transition time constant — Time constant for the transition to faulted state**

1e-3 s (default)

Time constant associated with the transition to the faulted state, as described in “Faults” on page 1-1022.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Enable temporal fault trigger — Select Yes to enable time-based fault triggering**

No (default) | Yes

Select Yes to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Simulation time for fault event — Time before entering faulted state**

1 s (default)

Set the simulation time at which you want the block to start entering the fault state.

**Dependencies**

Enabled when the **Enable temporal fault trigger** parameter is set to Yes.

**Enable behavioral fault trigger — Select Yes to enable behavioral fault triggering**

No (default) | Yes

Select Yes to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Maximum permissible voltage — Voltage threshold to fault transition**

100 V (default)

Define the voltage threshold to a fault transition. If the voltage value exceeds this threshold a certain number of times, specified by the **Number of events to fail when exceeding voltage** parameter value, then the block starts entering the fault state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Number of events to fail when exceeding voltage — Maximum number of times the voltage exceeds the threshold**

1 (default)

Because the physical mechanism underlying voltage-based failures depends on one or more partial discharge events occurring, this parameter allows you to set the number of voltage overshoots that the inductor can withstand before the fault transition begins. Note that the block does not check the time spent in the overvoltage condition, only the number of transitions.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Maximum permissible current — Current threshold to fault transition**

1 A (default)

Define the current threshold to a fault transition. If the current value exceeds this threshold for longer than the **Time to fail when exceeding current** parameter value, then the block starts entering the fault state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Time to fail when exceeding current — Maximum length of time the current exceeds the threshold**

1 s (default)

Set the maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

## Version History

Introduced in R2017a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Fault | Inductor | Three-Winding Mutual Inductor | Variable Inductor

# N-Channel IGBT

N-Channel insulated gate bipolar transistor

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The N-Channel IGBT block models an Insulated Gate Bipolar Transistor (IGBT). The block provides two main modeling options, by setting the **Modeling option** parameter to either:

- **Full I-V and capacitance characteristics** — This modeling option is a detailed component model suitable for simulating detailed switching characteristics and predicting component losses. This modeling option, in turn, provides two ways of modeling an IGBT:
  - As an equivalent circuit based on a PNP bipolar transistor and N-channel MOSFET. For more information on using this model, see “Representation by Equivalent Circuit” on page 1-1032, “Fine-Tuning the Current-Voltage Characteristics” on page 1-1037, and “Modeling Temperature Dependence” on page 1-1037.
  - By a 2D lookup table approximation to the I-V (current-voltage) curve. For details, see “Representation by 2D Lookup Table” on page 1-1033.
  - By a 3D lookup table approximation to the I-V (current-voltage) curve that includes temperature data. For details, see “Representation by 3D Lookup Table” on page 1-1034.

The gate capacitance in the detailed model is represented as a fixed gate-emitter capacitance  $C_{GE}$  and either a fixed or a nonlinear gate-collector capacitance  $C_{GC}$ . For details, see “Charge Model” on page 1-1034.

- **Simplified I-V characteristics and event-based timing** — This modeling option models the IGBT more simply by using just the on-state I-V data as a function of the collector-emitter voltage. In the off state (gate-emitter voltage less than **Threshold voltage,  $V_{th}$** ), the IGBT is modeled by a constant **Off-state conductance**. This simplified model is suitable when approximate dynamic characteristics are sufficient, and simulation speed is of paramount importance. For details, see “Event-Based IGBT Modeling Option” on page 1-1038.

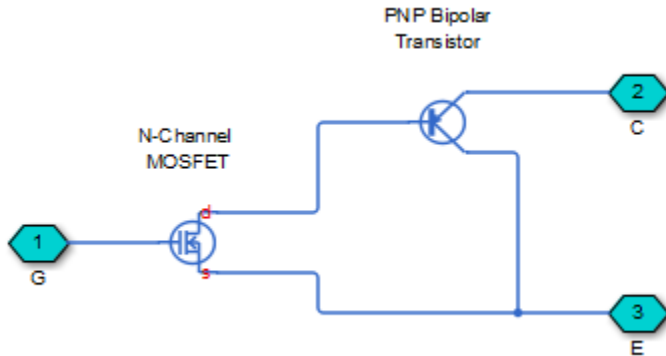
Together with the thermal port modeling options (see “Thermal Port” on page 1-1039), the block therefore provides you with four choices. To select the desired modeling option, set the **Modeling option** parameter to either:

- 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.

### Representation by Equivalent Circuit

The equivalent circuit of the detailed block modeling option consists of a PNP Bipolar Transistor block driven by an N-Channel MOSFET block, as shown in the following figure:



The MOSFET source is connected to the bipolar transistor collector, and the MOSFET drain is connected to the bipolar transistor base. The MOSFET uses the threshold-based equations shown in the N-Channel MOSFET block reference page. The bipolar transistor uses the equations shown in the PNP Bipolar Transistor block reference page, but with the addition of an emission coefficient parameter  $N$  that scales  $kT/q$ .

The N-Channel IGBT block uses the on and off characteristics you specify in the block dialog box to estimate the parameter values for the underlying N-Channel MOSFET and PNP bipolar transistor.

The block uses the off characteristics to calculate the base-emitter voltage,  $V_{be}$ , and the saturation current,  $I_S$ .

When the transistor is off, the gate-emitter voltage is zero and the IGBT base-collector voltage is large, so the PNP base and collector current equations simplify to:

$$I_b = 0 = -I_S \left[ \frac{1}{\beta_F} (e^{-qV_{be}/(NkT)} - 1) - \frac{1}{\beta_R} \right]$$

$$I_c = -I_S \left[ e^{-qV_{be}/(NkT)} \left( 1 + \frac{V_{bc}}{V_{AF}} \right) + \frac{1}{\beta_R} \right]$$

where  $N$  is the **Emission coefficient**,  $N$  parameter value,  $V_{AF}$  is the forward Early voltage, and  $I_c$  and  $I_b$  are defined as positive flowing into the collector and base, respectively. See the PNP Bipolar Transistor reference page for definitions of the remaining variables. The first equation can be solved for  $V_{be}$ .

The base current is zero in the off-condition, and hence  $I_c = -I_{ces}$ , where  $I_{ces}$  is the Zero gate voltage collector current. The base-collector voltage,  $V_{bc}$ , is given by  $V_{bc} = V_{ces} + V_{ces}$ , where  $V_{ces}$  is the voltage at which  $I_{ces}$  is measured. Hence we can rewrite the second equation as follows:

$$I_{ces} = I_S \left[ e^{-qV_{be}/(NkT)} \left( 1 + \frac{V_{ces} + V_{be}}{V_{AF}} \right) + \frac{1}{\beta_R} \right]$$

The block sets  $\beta_R$  and  $\beta_F$  to typical values of 1 and 50, so these two equations can be used to solve for  $V_{be}$  and  $I_S$ :

$$V_{be} = \frac{-NkT}{q} \log\left(1 + \frac{\beta_F}{\beta_R}\right)$$

$$I_S = \frac{I_C}{e^{-qV_{be}/(NkT)} + \frac{1}{\beta_R}}$$

---

**Note** The block does not require an exact value for  $\beta_F$  because it can adjust the MOSFET gain  $K$  to ensure the overall device gain is correct.

---

The block parameters **Collector-emitter saturation voltage, Vce(sat)** and **Collector current at which Vce(sat) is defined** are used to determine  $V_{be(sat)}$  by solving the following equation:

$$I_{ce(sat)} = I_S \left[ e^{-qV_{be(sat)}/(NkT)} \left( 1 + \frac{V_{ce(sat)} + V_{be(sat)}}{V_{AF}} \right) + \frac{1}{\beta_R} \right]$$

Given this value, the block calculates the MOSFET gain,  $K$ , using the following equation:

$$I_{tf} = TF \frac{dI_C}{dt}$$

$$I_{ds} = I_b - I_{tf} = K \left[ (V_{GE(sat)} - V_{th})V_{ds} - \frac{V_{ds}^2}{2} \right]$$

where  $V_{th}$  is the **Gate-emitter threshold voltage, Vge(th)** parameter value,  $V_{GE(sat)}$  is the **Gate-emitter voltage at which Vce(sat) is defined** parameter value, and  $TF$  is the **Total forward transit time** parameter value.

$V_{ds}$  is related to the transistor voltages as  $V_{ds} = V_{ce} - V_{be}$ . The block substitutes this relationship for  $V_{ds}$ , sets the base-emitter voltage and base current to their saturated values, and rearranges the MOSFET equation to give

$$K = \frac{I_{b(sat)}}{\left[ (V_{GE(sat)} - V_{th})(V_{be(sat)} + V_{ce(sat)}) - \frac{(V_{be(sat)} + V_{ce(sat)})^2}{2} \right]}$$

where  $V_{ce(sat)}$  is the **Collector-emitter saturation voltage, Vce(sat)** parameter value.

These calculations ensure the zero gate voltage collector current and collector-emitter saturation voltage are exactly met at these two specified conditions. However, the current-voltage plots are very sensitive to the emission coefficient  $N$  and the precise value of  $V_{th}$ . If the manufacturer datasheet gives current-voltage plots for different  $V_{GE}$  values, then the  $N$  and  $V_{th}$  can be tuned by hand to improve the match.

### Representation by 2D Lookup Table

For the lookup table representation of the detailed block modeling option, you provide tabulated values for collector current as a function of gate-emitter voltage and collector-emitter voltage. The main advantage of using this option is simulation speed. It also lets you parameterize the device from either measured data or from data obtained from another simulation environment. To generate your

own data from the equivalent circuit representation, you can use a test harness, such as shown in the “IGBT Characteristics” example.

The lookup table representation combines all of the equivalent circuit components (PNP transistor, N-channel MOSFET, collector resistor and emitter resistor) into one equivalent lookup table.

### Representation by 3D Lookup Table

For the temperature-dependent lookup table representation of the detailed block modeling option, you provide tabulated values for collector current as a function of gate-emitter voltage, collector-emitter voltage, and temperature.

The lookup table representation combines all of the equivalent circuit components (PNP transistor, N-channel MOSFET, collector resistor and emitter resistor) into one equivalent lookup table.

If the block thermal port is not exposed, then the **Device simulation temperature** parameter on the **Temperature Dependence** setting lets you specify the simulation temperature.

### Charge Model

The detailed modeling option of the block models capacitances either by fixed capacitance values, or by tabulated values as a function of the collector-emitter voltage. In either case, you can either directly specify the gate-emitter and gate-collector capacitance values, or let the block derive them from the input and reverse transfer capacitance values. Therefore, the **Parameterization** options for charge model on the **Junction Capacitance** setting are:

- Specify fixed input, reverse transfer and output capacitance — Provide fixed parameter values from datasheet and let the block convert the input and reverse transfer capacitance values to capacitance values, as described below. This is the default method.
- Specify fixed gate-emitter, gate-collector and collector-emitter capacitance — Provide fixed values for capacitance parameters directly.
- Specify tabulated input, reverse transfer and output capacitance — Provide tabulated capacitance and collector-emitter voltage values based on datasheet plots. The block converts the input and reverse transfer capacitance values to capacitance values, as described below.
- Specify tabulated gate-emitter, gate-collector and collector-emitter capacitance — Provide tabulated values for capacitances and collector-emitter voltage.

Use one of the tabulated capacitance options (Specify tabulated input, reverse transfer and output capacitance or Specify tabulated gate-emitter, gate-collector and collector-emitter capacitance) when the datasheet provides a plot of capacitances as a function of collector-emitter voltage. Using tabulated capacitance values will give more accurate dynamic characteristics, and avoids the need to iteratively tune parameters to fit the dynamics.

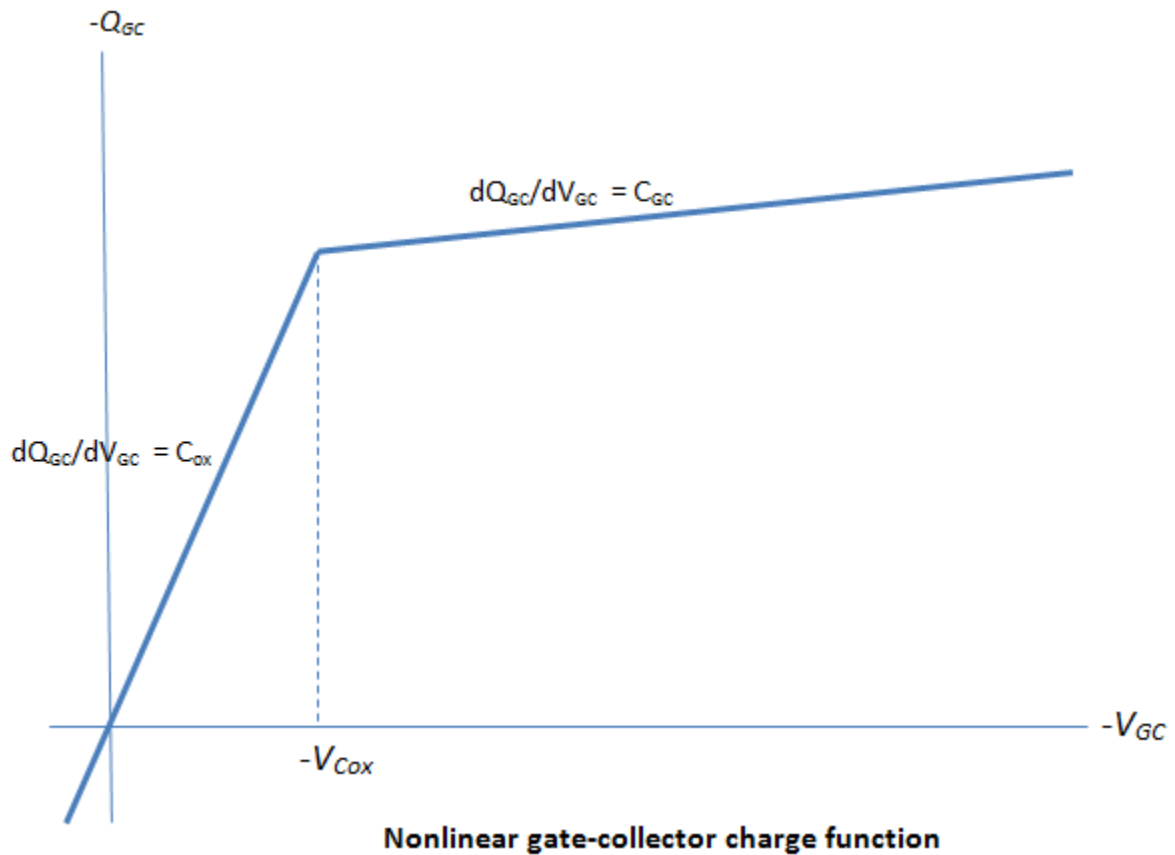
If you use the Specify fixed gate-emitter, gate-collector and collector-emitter capacitance or Specify tabulated gate-emitter, gate-collector and collector-emitter capacitance option, the **Junction Capacitance** setting lets you specify the **Gate-emitter capacitance**, **Gate-collector capacitance**, and **Collector-emitter capacitance** parameter values (fixed or tabulated) directly. Otherwise, the block derives them from the **Input capacitance**, **Cies**, **Reverse transfer capacitance**, **Cres**, and **Output capacitance**, **Coes** parameter values. These two parameterization methods are related as follows:

- $C_{GC} = C_{res}$

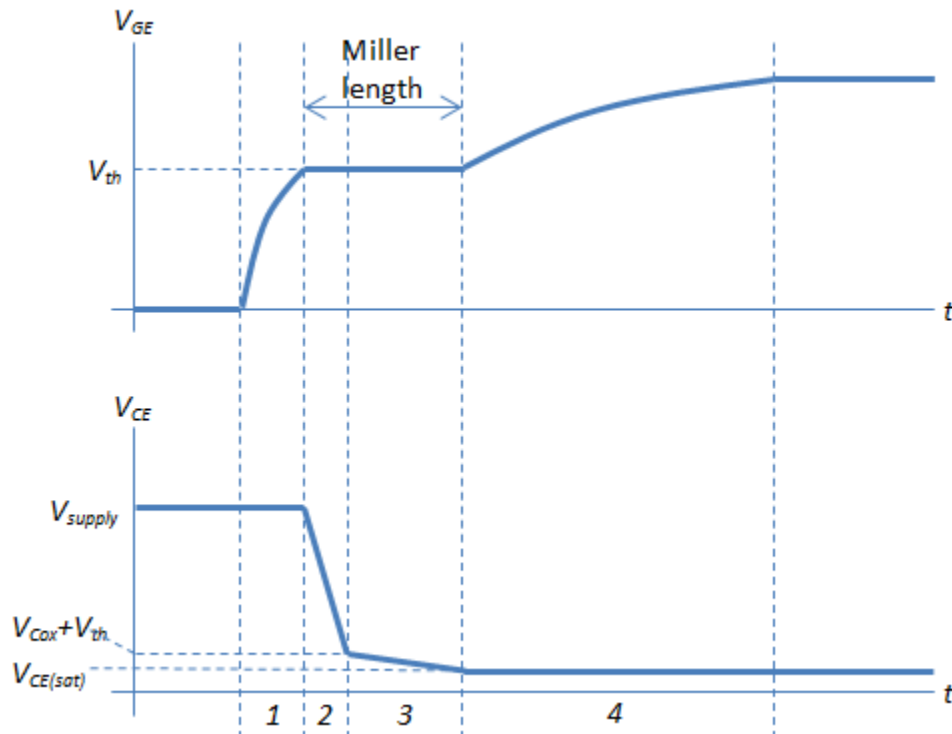


- $C_{GE} = C_{ies} - C_{res}$
- $C_{CE} = C_{oes} - C_{res}$

The two fixed capacitance options (Specify fixed input, reverse transfer and output capacitance or Specify fixed gate-emitter, gate-collector and collector-emitter capacitance) let you model gate capacitance as a fixed gate-emitter capacitance  $C_{GE}$  and either a fixed or a nonlinear gate-collector capacitance  $C_{GC}$ . If you select the Gate-collector charge function is nonlinear option for the **Charge-voltage linearity** parameter, then the gate-collector charge relationship is defined by the piecewise-linear function shown in the following figure.



With this nonlinear capacitance, the gate-emitter and collector-emitter voltage profiles take the form shown in the next figure, where the collector-emitter voltage fall has two regions (labeled 2 and 3) and the gate-emitter voltage has two time-constants (before and after the threshold voltage  $V_{th}$ ):



You can determine the capacitor values for  $C_{ies}$ ,  $C_{res}$ , and  $C_{ox}$  as follows, assuming that the IGBT gate is driven through an external resistance  $R_G$ :

- 1 Set  $C_{ies}$  to get correct time-constant for  $V_{GE}$  in Region 1. The time constant is defined by the product of  $C_{ies}$  and  $R_G$ . Alternatively, you can use a datasheet value for  $C_{ies}$ .
- 2 Set  $C_{res}$  so as to achieve the correct  $V_{CE}$  gradient in Region 2. The gradient is given by  $(V_{GE} - V_{th}) / (C_{res} \cdot R_G)$ .
- 3 Set  $V_{Cox}$  to the voltage at which the  $V_{CE}$  gradient changes minus the threshold voltage  $V_{th}$ .
- 4 Set  $C_{ox}$  to get correct Miller length and time constant in Region 4.

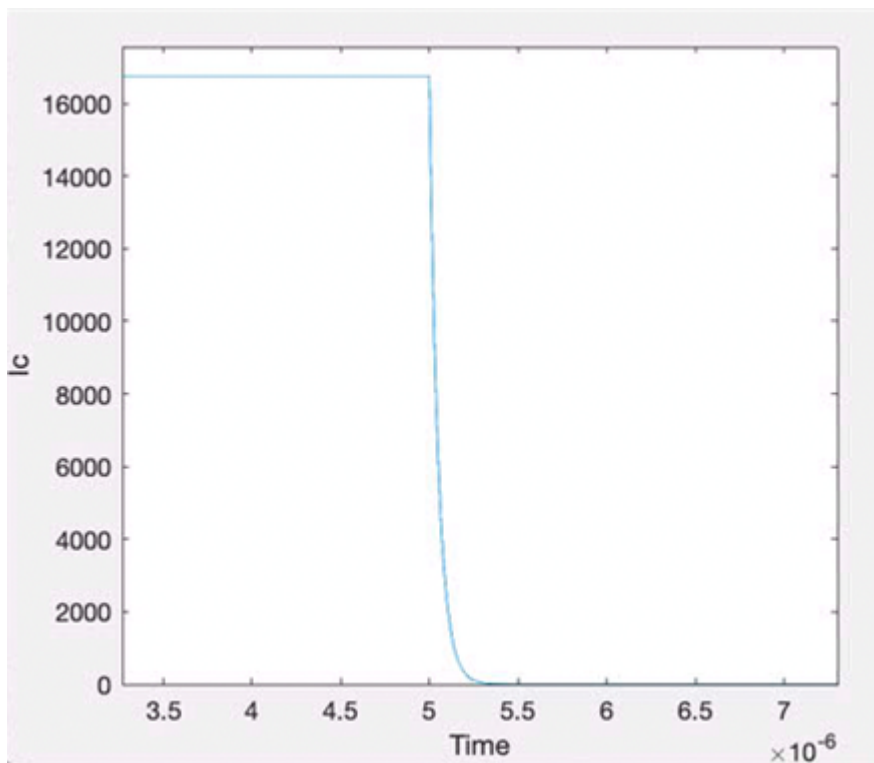
Because the underlying model is a simplification of an actual charge distribution, some iteration of these four steps may be required to get a best overall fit to measured data.

The collector current tail when the IGBT is turned off is determined by the **Total forward transit time** parameter if you set the **I-V characteristics defined by** parameter to Fundamental nonlinear equation.

If you set the **I-V characteristics defined by** parameter to Lookup table (2D, temperature independent) or Lookup table (3D, temperature dependent), then the collector current tail when the IGBT is turned off is determined by the **Turn-off tail current transit time** parameter:

$$i_c = i_{off} + (i_{on} - i_{off}) \exp\left(-\frac{t - t_{off}}{TT}\right)$$

Where  $i_{off}$  is the IGBT turn-off current,  $i_{on}$  is the IGBT turn-on current,  $t_{off}$  is the turn-off time, and  $TT$  is the value of the **Turn-off tail current transit time** parameter.



**Note** Because this block implementation includes a charge model, you must model the impedance of the circuit driving the gate to obtain representative turn-on and turn-off dynamics. Therefore, if you are simplifying the gate drive circuit by representing it as a controlled voltage source, you must include a suitable series resistor between the voltage source and the gate.

### Fine-Tuning the Current-Voltage Characteristics

For the equivalent circuit representation of the detailed model, use the parameters on the **Advanced** setting to fine-tune the current-voltage characteristics of the modeled device. To use these additional parameters effectively, you will need a manufacturer datasheet that provides plots of the collector current versus collector-emitter voltage for different values of gate-emitter voltage. The parameters on the **Advanced** setting have the following effects:

- The **Emission coefficient, N** parameter controls the shape of the current-voltage curves around the origin.
- The **Collector resistance, RC** and **Emitter resistance, RE** parameters affect the slope of the current-voltage curve at higher currents, and when fully turned on by a high gate-emitter voltage.
- The **Forward Early voltage, VAF** parameter affects the shape of the current-voltage curves for gate-emitter voltages around the **Gate-emitter threshold voltage, Vge(th)**.

### Modeling Temperature Dependence

For the 2D lookup table representation, the electrical equations do not depend on temperature. However, you can model temperature dependence by either using the 3D lookup table representation, or by using the equivalent circuit representation of the detailed model.

For the equivalent circuit representation, temperature dependence is modeled by the temperature dependence of the constituent components. See the N-Channel MOSFET and PNP Bipolar Transistor block reference pages for further information on the defining equations.

Some datasheets do not provide information on the zero gate voltage collector current,  $I_{ces}$ , at a higher measurement temperature. In this case, you can alternatively specify the energy gap,  $EG$ , for the device, using a typical value for the semiconductor type. For silicon, the energy gap is usually 1.11 eV.

### **Event-Based IGBT Modeling Option**

This implementation has much simpler equations than that with full I-V and capacitance characteristics. Use the event-based modeling option when the focus of the analysis is to understand overall circuit behavior rather than to verify the precise IGBT timing or losses characteristics.

The device is always in one of the following four states:

- Off
- Turning on
- On
- Turning off

In the off state, the relationship between collector current ( $i_c$ ) and collector-emitter voltage ( $v_{ce}$ ) is

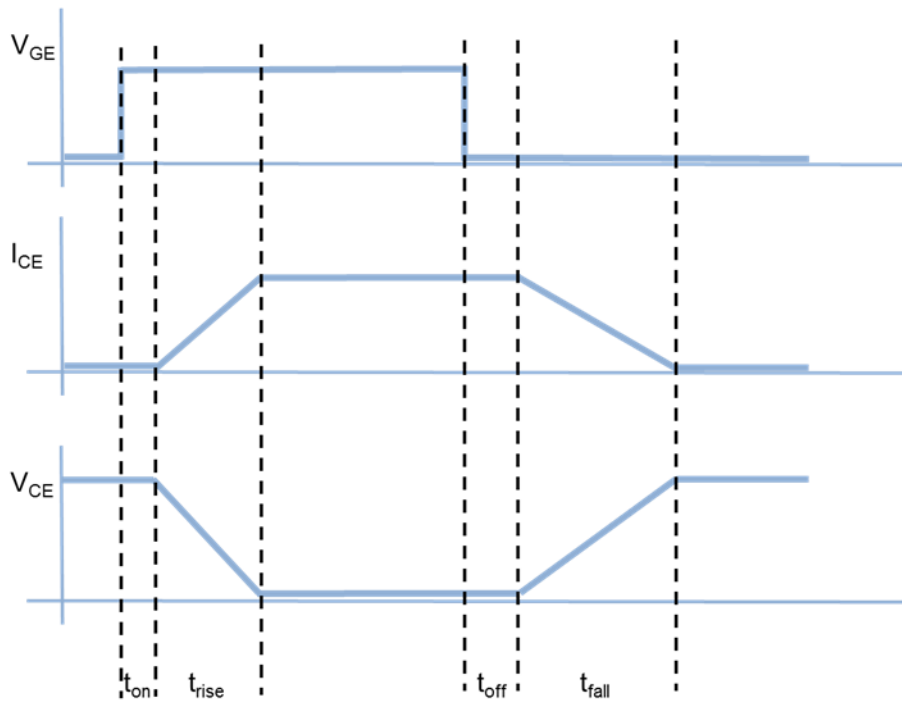
$$i_c = G_{off}v_{ce} \quad (1-44)$$

In the on state, the relationship between collector current ( $i_c$ ) and collector-emitter voltage ( $v_{ce}$ ) is

$$v_{ce} = \text{tablelookup}(i_c) \quad (1-45)$$

When turning on, the collector-emitter voltage is ramped down to zero over the rise time, the device moving into the on state when the voltage falls below the tabulated on-state value. Similarly when turning off, the collector-emitter voltage is ramped up over the (current) fall time to the specified blocking voltage value.

The following figure shows the resulting voltage and current profiles when driving a resistive load.



### Thermal Port

The block has an optional thermal port, hidden by default. To expose the thermal port, set the **Modeling option** parameter to:

- Full I-V and capacitance characteristics | Show thermal port — Detailed model. This action displays the thermal port **H** on the block icon, and exposes the **Thermal Port** parameters.
- Simplified I-V characteristics and event-based timing | Show thermal port — Simplified event-based model. This action displays the thermal port **H** on the block icon, exposes **Thermal Port** parameters and additional **Main** parameters. To simulate thermal effects, you must provide additional tabulated data for turn-on and turn-off losses and define the collector-emitter on-state voltage as a function of both current and temperature.

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

This table shows the relationship between the capacitances of the block and the initial targets:

Defined Capacitance	Initial Targets
Gate-emitter capacitance, $C_{ge}$	Set the initial target for the gate-emitter capacitance voltage only. Set the initial target of the collector-emitter capacitance voltage to 0 or set its priority to None.
Collector-emitter capacitance, $C_{ce}$	Set the initial target for the collector-emitter capacitance voltage only. Set the initial target of the gate-emitter capacitance voltage to 0 or set its priority to None.
Gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-emitter and gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial target for the gate-emitter and the collector-emitter capacitance.
Gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-collector and collector-emitter voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-emitter, gate collector and collector-emitter capacitances by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.

**Note** Inside your model, the number of initial targets with **Priority** equal to Low or High must match the number of differential variables. The differential variables come from the inductors and the capacitances in the model.

## Assumptions and Limitations

The detailed model is based on the following assumptions:

- This block does not allow you to specify initial conditions on the capacitances. If you select the **Start simulation from steady state** option in the Solver Configuration block, the block solves the initial voltages to be consistent with the calculated steady state. Otherwise, voltages are zero at the start of the simulation.
- You may need to use nonzero capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.
- The block does not account for temperature-dependent effects on the capacitances.

The simplified, event-based model is based on the following assumptions:

- When you use a pair of IGBTs in a bridge arm, normally the gate drive circuitry will prevent a device turning on until the corresponding device has turned off, thereby implementing a minimum dead band. If you need to simulate the case where there is no minimum dead band and both devices are momentarily partially on, use the detailed IGBT model modeling option (**Full I-V and capacitance characteristics**). The assumption used by the event-based modeling option that the collector-emitter voltages can be ramped between on and off states is not valid for such cases.
- A minimum pulse width is applied when turning on or off; at the point where the gate-collector voltage rises above the threshold, any subsequent gate voltage changes are ignored for a time equal to the sum of the turn-on delay and current rise time. Similarly at the point where the gate collector voltage falls below the threshold, any subsequent gate voltage changes are ignored for a time equal to the sum of the turn-off delay and current fall time. This feature is normally implemented in the gate drive circuitry.
- This model does not account for charge. Hence there is no current tail when turning off an inductive load.
- Representative modeling of the current spike during turn-on of an inductive load with preexisting freewheeling current requires tuning of the **Miller resistance** parameter.
- The tabulated turn-on switching loss uses the previous on-state current, not the current value (which is not known until the device reaches the final on state).
- Due to high model stiffness that can arise from the simplified equations, you may get minimum step size violation warnings when using this block. Open the Solver pane of the Configuration Parameters dialog box and increase the **Number of consecutive min steps** parameter value as necessary to remove these warnings.

## Ports

### Conserving

#### **C – Collector terminal**

electrical

Electrical conserving port associated with the PNP emitter terminal

#### **G – Gate terminal**

electrical

Electrical conserving port associated with the IGBT gate terminal

#### **E – Emitter terminal**

electrical

Electrical conserving port associated with the PNP collector terminal

## Parameters

### Modeling Option — IGBT representation

Full I-V and capacitance characteristics | No thermal port (default) | Full I-V and capacitance characteristics | Show thermal port | Simplified I-V characteristics and event-based timing | No thermal port | Simplified I-V characteristics and event-based timing | Show thermal port

Select the IGBT representation:

- 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.

### Main (Full I-V and Capacitance Characteristics)

This configuration of the **Main** setting corresponds to the detailed block modeling option, which is the default. If you are using the simplified, event-based modeling option of the block, see “Main (Simplified I-V Characteristics and Event-Based Timing)” on page 1-1052.

### I-V characteristics defined by — IGBT representation

Fundamental nonlinear equations (default) | Lookup table (2D, temperature independent) | Lookup table (3D, temperature dependent)

Select the IGBT representation:

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

### Zero gate voltage collector current, $I_{ces}$ — Zero gate voltage collector current

2 mA (default)

Collector current that flows when the gate-emitter voltage is set to zero, and a large collector-emitter voltage is applied, that is, the device is in the off-state. The value of the large collector-emitter voltage is defined by the parameter **Voltage at which  $I_{ces}$  is defined**.

### Dependencies

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

### Voltage at which $I_{ces}$ is defined — Voltage at which $I_{ces}$ is defined

600 V (default)



Voltage used when measuring the **Zero gate voltage collector current,  $I_{ces}$** .

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Gate-emitter threshold voltage,  $V_{ge(th)}$  – Gate-emitter threshold voltage**  
6 V (default)

Threshold voltage used in the MOSFET equations.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Collector-emitter saturation voltage,  $V_{ce(sat)}$  – Collector-emitter saturation voltage**  
2.6 V (default)

Collector-emitter voltage for a typical on-state as specified by the manufacturer.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Collector current at which  $V_{ce(sat)}$  is defined – Collector current at which  $V_{ce(sat)}$  is defined**  
400 A (default)

Collector-emitter current when the gate-emitter voltage is  $V_{ge(sat)}$  and collector-emitter voltage is  $V_{ce(sat)}$ .

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Gate-emitter voltage at which  $V_{ce(sat)}$  is defined – Gate-emitter voltage at which  $V_{ce(sat)}$  is defined**  
10 V (default)

Gate voltage used when measuring  $V_{ce(sat)}$  and  $I_{ce(sat)}$ .

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Measurement temperature – Measurement temperature**  
25 degC (default)

Temperature for which the parameters are quoted ( $T_{m1}$ ).

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Vector of gate-emitter voltages, Vge — Vector of gate-emitter voltages**

[-1 0 0.5 1 1.5 2 2.5 3 3.5 4] V (default)

Vector of gate-emitter voltages, to be used for table lookup. The vector values must be strictly increasing. The values can be nonuniformly spaced.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to either Lookup table (2D, temperature independent) or Lookup table (3D, temperature dependent).

**Vector of collector-emitter voltages, Vce — Vector of collector-emitter voltages**

[-2 6 7 8 10 12 15 20] V (default)

Vector of collector-emitter voltages, to be used for table lookup. The vector values must be strictly increasing. The values can be nonuniformly spaced.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to either Lookup table (2D, temperature independent) or Lookup table (3D, temperature dependent).

**Tabulated collector currents, Ic(Vge,Vce) — Tabulated collector currents temperature independent**

```
[-1.015e-05,
1.35e-08, .00047135, .0005092, .0005105, .00051175, .00051299, .00051423, .00
051548, .00051672; -9.9869e-06,
1.35e-08, .00047135, .0005092, .0005105, .00051175, .00051299, .00051423, .00
051548, .00051672; -9.955e-06, 1.35e-08, .0065225, 3.3324, 48.154, 93.661,
105.52, 105.72, 105.93, 106.14; -9.955e-06, 1.35e-08, .0065235, 3.5783,
70.264, 166.33, 252.4, 317.67, 353.38, 357.39; -9.955e-06, 1.35e-08, .006524,
3.7206, 89.171, 228.09, 371.63, 511.02, 642.69, 764.04; -9.9549e-06,
1.35e-08, .0065242, 3.7716, 97.793, 256.21, 424.27, 592.92, 759.2, 921.52;
-9.9549e-06, 1.35e-08, .0065243, 3.8067, 104.52, 278.11, 464.6, 654.37,
844.57, 1033.9; -9.9549e-06, 1.35e-08, .0065244, 3.8324, 109.92, 295.67,
496.54, 702.28, 909.96, 1118.3] A (default)
```

Tabulated values for collector current as a function of gate-emitter voltage and collector-emitter voltage, to be used for 2D table lookup. Each value in the matrix specifies the collector current for a specific combination of gate-emitter voltage and collector-emitter voltage. The matrix size must match the dimensions defined by the gate-emitter voltage and collector-emitter voltage vectors. The default values, in A, are:

```
[-1.015e-5 1.35e-8 4.7135e-4 5.092e-4 5.105e-4 5.1175e-4 5.1299e-4 5.1423e-4 5.1548e-4 5.1672e-4;
-9.9869e-6 1.35e-8 4.7135e-4 5.092e-4 5.105e-4 5.1175e-4 5.1299e-4 5.1423e-4 5.1548e-4 5.1672e-4;
-9.955e-6 1.35e-8 0.0065225 3.3324 48.154 93.661 105.52 105.72 105.93 106.14;
-9.955e-6 1.35e-8 0.0065235 3.5783 70.264 166.33 252.4 317.67 353.38 357.39;
-9.955e-6 1.35e-8 0.006524 3.7206 89.171 228.09 371.63 511.02 642.69 764.04;
-9.9549e-6 1.35e-8 0.0065242 3.7716 97.793 256.21 424.27 592.92 759.2 921.52;
-9.9549e-6 1.35e-8 0.0065243 3.8067 104.52 278.11 464.6 654.37 844.57 1.0339e+3;
-9.9549e-6 1.35e-8 0.0065244 3.8324 109.92 295.67 496.54 702.28 909.96 1.1183e+3]
```

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Lookup table (2D, temperature independent).

**Vector of temperatures, T — Vector of temperatures**

[25 125] degC (default)

Vector of temperatures, to be used for table lookup. The vector values must be strictly increasing. The values can be nonuniformly spaced.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Lookup table (3D, temperature dependent).

**Tabulated collector currents,  $I_c(V_{ge}, V_{ce}, T)$  — Tabulated collector currents temperature dependent**

zeros(8, 10, 2) A (default)

Tabulated values for collector current as a function of gate-emitter voltage, collector-emitter voltage, and temperature, to be used for 3D table lookup. Each value in the matrix specifies the collector current for a specific combination of gate-emitter voltage and collector-emitter voltage at a specific temperature. The matrix size must match the dimensions defined by the gate-emitter voltage, collector-emitter voltage, and temperature vectors.

**Dependencies**

To enable this parameter, set **I-V characteristics defined by** to Lookup table (3D, temperature dependent).

**Junction Capacitance (Full I-V and Capacitance Characteristics)****Parameterization — Junction capacitance parameterization**

Specify fixed input, reverse transfer, and output capacitance (default) | Specify fixed gate-emitter, gate-collector, and collector-emitter capacitance | Specify tabulated input, reverse transfer, and output capacitance | Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance

Select one of the following methods for block parameterization:

- Specify fixed input, reverse transfer, and output capacitance — Provide fixed parameter values from datasheet and let the block convert the input, output, and reverse transfer capacitance values to junction capacitance values, as described in “Charge Model” on page 1-1034. This is the default method.
- Specify fixed gate-emitter, gate-collector, and collector-emitter capacitance — Provide fixed values for junction capacitance parameters directly.
- Specify tabulated input, reverse transfer, and output capacitance — Provide tabulated capacitance and collector-emitter voltage values based on datasheet plots. The block converts the input, output, and reverse transfer capacitance values to junction capacitance values, as described in “Charge Model” on page 1-1034.
- Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance — Provide tabulated values for junction capacitances and collector-emitter voltage.

**Input capacitance, Cies – Input capacitance**

26.4 nF (default)

Gate-emitter capacitance with the collector shorted to the emitter.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Input capacitance, Cies (Vce) – Input capacitance with tabulated input**

[80 40 32 28 27.5 27 26.5 26.5 26.5] nF (default)

Gate-emitter capacitance with the collector shorted to the emitter, tabulated with the collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Reverse transfer capacitance, Cres – Reverse transfer capacitance**

2.7 nF (default)

Collector-gate capacitance with the emitter connected to ground.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Reverse transfer capacitance, Cres (Vce) – Reverse transfer capacitance with tabulated input**

[55 9 5.5 3.1 2.5 2.1 1.9 1.8 1.7] nF (default)

Collector-gate capacitance with the emitter connected to ground, tabulated with the collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Output capacitance, Coes – Output capacitance**

0 nF (default)

Collector-emitter capacitance with the gate and emitter shorted.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Output capacitance, Coes (Vce) – Output capacitance with tabulated input**

[60 20 12 8 6 4.8 4 3.5 3.1] nF (default)

Collector-emitter capacitance with the gate and emitter shorted, tabulated with collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Gate-emitter capacitance — Gate-emitter capacitance**

23.7 nF (default)

Capacitance placed between the gate and the emitter.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed gate-emitter, gate-collector, and collector-emitter capacitance.

**Gate-emitter capacitance,  $C_{ge}(V_{ge}, V_{ce})$  — Gate-emitter capacitance with tabulated values**

[25 31 26.5 24.9 25 24.9 24.6 24.7 24.8] nF (default)

Capacitance placed between the gate and the emitter, tabulated with gate-emitter voltage and collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance.

**Gate-collector capacitance — Gate-collector capacitance**

2.7 nF (default)

Capacitance placed between the gate and the collector.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed gate-emitter, gate-collector, and collector-emitter capacitance.

**Gate-collector capacitance,  $C_{gc}(V_{ge}, V_{ce})$  — Gate-collector capacitance with tabulated values**

[55 9 5.5 3.1 2.5 2.1 1.9 1.8 1.7] nF (default)

Capacitance placed between the gate and the collector, tabulated with the gate-emitter voltage and the collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance.

**Corresponding gate-emitter voltages for tabulated gate capacitance,  $V_{ge}$  — Corresponding collector-emitter voltages**

0 V (default)

Gate-emitter voltages corresponding to the tabulated gate capacitance values.

If this value is a scalar, then **Gate-emitter capacitance,  $C_{ge}(V_{ge}, V_{ce})$**  and **Gate-collector capacitance,  $C_{gc}(V_{ge}, V_{ce})$**  are vectors of scalars. If this value is a vector, then **Gate-emitter**

**capacitance,  $C_{ge}(V_{ge}, V_{ce})$  and Gate-collector capacitance,  $C_{gc}(V_{ge}, V_{ce})$**  are matrices of scalars.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance.

**Collector-emitter capacitance — Collector-emitter capacitance**

0 nF (default)

Capacitance placed between the collector and the emitter.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed gate-emitter, gate-collector, and collector-emitter capacitance.

**Collector-emitter capacitance,  $C_{ce}(V_{ce})$  — Collector-emitter capacitance with tabulated values**

[5 11 6.5 4.9 3.5 2.7 2.1 1.7 1.4] nF (default) | nF

Capacitance placed between the collector and the emitter, tabulated with the collector-emitter voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance.

**Corresponding collector-emitter voltages — Corresponding collector-emitter voltages**

[0 1 2 5 10 15 20 25 30] V (default)

Collector-emitter voltages corresponding to the tabulated capacitance values.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-emitter, gate-collector, and collector-emitter capacitance.

**Charge-voltage linearity — Charge-voltage linearity**

Gate-collector capacitance is constant (default) | Gate-collector charge function is nonlinear

Whether the gate-drain capacitance is fixed or nonlinear:

- **Gate-collector capacitance is constant** — The capacitance value is constant and defined according to the selected parameterization option, either directly or derived from a datasheet. This is the default method.
- **Gate-collector charge function is nonlinear** — The gate-collector charge relationship is defined according to the piecewise-nonlinear function described in “Charge Model” on page 1-1034. Two additional parameters appear to let you define the gate-collector charge function.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify fixed input, reverse transfer and output capacitance or Specify fixed gate-emitter, gate-collector and output capacitance.

**Gate-collector oxide capacitance — Gate-collector oxide capacitance**

20 nF (default)

Gate-collector capacitance when the device is on and the collector-gate voltage is small.

**Dependencies**

To enable this parameter, set **Charge-voltage linearity** to Gate-collector charge function is nonlinear.

**Collector-gate voltage below which oxide capacitance becomes active — Collector-gate voltage below which oxide capacitance becomes active**

-5 V (default)

Collector-gate voltage at which the collector-gate capacitance switches between off-state ( $C_{GC}$ ) and on-state ( $C_{ox}$ ) capacitance values.

**Dependencies**

To enable this parameter, set **Charge-voltage linearity** to Gate-collector charge function is nonlinear.

**Total forward transit time — Total forward transit time**

0 us (default)

Forward transit time for the PNP transistor used as part of the underlying IGBT model. It affects how quickly charge is removed from the channel when the IGBT is turned off.

**Dependencies**

To enable this parameter, in the **Main** settings, set **I-V characteristics defined by** to Fundamental nonlinear equations.

**Turn-off tail current transit time — Total forward transit time**

0 us (default)

Turn-off transit time for the tail current.

**Dependencies**

To enable this parameter, in the **Main** settings, set **I-V characteristics defined by** to Lookup table (2D, temperature independent) or Lookup table (3D, temperature dependent).

**Advanced (Full I-V and Capacitance Characteristics)**

The lookup table representation combines all the equivalent circuit components into one lookup table, and therefore this setting is empty. If you use the equivalent circuit representation, this setting has the following parameters.

**Emission coefficient, N – Emission coefficient**

1 (default)

Emission coefficient or ideality factor of the bipolar transistor.

**Forward Early voltage, VAF – Forward Early voltage**

200 V (default)

Forward Early voltage for the PNP transistor used in the IGBT model. For more information, see the PNP Bipolar Transistor block reference page.

**Collector resistance, RC – Collector resistance**

0.001 Ohm (default)

Resistance at the collector.

**Emitter resistance, RE – Emitter resistance**

0.0010hm (default)

Resistance at the emitter.

**Internal gate resistance, RG – Internal gate resistance**

0.0010hm (default)

Resistance of the internal gate resistor at the measurement temperature. This is not the value of the external circuit series gate resistance, which you should model externally to the IGBT.

**Forward current transfer ratio, BF – Forward current transfer ratio**

50 (default)

Ideal maximum forward current gain for the PNP transistor used in the IGBT model. See the PNP Bipolar Transistor block reference page for more information.

**Temperature Dependence (Full I-V and Capacitance Characteristics)**

For the 2D lookup table representation, the electrical equations do not depend on temperature and therefore this setting is empty. For the 3D lookup table representation with exposed thermal port, this setting is also empty because the 3D matrix on the **Main** setting captures the temperature dependence. If the block thermal port is not exposed for the 3D lookup table representation, then this setting contains only the **Device simulation temperature** parameter. If you use the equivalent circuit representation, this setting has the following parameters.

**Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Specify  $I_{ces}$  and  $V_{ce(sat)}$  at second measurement temperature | Specify  $V_{ce(sat)}$  at second measurement temperature plus the energy gap, EG

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature — Temperature dependence is not modeled, and none of the other parameters on this setting are visible. This is the default method.
- Specify  $I_{ces}$  and  $V_{ce(sat)}$  at second measurement temperature — Model temperature-dependent effects by providing values for the zero gate voltage collector current,  $I_{ces}$ , and collector-emitter voltage,  $V_{ce(sat)}$ , at the second measurement temperature.



- Specify  $V_{ce(sat)}$  at second measurement temperature plus the energy gap, EG
  - Use this option when the datasheet does not provide information on the zero gate voltage collector current,  $I_{ces}$ , at a higher measurement temperature.

### Energy gap, EG – Energy gap

1.11 eV (default)

Energy gap value.

#### Dependencies

To enable this parameter, in the **Temperature Dependence** setting, set **Parameterization** to Specify  $V_{ce(sat)}$  at second measurement temperature plus the energy gap, EG.

### Zero gate voltage collector current, $I_{ces}$ , at second measurement temperature – Zero gate voltage collector current, $I_{ces}$ , at second measurement temperature

100 mA (default)

Zero gate collector current value at the second measurement temperature.

#### Dependencies

To enable this parameter, in the **Temperature Dependence** setting, set **Parameterization** to Specify  $I_{ces}$  and  $V_{ce(sat)}$  at second measurement temperature.

### Collector-emitter saturation voltage, $V_{ce(sat)}$ , at second measurement temperature – Collector-emitter saturation voltage, $V_{ce(sat)}$ , at second measurement temperature

3 V (default)

Collector-emitter saturation voltage value at the second measurement temperature, and when the collector current and gate-emitter voltage are as defined by the corresponding parameters on the **Main** setting.

### Second measurement temperature – Second measurement temperature

125 degC (default)

Second temperature  $T_{m2}$  at which **Zero gate voltage collector current,  $I_{ces}$ , at second measurement temperature** and **Collector-emitter saturation voltage,  $V_{ce(sat)}$ , at second measurement temperature** are measured.

### Saturation current temperature exponent, $X_{TI}$ – Saturation current temperature exponent

3 (default)

Saturation current exponent value for your device type. If you have graphical data for the value of  $I_{ces}$  as a function of temperature, you can use it to fine-tune the value of  $X_{TI}$ .

### Mobility temperature exponent, $BEX$ – Mobility temperature exponent

-1.5 (default)

Mobility temperature coefficient value. You can use the default value for most devices. If you have graphical data for  $V_{ce(sat)}$  at different temperatures, you can use it to fine-tune the value of  $BEX$ .

**Internal gate resistance temperature coefficient – Internal gate resistance temperature coefficient**

0 1/K (default)

Represents the fractional rate of change ( $\alpha$ ) of internal gate resistance (RG) with temperature. Thus the gate resistance is  $R = R_{\text{meas}}(1 + \alpha (T_s - T_{m1}))$ , where  $R_{\text{meas}}$  is the **Internal gate resistance, RG** parameter value.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

**Main (Simplified I-V Characteristics and Event-Based Timing)**

This configuration of the **Main** setting corresponds to the simplified, event-based block modeling option. If you are using the detailed modeling option of the block, see “Main (Full I-V and Capacitance Characteristics)” on page 1-1042.

**Vector of collector currents, Ic – Vector of collector currents**

[0, 10, 50, 100, 200, 400, 600] A (default)

Collector currents for which the on-state collector-emitter voltages are defined. The first element must be zero.

**Vector of temperatures, Tj – Vector of temperatures**

[298.15, 398.15] K (default)

Temperature values at which the collector-emitter and turn-on/turn-off losses are quoted.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Corresponding on-state collector-emitter voltages – Corresponding on-state collector-emitter voltages**

[0, 1.1, 1.3, 1.45, 1.75, 2.25, 2.7] V (default)

Collector-emitter voltages corresponding to the vector of collector currents. The first element must be zero.

**Dependencies**

To enable this parameter, set **Modeling option** to Simplified I-V characteristics and event-based timing | No thermal port.

**Collector-emitter on-state voltages, Vce=fcn(Tj,Ic) – Collector-emitter on-state voltages**

[0, 1.1, 1.3, 1.45, 1.75, 2.25, 2.7; 0, 1.0, 1.15, 1.35, 1.7, 2.35, 3.0] V (default)

Collector-emitter voltages when in the on state, defined as a function of both temperature and current.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Miller resistance – Miller resistance**

0.1 Ohm (default)

When the device turns on, it has a constant-value Miller resistance in series with the demanded voltage ramp. This resistance represents the partial conductance path through the device during turn on, and can be used to match the voltage spike observed when reconnecting a current-carrying inductor and corresponding freewheeling diode. A typical value is 10 to 50 times the effective on-state resistance.

**Off-state conductance – Off-state conductance**

1e-5 Ohm (default)

Conductance when the device is in the off state.

**Gate-emitter threshold voltage,  $V_{ge(th)}$  – Threshold voltage**

6 V (default)

Gate-emitter voltage must be greater than this value for the device to turn on.

**Switching Losses (Simplified I-V Characteristics and Event-Based Timing)****Vector of temperatures for switching losses,  $T_j$  – Switching losses temperatures**

[298.15, 398.15] K (default)

Temperatures at which switching losses are tabulated.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Vector of collector currents for switching losses,  $I_c$  – Switching losses collector currents**

[0, 10, 50, 100, 200, 400, 600] A (default)

Collector currents at which switching losses are tabulated.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Turn-on switching losses,  $E_{on}=f_{cn}(T_j, I_c)$  – Turn-on switching losses**

[0, 0.2, 1, 2, 4, 8, 15; 0, 0.3, 1.3, 2.5, 5, 11, 18]\*1e-3 J (default)

Energy loss when turning the device on, defined as a function of temperature and final on-state current.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Turn-off switching losses,  $E_{off}=f_{cn}(T_j, I_c)$  — Turn-off switching losses**

[0, 0.3, 1.5, 3, 6, 15, 25; 0, 0.7, 3.3, 6.5, 13, 25, 35]\*1e-3 J (default)

Energy loss when turning the device off, defined as a function of temperature and final on-state current.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

**Dynamics (Simplified I-V Characteristics and Event-Based Timing)****Turn-on delay — Turn-on delay**

0.07 us (default)

Time before which the device starts to ramp on.

**Current rise time — Current rise time**

0.7 us (default)

Time taken for the current to ramp up when driving a resistive load.

**Turn-off delay — Turn-off delay**

0.2 us (default)

Time before which the device starts to ramp off.

**Current fall time — Current fall time**

0.5 us (default)

Time taken for the current to ramp down when driving a resistive load.

**Off-state voltage for rise and fall times — Off-state voltage for rise and fall times**

300 V (default)

Off-state collector-emitter voltage used when specifying the rise and fall times. The default value is 300 V.

**Dependencies**

To enable this parameter, set **Modeling option** to Simplified I-V characteristics and event-based timing | No thermal port.

**Off-state voltage for rise and fall times — Off-state voltage for rise and fall times**

300 V (default)

Off-state collector-emitter voltage used when specifying the rise times, fall times, and the losses data.

**Dependencies**

To enable this parameter, expose the thermal port by setting **Modeling option** to Simplified I-V characteristics and event-based timing | Show thermal port.

## Version History

Introduced in R2008a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Functions**

`semiconductorSubcircuit2lookup`

## N-Channel JFET

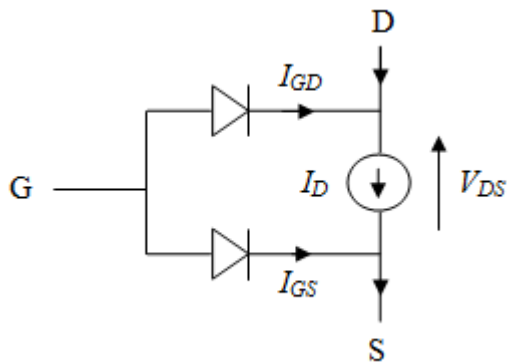
N-Channel junction field-effect transistor

**Library:** Simscape / Electrical / Semiconductors & Converters



### Description

The N-Channel JFET block uses the Shichman and Hodges equations to represent an N-Channel JFET using a model with the following structure:



G is the transistor gate, D is the transistor drain, and S is the transistor source. The drain current,  $I_D$ , depends on the region of operation and whether the transistor is operating in normal or inverse mode.

- In normal mode ( $V_{DS} \geq 0$ ), the block provides the following relationship between the drain current  $I_D$  and the drain-source voltage  $V_{DS}$ .

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Off	$V_{GS} - V_{t0} \leq 0$	$I_D = 0$
Linear	$0 < V_{DS} < V_{GS} - V_{t0}$	$I_D = \beta V_{DS}(2(V_{GS} - V_{t0}) - V_{DS})(1 + \lambda V_{DS})$
Saturated	$0 < V_{GS} - V_{t0} \leq V_{DS}$	$I_D = \beta (V_{GS} - V_{t0})^2 (1 + \lambda V_{DS})$

- In inverse mode ( $V_{DS} < 0$ ), the block provides the following relationship between the drain current  $I_D$  and the drain-source voltage  $V_{DS}$ .

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Off	$V_{GD} - V_{t0} \leq 0$	$I_D = 0$
Linear	$0 < -V_{DS} < V_{GD} - V_{t0}$	$I_D = \beta V_{DS}(2(V_{GD} - V_{t0}) + V_{DS})(1 - \lambda V_{DS})$

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Saturated	$0 < V_{GD} - V_{t0} \leq -V_{DS}$	$I_D = \beta (V_{GD} - V_{t0})^2 (1 - \lambda V_{DS})$

In the preceding equations:

- $V_{GS}$  is the gate-source voltage.
- $V_{GD}$  is the gate-drain voltage.
- $V_{t0}$  is the threshold voltage. If you select **Specify using equation parameters directly** for the **Parameterization** parameter,  $V_{t0}$  is the **Threshold voltage** parameter value. Otherwise, the block calculates  $V_{t0}$  from the datasheet parameters you specify.
- $\beta$  is the transconductance parameter. If you select **Specify using equation parameters directly** for the **Parameterization** parameter,  $\beta$  is the **Transconductance parameter** parameter value. Otherwise, the block calculates  $\beta$  from the datasheet parameters you specify.
- $\lambda$  is the channel-length modulation parameter. If you select **Specify using equation parameters directly** for the **Parameterization** parameter,  $\lambda$  is the **Channel-length modulation** parameter value. Otherwise, the block calculates  $\lambda$  from the datasheet parameters you specify.

The currents in each of the diodes satisfy the exponential diode equation

$$I_{GD} = IS \cdot \left( e^{\frac{qV_{GD}}{kT_{m1}}} - 1 \right)$$

$$I_{GS} = IS \cdot \left( e^{\frac{qV_{GS}}{kT_{m1}}} - 1 \right)$$

where:

- $IS$  is the saturation current. If you select **Specify using equation parameters directly** for the **Parameterization** parameter,  $IS$  is the **Saturation current** parameter value. Otherwise, the block calculates  $IS$  from the datasheet parameters you specify.
- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $T_{m1}$  is the measurement temperature. The value comes from the **Measurement temperature** parameter.

The block models gate junction capacitance as a fixed gate-drain capacitance  $C_{GD}$  and a fixed gate-source capacitance  $C_{GS}$ . If you select **Specify using equation parameters directly** for the **Parameterization** parameter, you specify these values directly using the **Gate-drain junction capacitance** and **Gate-source junction capacitance** parameters. Otherwise, the block derives them from the **Input capacitance Ciss** and **Reverse transfer capacitance Crss** parameter values. The two parameterizations are related as follows:

- $C_{GD} = Crss$
- $C_{GS} = Ciss - Crss$

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. You can optionally include modeling the

dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the junction capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The measurement temperature value,  $T_{m1}$ , is replaced with the simulation temperature,  $T_s$ . The transconductance,  $\beta$ , and the threshold voltage,  $V_{t0}$ , become a function of temperature according to the following equations:

$$\beta_{T_s} = \beta_{T_{m1}} \left( \frac{T_s}{T_{m1}} \right)^{BEX}$$

$$V_{t0s} = V_{t01} + \alpha (T_s - T_{m1})$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $\beta_{T_{m1}}$  is JFET transconductance at the measurement temperature.
- $\beta_{T_s}$  is JFET transconductance at the simulation temperature. This is the transconductance value used in the JFET equations when temperature dependence is modeled.
- $V_{t01}$  is the threshold voltage at measurement temperature.
- $V_{t0s}$  is the threshold voltage at simulation temperature. This is the threshold voltage value used in the JFET equations when temperature dependence is modeled.
- $BEX$  is the mobility temperature exponent. A typical value of  $BEX$  is -1.5.
- $\alpha$  is the gate threshold voltage temperature coefficient,  $dV_{th}/dT$ .

For most JFETS, you can use the default value of -1.5 for  $BEX$ . Some datasheets quote the value for  $\alpha$ , but most typically they provide the temperature dependence for the saturated drain current,  $I_{dss}$ . Depending on the block parameterization method, you have two ways of specifying  $\alpha$ :

- If you parameterize the block from a datasheet, you have to provide  $I_{dss}$  at a second measurement temperature. The block then calculates the value for  $\alpha$  based on this data.
- If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

If you have more data comprising drain current as a function of gate-source voltage for fixed drain-source voltage plotted at more than one temperature, then you can also use Simulink Design Optimization software to help tune the values for  $\alpha$  and  $BEX$ .

In addition, the saturation current term,  $IS$ , in the gate-drain and gate-source current equations depends on temperature

$$IS_{T_s} = IS_{T_{m1}} \cdot (T_s/T_{m1})^{XTI} \cdot \exp\left(-\frac{EG}{kT_s}(1 - T_s/T_{m1})\right)$$

where:

- $IS_{T_{m1}}$  is the saturation current at the measurement temperature.
- $IS_{T_s}$  is the saturation current at the simulation temperature. This is the saturation current value used in the gate diode equations when temperature dependence is modeled.



- $EG$  is the energy gap.
- $k$  is the Boltzmann constant ( $1.3806503e-23$  J/K).
- $XTI$  is the saturation current temperature exponent.

Similar to  $\alpha$ , you have two ways of specifying  $EG$  and  $XTI$ :

- If you parameterize the block from a datasheet, you have to specify the gate reverse current,  $I_{gss}$ , at a second measurement temperature. The block then calculates the value for  $EG$  based on this data and assuming a p-n junction nominal value of 3 for  $XTI$ .
- If you parameterize by specifying equation parameters, you have to provide the values for  $EG$  and  $XTI$  directly. This option gives you most flexibility to match device behavior, for example, if you have a graph of  $I_{gss}$  as a function of temperature. With this data you can use Simulink Design Optimization software to help tune the values for  $EG$  and  $XTI$ .

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- This block does not allow you to specify initial conditions on the junction capacitances. If you select the **Start simulation from steady state** option in the Solver Configuration block, the block solves the initial voltages to be consistent with the calculated steady state. Otherwise, voltages are zero at the start of the simulation.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.
- The block does not account for temperature-dependent effects on the junction capacitances.
- When you specify  $I_{dss}$  at a second measurement temperature, it must be quoted for the same working point (that is, the same drain current and gate-source voltage) as for the  $I_{dss}$  value on the **Main** tab. Inconsistent values for  $I_{dss}$  at the higher temperature will result in unphysical values for  $\alpha$  and unrepresentative simulation results.

- You may need to tune the value of *BEX* to replicate the  $I_D$ - $V_{GS}$  relationship (if available) for a given device. The value of *BEX* affects whether the  $I_D$ - $V_{GS}$  curves for different temperatures cross each other, or not, for the ranges of  $I_D$  and  $V_{GS}$  considered.

## Ports

### Conserving

#### **G – Gate terminal**

electrical

Electrical conserving port associated with the transistor gate terminal

#### **D – Drain terminal**

electrical

Electrical conserving port associated with the transistor drain terminal

#### **S – Source terminal**

electrical

Electrical conserving port associated with the transistor source terminal

#### **H – Thermal port**

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### **Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

#### **Parameterization – Block parameterization**

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- **Specify from a datasheet** – Provide parameters that the block converts to equations that describe the transistor. This is the default method.
- **Specify using equation parameters directly** – Provide equation parameters  $\beta$ ,  $I_S$ ,  $V_{t0}$ , and  $\lambda$ .

#### **Gate reverse current, I\_gss – Gate reverse current**

-1 nA (default)

The reverse current that flows in the diode when the drain and source are short-circuited and a large negative gate-source voltage is applied.

#### Dependencies

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter.

#### Saturated drain current, $I_{dss}$ — Saturated drain current

3 .mA (default)

The current that flows when a large positive drain-source voltage is applied for a specified gate-source voltage. For a depletion-mode device, this gate-source voltage may be zero, in which case  $I_{dss}$  may be referred to as the zero-gate voltage drain current.

#### Dependencies

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter.

#### $I_{dss}$ measurement point, $[V_{gs} V_{ds}]$ — $I_{dss}$ measurement point

[0, 15] V (default)

A vector of the values of  $V_{GS}$  and  $V_{DS}$  at which  $I_{dss}$  is measured. Normally  $V_{GS}$  is zero.  $V_{DS}$  should be greater than zero.

#### Dependencies

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter.

#### Small-signal parameters, $[g_{fs} g_{os}]$ — Small-signal parameters

[3000, 10] uSuS (default)

A vector of the values of  $g_{fs}$  and  $g_{os}$ .  $g_{fs}$  is the forward transfer conductance, that is, the conductance for a fixed drain-source voltage.  $g_{os}$  is the output conductance, that is, the conductance for a fixed gate-source voltage.

#### Dependencies

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter.

#### Small-signal measurement point, $[V_{gs} V_{ds}]$ — Small-signal measurement point

[0, 15] V (default)

A vector of the values of  $V_{GS}$  and  $V_{DS}$  at which  $g_{fs}$  and  $g_{os}$  are measured.  $V_{DS}$  should be greater than zero. For depletion-mode devices,  $V_{GS}$  is typically zero.

#### Dependencies

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter.

#### Transconductance parameter — Transconductance parameter

6.2e-4 AV<sup>2</sup> (default)

The derivative of drain current with respect to gate voltage.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Saturation current — Saturation current**

5e-10 A (default)

The magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Threshold voltage — Threshold voltage**

-2.3 V (default)

The gate-source voltage above which the transistor produces a nonzero drain current. For an enhancement device,  $V_{t0}$  should be positive. For a depletion mode device,  $V_{t0}$  should be negative.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Channel-length modulation — Channel-length modulation**

0.003 1/V (default)

The channel-length modulation.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Measurement temperature — Measurement temperature**

25 degC (default)

The temperature for which the datasheet parameters are quoted.

**Ohmic Resistance****Source ohmic resistance — Source ohmic resistance**

0.0001 Ohm (default) | nonnegative scalar

The transistor source resistance.

**Drain ohmic resistance — Drain ohmic resistance**

0.01 Ohm (default) | nonnegative scalar

The transistor drain resistance.

## Junction Capacitance

### Parameterization — Junction capacitance parameterization

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- Specify from a datasheet — Provide parameters that the block converts to junction capacitance values. This is the default method.
- Specify using equation parameters directly — Provide junction capacitance parameters directly.

### Input capacitance, Ciss — Input capacitance

4.5 pF (default)

The gate-source capacitance with the drain shorted to the source.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Reverse transfer capacitance, Crss — Reverse transfer capacitance

1.5 pF (default)

The drain-gate capacitance with the source connected to ground.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Gate-source junction capacitance — Gate-source junction capacitance

3 pF (default)

The value of the capacitance placed between the gate and the source.

#### Dependencies

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Gate-drain junction capacitance — Gate-drain junction capacitance

1.5 pF (default)

The value of the capacitance placed between the gate and the drain.

#### Dependencies

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter in the **Junction Capacitance** tab.

## Temperature Dependence

### Parameterization — Temperature dependence parameterization

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- **None** – Simulate at parameter measurement temperature – Temperature dependence is not modeled. This is the default method.
- **Model temperature dependence** – Model temperature-dependent effects. You also have to provide a set of additional parameters depending on the block parameterization method. If you parameterize the block from a datasheet, you have to provide values for  $I_{gss}$  and  $I_{dss}$  at second measurement temperature. If you parameterize by directly specifying equation parameters, you have to provide the values for  $EG$ ,  $XTI$ , and the gate threshold voltage temperature coefficient,  $dV_{t0}/dT$ . Regardless of the block parameterization method, you also have to provide values for  $BEX$  and for the simulation temperature,  $T_s$ .

**Gate reverse current,  $I_{gss}$ , at second measurement temperature – Gate reverse current at second measurement temperature**

-200 nA (default)

The value of the gate reverse current,  $I_{gss}$ , at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Saturated drain current,  $I_{dss}$ , at second measurement temperature – Saturated drain current at second measurement temperature**

2.5 mA (default)

The value of the saturated drain current,  $I_{dss}$ , at the second measurement temperature, and when the  $I_{dss}$  measurement point is the same as defined by the  **$I_{dss}$  measurement point, [V<sub>gs</sub> V<sub>ds</sub>]** parameter on the **Main** tab.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Second temperature  $T_{m2}$  at which **Gate reverse current,  $I_{gss}$ , at second measurement temperature** and **Saturated drain current,  $I_{dss}$ , at second measurement temperature** are measured.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Energy gap,  $EG$  – Energy gap**

1.11 eV (default)

Energy gap value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Saturation current temperature coefficient value.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Gate threshold voltage temperature coefficient,  $dV_{t0}/dT$  – Gate threshold voltage temperature coefficient**

-2 mV/K (default)

The rate of change of gate threshold voltage with temperature.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Mobility temperature exponent, BEX – Mobility temperature exponent**

-1.5 (default)

Mobility temperature coefficient value. You can use the default value for most JFETs. See the “Assumptions and Limitations” on page 1-1059 section for additional considerations.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

## Version History

**Introduced in R2008a****Thermal network simlog paths***Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

## References

- [1] H. Shichman and D. A. Hodges, *Modeling and simulation of insulated-gate field-effect transistor switching circuits*. IEEE J. Solid State Circuits, SC-3, 1968.
- [2] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993. Chapter 2.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

P-Channel JFET



## N-Channel LDMOS FET

N-Channel laterally diffused metal oxide semiconductor or vertically diffused metal oxide semiconductor transistors suitable for high voltage

**Library:** Simscape / Electrical / Semiconductors & Converters

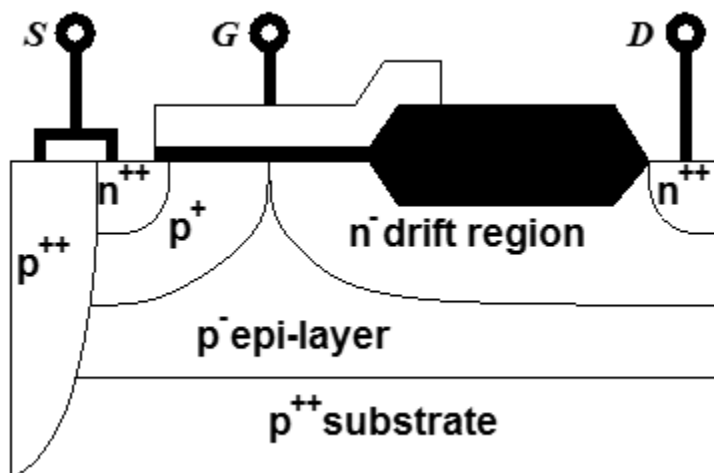


### Description

The N-Channel LDMOS FET block lets you model LDMOS (or VDMOS) transistors suitable for high voltage. The model is based on surface potential and includes 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 option with exposed thermal ports (see “Thermal Port” on page 1-1071), dynamic self-heating

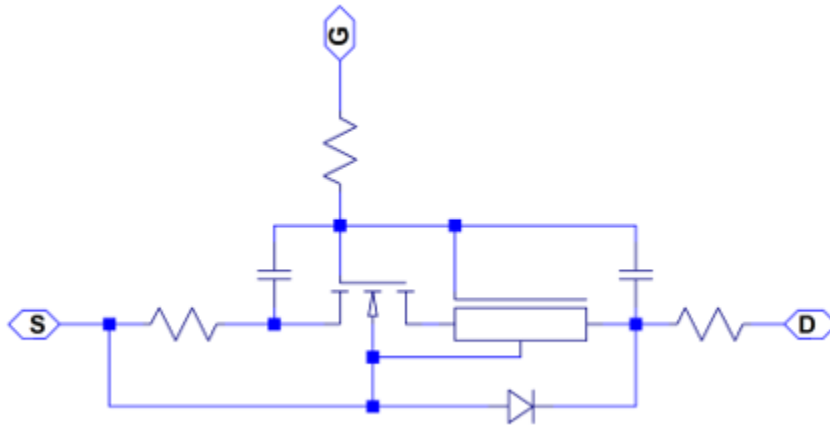
The physical structure of the model is shown in the following figure.



The channel region is in the p+ region, from the heavily n-doped source well to the end of the p+ region. The drift region is a lightly doped drain extension. Further down, there is a p-type epi-layer,

and then the entire structure is on a heavily p-doped substrate. The gate oxide is thin over the entire channel region and over part of the drift region. Further into the drift region, the gate oxide has a greater thickness in the local-oxidation-of-silicon (LOCOS) region.

The next figure shows the equivalent circuit of the model.



The modeling approach is similar to [1]. The overlaps of the gate contact with the source and drain n-wells are modeled as lumped linear capacitances. The channel (p+) region is modeled using the surface-potential-based MOSFET model. The pn-junction between the source/bulk and drain is modeled using an ideal diode, including both junction and diffusion capacitances. The drift region underneath the thin gate oxide is modeled according to a surface-potential formulation, which includes:

- The current due to the accumulation layer at the semiconductor-oxide interface
- The current due to the electrons flowing towards the drain deeper inside the drift region

The space-charge region between the epi-layer and the drift region is represented using a pinching effect on the current flowing through the bulk of the drift region. The LOCOS part of the drift region is modeled as a lumped, series resistor, and there are also series resistances added to the source and gate contacts.

For detailed description of the channel model, see the surface-potential-based model of the N-Channel MOSFET block. The drift region model is similarly derived from the surface potential using the Poisson equation. For an n-type semiconductor under the gradual-channel approximation, the defining equations are:

$$\frac{\partial^2 \psi}{\partial y^2} \approx \frac{qN_D}{\epsilon_{Si}} \left[ -1 - \exp\left(\frac{-\psi - 2\phi_B}{\phi_T}\right) + \exp\left(\frac{\psi - V_{CB}}{\phi_T}\right) \right]$$

$$\phi_T = \frac{k_B T}{q}$$

where:

- $\psi$  is the electrostatic potential.
- $q$  is the magnitude of the electronic charge.

- $N_D$  is the doping density of the drift region.
- $\epsilon_{Si}$  is the dielectric permittivity of the semiconductor material (for example, silicon).
- $\phi_B$  is the difference between the intrinsic Fermi level and the Fermi level deep in the drift region.
- $V_{CB}$  is the quasi-Fermi potential of the drift region referenced to the bulk.
- $\phi_T$  is the thermal voltage.
- $k_B$  is Boltzmann's constant.
- $T$  is temperature.

If we neglect inversion for the DC current model, we obtain the following current expression:

$$I_D = \frac{1}{1 + \theta_{sat} V_{DK}} \left[ \frac{f_{lin}}{R_D} V_{DK} + \frac{\beta}{2} \cdot \frac{V_{GK}^2 - V_{GD}^2}{1 + \frac{\theta_{surf}}{2} (V_{GK} + V_{GD})} \right]$$

where:

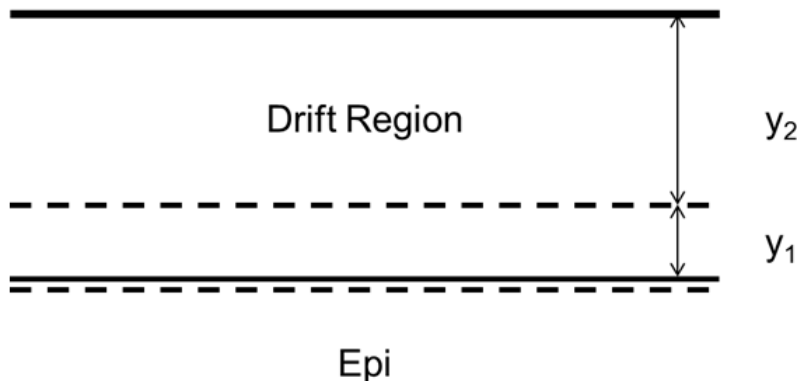
- $I_D$  is the drain current.
- $\theta_{sat}$  is the velocity saturation.
- $V_{ij}$  is the voltage difference between nodes  $i$  and  $j$ , where subscripts D and K refer to the drain and to the junction of the channel and drift regions, respectively, and subscript G refers to the gate with a correction due to the flatband voltage being applied.
- $f_{lin}/R_D$  represents the conductance of the bulk of the drift region, including the effect of pinching due to depletion from the epi-drift interface.
- $\beta$  is the gain of the accumulation layer at the interface between the drift region and the thin gate oxide.
- $\theta_{surf}$  is the parameter that accounts for scattering in the accumulation layer due to the vertical electric field.

The pinching off of the bulk part of the drift region is described by

$$f_{lin} = 1 - \lambda_D \frac{\sqrt{V_{bi} + V_{SB}} - \sqrt{V_{bi}}}{\sqrt{V_{bi}}}$$

where:

- $\lambda_D$  is the parameter representing the n-side vertical depth of the space-charge region along the epi-drift interface at zero bias divided by the vertical depth of the undepleted part of the drift region at zero bias.



In the figure, the top solid line is the semiconductor surface. The lower solid line is the junction between the drift region and the epi layer. The dashed lines show the extent of the space-charge region around the drift-epi interface.  $\lambda_D$  is  $y_1/y_2$  at zero bias.

- $V_{bi}$  is the built-in voltage for the epi-drift diode.
- $V_{SB}$  is the source-body voltage, used as an approximation to the bias applied across the epi-drift diode. Using this voltage instead of  $V_{KB}$  is more numerically stable, and is justified because most of the drain-source voltage drops across the drift region in the transistor on-state.

The charge model is similar to that of the surface-potential-based MOSFET model, with additional expressions to account for the charge in the drift region. The block uses the derived equations as described in [1], which include both inversion and accumulation in the drift region.

### Modeling Body Diode

The block models the body diode as an ideal, exponential diode with both junction and diffusion capacitances:

$$I_{dio} = I_s \left[ \exp\left(-\frac{V_{DB}}{n\phi_T}\right) - 1 \right]$$

$$C_j = \frac{C_{j0}}{\sqrt{1 + \frac{V_{DB}}{V_{bi}}}}$$

$$C_{diff} = \frac{\tau I_s}{n\phi_T} \exp\left(-\frac{V_{DB}}{n\phi_T}\right)$$

where:

- $I_{dio}$  is the current through the diode.
- $I_s$  is the reverse saturation current.
- $V_{DB}$  is the drain-body voltage.
- $n$  is the ideality factor.
- $\phi_T$  is the thermal voltage.
- $C_j$  is the junction capacitance of the diode.
- $C_{j0}$  is the zero-bias junction capacitance.
- $V_{bi}$  is the built-in voltage.
- $C_{diff}$  is the diffusion capacitance of the diode.
- $\tau$  is the transit time.

The capacitances are defined through an explicit calculation of charges, which are then differentiated to give the capacitive expressions above. The block computes the capacitive diode currents as time derivatives of the relevant charges, similar to the computation in the surface-potential-based MOSFET model.

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. To model the dependence on

temperature during simulation, select **Model temperature dependence** for the **Parameterization** parameter on the **Temperature Dependence** tab.

The model includes temperature effects on the capacitance characteristics, as well as modeling the dependence of the transistor static behavior on temperature during simulation.

The **Measurement temperature** parameter on the **Main** tab specifies temperature  $T_{m1}$  at which the other device parameters have been extracted. The **Temperature Dependence** parameters provides the simulation temperature,  $T_s$ , and the temperature-scaling coefficients for the other device parameters. For more information, see “Temperature Dependence” on page 1-1075.

## Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

The thermal option of this block includes dynamic self-heating. This lets you simulate the effect of self-heating on the electrical characteristics of the device.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### G — Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal

#### D — Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal

#### S — Source terminal

electrical

Electrical conserving port associated with the transistor source terminal

## H — Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

### Gain, [channel drift\_region] — Gain

[11.6, .01] A/V (default) | positive vector

The gain,  $\beta$ , of the MOSFET regions. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. This parameter primarily defines the linear region of operation on an  $I_D$ - $V_{DS}$  characteristic.

### Flatband voltage, [channel drift\_region] — Flatband voltage

[-1.05, -.1] V (default)

The flatband voltage,  $V_{FB}$ , defines the gate bias that must be applied in order to achieve the flatband condition at the surface of the silicon. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The default value is [-1.05, -0.1] V. You can also use this parameter to arbitrarily shift the threshold voltage due to material work function differences, and to trapped interface or oxide charges. In practice, however, it is usually recommended to modify the threshold voltage by using the **Body factor** and **Surface potential at strong inversion** parameters first, and only use this parameter for fine-tuning.

The threshold voltage for the channel region, for a short-circuited source-bulk connection, is approximately

$$V_T = V_{FB} + 2\phi_B + 2\phi_T + \gamma\sqrt{2\phi_B + 2\phi_T}$$

where  $2\phi_B$  is the surface potential at strong inversion and  $\gamma$  is the body factor, both at the channel region.

### Body factor, [channel drift\_region] — Body factor

[3.4, 2.5]  $V^{1/2}$  (default)

Body factor,  $\gamma$ , in the surface-potential equation. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.

For the channel region, the body factor is

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_A}}{C_{ox}}$$

See the N-Channel MOSFET block reference page for details on this equation. The drift region equation is similar, except that  $N_A$  is replaced by the doping density,  $N_D$ . The channel-region parameter value primarily impacts the threshold voltage. For the drift region, this parameter primarily affects the charge model, and also has a minor effect on the pinch-off behavior of the bulk current through the drift region.

**Surface potential at strong inversion, [channel drift\_region] – Surface potential at strong inversion**

[.95, .95] V (default)

The  $2\phi_B$  term in the surface-potential equation. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.

The channel-region parameter value also primarily impacts the threshold voltage. For the drift region, this parameter affects the charge model only.

**Velocity saturation factor, [channel drift\_region] – Velocity saturation factor**

[0, .1] 1/V (default)

Velocity saturation,  $\theta_{sat}$ , in the drain-current equation. Use this parameter in cases where a good fit to linear operation leads to a saturation current that is too high. By increasing this parameter value, you reduce the saturation current. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The default value is [0.0, 0.1] 1/V, which means that velocity saturation in the channel region is off by default.

**Drift region surface scattering factor – Drift region surface scattering factor**

0 V (default)

Surface scattering factor,  $\theta_{surf}$ , in the drain-current equation. This parameter applies to the drift region only and accounts for scattering in the accumulation layer due to the vertical electric field.

**Channel-length modulation factor – Channel-length modulation factor**

0 (default)

The factor,  $\alpha$ , multiplying the logarithmic term in the  $G_{AL}$  equation. See the N-Channel MOSFET block reference page for details on this equation. This parameter describes the onset of channel-length modulation. For device characteristics that exhibit a positive conductance in saturation, increase the parameter value to fit this behavior. This parameter applies to the channel region only. The default value is 0, which means that channel-length modulation is off by default.

**Channel-length modulation voltage – Channel-length modulation voltage**

5e-2 V (default)

The voltage  $V_p$  in the  $G_{AL}$  equation. See the N-Channel MOSFET block reference page for details on this equation. This parameter controls the drain-voltage at which channel-length modulation starts to become active. This parameter applies to the channel region only.

**Linear-to-saturation transition coefficient – Linear-to-saturation transition coefficient**

8 (default)

This parameter controls how smoothly the MOSFET transitions from linear into saturation, particularly when velocity saturation is enabled. This parameter can usually be left at its default value, but you can use it to fine-tune the knee of the  $I_D$ - $V_{DS}$  characteristic. This parameter applies both to the channel and drift regions. The expected range for this parameter value is between 2 and 8.

### Measurement temperature — Measurement temperature

25 degC (default)

Temperature  $T_{m1}$  at which the block parameters are measured. If the **Device simulation temperature** parameter on the **Temperature Dependence** tab differs from this value, then device parameters will be scaled from their defined values according to the simulation and reference temperatures. For more information, see “Temperature Dependence” on page 1-1075.

### Ohmic Resistance

#### Source ohmic resistance — Source ohmic resistance

1e-4 Ohm (default) | nonnegative scalar

The transistor source resistance, that is, the series resistance associated with the source contact.

#### Drain ohmic resistance — Drain ohmic resistance

0.07 Ohm (default) | nonnegative scalar

The transistor drain resistance, that is, the series resistance associated with the drain contact and with the LOCOS part of the drift region, which is not heavily impacted by the applied gate voltage.

#### Gate ohmic resistance — Gate ohmic resistance

8.4 Ohm (default) | nonnegative scalar

The transistor gate resistance, that is, the series resistance associated with the gate contact.

#### Drift region low-bias resistance for gated region — Drift region low-bias resistance for gated region

0.1 Ohm (default) | nonnegative scalar

Resistance  $R_D$  in the drain-current equation. It represents the resistance of the bulk part of the drift region in the absence of depletion from the top and bottom interfaces.

#### Drift region depletion layer thickness factor — Drift region depletion layer thickness factor

0.2 (default)

Parameter  $\lambda_D$  in the drain-current equation. It is the ratio of vertical depths  $y_1$  and  $y_2$  at zero bias, where  $y_1$  represents the space-charge region and  $y_2$  represents the undepleted part of the drift region.

### Capacitances

#### Oxide capacitance, [channel drift\_region] — Oxide capacitance

[1600, 1000] pF (default)

The parallel plate gate-channel and gate-drift-region capacitance. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.



**Gate-source overlap capacitance — Gate-source overlap capacitance**

15 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the source well.

**Gate-drain overlap capacitance — Gate-drain overlap capacitance**

15 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the drain well.

**Body Diode****Reverse saturation current — Reverse saturation current**

1e-13 A (default)

The current designated by the  $I_s$  symbol in the body-diode equations.

**Built-in voltage — Built-in voltage**

0.6 V (default)

The built-in voltage of the diode, designated by the  $V_{bi}$  symbol in the body-diode equations.

**Ideality factor — Ideality factor**

1 (default)

The factor designated by the  $n$  symbol in the body-diode equations.

**Zero-bias junction capacitance — Zero-bias junction capacitance**

1800 pF (default)

The capacitance between the drain and bulk contacts at zero-bias due to the body diode alone. It is designated by the  $C_{j0}$  symbol in the body-diode equations.

**Transit time — Transit time**

50e-9 s (default)

The time designated by the  $\tau$  symbol in the body-diode equations.

**Temperature Dependence****Parameterization — Temperature dependence parameterization**

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None — Simulate at parameter measurement temperature — Temperature dependence is not modeled. This is the default method.
- Model temperature dependence — Model temperature-dependent effects. Provide a value for the device simulation temperature,  $T_s$ , and the temperature-scaling coefficients for other block parameters.

**Device simulation temperature — Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated

### Gain temperature exponent, [channel drift\_region] – Gain temperature exponent [1.3, 1.3] (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. Both in the channel and the drift region, the MOSFET gain,  $\beta$ , is assumed to scale exponentially with temperature,  $\beta = \beta_{m1} (T_{m1}/T_s)^{\eta_\beta}$ .  $\beta_{m1}$  is the value of the channel or drift region gain, as specified by the **Gain, [channel drift\_region]** parameter from the **Main** tab.  $\eta_\beta$  is the corresponding element of the **Gain temperature exponent, [channel drift\_region]** parameter.

### Flatband voltage temperature coefficient, [channel drift\_region] – Flatband voltage temperature coefficient [.0005, .0005] V/K (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The flatband voltage,  $V_{FB}$ , is assumed to scale linearly with temperature,  $V_{FB} = V_{FBm1} + (T_s - T_{m1})S_{TV_{FB}}$ .  $V_{FBm1}$  is the value of the channel or drift region flatband voltage, as specified by the **Flatband voltage, [channel drift\_region]** parameter from the **Main** tab.  $S_{TV_{FB}}$  is the corresponding element of the **Flatband voltage temperature coefficient, [channel drift\_region]** parameter.

### Surface potential at strong inversion temperature coefficient – Surface potential -8.5e-4 V/K (default)

The surface potential at strong inversion,  $2\phi_B$ , is assumed to scale linearly with temperature,  $2\phi_B = 2\phi_{Bm1} + (T_s - T_{m1})S_{T\phi_B}$ .  $2\phi_{Bm1}$  is the value of the **Surface potential at strong inversion** parameter from the **Main** tab and  $S_{T\phi_B}$  is the **Surface potential at strong inversion temperature coefficient**.

### Velocity saturation temperature exponent, [channel drift\_region] – Velocity saturation temperature exponent [1.04, 1.04] (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The velocity saturation,  $\theta_{sat}$ , is assumed to scale exponentially with temperature,  $\theta_{sat} = \theta_{sat,m1} (T_{m1}/T_s)^{\eta_\theta}$ .  $\theta_{sat,m1}$  is the value of the channel or drift region velocity saturation factor, as specified by the **Velocity saturation factor, [channel drift\_region]** parameter from the **Main** tab.  $\eta_\theta$  is the corresponding element of the **Velocity saturation temperature exponent, [channel drift\_region]** parameter.

### Ohmic resistance temperature exponent – Ohmic resistance temperature exponent 0.95 (default)

The series resistances are assumed to correspond to semiconductor resistances. Therefore, they decrease exponentially with increasing temperature.  $R_i = R_{i,m1} (T/T_s)^{\eta_R}$ , where  $i$  is  $S_{m1}$ ,  $D$ , or  $G$ , for the source, drain, or gate series resistance, respectively.  $R_{i,m1}$  is the value of the corresponding series resistance parameter from the **Ohmic Resistance** tab and  $\eta_R$  is the **Ohmic resistance temperature exponent**.

### Drift region low-bias resistance temperature exponent for gated portion – Drift region low-bias resistance temperature exponent for gated portion 0.95 (default)

Resistance  $R_D$ , the low-bias resistance of the bulk part of the drift region, scales similarly to the other series resistances. A separate value of the temperature exponent for this resistance provides an extra degree of freedom.

### Body diode reverse saturation current temperature exponent – Body diode reverse saturation current temperature exponent

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$

$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** tab,  $k_B$  is Boltzmann's constant (8.617x10<sup>-5</sup>eV/K), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

## Version History

### Introduced in R2016b

#### Thermal network simlog paths

*Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

## References

- [1] Aarts, A., N. D'Halleweyn, and R. Van Langevelde. "A Surface-Potential-Based High-Voltage Compact LDMOS Transistor Model." *IEEE Transactions on Electron Devices*. 52(5):999 - 1007. June 2005.
- [2] Van Langevelde, R., A. J. Scholten, and D. B. M. Klaassen. "Physical Background of MOS Model 11. Level 1101." *Nat.Lab. Unclassified Report 2003/00239*. April 2003.
- [3] Oh, S-Y., D. E. Ward, and R. W. Dutton. "Transient analysis of MOS transistors." *IEEE J. Solid State Circuits*. SC-15, pp. 636-643, 1980.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

N-Channel MOSFET | P-Channel LDMOS FET

# N-Channel MOSFET

N-Channel metal oxide semiconductor field effect transistor using either Shichman-Hodges equation or surface-potential-based model

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The N-Channel MOSFET block provides two main modeling options:

- Based on threshold voltage — Uses 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. For details, see “Threshold-Based Model” on page 1-1080.

This modeling option provides four ways of parameterizing an N-Channel MOSFET:

- By specifying parameters from a datasheet.
- By specifying equation parameters directly.
- By a 2-D lookup table approximation to the I-V (current-voltage) curve. For details, see “Representation by 2-D Lookup Table” on page 1-1084.
- By a 3-D lookup table approximation to the I-V (current-voltage) curve that includes temperature data. For details, see “Representation by 3-D Lookup Table” on page 1-1086.
- Based on surface potential — Uses the surface-potential equation to represent the device. This modeling approach provides a greater level of model fidelity than the simple square-law (threshold-voltage-based) models can provide. The trade-off is that there are more parameters that require extraction. For details, see “Surface-Potential-Based Model” on page 1-1086.

Together with the thermal port options (see “Thermal Port” on page 1-1093), the block therefore provides you with four choices. To select the desired option, set the **Modeling option** parameter to either:

- **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.
- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port.
- **Surface-potential-based** — Model based on surface potential. This model does not simulate thermal effects.
- **Surface-potential-based with thermal** — Thermal modeling option of the model based on surface potential.

### Threshold-Based Model

The threshold-based option of the block uses the Shichman and Hodges equations [1] for an insulated-gate field-effect transistor to represent an N-Channel MOSFET.

The drain-source current,  $I_{DS}$ , depends on the region of operation:

- In the off region ( $V_{GS} < V_{th}$ ), the drain-source current is:

$$I_{DS} = 0$$

- In the linear region ( $0 < V_{DS} < V_{GS} - V_{th}$ ), the drain-source current is:

$$I_{DS} = K((V_{GS} - V_{th})V_{DS} - V_{DS}^2/2)(1 + \lambda |V_{DS}|)$$

- In the saturated region ( $0 < V_{GS} - V_{th} < V_{DS}$ ), the drain-source current is:

$$I_{DS} = (K/2)(V_{GS} - V_{th})^2(1 + \lambda |V_{DS}|)$$

In the preceding equations:

- $K$  is the transistor gain.
- $V_{DS}$  is the positive drain-source voltage.
- $V_{GS}$  is the gate-source voltage.
- $V_{th}$  is the threshold voltage. For the four terminal parameterization,  $V_{th}$  is obtained using these equations:

$V_{BS}$ Range	$V_{th}$ Equation
$V_{BS} \leq 0$	$V_{th} = V_{T0} + \gamma(-\sqrt{2\phi_B}) + \gamma(\sqrt{2\phi_B - V_{BS}})$
$0 < V_{BS} \leq 4\phi_B$	$V_{th} = V_{T0} - \frac{\gamma V_{BS}}{\sqrt{2\phi_B}}$
$V_{BS} > 4\phi_B$	$V_{th} = V_{T0} + \gamma(-\sqrt{2\phi_B})$

- $\lambda$  is the channel modulation.

### Charge Model for Threshold-Based Modeling Option

The block models capacitances either by fixed capacitance values, or by tabulated values as a function of the drain-source voltage. In either case, you can either directly specify the gate-source and gate-drain capacitance values, or let the block derive them from the input and reverse transfer capacitance values. Therefore, the **Parameterization** options for charge model on the **Capacitance** setting are:

- Specify fixed input, reverse transfer and output capacitance — Provide fixed parameter values from datasheet and let the block convert the input and reverse transfer capacitance values to capacitance values, as described below. This is the default method.
- Specify fixed gate-source, gate-drain and drain-source capacitance — Provide fixed values for capacitance parameters directly.
- Specify tabulated input, reverse transfer and output capacitance — Provide tabulated capacitance and drain-source voltage values based on datasheet plots. The block converts the input and reverse transfer capacitance values to capacitance values, as described below.

- **Specify tabulated gate-source, gate-drain and drain-source capacitance** — Provide tabulated values for capacitances and drain-source voltage.

Use one of the tabulated capacitance options (**Specify tabulated input, reverse transfer and output capacitance** or **Specify tabulated gate-source, gate-drain and drain-source capacitance**) when the datasheet provides a plot of capacitances as a function of drain-source voltage. Using tabulated capacitance values gives more accurate dynamic characteristics and avoids the need for interactive tuning of parameters to fit the dynamics.

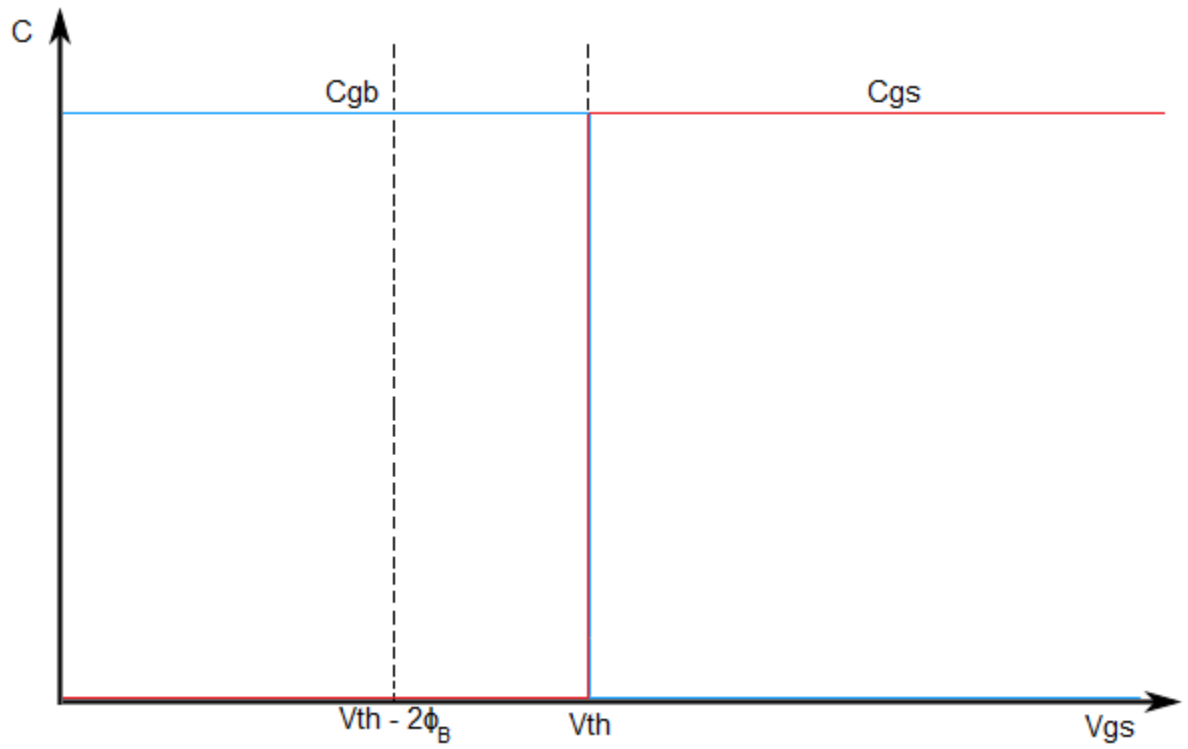
If you use the **Specify fixed gate-source, gate-drain and drain-source capacitance** or **Specify tabulated gate-source, gate-drain and drain-source capacitance** option, the **Capacitance** setting lets you specify the **Gate-drain capacitance, Cgd**, **Gate-source capacitance, Cgs**, and **Drain-source capacitance, Cds** parameter values (fixed or tabulated) directly. Otherwise, the block derives them from the **Input capacitance, Ciss**, **Reverse transfer capacitance, Crss**, and **Output capacitance, Coss** parameter values. These two parameterization methods are related as follows:

- $C_{GD} = Crss$
- $C_{GS} = Ciss - Crss$
- $C_{DS} = Coss - Crss$

For the four terminals parameterization, the **Input capacitance, Ciss**, **Reverse transfer capacitance, Crss**, and **Output capacitance, Coss** are obtained using these equations:

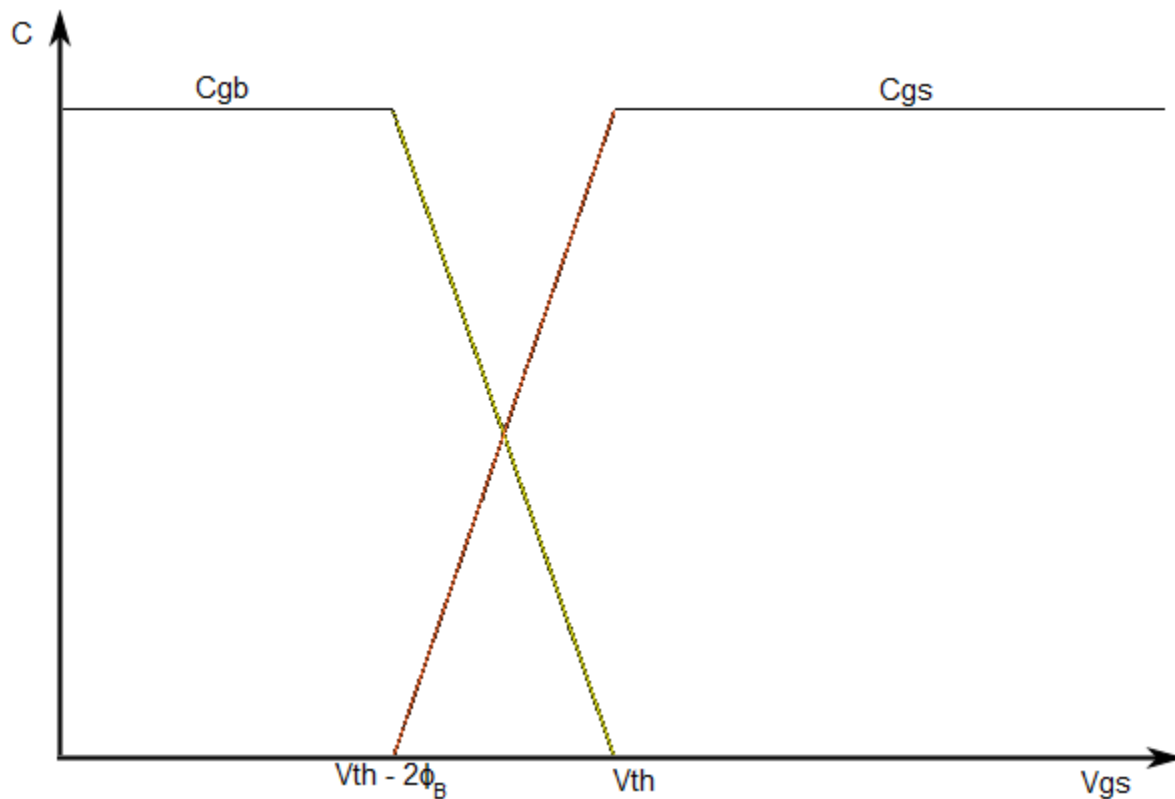
- $C_{GD} = Crss$
- $C_{GS} + C_{GB} = Ciss - Crss$
- $C_{DB} = Coss - Crss$

A simplified Meyer's capacitance model is used to describe the gate-source capacitance,  $C_{GS}$ , the gate-bulk capacitance,  $C_{GB}$ , and the gate-drain capacitance,  $C_{GD}$ . These figures show how the gate-bulk and gate-source capacitances change instantaneously, while the

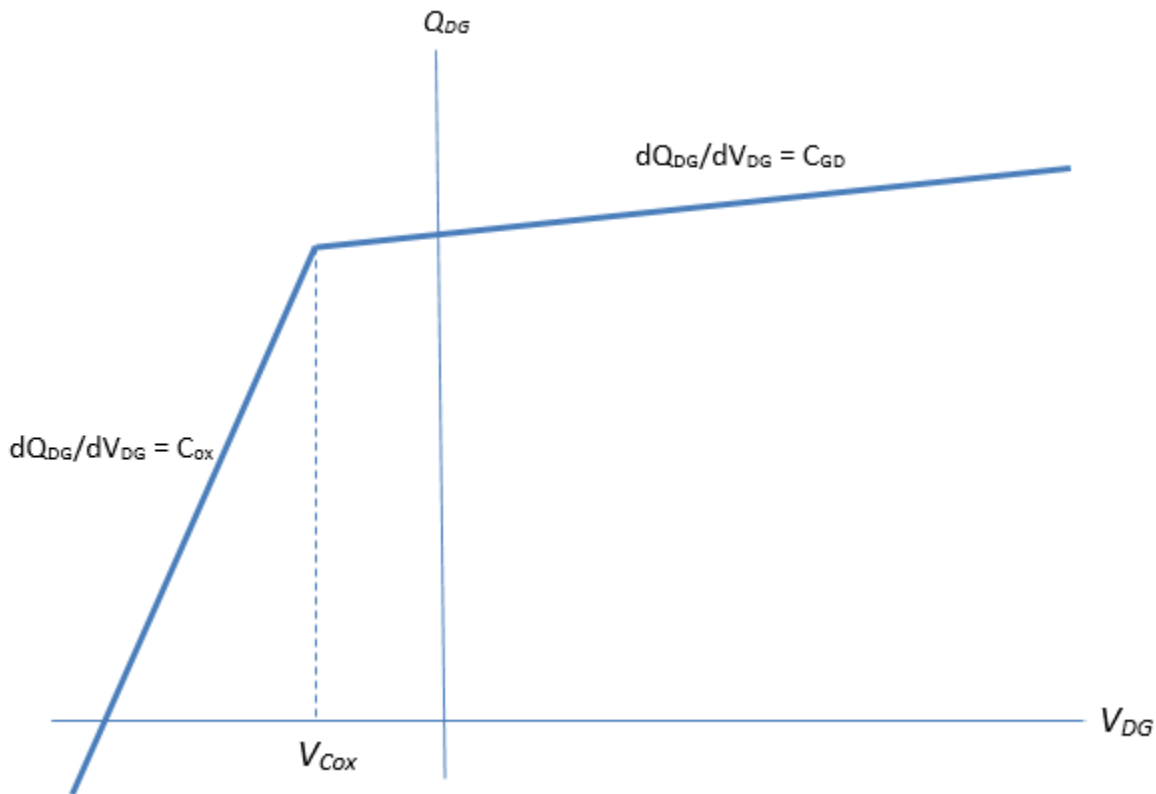


Gate-bulk and gate-source capacitance change instantaneously.





The two fixed capacitance options (Specify fixed input, reverse transfer and output capacitance or Specify fixed gate-source, gate-drain and drain-source capacitance) let you model gate capacitance as a fixed gate-source capacitance  $C_{GS}$  and either a fixed or a nonlinear gate-drain capacitance  $C_{GD}$ . If you select the Gate-drain charge function is nonlinear option for the **Gate-drain charge-voltage linearity** parameter, then the gate-drain charge relationship is defined by the piecewise-linear function shown in the following figure.



**Nonlinear gate-drain charge function**

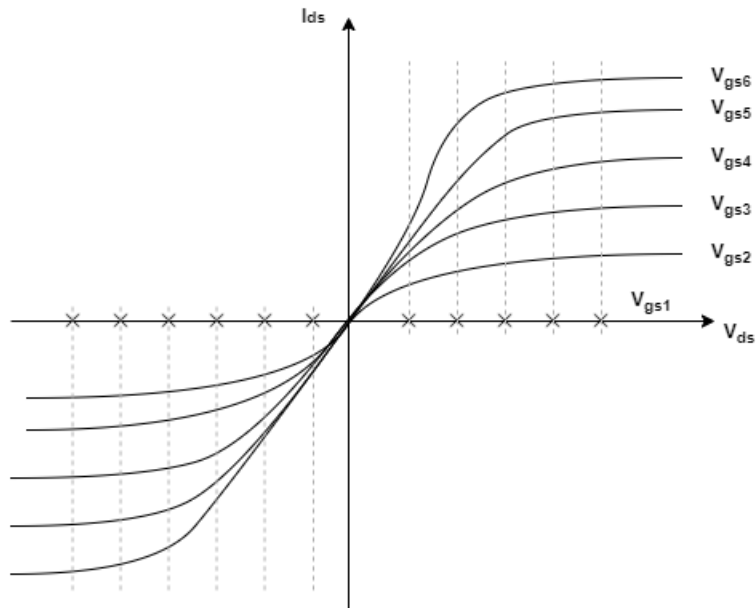
For instructions on how to map a time response to device capacitance values, see the N-Channel IGBT block reference page. However, this mapping is only approximate because the Miller voltage typically varies more from the threshold voltage than in the case for the IGBT.

**Note** Because this block implementation includes a charge model, you must model the impedance of the circuit driving the gate to obtain representative turn-on and turn-off dynamics. Therefore, if you are simplifying the gate drive circuit by representing it as a controlled voltage source, you must include a suitable series resistor between the voltage source and the gate.

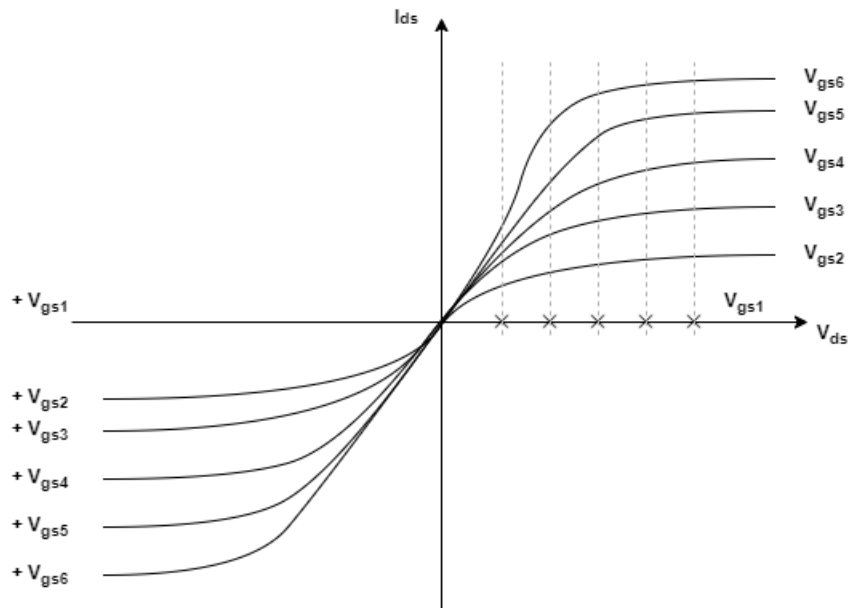
### Representation by 2-D Lookup Table

For the lookup table representation of the detailed block modeling option, you provide tabulated values for drain-source currents as a function of gate-source voltage and drain-source voltage. The main advantage of using this option is simulation speed. It also lets you parameterize the device from either measured data or from data obtained from another simulation environment.

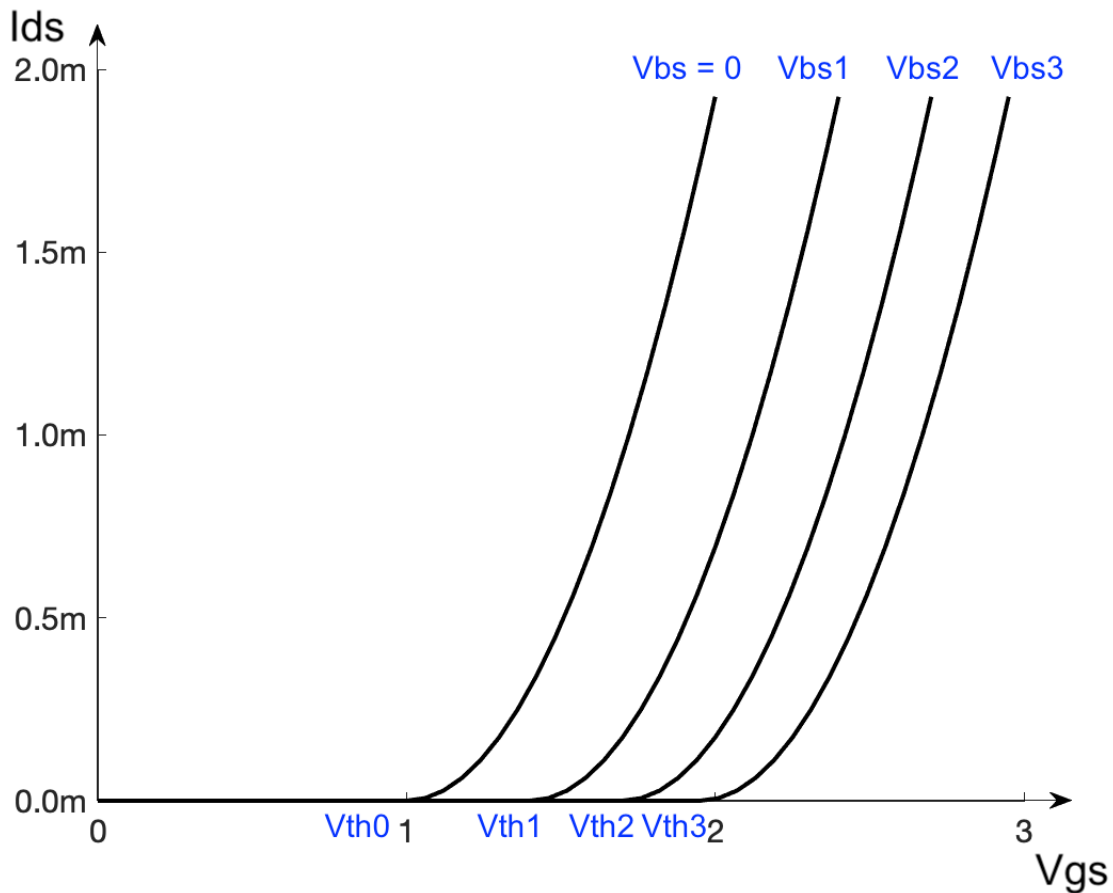
This figure shows the implementation of the 2-D lookup table option when you set **Ids-Vds parameterization** to Provide negative and positive Vds data:



This figure shows the implementation of the 2-D lookup table option when you set **Ids-Vds parameterization** to Provide positive Vds data only:



For the four terminal MOSFET, the surface potential and body factor values are calculated based on the nearest threshold voltage as shown in this picture:



### Representation by 3-D Lookup Table

For the temperature-dependent lookup table representation of the detailed block modeling option, you provide tabulated values for drain-source currents as a function of gate-source voltage, drain-source voltage, and temperature.

### Surface-Potential-Based Model

The surface-potential-based modeling option of the block provides a greater level of model fidelity than the simple square-law (threshold-voltage-based) model. The surface-potential-based block modeling option includes the following effects:

- Fully nonlinear capacitance model (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 modeling option, dynamic self-heating (that is, you can simulate the effect of self-heating on the electrical characteristics of the device)

This model is a minimal version of the world-standard PSP model (see <https://briefs.techconnect.org/papers/introduction-to-psp-mosfet-model/>), including only certain effects from the PSP model in order

to strike a balance between model fidelity and complexity. For details of the physical background to the phenomena included in this model, see [2].

The basis of the model is Poisson equation:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \frac{qN_A}{\epsilon_{Si}} \left[ 1 - \exp\left(\frac{-\psi}{\phi_T}\right) + \exp\left(\frac{\psi - 2\phi_B - V_{CB}}{\phi_T}\right) \right]$$

$$\phi_T = \frac{k_B T}{q}$$

where:

- $\psi$  is the electrostatic potential.
- $q$  is the magnitude of the electronic charge.
- $N_A$  is the density of acceptors in the substrate.
- $\epsilon_{Si}$  is the dielectric permittivity of the semiconductor material (for example, silicon).
- $\phi_B$  is the difference between the intrinsic Fermi level and the Fermi level in the bulk silicon.
- $V_{CB}$  is the quasi-Fermi potential of the surface layer referenced to the bulk.
- $\phi_T$  is the thermal voltage.
- $k_B$  is Boltzmann's constant.
- $T$  is temperature.

Poisson equation is used to derive the surface-potential equation:

$$(V_{GB} - V_{FB} - \psi_s)^2 = \gamma^2 \left[ \psi_s + \phi_T \left( \exp\left(\frac{-\psi_s}{\phi_T}\right) - 1 \right) + \phi_T \exp\left(-\frac{2\phi_B + V_{CB}}{\phi_T}\right) \left( \exp\left(\frac{\psi_s}{\phi_T}\right) - 1 \right) \right]$$

where:

- $V_{GB}$  is the applied gate-body voltage.
- $V_{FB}$  is the flatband voltage.
- $\psi_s$  is the surface potential.
- $\gamma$  is the body factor,

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_A}}{C_{ox}}$$

- $C_{ox}$  is the capacitance per unit area.

The block uses an explicit approximation to the surface-potential equation, to avoid the need for numerical solution to this implicit equation.

Once the surface potential is known, the drain current  $I_D$  is given by

$$I_D = \frac{W\mu_0}{LG_{\Delta L}G_{mob}} \sqrt{1 + (\theta_{sat}\Delta\psi)^2} \left[ -\bar{Q}_{inv}\Delta\psi + \phi_T(Q_{invL} - Q_{inv0}) \right]$$

where:

- $W$  is the device width.
- $L$  is the channel length.
- $\mu_0$  is the low-field mobility.
- $\theta_{sat}$  is the velocity saturation.
- $\Delta\psi$  is the difference in the surface potential between the drain and the source.
- $Q_{inv0}$  and  $Q_{invL}$  are the inversion charge densities at the source and drain, respectively.
- $\bar{Q}_{inv}$  is the average inversion charge density across the channel.
- $G_{mob}$  is the mobility reduction factor. For more information, see the **Surface roughness scattering factor** parameter description in the “Main (Surface-Potential-Based Modeling Option)” on page 1-1102 section.
- $G_{\Delta L}$  is the channel-length modulation.

$$G_{\Delta L} = 1 - \frac{\Delta L}{L} = 1 - \alpha \ln \left[ \frac{V_{DB} - V_{DB,eff} + \sqrt{(V_{DB} - V_{DB,eff})^2 + V_p^2}}{V_p} \right]$$

where:

- $\alpha$  is the channel-length modulation factor.
- $V_{DB}$  is the drain-body voltage.
- $V_{DB,eff}$  is the drain-body voltage clipped to a maximum value corresponding to velocity saturation or pinch-off (whichever occurs first).
- $V_p$  is the channel-length modulation voltage.

The block computes the inversion charge densities directly from the surface potential.

The block also computes the nonlinear capacitances from the surface potential. Source and drain charge contributions are assigned via a bias-dependent Ward-Dutton charge-partitioning scheme, as described in [3]. These charges are computed explicitly, so this model is charge-conserving. The capacitive currents are computed by taking the time derivatives of the relevant charges. In practice, the charges within the simulation are normalized to the oxide capacitance and computed in units of volts.

The MOSFET gain,  $\beta$ , is given by

$$\beta = \frac{W\mu_0 C_{ox}}{L}$$

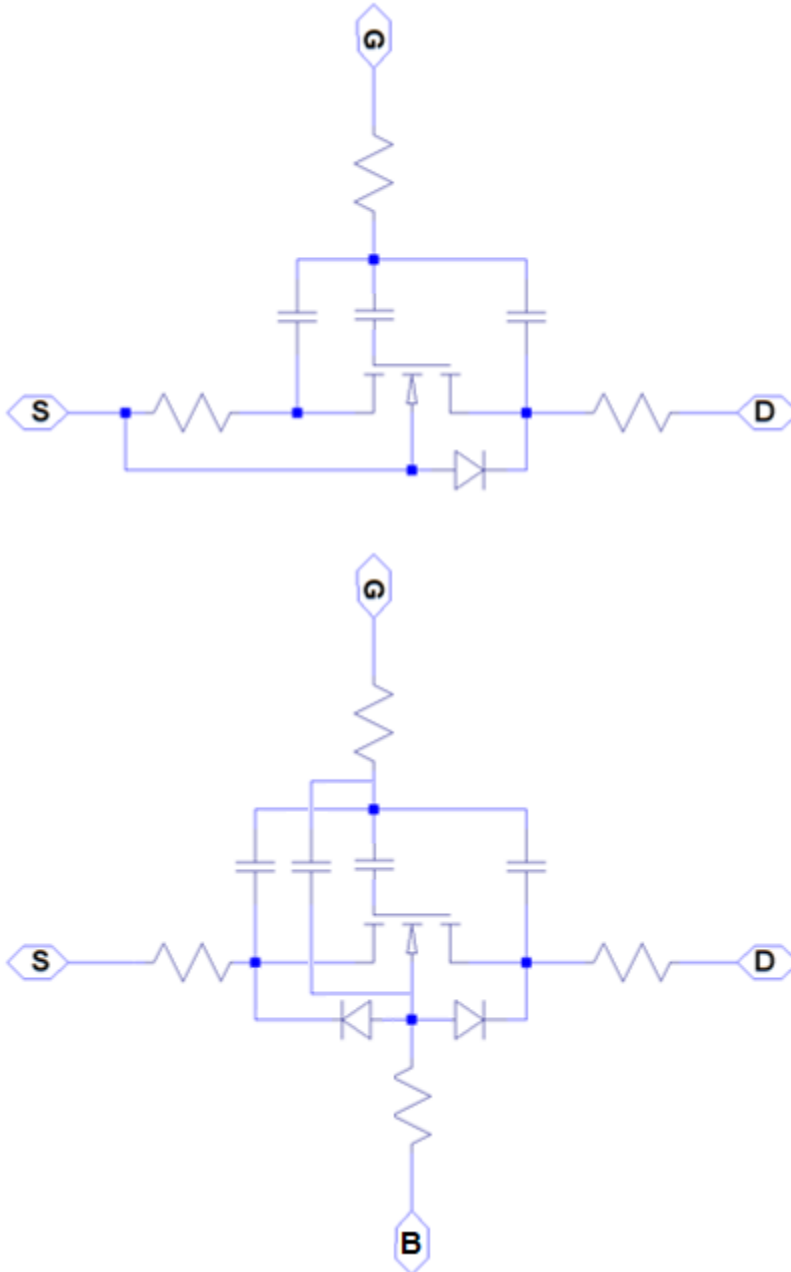
The threshold voltage for a short-circuited source-bulk connection is approximately given by

$$V_T = V_{FB} + 2\phi_B + 2\phi_T + \gamma\sqrt{2\phi_B + 2\phi_T}$$

where:

- $2\phi_B$  is the surface potential at strong inversion.

The overall three and four terminal models consist of an intrinsic MOSFET defined by the surface-potential formulation, a body diode, series resistances, and fixed overlap capacitances, as shown in the schematics.



### Modeling Body Diode

The block models the body diode either as an ideal, exponential diode or as a tabulated diode.

#### Exponential Diode

When you set **Model body diode** to **Exponential**, the junction and diffusion capacitances are:

$$I_{dio} = I_s \left[ \exp\left(-\frac{V_{DB}}{n\phi_T}\right) - 1 \right]$$

$$C_j = \frac{C_{j0}}{\sqrt{1 + \frac{V_{DB}}{V_{bi}}}}$$

$$C_{diff} = \frac{\tau I_s}{n\phi_T} \exp\left(-\frac{V_{DB}}{n\phi_T}\right)$$

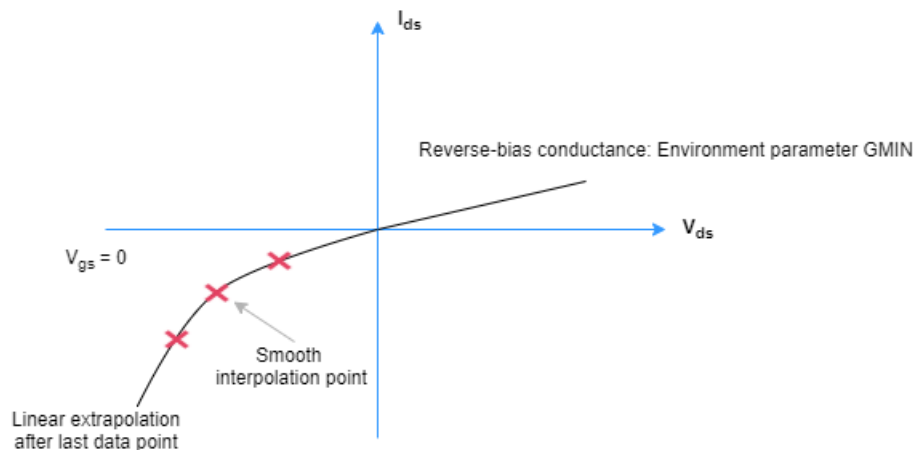
where:

- $I_{dio}$  is the current through the diode.
- $I_s$  is the reverse saturation current.
- $V_{DB}$  is the drain-body voltage.
- $n$  is the ideality factor.
- $\phi_T$  is the thermal voltage.
- $C_j$  is the junction capacitance of the diode.
- $C_{j0}$  is the zero-bias junction capacitance.
- $V_{bi}$  is the built-in voltage.
- $C_{diff}$  is the diffusion capacitance of the diode.
- $\tau$  is the transit time.

The capacitances are defined through an explicit calculation of charges, which are then differentiated to give the capacitive expressions above. The block computes the capacitive diode currents as time derivatives of the relevant charges, similar to the computation in the surface-potential-based MOSFET model.

### Tabulated Diode

To model a tabulated diode, set the **Model body diode** parameter to Tabulated I-V curve. This figure shows the implementation of the tabulated diode option:



When choosing this parameterization, you must provide the data for the forward bias only.

The block implements the diode using a smooth interpolation option. If the diode exceeds the provided tabulated data range, the block uses a linear extrapolation technique at the last voltage-current data point.



---

**Note** The tabulated diode does not model the reverse breakdown.

---

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. To model the dependence on temperature during simulation, select **Model temperature dependence** for the **Parameterization** parameter in the **Temperature Dependence** setting.

### Threshold-Based Model

For threshold-based modeling option, you can include modeling the dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The gain,  $K$ , and the threshold voltage,  $V_{th}$ , become a function of temperature according to the following equations:

$$K_{T_s} = K_{T_{m1}} \left( \frac{T_s}{T_{m1}} \right)^{BEX}$$

$$V_{th_s} = V_{th1} + \alpha (T_s - T_{m1})$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $K_{T_{m1}}$  is the transistor gain at the measurement temperature.
- $K_{T_s}$  is the transistor gain at the simulation temperature. This is the transistor gain value used in the MOSFET equations when temperature dependence is modeled.
- $V_{th1}$  is the threshold voltage at the measurement temperature.
- $V_{th_s}$  is the threshold voltage at the simulation temperature. This is the threshold voltage value used in the MOSFET equations when temperature dependence is modeled.
- $BEX$  is the mobility temperature exponent. A typical value of  $BEX$  is -1.5.
- $\alpha$  is the gate threshold voltage temperature coefficient,  $dV_{th}/dT$ .

For the four terminals parameterization,  $V_{th}$  is obtained using these equations:

$V_{BS}$ Range	$V_{th}$ Equation
$V_{BS} \leq 0$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma}{2\sqrt{2\phi_B}} \frac{d2\phi_B}{dT} + \frac{\gamma}{2\sqrt{2\phi_B - V_{BS}}} \frac{d2\phi_B}{dT}$
$0 < V_{BS} \leq 4\phi_B$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma V_{BS}}{4} (2\phi_B)^{-\frac{3}{2}} \frac{d2\phi_B}{dT}$
$V_{BS} > 4\phi_B$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma}{2\sqrt{2\phi_B}} \frac{d2\phi_B}{dT}$

Where:

- $\phi_B = \frac{kT}{q} \ln\left(\frac{N_B}{n_i}\right)$  is the surface potential and  $\frac{d^2\phi_B}{dT} = \frac{1}{T} \left[ 2\phi_B - \left( \frac{E_g(0)}{q} + \frac{3kT}{q} \right) \right]$ .
- $E_g(0)$  is the extrapolated zero degree band-gap, which is equal to 1.16 eV for silicon.
- $V_{BS}$  is the bulk-source voltage.

For most MOSFETS, you can use the default value of -1.5 for  $BEX$ . Some datasheets quote the value for  $\alpha$ , but most typically they provide the temperature dependence for drain-source on resistance,  $R_{DS(on)}$ . Depending on the block parameterization method, you have two ways of specifying  $\alpha$ :

- If you parameterize the block from a datasheet, you have to provide  $R_{DS(on)}$  at a second measurement temperature. The block then calculates the value for  $\alpha$  based on this data.
- If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

If you have more data comprising drain current as a function of gate-source voltage for more than one temperature, then you can also use Simulink Design Optimization software to help tune the values for  $\alpha$  and  $BEX$ .

### Surface-Potential-Based Model

The surface-potential-based model includes temperature effects on the capacitance characteristics, as well as modeling the dependence of the transistor static behavior on temperature during simulation.

The **Measurement temperature** parameter on the **Main** setting specifies temperature  $T_{m1}$  at which the other device parameters have been extracted. The **Temperature Dependence** setting provides the simulation temperature,  $T_s$ , and the temperature-scaling coefficients for the other device parameters. For more information, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1115.

### Faults

The N-Channel MOSFET block models five types of fault:

- `Open circuit` — Failure due to metallization burnout
- `Drain-source short` — Failure due to avalanche breakdown on drain-source channel
- `Drain-bulk short or source-bulk short` — Failure due to avalanche breakdown on drain-bulk or source-bulk channels
- `Gate oxide short` — Failure of the gate oxide dielectric layer
- `Parameter shift` — Failure due to aging

The block can trigger fault events at a specific time or when the current or voltage exceed the limit for longer than a specific time interval.

To enable these trigger mechanisms, set the **Fault trigger** parameter to either `Temporal` or `Behavioral`.

#### Parameter Shift Fault

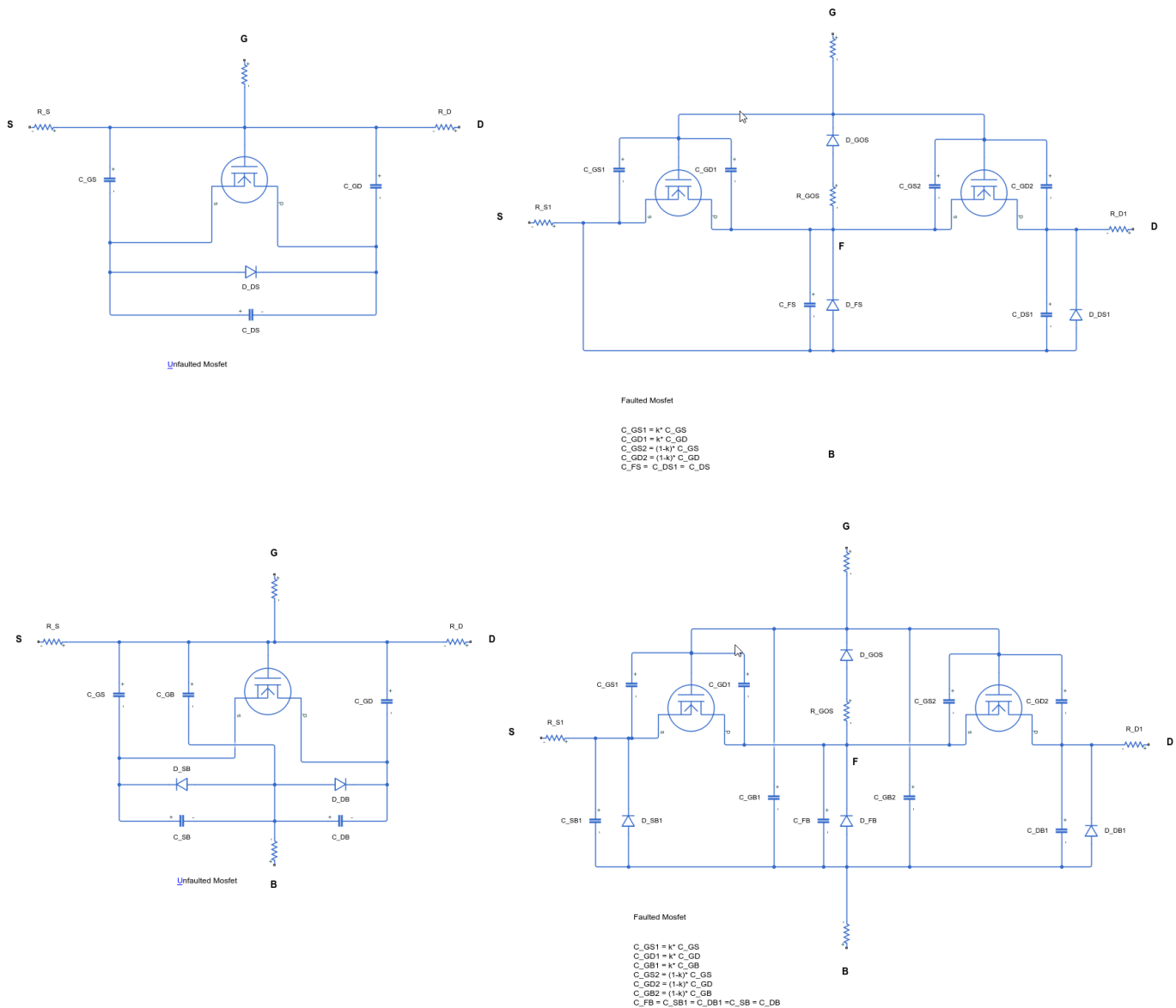
If, in the **Faults** settings, you set the **Failure mode** parameter to `Parameter shift`, the MOSFET fails due to the aging of its components. This equation defines the value of the shifted parameters:

$$Parameter(t) = Parameter_{faulted} - (Parameter_{faulted} - Parameter_{unfaulted}) \operatorname{sech}\left(\frac{t - t_{th}}{\tau}\right),$$

where  $t_{th}$  is the time threshold when the fault is triggered and  $\tau$  is the value of the **Fault transition time constant, tau** parameter.

### Gate Oxide Short Fault

If, in the **Faults** settings, you set the **Failure mode** parameter to Gate oxide short, the gate oxide dielectric layer fails. These figures show the equivalent circuits for the three terminal or four terminal MOSFET in the unfaulted and faulted state:



### Thermal Port

The block has an optional thermal port, hidden by default. To expose the thermal port, set the **Modeling option** parameter to:

- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port
- **Surface-potential-based with thermal** — Model based on surface potential and with exposed thermal port

This action displays the thermal port **H** on the block icon, and exposes the **Thermal Port** parameters.

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

This table shows the relationship between the capacitances of the block and the initial targets:

Defined Capacitance	Initial Targets
Gate-emitter capacitance, $C_{ge}$	Set the initial target for the gate-emitter capacitance voltage only. Set the initial target of the collector-emitter capacitance voltage to 0 or set its priority to None.
Collector-emitter capacitance, $C_{ce}$	Set the initial target for the collector-emitter capacitance voltage only. Set the initial target of the gate-emitter capacitance voltage to 0 or set its priority to None.
Gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-emitter and gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial target for the gate-emitter and the collector-emitter capacitance.

Defined Capacitance	Initial Targets
Gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-collector and collector-emitter voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-emitter, gate collector and collector-emitter capacitances by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.

**Note** Inside your model, the number of initial targets with **Priority** equal to Low or High must match the number of differential variables. The differential variables come from the inductors and the capacitances in the model.

## Assumptions and Limitations

When modeling temperature dependence for the threshold-based block modeling option, consider the following:

- The block does not account for temperature-dependent effects on the capacitances.
- When you specify  $R_{DS(on)}$  at a second measurement temperature, it must be quoted for the same working point (that is, the same drain current and gate-source voltage) as for the other  $R_{DS(on)}$  value. Inconsistent values for  $R_{DS(on)}$  at the higher temperature will result in unphysical values for  $\alpha$  and unrepresentative simulation results. Typically  $R_{DS(on)}$  increases by a factor of about 1.5 for a hundred degree increase in temperature.
- You may need to tune the values of  $BEX$  and threshold voltage,  $V_{th}$ , to replicate the  $I_{DS}-V_{GS}$  relationship (if available) for a given device. Increasing  $V_{th}$  moves the  $I_{DS}-V_{GS}$  plots to the right. The value of  $BEX$  affects whether the  $I_{DS}-V_{GS}$  curves for different temperatures cross each other, or not, for the ranges of  $V_{DS}$  and  $V_{GS}$  considered. Therefore, an inappropriate value can result in the different temperature curves appearing to be reordered. Quoting  $R_{DS(on)}$  values for higher currents, preferably close to the current at which it will operate in your circuit, will reduce sensitivity to the precise value of  $BEX$ .

## Ports

### Conserving

#### G – Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal

#### D – Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal

**S – Source terminal**

electrical

Electrical conserving port associated with the transistor source terminal

**B – Body terminal**

electrical

Electrical conserving port associated with the transistor body terminal

**Dependencies**

To enable this port, set **Parameterization** to Four.

**Parameters****Modeling Option – MOSFET representation**

Threshold-based (default) | Threshold-based with thermal | Surface-potential-based | Surface-potential-based with thermal

Select the MOSFET representation:

- **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.
- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port.
- **Surface-potential-based** — Model based on surface potential. This model does not simulate thermal effects.
- **Surface-potential-based with thermal** — Thermal modeling option of the model based on surface potential.

**Main (Threshold-Based Modeling Option)**

This configuration of the **Main** parameters corresponds to the threshold-based block modeling option, which is the default. If you are using the surface-potential-based modeling option of the block, see “Main (Surface-Potential-Based Modeling Option)” on page 1-1102.

**Number of terminals – Terminal parameterization**

Three (default) | Four

Number of terminals of the block.

**Parameterization – Block parameterization**

Specify from a datasheet (default) | Specify using equation parameters directly | Lookup table (2-D, temperature independent) | Lookup table (3-D, temperature dependent)

Select one of the following methods for block parameterization:

- **Specify from a datasheet** — Provide the drain-source on resistance and the corresponding drain current and gate-source voltage. The block calculates the transistor gain for the Shichman and Hodges equations from this information.

- Specify using equation parameters directly — Provide the transistor gain.
- Lookup table (2-D, temperature independent) — Use 2-D table lookup for drain-source current as a function of gate-source voltage and drain-source voltage.
- Lookup table (3-D, temperature dependent) — Use 3-D table lookup for drain-source current as a function of gate-source voltage, drain-source voltage, and temperature.

#### **Drain-source on resistance, $R_{DS(on)}$ — Drain-source on resistance**

0.025 Ohm (default)

Ratio of the drain-source voltage to the drain current for specified values of drain current and gate-source voltage.  $R_{DS(on)}$  should have a positive value.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.

#### **Drain current, $I_{DS}$ , for $R_{DS(on)}$ — Drain current**

6A (default)

Drain current the block uses to calculate the value of the drain-source resistance.  $I_{DS}$  should have a positive value.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.

#### **Gate-source voltage, $V_{GS}$ , for $R_{DS(on)}$ — Gate-source voltage, $V_{GS}$**

10 V (default)

Gate-source voltage the block uses to calculate the value of the drain-source resistance.  $V_{GS}$  should have a positive value.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.

#### **Gain, $K$ — Positive constant gain coefficient**

5.0 A/V<sup>2</sup> (default)

Positive constant gain coefficient for the Shichman and Hodges equations.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify using equation parameters directly.

#### **Gate-source threshold voltage, $V_{th}$ — Gate-source threshold voltage**

1.7 V (default)

Gate-source threshold voltage  $V_{th}$  in the Shichman and Hodges equations. For an enhancement device,  $V_{th}$  should be positive. For a depletion mode device,  $V_{th}$  should be negative.

#### **Dependencies**

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Gate-source threshold voltage at zero bulk-source voltage, Vth0 — Gate-source threshold voltage at zero bulk-source voltage**

1.7 V (default)

Gate-source threshold voltage at zero bulk-source voltage  $V_{th0}$  in the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Channel modulation, L — Channel modulation**

0 1/V (default)

The channel-length modulation, usually denoted by the mathematical symbol  $\lambda$ . When in the saturated region, it is the rate of change of drain current with drain-source voltage. The effect on drain current is typically small, and the effect is neglected if calculating transistor gain  $K$  from drain-source on-resistance,  $R_{DS(on)}$ . A typical value is 0.02, but the effect can be ignored in most circuit simulations. However, in some circuits a small nonzero value may help numerical convergence.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Gate-source threshold voltage at first non-zero bulk-source voltage, Vth1 — Gate-source threshold voltage at first non-zero bulk-source voltage**

1.9071 V (default)

Gate-source threshold voltage at first non-zero bulk-source voltage  $V_{th1}$  in the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**First bulk-source voltage, Vbs1 — First bulk-source voltage**

-1 V (default)

First bulk-source voltage,  $V_{bs1}$

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Gate-source threshold voltage at second non-zero bulk-source voltage, Vth2 — Gate-source threshold voltage at second non-zero bulk-source voltage**

2.066 V (default)

Gate-source threshold voltage at second non-zero bulk-source voltage  $V_{th2}$  in the Shichman and Hodges equations.



**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Second bulk-source voltage, V<sub>bs2</sub> — Second bulk-source voltage**

-2 V (default)

Second bulk-source voltage,  $V_{bs2}$

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Body factor — Body factor**

$0.5 \text{ V}^{0.50000}$  (default)

Body factor,  $\gamma$ .

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify using equation parameters directly.

**Surface potential — Surface potential**

0.5 V (default)

Surface potential.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify using equation parameters directly.

**Measurement temperature — Measurement temperature**

25 °C (default)

Temperature  $T_{m1}$  at which **Drain-source on resistance, R<sub>DS(on)</sub>** is measured.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Vector of gate-source voltages, V<sub>gs</sub> — Vector of gate-source voltages**

[-.5, 2, 3, 4, 5] V (default)

Vector of gate-source voltages.

**Dependencies**

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

**Vector of drain-source voltages, V<sub>ds</sub> — Vector of drain-source voltages**

[0, 2, 4, 8, 12, 16, 20] V (default)

Vector of drain-source voltages, to be used for table lookup. The vector values must be strictly increasing.

#### Dependencies

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

#### Vector of bulk-source voltages, $V_{bs}$ — Vector of drain-source voltages

[0, -4, -6, -12] V (default)

Vector of bulk-source voltages, to be used for table lookup.

#### Dependencies

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

#### Tabulated drain-source currents, $I_{ds}(V_{gs}, V_{ds})$ — Tabulated drain-source currents temperature independent

```
[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;
0, .0055, .008, .009, .0095, .01, .0101;
0, .0115, .016, .018, .019, .0195, .02;
0, .0152, .0225, .0265, .028, .0295, .03] A (default)
```

Tabulated values for drain-source currents as a function of gate-source voltage and drain-source voltage, to be used for 2-D table lookup. Each value in the matrix specifies the drain-source current for a specific combination of gate-source voltage and drain-source voltage. The matrix size must match the dimensions defined by the gate-source voltage and drain-source voltage vectors.

#### Dependencies

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Lookup table (2-D, temperature independent).

#### Vector of temperatures, $T$ — Vector of temperatures

[25 125] degC (default)

Vector of temperatures, to be used for table lookup. The vector values must be strictly increasing. The values can be nonuniformly spaced.

#### Dependencies

To enable this parameter, set **Parameterization** to Lookup table (3-D, temperature dependent).

#### Tabulated drain-source currents, $I_{ds}(V_{gs}, V_{ds}, T)$ — Tabulated drain-source currents temperature dependent

zeros(5, 7, 2) A (default)

Tabulated values for drain-source currents as a function of gate-source voltage, drain-source voltage, and temperature, to be used for 2-D table lookup. Each value in the matrix specifies the drain-source current for a specific combination of gate-source voltage and drain-source voltage. The matrix size must match the dimensions defined by the gate-source voltage, drain-source voltage, and temperature vectors.

**Dependencies**

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Lookup table (3-D, temperature dependent).

**Tabulated gate-source threshold voltage,  $V_{th}(V_{bs})$  — Vector of gate-source threshold voltage temperature independent**

[.4, 1.2, 1.4, 1.45] V (default)

Tabulated values for gate-source threshold voltage as a function of bulk-source voltage, to be used for 2-D table lookup. The vector values must be strictly increasing.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (2-D, temperature independent).

**Tabulated drain-source currents at zero bulk-source voltage,  $I_{ds}(V_{gs}, V_{ds})$  — Tabulated drain-source currents at zero bulk-source voltage, temperature independent**

[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;  
0, .0115, .016, .018, .019, .0195, .02;  
0, .0152, .0225, .0265, .028, .0295, .03] A (default)

Tabulated values for drain-source currents at zero bulk-source voltage, as a function of gate-source voltage and drain-source voltage, to be used for 2-D table lookup. Each value in the matrix specifies the drain-source current for a specific combination of gate-source voltage and drain-source voltage. The matrix size must match the dimensions defined by the gate-source voltage and drain-source voltage vectors.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (2-D, temperature independent).

**Tabulated gate-source threshold voltage,  $V_{th}(V_{bs}, T)$  — Vector of gate-source threshold voltage temperature dependent**

[.4, .35; 1.2, 1.15; 1.4, 1.35; 1.45, 1.4] V (default)

Tabulated values for gate-source threshold voltage as a function of bulk-source voltage and temperature, to be used for 3-D table lookup. The vector values must be strictly increasing.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (3-D, temperature dependent).

**Tabulated drain-source currents at zero bulk-source voltage,  $I_{ds}(V_{gs}, V_{ds}, T)$  — Tabulated drain-source currents at zero bulk-source voltage, temperature dependent**

zeros(5, 7, 2) A (default)

Tabulated values for drain-source currents at zero bulk-source voltage, as a function of gate-source voltage, drain-source voltage, and temperature, to be used for 3-D table lookup. Each value in the matrix specifies the drain-source current for a specific combination of gate-source voltage, drain-source voltage, and temperature. The matrix size must match the dimensions defined by the gate-source voltage, drain-source voltage, and temperature vectors.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (3-D, temperature dependent).

**Ids-Vds parameterization — Ids-Vds table data as symmetric data**

Provide negative and positive Vds data (default) | Provide positive Vds data only

Whether to provide Ids-Vds table data as symmetric data. If you choose Provide negative and positive Vds data, the data is not symmetric. If you choose Provide positive Vds data only, the block rotates and flips the positive data to obtain the negative data.

**Dependencies**

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

**Main (Surface-Potential-Based Modeling Option)**

This configuration of the **Main** setting corresponds to the surface-potential-based block modeling option. If you are using the threshold-based modeling option of the block, based on the Shichman and Hodges equations, see “Main (Threshold-Based Modeling Option)” on page 1-1096.

**Number of terminals — Terminal parameterization**

Three (default) | Four

Number of terminals of the block.

**Gain — MOSFET gain**

18 A/V<sup>2</sup> (default) | positive scalar

MOSFET gain,  $\beta$ . This parameter primarily defines the linear region of operation on an  $I_D$ - $V_{DS}$  characteristic.

**Flatband voltage — Flatband voltage**

-1.1 V (default)

Flatband voltage,  $V_{FB}$ , defines the gate bias that must be applied in order to achieve the flatband condition at the surface of the silicon. The default value is -1.1 V. You can also use this parameter to arbitrarily shift the threshold voltage due to material work function differences, and to trapped interface or oxide charges. In practice, however, it is usually recommended to modify the threshold voltage by using the **Body factor** and **Surface potential at strong inversion** parameters first, and only use this parameter for fine-tuning.

**Body factor — Body factor**

3.5 V<sup>1/2</sup> (default)

Body factor,  $\gamma$ , in the surface-potential equation. This parameter primarily impacts the threshold voltage.

**Surface potential at strong inversion — Surface potential at strong inversion**

1 V (default)

$2\phi_B$  term in the surface-potential equation. This parameter also primarily impacts the threshold voltage.

**Velocity saturation factor – Velocity saturation factor**

0.4 1/V (default)

Velocity saturation,  $\theta_{sat}$ , in the drain-current equation. Use this parameter in cases where a good fit to linear operation leads to a saturation current that is too high. By increasing this parameter value, you reduce the saturation current. For high-voltage devices, it is often the case that a good fit to linear operation leads to a saturation current that is too low. In such a case, either increase both the gain and the drain ohmic resistance or use an N-Channel LDMOS FET block instead.

**Channel-length modulation factor – Channel-length modulation factor**

0 (default)

Factor,  $\alpha$ , multiplying the logarithmic term in the  $G_{\Delta L}$  equation. This parameter describes the onset of channel-length modulation. For device characteristics that exhibit a positive conductance in saturation, increase the parameter value to fit this behavior. The default value is 0, which means that channel-length modulation is off by default.

**Channel-length modulation voltage – Channel-length modulation voltage**

5e-2 V (default)

Voltage  $V_p$  in the  $G_{\Delta L}$  equation. This parameter controls the drain-voltage at which channel-length modulation starts to become active

**Surface roughness scattering factor – Surface roughness scattering factor**

0 1/V (default)

Strength of the mobility reduction. The mobility is  $\mu = \mu_0/G_{mob}$ , where  $\mu_0$  is the low-field mobility without the effect of surface scattering. The mobility reduction factor,  $G_{mob}$ , is given by

$G_{mob} = \sqrt{1 + (\theta_{sr}V_{eff})^4}$ , where  $\theta_{sr}$  is the surface roughness scattering factor and  $V_{eff}$  is a voltage that is indicative of the effective vertical electric field strength in the channel,  $E_{eff}$ . For high vertical electric fields, the mobility is roughly proportional to  $E_{eff}^2$  for electrons.

**Linear-to-saturation transition coefficient – Linear-to-saturation transition coefficient**

8 (default)

Smoothness of the MOSFET transitions from linear into saturation, particularly when velocity saturation is enabled. This parameter can usually be left at its default value, but you can use it to fine-tune the knee of the  $I_D$ - $V_{DS}$  characteristic. The expected range for this parameter value is between 2 and 8.

**Measurement temperature – Measurement temperature**

25 °C (default)

Temperature  $T_{m1}$  at which the block parameters are measured. If the **Device simulation temperature** parameter on the **Temperature Dependence** setting differs from this value, then device parameters will be scaled from their defined values according to the simulation and reference temperatures. For more information, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1115.

## Ohmic Resistance

If you set **Modeling option** to Threshold-based or Threshold-based with thermal, to enable this setting set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

### Source ohmic resistance, $R_s$ — Transistor source resistance

0.0001 Ohm (default) | nonnegative scalar

Transistor source resistance, that is, the series resistance associated with the source contact. The default value for threshold-based variants is  $1e-4$  Ohm. The default value for surface-potential-based variants is  $2e-3$  Ohm.

### Drain ohmic resistance, $R_d$ — Transistor drain resistance

0.01 Ohm (default) | nonnegative scalar

Transistor drain resistance, that is, the series resistance associated with the drain contact. The value must be greater than or equal to 0. The default value for threshold-based variants is 0.01 Ohm. The default value for surface-potential-based variants is 0.17 Ohm.

### Gate ohmic resistance, $R_g$ — Transistor gate resistance

8.4 Ohm (default) | nonnegative scalar

Transistor gate resistance, that is, the series resistance associated with the gate contact.

#### Dependencies

To enable this parameter, set **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

### Body ohmic resistance, $R_b$ — Transistor body resistance

0.001 Ohm (default) | nonnegative scalar

Transistor body resistance, that is, the series resistance associated with the body contact.

#### Dependencies

To enable this parameter, set:

- **Number of terminals** to Four.
- **Modeling option** to Threshold-based or Threshold-based with thermal.

### Bulk ohmic resistance — Transistor bulk resistance

$2e-3$  Ohm (default) | nonnegative scalar

Transistor body resistance, that is, the series resistance associated with the bulk contact.

#### Dependencies

To enable this parameter, set:

- **Number of terminals** to Four.
- **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

## Capacitance

To enable this setting, set **Modeling option** to Threshold-based or Threshold-based with thermal.

### Parameterization — Capacitance parameterization

Specify fixed input, reverse transfer, and output capacitance (default) | Specify fixed gate-source, gate-drain, and drain-source capacitance | Specify tabulated input, reverse transfer, and output capacitance | Specify tabulated gate-source, gate-drain, and drain-source capacitance

Select one of the following methods for capacitance parameterization:

- **Specify fixed input, reverse transfer, and output capacitance** — Provide fixed parameter values from datasheet and let the block convert the input, output, and reverse transfer capacitance values to capacitance values, as described in “Charge Model for Threshold-Based Modeling Option” on page 1-1080.
- **Specify fixed gate-source, gate-drain, and drain-source capacitance** — Provide fixed values for capacitance parameters directly.
- **Specify tabulated input, reverse transfer, and output capacitance** — Provide tabulated capacitance and drain-source voltage values based on datasheet plots. The block converts the input, output, and reverse transfer capacitance values to capacitance values, as described in “Charge Model for Threshold-Based Modeling Option” on page 1-1080.
- **Specify tabulated gate-source, gate-drain, and drain-source capacitance** — Provide tabulated values for capacitances and drain-source voltage.

### Input capacitance, $C_{iss}$ — Input gate-source capacitance

350 pF (default)

Gate-source capacitance with the drain shorted to the source.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

### Input capacitance, $C_{iss}(V_{ds})$ — Input gate-source capacitance with tabulated input

[720 700 590 470 390 310] pF (default)

Gate-source capacitance with the drain shorted to the source, tabulated with drain-source voltage.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

### Reverse transfer capacitance, $C_{rss}$ — Reverse transfer capacitance

80 pF (default)

Drain-gate capacitance with the source connected to ground, also known as the Miller capacitance.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Reverse transfer capacitance, Crss(Vds) – Reverse transfer capacitance with tabulated inputs**

[450 400 300 190 95 55] pF (default)

Drain-gate capacitance with the source connected to ground, also known as the Miller capacitance, tabulated with drain-source voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Output capacitance, Coss – Output capacitance**

0 pF (default)

Drain-source capacitance with the gate and source shorted.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Output capacitance, Coss(Vds) – Output capacitance with tabulated inputs**

[900 810 690 420 270 170] pF (default)

Drain-source capacitance with the gate and source shorted, tabulated with drain-source voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Gate-source capacitance, Cgs – Gate-source capacitance**

270 pF (default)

Capacitance placed between the gate and the source.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

**Gate-source capacitance, Cgs(Vgs, Vds) – Gate-source capacitance with tabulated values**

[270 300 290 280 295 255] pF (default)

Capacitance placed between the gate and the source, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

**Gate-drain capacitance, Cgd – Gate-drain capacitance**

80 pF (default)

Capacitance placed between the gate and the drain.



**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

**Gate-drain capacitance,  $C_{gd}(V_{gs}, V_{ds})$  — Gate-drain capacitance with tabulated values**  
[450 400 300 190 95 55] pF (default)

Capacitance placed between the gate and the drain, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

**Gate-bulk capacitance,  $C_{gb}$  — Gate-bulk capacitance**  
0 pF (default)

Capacitance placed between the gate and the bulk.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance and **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.

**Gate-bulk capacitance,  $C_{gb}(V_{gs}, V_{ds})$  — Gate-bulk capacitance with tabulated values**  
[450 400 300 190 95 55] pF (default)

Capacitance placed between the gate and the bulk, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance and **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.

**Corresponding gate-source voltages,  $V_{gs}$  — Corresponding gate-source voltages**  
0 V (default)

Gate-source voltages corresponding to the tabulated capacitance values.

If this value is a scalar, then **Gate-source capacitance,  $C_{gs}(V_{gs}, V_{ds})$**  and **Gate-drain capacitance,  $C_{gd}(V_{gs}, V_{ds})$**  are vectors of scalars. If this value is a vector, then **Gate-source capacitance,  $C_{gs}(V_{gs}, V_{ds})$**  and **Gate-drain capacitance,  $C_{gd}(V_{gs}, V_{ds})$**  are matrices of scalars.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

**Drain-source capacitance,  $C_{ds}$  — Drain-source capacitance**  
0 pF (default)

Capacitance placed between the drain and the source.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

#### Drain-source capacitance, $C_{ds}(v_{ds})$ — Drain-source capacitance with tabulated values [450 410 390 230 175 115] pF (default)

Capacitance placed between the drain and the source, tabulated with the drain-source voltage.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

#### Corresponding drain-source voltages, $V_{ds}$ — Corresponding drain-source voltages [.1, .3, 1, 3, 10, 30] V (default)

Drain-source voltages corresponding to the tabulated capacitance values.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

#### Gate-source voltage, $V_{gs}$ , for tabulated capacitances — Gate-source voltage for tabulated capacitances 0 V (default)

For tabulated capacitance models, this parameter controls the voltage dependence of the **Reverse transfer capacitance,  $C_{rss}$**  or the **Gate-drain capacitance,  $C_{gd}$**  parameter (depending on the selected parameterization option). These capacitances are a function of the drain-gate voltage. The block calculates drain-gate voltages by subtracting this gate-source voltage value from the values specified for the **Corresponding drain-source voltages** parameter.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

#### Gate-drain charge-voltage linearity — Charge-voltage linearity Gate-drain capacitance is constant (default) | Gate-drain charge function is nonlinear

The two fixed capacitance options let you model gate capacitance as a fixed gate-source capacitance  $C_{GS}$  and either a fixed or a nonlinear gate-drain capacitance  $C_{GD}$ . Select whether the gate-drain capacitance is fixed or nonlinear:

- **Gate-drain capacitance is constant** — The capacitance value is constant and defined according to the selected parameterization option, either directly or derived from a datasheet.
- **Gate-drain charge function is nonlinear** — The gate-drain charge relationship is defined according to the piecewise-nonlinear function described in “Charge Model for Threshold-Based Modeling Option” on page 1-1080. Two additional parameters appear to let you define the gate-drain charge function.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer and output capacitance or to Specify fixed gate-source, gate-drain and drain-source capacitance.

**Gate-drain oxide capacitance — Gate-drain oxide capacitance**

200 pF (default)

Gate-drain capacitance when the drain-gate voltage is less than the **Drain-gate voltage at which oxide capacitance becomes active** parameter value.

**Dependencies**

To enable this parameter, set **Gate-drain charge-voltage linearity** to Gate-drain charge function is nonlinear.

**Drain-gate voltage at which oxide capacitance becomes active — Drain-gate voltage at which oxide capacitance becomes active**

-0.5 V (default)

Drain-gate voltage at which the drain-gate capacitance switches between off-state ( $C_{GD}$ ) and on-state ( $C_{ox}$ ) capacitance values.

**Dependencies**

To enable this parameter, set **Gate-drain charge-voltage linearity** to Gate-drain charge function is nonlinear.

**Gate-bulk and gate-source charge-voltage linearity — Gate-bulk and gate-source charge-voltage linearity**

Gate-bulk and gate-source capacitance change instantly (default) | Gate-bulk and gate-source capacitance change gradually | Separate gate-bulk and gate-source capacitance

Gate-bulk and gate-source charge-voltage linearity.

**Dependencies**

To enable this parameter, in the **Main** setting, set **Number of terminals** to Four.

To enable the Separate gate-bulk and gate-source capacitance option, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance

**Channel Capacitances**

To enable this setting, set **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

**Oxide capacitance — Oxide capacitance**

1500 pF (default)

The parallel plate gate-channel capacitance.

**Gate-source overlap capacitance – Gate-source overlap capacitance**

100 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the source well.

**Gate-drain overlap capacitance – Gate-drain overlap capacitance**

14 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the drain well.

**Body Diode****Model body diode – Body diode modeling option**

No (default) | Exponential | Tabulated I-V curve

Whether to model the body diode.

**Table type – Tabulated function**Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

To enable this parameter, set **Model body diode** to Tabulated I-V curve.

**Reverse saturation current,  $I_s$  – Reverse saturation current**

0 A (default)

The current designated by the  $I_s$  symbol in the body-diode equations. The default value for threshold-based modeling option is 0 A. The default value for surface-potential-based modeling option is  $5.2e-13$  A.

To enable conduction through the body diode, for applications where the MOSFET current changes sign during the simulation, such as when the MOSFET is driving an inductive load, set this parameter to a non-zero value.

For applications where the MOSFET current never changes sign, such as in a small-signal amplifier, set this parameter to 0 to improve simulation speed.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Built-in voltage,  $V_{bi}$  – Built-in voltage**

0.6 V (default)

The built-in voltage of the diode, designated by the  $V_{bi}$  symbol in the body-diode equations. Built-in voltage has an impact only on the junction capacitance equation. It does not affect the conduction current.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Ideality factor, n – Ideality factor**

1 (default)

The factor designated by the  $n$  symbol in the body-diode equations.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Zero-bias junction capacitance, Cj0 – Zero-bias junction capacitance**

0 pF (default)

The capacitance between the drain and bulk contacts at zero-bias due to the body diode alone. It is designated by the  $C_{j0}$  symbol in the body-diode equations. The default value for threshold-based modeling option is 0 pF. The default value for surface-potential-based modeling option is 480 pF.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Transit time, TT – Transit time**

50e-9 s (default)

The time designated by the  $\tau$  symbol in the body-diode equations.

When the **Reverse saturation current** and **Transit time** parameters are both non-zero, this block includes the reverse recovery inside the body diode model.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Measurement temperature – Measurement temperature**

25 °C (default)

Temperature at which the block parameters are measured.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential and, in the **Main** setting, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent)

**Diode forward voltages, Vf(If) – Diode forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85] V (default)

Tabulated values of the minimum voltage that needs to be applied for the diode to become forward-biased.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify from equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body diode** setting, set:

- **Model body diode** to Tabulated I-V curve.

**Table type** to Table in Vf(If) form.

#### **Diode forward currents, If — Diode forward currents temperature independent**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Tabulated values of the forward currents.

#### **Dependencies**

To enable this parameter, in the **Body diode** setting, set **Model body diode** to Tabulated I-V curve and **Table type** to Table in Vf(If) form or Table in Vf(Tj,If) form.

#### **Diode forward currents, If(Vf) — Diode forward currents tabulated with forward voltages**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2] A (default)

Tabulated values of the forward current, as a function of the forward voltages.

#### **Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify from equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in If(Vf) form.

#### **Diode forward voltages, Vf — Diode forward voltages temperature independent**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Tabulated values of the forward voltages.

#### **Dependencies**

To enable this parameter, in the **Body diode** setting, set **Model body diode** to Tabulated I-V curve and **Table type** to Table in If(Vf) form or Table in If(Tj,Vf) form.

#### **Diode junction temperatures, Tj — Diode junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures.

#### **Dependencies**

To enable this parameter, set **Model body diode** to Tabulated I-V curve and, in the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).

#### **Diode forward currents, If(Tj,Vf) — Diode forward currents temperature dependent**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default)

Tabulated values of the forward current, as a function of the forward voltages and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in  $I_f(T_j, V_f)$  form.

**Diode forward voltages,  $I_f(T_j, V_f)$  – Diode forward voltages temperature dependent**  
 [.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default)

Tabulated values of the forward voltage, as a function of the forward currents and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in  $V_f(T_j, I_f)$  form.

**Temperature Dependence (Threshold-Based Modeling Option)**

This configuration of the **Temperature Dependence** setting corresponds to the threshold-based block modeling option, which is the default. If you are using the surface-potential-based modeling option of the block, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1115

**Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled. This is the default method.
- Model temperature dependence – Model temperature-dependent effects. Provide a value for simulation temperature,  $T_s$ , a value for  $BEX$ , and a value for the measurement temperature  $T_{m1}$  (using the **Measurement temperature** parameter on the **Main** setting). You also have to provide a value for  $\alpha$  using one of two methods, depending on the value of the **Parameterization** parameter on the **Main** setting. If you parameterize the block from a datasheet, you have to provide  $R_{DS(on)}$  at a second measurement temperature, and the block will calculate  $\alpha$  based on that. If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

**Drain-source on resistance, R\_DS(on), at second measurement temperature – Drain-source on resistance at second measurement temperature**

0.037 Ohm (default)

The ratio of the drain-source voltage to the drain current for specified values of drain current and gate-source voltage at second measurement temperature. It must be quoted for the same working point (drain current and gate-source voltage) as the **Drain-source on resistance, R\_DS(on)** parameter on the **Main** setting.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** setting.

**Second measurement temperature – Second temperature**

125 °C (default)

Second temperature  $T_{m2}$  at which **Drain-source on resistance, R\_DS(on), at second measurement temperature** is measured.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** setting.

**Gate threshold voltage temperature coefficient, dVth/dT – Gate-source voltage, Vgs**

-6 mV/K (default)

The rate of change of gate threshold voltage with temperature.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** setting.

**Mobility temperature exponent, BEX – Mobility temperature exponent**

-1.5 (default)

Mobility temperature coefficient value. You can use the default value for most MOSFETs. See the "Assumptions and Limitations" on page 1-1095 section for additional considerations.

**Body diode reverse saturation current temperature exponent – Body diode reverse saturation current temperature exponent**

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$



$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** setting,  $k_B$  is Boltzmann's constant ( $8.617 \times 10^{-5} \text{ eV/K}$ ), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

### Device simulation temperature – Device simulation temperature

25 °C (default)

Temperature  $T_s$  at which the device is simulated.

### Temperature Dependence (Surface-Potential-Based Modeling Option)

This configuration of the **Temperature Dependence** setting corresponds to the surface-potential-based block modeling option. If you are using the threshold-based modeling option of the block, see “Temperature Dependence (Threshold-Based Modeling Option)” on page 1-1113

### Parameterization – Temperature dependence parameterization

None – Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled.
- Model temperature dependence – Model temperature-dependent effects. Provide a value for the device simulation temperature,  $T_s$ , and the temperature-scaling coefficients for other block parameters.

### Gain temperature exponent – Gain temperature exponent

1.3 (default)

The MOSFET gain,  $\beta$ , is assumed to scale exponentially with temperature,  $\beta = \beta_{m1} (T_{m1}/T_s)^{\eta_\beta}$ .  $\beta_{m1}$  is the value of the **Gain** parameter from the **Main** setting and  $\eta_\beta$  is the **Gain temperature exponent**.

### Flatband voltage temperature coefficient – Flatband voltage temperature coefficient

$5 \times 10^{-4} \text{ V/K}$  (default)

The flatband voltage,  $V_{FB}$ , is assumed to scale linearly with temperature,  $V_{FB} = V_{FBm1} + (T_s - T_{m1})S_{TV_{FB}}$ .  $V_{FBm1}$  is the value of the **Flatband voltage** parameter from the **Main** setting and  $S_{TV_{FB}}$  is the **Flatband voltage temperature coefficient**.

### Surface potential at strong inversion temperature coefficient – Surface potential at strong inversion temperature coefficient

$-8.5 \times 10^{-4} \text{ V/K}$  (default)

The surface potential at strong inversion,  $2\phi_B$ , is assumed to scale linearly with temperature,  $2\phi_B = 2\phi_{Bm1} + (T_s - T_{m1})S_{T\phi_B}$ .  $2\phi_{Bm1}$  is the value of the **Surface potential at strong inversion** parameter from the **Main** setting and  $S_{T\phi_B}$  is the **Surface potential at strong inversion temperature coefficient**.

### Velocity saturation temperature exponent – Velocity saturation temperature exponent

1.04 (default)

The velocity saturation,  $\theta_{sat}$ , is assumed to scale exponentially with temperature,  $\theta_{sat} = \theta_{sat,m1} (T_{m1}/T_s)^{\eta_\theta}$ .  $\theta_{sat,m1}$  is the value of the **Velocity saturation factor** parameter from the **Main** setting and  $\eta_\theta$  is the **Velocity saturation temperature exponent**.

### Surface roughness scattering temperature exponent – Surface roughness scattering temperature exponent

0.65 (default)

This parameter leads to a temperature-dependent reduction in the MOSFET transconductance at high gate voltage. The surface roughness scattering,  $\theta_{sr}$ , is assumed to scale exponentially with temperature,  $\theta_{sr} = \theta_{sr,m1} (T_{m1}/T_s)^{\eta_{sr}}$ .  $\theta_{sr,m1}$  is the value of the **Surface roughness scattering factor** parameter from the **Main** setting and  $\eta_{sr}$  is the **Surface roughness scattering temperature exponent**.

### Resistance temperature exponent – Resistance temperature exponent

0.95 (default)

The series resistances are assumed to correspond to semiconductor resistances. Therefore, they decrease exponentially with increasing temperature.  $R_i = R_{i,m1} (T_{m1}/T_s)^{\eta_R}$ , where  $i$  is S, D, or G, for the source, drain, or gate series resistance, respectively.  $R_{i,m1}$  is the value of the corresponding series resistance parameter from the **Ohmic Resistance** setting and  $\eta_R$  is the **Resistance temperature exponent**.

### Body diode reverse saturation current temperature exponent – Body diode reverse saturation current temperature exponent

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$

$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** setting,  $k_B$  is Boltzmann's constant (8.617x10<sup>-5</sup>eV/K), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

### Device simulation temperature – Device simulation temperature

25 °C (default)

Temperature  $T_s$  at which the device is simulate.

### Faults

#### Enable faults – Faults modeling option

Off (default) | On

Whether to enable faults in MOSFET.

**Fault trigger – Fault trigger option**

Temporal (default) | Behavioral

Whether to trigger faults on temporal or behavioral conditions. Set this parameter to:

- **Temporal** — Trigger fault at a specific time.
- **Behavioral** — Trigger fault when a current limit or a voltage limit are exceeded for longer than a specific time interval.

**Simulation time for fault event – Time before entering faulted state**

inf s (default)

Simulation time at which you want the block to enter the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Temporal.

**Maximum permissible gate-source voltage – Gate-source voltage threshold to fault transition**

inf V (default) | positive scalar

Maximum permissible gate-source voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Maximum permissible drain-source voltage – Drain-source voltage threshold to fault transition**

inf V (default) | positive scalar

Maximum permissible drain-source voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Maximum permissible drain-bulk voltage – Drain-bulk voltage threshold to fault transition**

inf V (default) | positive scalar

Maximum permissible drain-bulk voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Fault trigger** to Behavioral.

**Maximum permissible drain-source current – Current threshold to fault transition**

inf A (default) | positive scalar

Maximum permissible drain-source current value. If the current exceeds this value for longer than the value of the **Time to fail when exceeding drain-source current** parameter, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Time to fail when exceeding drain-source current — Time to fail when exceeding drain-source current**

1 s (default) | positive scalar

Amount of time that the drain-source current must continuously exceed the maximum permissible drain-source current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Maximum permissible body diode current — Body diode current threshold to fault transition**

inf A (default) | positive scalar

Maximum permissible body diode current value. If the current exceeds this value for longer than the value of the **Time to fail when exceeding body diode current** parameter, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

**Time to fail when exceeding body diode current — Time to fail when exceeding body diode current**

1 s (default) | positive scalar

Amount of time that the body diode current must continuously exceed the maximum permissible body diode current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

**Maximum diode recovery, dV/dt — Maximum diode recovery**

inf V/s (default) | positive scalar

Maximum amount of diode recovery per second.

**Dependencies**

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

**Maximum permissible temperature — Temperature threshold to fault transition**

inf K (default) | positive scalar

Maximum permissible temperature value. If the temperature exceeds this value for longer than the value of the **Time to fail when exceeding temperature** parameter, then the block enters the faulted state.

#### Dependencies

To enable this parameter, set **Modeling option** to Threshold-based with thermal, **Enable faults** to On, and **Fault trigger** to Behavioral.

#### Time to fail when exceeding temperature — Time to fail when exceeding temperature

1 s (default) | positive scalar

Amount of time that the temperature must continuously exceed the maximum permissible temperature before a behavioral fault is triggered.

#### Dependencies

To enable this parameter, set **Modeling option** to Threshold-based with thermal, **Enable faults** to On, and **Fault trigger** to Behavioral.

#### Fault transition time constant, tau — Fault transition time constant

1e-3 s (default)

Time constant for transitioning to faulted state.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

#### Failure mode — Failure category

Open circuit (default) | Drain-source short | Drain-bulk short or source-bulk short | Gate oxide short | Parameter shift

Type of failure. You can set this parameter to:

- Open circuit — Failure due to metallization burnout.
- Drain-source short — Failure due to avalanche breakout on drain-source channel.
- Drain-bulk short or source-bulk short — Failure due to avalanche breakout on drain-bulk or source-bulk channels.
- Gate oxide short — Failure of the gate oxide dielectric layer.
- Parameter shift — Failure due to aging.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

To enable the Drain-source short option, in the **Main** setting, set **Number of terminals** to Three.

To enable the Drain-bulk short or source-bulk short option, in the **Main** setting, set **Number of terminals** to Four.

#### Faulted series gate resistance, Rg — Faulted series gate resistance

1e6 Ohm (default)

Series gate resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series drain resistance, Rd — Faulted series drain resistance**

1000 Ohm (default)

Series drain resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series source resistance, Rs — Faulted series source resistance**

1000 Ohm (default)

Series source resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series bulk resistance, Rb — Faulted series source resistance**

1000 Ohm (default)

Series source resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Open circuit.

**Faulted parallel drain-source conductance, Gds — Faulted parallel drain-source conductance**

1000 1/Ohm (default)

Parallel drain-source conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.
- **Enable faults** to On.
- **Failure mode** to Drain-source short.

**Faulted parallel drain-bulk conductance, Gdb — Faulted parallel drain-bulk conductance**

1000 1/Ohm (default)

Parallel drain-bulk conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Drain-bulk short or source-bulk short.

**Faulted parallel source-bulk conductance, Gsb — Faulted parallel source-bulk conductance**

1000 1/Ohm (default)

Parallel source-bulk conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Drain-bulk short or source-bulk short.

**Location of fault node (percentage of total channel length from source) — Fault node location**

0 (default)

Location of the fault node, as a percentage of the total channel length from the source.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Gate oxide short conductance, G — Gate oxide short conductance**

1000 1/Ohm (default)

Gate oxide short conductance.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Model gate oxide short diode — Model option for gate oxide short diode**

No (default) | Yes

Whether to model the gate oxide short diode.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Gate oxide short diode reverse saturation current, Is — Reverse saturation current of gate oxide short diode**

0 A (default)

Reverse saturation current of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode built-in voltage,  $V_{bi}$  — Built-in voltage of gate oxide short diode**

0.6 V (default)

Built-in voltage of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode ideality factor,  $n$  — Ideality factor of gate oxide short diode**

1 (default)

Ideality factor of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode zero-bias junction capacitance,  $C_{j0}$  — Zero-bias junction capacitance of gate oxide short diode**

0 pF (default)

Zero-bias junction capacitance of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode transit time,  $T_T$  — Transit time of gate oxide short diode**

50e-9 s (default)

Transit time of the gate oxide short diode.



**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Shifted gain as percentage of unshifted — Shifted gain as percentage of unshifted**  
100 (default)

Shifted gain as percentage of unshifted.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Threshold voltage shift value,  $dV_{th}$  — Threshold voltage shift value**  
0 V (default)

Shift value of Threshold voltage.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted channel modulation,  $L$  — Faulted channel modulation**  
0 1/V (default)

Channel modulation in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Shifted body factor as percentage of unshifted — Shifted body factor as percentage of unshifted**  
100 (default)

Shifted body factor as percentage of unshifted.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Surface potential shift value, dPhib — Surface potential shift value**

0 V (default)

Shift value of surface potential.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted source ohmic resistance, Rs — Faulted source ohmic resistance**

0.0001 Ohm (default)

Transistor source resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted drain ohmic resistance, Rd — Faulted drain ohmic resistance**

0.01 Ohm (default)

Transistor drain resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted body ohmic resistance, Rb — Faulted body ohmic resistance**

0.001 Ohm (default)

Transistor body resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted vector of gate-source voltages, Vgs — Faulted vector of gate-source voltages**

[-.5, 2, 3, 4, 5] V (default)

Gate-source voltages in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted vector of drain-source voltages, Vds — Faulted vector of drain-source voltages**

[0, 2, 4, 8, 12, 16, 20] V (default)

Drain-source voltages in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted vector of bulk-source voltages, Vbs — Faulted vector of bulk-source voltages**

[0, -4, -6, -12] V (default)

Bulk-source voltages in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated gate-source threshold voltage,  $V_{th}(V_{bs})$  – Faulted tabulated gate-source threshold voltage**

[.4, 1.2, 1.4, 1.45] V (default)

Tabulated gate-source threshold voltage in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated gate-source threshold voltage,  $V_{th}(V_{bs}, T)$  – Faulted tabulated gate-source threshold voltage dependent on temperature**

[.4, .35; 1.2, 1.15; 1.4, 1.35; 1.45, 1.4] V (default)

Gate-source threshold voltage in faulted state, tabulated on bulk-source voltage and temperature.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated drain-source currents,  $I_{ds}(V_{gs}, V_{ds})$  – Faulted tabulated drain-source currents**

[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;  
0, .0115, .016, .018, .019, .0195, .02;  
0, .0152, .0225, .0265, .028, .0295, .03] A (default)

Tabulated drain-source currents in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.

- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated drain-source currents,  $I_{ds}(V_{gs}, V_{ds}, T)$  — Faulted tabulated drain-source currents, dependent on temperature**

zeros(5, 7, 2) A (default)

Drain-source currents in faulted state, tabulated on gate-source voltage, drain-source voltage, and temperature.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated drain-source currents at zero bulk-source voltage,  $I_{ds}(V_{gs}, V_{ds})$  — Faulted tabulated drain-source currents at zero bulk-source voltage**

[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;  
0, .0115, .016, .018, .019, .0195, .02;  
0, .0152, .0225, .0265, .028, .0295, .03] A (default)

Tabulated drain-source currents in faulted state, when the bulk-source voltage is zero.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated drain-source currents at zero bulk-source voltage,  $I_{ds}(V_{gs}, V_{ds}, T)$  — Faulted tabulated drain-source currents at zero bulk-source voltage, dependent on temperature**

zeros(5, 7, 2) A (default)

Tabulated drain-source currents in faulted state, when the bulk-source voltage is zero, tabulated on gate-source voltage, drain-source voltage, and temperature.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted gate-source capacitance as percentage of unfaulted — Faulted gate-source capacitance as percentage of unfaulted**

100 (default)

Gate-source capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-drain capacitance as percentage of unfaulted — Faulted gate-drain capacitance as percentage of unfaulted**

100 (default)

Gate-drain capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to 0n.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-bulk capacitance as percentage of unfaulted — Faulted gate-bulk capacitance as percentage of unfaulted**

100 (default)

Gate-bulk capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set **Enable faults** to 0n.

**Condition Set Two**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to 0n.
  - **Failure mode** to Open circuit, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted drain-source capacitance as percentage of unfaulted — Faulted drain-source capacitance as percentage of unfaulted**

100 (default)

Drain-source capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-drain oxide capacitance as percentage of unfaulted — Faulted gate-drain oxide capacitance as percentage of unfaulted**

100 (default)

Gate-drain oxide capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:
  - **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
  - **Gate-drain charge-voltage linearity** to Gate-drain charge function is non linear.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted drain-gate voltage at which oxide capacitance becomes active — Faulted drain-gate voltage at which oxide capacitance becomes active**

-0.5 V (default)

Drain-gate voltage in faulted state at which oxide capacitance becomes active.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:



- **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- **Gate-drain charge-voltage linearity** to Gate-drain charge function is non linear.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted input capacitance, $C_{iss}(V_{ds})$ – Faulted input capacitance**

[720, 700, 590, 470, 390, 310] pF (default)

Input capacitance in faulted state.

##### **Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted reverse transfer capacitance, $C_{rss}(V_{ds})$ – Faulted reverse transfer capacitance**

[450, 400, 300, 190, 95, 55] pF (default)

Reverse transfer capacitance in faulted state.

##### **Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted output capacitance, $C_{oss}(V_{ds})$ – Faulted output capacitance**

[900, 810, 690, 420, 270, 170] pF (default)

Output capacitance in faulted state.

##### **Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:

- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted corresponding drain-source voltages, Vds — Faulted corresponding drain-source voltages**

[.1, .3, 1, 3, 10, 30] V (default)

Corresponding drain-source voltages in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-source voltages, Vgs, for tabulated capacitances — Faulted gate-source voltages for tabulated capacitances**

0 V (default)

Gate-source voltages for tabulated capacitances, in faulted state.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Condition Set Two**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Gate-bulk and gate-source capacitance change instantly or Gate-bulk and gate-source capacitance change gradually.
- In the **Faults** setting, set:
  - **Enable faults** to On.

- **Failure mode** to Parameter shift.

**Faulted gate-source capacitance,  $C_{gs}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[270, 300, 290, 280, 295, 255] pF (default)

Gate-source capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-drain capacitance,  $C_{gd}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[450, 400, 300, 190, 95, 55] pF (default)

Gate-drain capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-bulk capacitance,  $C_{gb}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[270, 300, 290, 280, 295, 255] pF (default)

Gate-bulk capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.

- **Failure mode** to Parameter shift.

**Faulted corresponding gate-source voltages,  $V_{gs}$  — Faulted corresponding gate-source voltages**

0 V (default)

Corresponding gate-source voltages in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode reverse saturation current,  $I_s$  — Faulted diode reverse saturation current**

0 A (default)

Reverse saturation current of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode built-in voltage,  $V_{bi}$  — Faulted diode built-in voltage**

0.6 V (default)

Built-in voltage of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode ideality factor, n – Faulted diode ideality factor**

1 (default)

Ideality factor of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode zero-bias junction capacitance as percentage of unfaulted – Faulted diode zero-bias junction capacitance as percentage of unfaulted**

100 (default)

Faulted diode zero-bias junction capacitance as percentage of unfaulted.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode transit time, TT – Faulted diode transit time**

50e-9 s (default)

Transit time of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward currents, If(Vf) – Faulted diode forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2] A (default)

Forward currents of the diode in faulted state, tabulated with the forward voltages.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify using equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(V_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f$  — Faulted diode forward voltages**  
[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Forward voltages of the diode in faulted state.

#### **Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(V_f)$  form or Table in  $I_f(T_j, V_f)$  form
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f(I_f)$  — Faulted diode forward voltages**  
[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85] V (default)

Forward voltages of the diode in faulted state, tabulated with the forward currents.

#### **Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify using equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $V_f(I_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward currents,  $I_f$  — Faulted diode forward currents**  
[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Forward currents of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $V_f(I_f)$  form and Table in  $V_f(T_j, I_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward currents,  $I_f(T_j, V_f)$  – Faulted diode forward currents with tabulated inputs**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default)

Forward currents of the diode in faulted state, tabulated with the forward voltages and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(T_j, V_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f(T_j, I_f)$  – Faulted diode forward voltages with tabulated inputs**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default)

Forward voltages of the diode in faulted state, tabulated with the forward currents and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

- **Table type** to Table in Vf(Tj,If) form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

## Version History

Introduced in R2008a

## References

- [1] Shichman, H. and D. A. Hodges. "Modeling and simulation of insulated-gate field-effect transistor switching circuits." *IEEE J. Solid State Circuits*. SC-3, 1968.
- [2] Van Langevelde, R., A. J. Scholten, and D. B. M. Klaassen. "Physical Background of MOS Model 11. Level 1101." *Nat.Lab. Unclassified Report 2003/00239*. April 2003.
- [3] Oh, S-Y., D. E. Ward, and R. W. Dutton. "Transient analysis of MOS transistors." *IEEE J. Solid State Circuits*. SC-15, pp. 636-643, 1980.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

P-Channel MOSFET

### Functions

semiconductorSubcircuit2lookup



# Negative Supply Rail

Ideal negative supply rail

**Library:** Simscape / Electrical / Sources



## Description

The Negative Supply Rail block represents an ideal negative supply rail. Use this block instead of the SimscapeDC Voltage Source block to define the output voltage relative to the SimscapeElectrical Reference block that must appear in each model.

---

**Note** Do not attach more than one Negative Supply Rail block to any connected line.

---

## Ports

### Conserving

- — **Negative voltage**

electrical

Electrical conserving port associated to the negative supply rail negative voltage.

## Parameters

**Constant voltage — Constant output voltage**

-1 V (default)

The voltage at the output port relative to the Electrical Reference block ground port.

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

DC Voltage Source | Positive Supply Rail

## Neutral Port (Three-Phase)

Connect phases of three-phase system to electrical conserving port

**Library:** Simscape / Electrical / Connectors & References



### Description

The Neutral Port (Three-Phase) block connects the phases of a three-phase system to an electrical conserving port.

---

**Note** If you do not need to connect the neutral port to other blocks, use a Floating Neutral (Three-Phase) block instead. If you want to ground the neutral port, use a Grounded Neutral (Three-Phase) block.

---

### Ports

#### Conserving

**~ — Three-phase port**

electrical

Expandable composite (**a,b, c**) three-phase port.

**n — Neutral point**

electrical

Electrical conserving port associated with the neutral point

### Parameters

**Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Parasitic ground conductance — Parasitic ground conductance**

1e-12 1/0hm (default)

Parasitic conductance to ground. A nonzero value is required for the simulation of some circuit topologies.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Floating Neutral (Three-Phase) | Grounded Neutral (Three-Phase) | Open Circuit (Three-Phase)

## **Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Nonlinear Inductor

Inductor with nonideal core

**Library:** Simscape / Electrical / Passive



## Description

The Nonlinear Inductor block represents an inductor with a nonideal core. A core may be nonideal due to its magnetic properties and dimensions. The block provides the following parameterization options:

- Single inductance (linear) on page 1-1142
- Single saturation point on page 1-1142
- Magnetic flux versus current characteristic on page 1-1143
- Magnetic flux density versus magnetic field strength characteristic on page 1-1143
- Magnetic flux density versus magnetic field strength characteristic with hysteresis on page 1-1144

### Single Inductance (Linear)

The relationships between voltage, current and flux are defined by the following equations:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = \frac{L}{N_w} i_L$$

where:

- $v$  is the terminal voltage.
- $i$  is the terminal current.
- $i_L$  is the current through inductor.
- $G_p$  is the parasitic parallel conductance.
- $N_w$  is the number of winding turns.
- $\Phi$  is the magnetic flux.
- $L$  is the unsaturated inductance.

### Single Saturation Point

The relationships between voltage, current and flux are defined by the following equations:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = \frac{L}{N_w} i_L \text{ (for unsaturated)}$$

$$\Phi = \frac{L_{sat}}{N_w} i_L \pm \Phi_{offset} \text{ (for saturated)}$$

where:

- $v$  is the terminal voltage.
- $i$  is the terminal current.
- $i_L$  is the current through inductor.
- $G_p$  is the parasitic parallel conductance.
- $N_w$  is the number of winding turns.
- $\Phi$  is the magnetic flux.
- $\Phi_{offset}$  is the magnetic flux saturation offset.
- $L$  is the unsaturated inductance.
- $L_{sat}$  is the saturated inductance.

### Magnetic Flux Versus Current Characteristic

The relationships between voltage, current and flux are defined by the following equations:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = f(i_L)$$

where:

- $v$  is the terminal voltage.
- $i$  is the terminal current.
- $i_L$  is the current through inductor.
- $G_p$  is the parasitic parallel conductance.
- $N_w$  is the number of winding turns.
- $\Phi$  is the magnetic flux.

Magnetic flux is determined by one-dimensional table lookup, based on the vector of current values and the vector of corresponding magnetic flux values that you provide. You can construct these vectors using either negative and positive data, or positive data only. If using positive data only, the vector must start at 0, and the negative data will be automatically calculated by rotation about (0,0).

### Magnetic Flux Density Versus Magnetic Field Strength Characteristic

The relationships between voltage, current and flux are defined by the following equations:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = B \cdot A_e$$

$$B = f(H)$$

$$H = \frac{N_w}{l_e} i_L$$

where:

- $v$  is the terminal voltage.
- $i$  is the terminal current.
- $i_L$  is the current through inductor.
- $G_p$  is the parasitic parallel conductance.
- $N_w$  is the number of winding turns.
- $\Phi$  is the magnetic flux.
- $B$  is the magnetic flux density.
- $H$  is the magnetic field strength.
- $l_e$  is the effective core length.
- $A_e$  is the effective core cross-sectional area.

Magnetic flux density is determined by one-dimensional table lookup, based on the vector of magnetic field strength values and the vector of corresponding magnetic flux density values that you provide. You can construct these vectors using either negative and positive data, or positive data only. If using positive data only, the vector must start at 0, and the negative data will be automatically calculated by rotation about (0,0).

### **Magnetic Flux Density Versus Magnetic Field Strength Characteristic with Hysteresis**

The relationships between voltage, current and flux are defined by the following equations:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = B \cdot A_e$$

$$B = \mu_0(H + M)$$

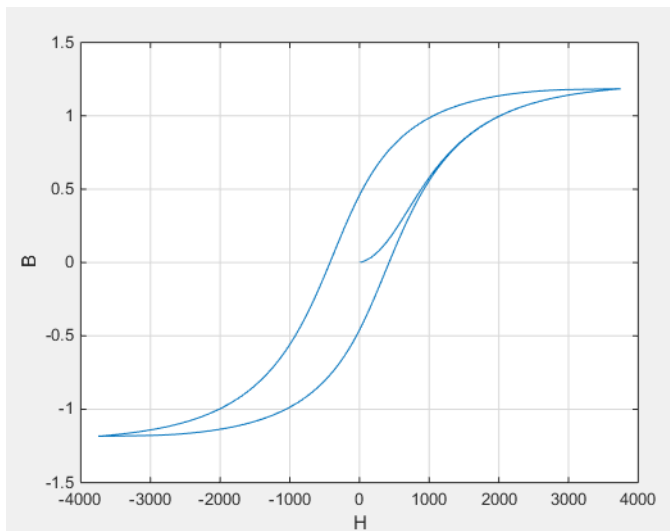
$$H = \frac{N_w}{l_e} i_L$$

where:

- $v$  is the terminal voltage.
- $i$  is the terminal current.
- $i_L$  is the current through inductor.

- $G_p$  is the parasitic parallel conductance.
- $N_w$  is the number of winding turns.
- $\Phi$  is the magnetic flux.
- $B$  is the magnetic flux density.
- $\mu_0$  is the magnetic constant, permeability of free space.
- $H$  is the magnetic field strength.
- $M$  is the magnetization of the inductor core.
- $l_e$  is the effective core length.
- $A_e$  is the effective core cross-sectional area.

The magnetization acts to increase the magnetic flux density, and its value depends on both the current value and the history of the field strength  $H$ . The Jiles-Atherton [1 on page 1-1152, 2 on page 1-1152] equations are used to determine  $M$  at any given time. The figure below shows a typical plot of the resulting relationship between  $B$  and  $H$ .



In this case the magnetization starts as zero, and hence the plot starts at  $B = H = 0$ . As the field strength increases, the plot tends to the positive-going hysteresis curve; then on reversal the rate of change of  $H$ , it follows the negative-going hysteresis curve. The difference between positive-going and negative-going curves is due to the dependence of  $M$  on the trajectory history. Physically the behavior corresponds to magnetic dipoles in the core aligning as the field strength increases, but not then fully recovering to their original position as field strength decreases.

The starting point for the Jiles-Atherton equation is to split the magnetization effect into two parts, one that is purely a function of effective field strength ( $H_{eff}$ ) and the other an irreversible part that depends on past history:

$$M = cM_{an} + (1 - c)M_{irr}$$

The  $M_{an}$  term is called the anhysteretic magnetization because it exhibits no hysteresis. It is described by the following function of the current value of the effective field strength,  $H_{eff}$ :

$$M_{an} = M_s \left( \coth\left(\frac{H_{eff}}{a}\right) - \frac{a}{H_{eff}} \right)$$

This function defines a saturation curve with limiting values  $\pm M_s$  and point of saturation determined by the value of  $a$ , the anhysteretic shape factor. It can be approximately thought of as describing the average of the two hysteretic curves. In the Nonlinear Inductor block, you provide values for  $dM_{an}/dH_{eff}$  when  $H_{eff} = 0$  and a point  $[H_1, B_1]$  on the anhysteretic B-H curve, and these are used to determine values for  $a$  and  $M_s$ .

The parameter  $c$  is the coefficient for reversible magnetization, and dictates how much of the behavior is defined by  $M_{an}$  and how much by the irreversible term  $M_{irr}$ . The Jiles-Atherton model defines the irreversible term by a partial derivative with respect to field strength:

$$\frac{dM_{irr}}{dH} = \frac{M_{an} - M_{irr}}{K\delta - \alpha(M_{an} - M_{irr})}$$

$$\delta = \begin{cases} 1 & \text{if } H \geq 0 \\ -1 & \text{if } H < 0 \end{cases}$$

Comparison of this equation with a standard first order differential equation reveals that as increments in field strength,  $H$ , are made, the irreversible term  $M_{irr}$  attempts to track the reversible term  $M_{an}$ , but with a variable tracking gain of  $1/(K\delta - \alpha(M_{an} - M_{irr}))$ . The tracking error acts to create the hysteresis at the points where  $\delta$  changes sign. The main parameter that shapes the irreversible characteristic is  $K$ , which is called the bulk coupling coefficient. The parameter  $\alpha$  is called the inter-domain coupling factor, and is also used to define the effective field strength used when defining the anhysteretic curve:

$$H_{eff} = H + \alpha M$$

The value of  $\alpha$  affects the shape of the hysteresis curve, larger values acting to increase the B-axis intercepts. However, notice that for stability the term  $K\delta - \alpha(M_{an} - M_{irr})$  must be positive for  $\delta > 0$  and negative for  $\delta < 0$ . Therefore not all values of  $\alpha$  are permissible, a typical maximum value being of the order  $1e-3$ .

### Procedure for Finding Approximate Values for Jiles-Atherton Equation Coefficients

You can determine representative parameters for the equation coefficients by using the following procedure:

- 1 Provide a value for the **Anhysteretic B-H gradient when H is zero** parameter ( $dM_{an}/dH_{eff}$  when  $H_{eff} = 0$ ) plus a data point  $[H_1, B_1]$  on the anhysteretic B-H curve. From these values, the block initialization determines values for  $a$  and  $M_s$ .
- 2 Set the **Coefficient for reversible magnetization, c** parameter to achieve correct initial B-H gradient when starting a simulation from  $[H \ B] = [0 \ 0]$ . The value of  $c$  is approximately the ratio of this initial gradient to the **Anhysteretic B-H gradient when H is zero**. The value of  $c$  must be greater than 0 and less than 1.
- 3 Set the **Bulk coupling coefficient, K** parameter to the approximate magnitude of  $H$  when  $B = 0$  on the positive-going hysteresis curve.
- 4 Start with  $\alpha$  very small, and gradually increase to tune the value of  $B$  when crossing  $H = 0$  line. A typical value is in the range of  $1e-4$  to  $1e-3$ . Values that are too large will cause the gradient of the B-H curve to tend to infinity, which is nonphysical and generates a run-time assertion error.

Sometimes you need to iterate on these four steps to get a good match against a predefined B-H curve.



## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the inductor positive voltage.

#### - – Negative terminal

electrical

Electrical conserving port associated with the inductor negative voltage.

## Parameters

### Main

#### Parameterized by – Block parameterization

Single saturation point (default) | Single inductance (linear) | Magnetic flux versus current characteristic | Magnetic flux density versus magnetic field strength characteristic | Magnetic flux density versus magnetic field strength characteristic with hysteresis

Select one of the following methods for block parameterization:

- **Single inductance (linear)** — Provide the values for number of turns, unsaturated inductance, and parasitic parallel conductance.
- **Single saturation point** — Provide the values for number of turns, unsaturated and saturated inductances, saturation magnetic flux, and parasitic parallel conductance. This is the default option.
- **Magnetic flux versus current characteristic** — In addition to the number of turns and the parasitic parallel conductance value, provide the current vector and the magnetic flux vector, to populate the magnetic flux versus current lookup table.
- **Magnetic flux density versus magnetic field strength characteristic** — In addition to the number of turns and the parasitic parallel conductance value, provide the values for effective core length and cross-sectional area, as well as the magnetic field strength vector and the magnetic flux density vector, to populate the magnetic flux density versus magnetic field strength lookup table.
- **Magnetic flux density versus magnetic field strength characteristic with hysteresis** — In addition to the number of turns and the effective core length and cross-sectional area, provide the values for the initial anhysteretic B-H curve gradient, the magnetic flux density and field strength at a certain point on the B-H curve, as well as the coefficient for the reversible magnetization, bulk coupling coefficient, and inter-domain coupling factor, to define magnetic flux density as a function of both the current value and the history of the field strength.

#### Number of turns – Total number of turns

10 (default)

The total number of turns of wire wound around the inductor core.

#### Unsaturated inductance – Unsaturated inductance

2e-4 H (default)

The value of inductance used when the inductor is operating in its linear region.

**Dependencies**

This parameter is visible only when you select `Single inductance (linear)` or `Single saturation point` for the **Parameterized by** parameter.

**Saturated inductance – Saturated inductance**

1e-4 H (default)

The value of inductance used when the inductor is operating beyond its saturation point.

**Dependencies**

This parameter is visible only when you select `Single saturation point` for the **Parameterized by** parameter.

**Saturation magnetic flux – Saturation magnetic flux**

1.3e-05 Wb (default)

The value of magnetic flux at which the inductor saturates.

**Dependencies**

This parameter is visible only when you select `Single saturation point` for the **Parameterized by** parameter.

**Current vector, i – Current data vector**

[0, .64, 1.28, 1.92, 2.56, 3.2] A (default)

The current data used to populate the magnetic flux versus current lookup table.

**Dependencies**

This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Parameterized by** parameter.

**Magnetic flux vector, phi – Magnetic flux data**

[0, 1.29, 2, 2.27, 2.36, 2.39] .\* 1e-5 Wb (default)

The magnetic flux data used to populate the magnetic flux versus current lookup table.

**Dependencies**

This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Parameterized by** parameter.

**Magnetic field strength vector, H – Magnetic field strength vector**

[0, 200, 400, 600, 800, 1000] A/m (default)

The magnetic field strength data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Parameterized by** parameter.

**Magnetic flux density vector, B – Magnetic flux density data**

[0, .81, 1.25, 1.42, 1.48, 1.49] T (default)

The magnetic flux density data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic for the **Parameterized by** parameter.

**Effective length – Effective core length**

0.032 m (default)

The effective core length, that is, the average distance of the magnetic path.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic or Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Effective cross-sectional area – Effective cross-sectional area**1.6e-5 m<sup>2</sup> (default)

The effective core cross-sectional area, that is, the average area of the magnetic path.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic or Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Anhyseretic B-H gradient when H is zero – Anhyseretic B-H gradient around zero field strength**

0.005 m\*T/A (default)

The gradient of the anhyseretic (no hysteresis) B-H curve around zero field strength. Set it to the average gradient of the positive-going and negative-going hysteresis curves.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Flux density point on anhyseretic B-H curve – Flux density point on anhyseretic B-H curve**

1.49 T (default)

Specify a point on the anhyseretic curve by providing its flux density value. Picking a point at high field strength where the positive-going and negative-going hysteresis curves align is the most accurate option.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Corresponding field strength — Corresponding field strength**

1000 A/m (default)

The corresponding field strength for the point that you define by the **Flux density point on anhysteretic B-H curve** parameter.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Coefficient for reversible magnetization, c — Coefficient for reversible magnetization**

0.1 (default)

The proportion of the magnetization that is reversible. The value should be greater than zero and less than one.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Bulk coupling coefficient, K — Bulk coupling coefficient**

200 A/m (default)

The Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Inter-domain coupling factor, alpha — Inter-domain coupling factor**

1e-4 (default)

The Jiles-Atherton parameter that primarily affects the points at which the B-H curves intersect the zero field strength line. Typical values are in the range of 1e-4 to 1e-3.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Averaging period for power logging — Averaging period for power logging**

0 s (default)

Averaging period for the hysteresis losses calculation. These losses are proportional to the area enclosed by the B-H trajectory. If the block is excited at a known, fixed frequency, you can set this value to the corresponding excitation period to calculate the hysteresis loss. In this case, the block logs the hysteresis loss once per AC cycle to the variable `power_dissipated`. If you are using a fixed-step solver, this value must be an integer multiple of the simulation step size.

If the block is not excited at a known, fixed frequency, set this parameter to 0. In this case, the block sets `power_dissipated` to zero, and you can calculate the actual hysteresis loss by post-processing the logged variable `power_instantaneous`.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter on the **Main** tab.

**Parasitic parallel conductance — Parasitic parallel conductance**

1e-9 1/Ohm (default)

Use this parameter to represent small parasitic effects. A small parallel conductance may be required for the simulation of some circuit topologies.

**Interpolation option — Interpolation option**

Linear (default) | Smooth

The lookup table interpolation option. Select one of the following interpolation methods:

- **Linear** — Select this option to get the best performance.
- **Smooth** — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

**Dependencies**

This parameter is visible only when you select Magnetic flux versus current characteristic or Magnetic flux density versus magnetic field strength characteristic for the **Parameterized by** parameter on the **Main** tab.

**Initial Conditions****Specify initial state by — Initial state specification option**

Current (default) | Magnetic flux

Select the appropriate initial state specification option:

- **Current** — Specify the initial state of the inductor by the initial current through the inductor ( $i_L$ ). This is the default option.
- **Magnetic flux** — Specify the initial state of the inductor by the magnetic flux.

**Dependencies**

This parameter is not visible when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter.

**Initial current — Initial current**

0 A (default)

The initial current value used to calculate the value of magnetic flux at time zero. This is the current passing through the inductor. Component current consists of current passing through the inductor and current passing through the parasitic parallel conductance.

**Dependencies**

This parameter is visible only when you select Current for the **Specify initial state by** parameter.

**Initial magnetic flux — Initial magnetic flux**

0 Wb (default)

The value of magnetic flux at time zero.

**Dependencies**

This parameter is visible only when you select Magnetic flux for the **Specify initial state by** parameter.

**Initial magnetic flux density — Initial magnetic flux density**

0 T (default)

The value of magnetic flux density at time zero.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter on the **Main** tab.

**Initial field strength — Initial field strength**

0 A/m (default)

The value of magnetic field strength at time zero.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Parameterized by** parameter on the **Main** tab.

## Version History

Introduced in R2012b

## References

- [1] Jiles, D. C. and D. L. Atherton. "Theory of ferromagnetic hysteresis." *Journal of Magnetism and Magnetic Materials* . Vol. 61, 1986, pp. 48-60.
- [2] Jiles, D. C. and D. L. Atherton. "Ferromagnetic hysteresis." *IEEE Transactions on Magnetics* . Vol. 19, No. 5, 1983, pp. 2183-2184.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

# Nonlinear Reluctance

Nonlinear reluctance with magnetic hysteresis

**Library:** Simscape / Electrical / Passive



## Description

The Nonlinear Reluctance block models linear or nonlinear reluctance with magnetic hysteresis. Use this block to build custom inductances and transformers that exhibit magnetic hysteresis.

The length and area parameters in the **Geometry** settings let you define the geometry for the part of the magnetic circuit that you are modeling. The block uses the geometry information to map the magnetic domain Through and Across variables to flux density and field strength.

## Equations for Linear Reluctance Parameterization

The equations for the linear reluctance parameterization are:

$$B = \mu_0 \mu_r H$$

$$mmf = l_{eff} H$$

$$\varphi = s_{eff} B$$

where:

- $B$  is the flux density.
- $\mu_0$  is the permeability in a vacuum.
- $\mu_r$  is the relative magnetic permeability.
- $H$  is the field strength.
- $mmf$  is the magnetomotive force (mmf) across the component.
- $l_{eff}$  is the effective length of the section being modeled.
- $\varphi$  is magnetic flux.
- $s_{eff}$  is the effective cross-sectional area of the section being modeled.

## Equations for Reluctance with Single Saturation Point Parameterization

This parameterization models a switch-linear reluctance. In the unsaturated state, the material has a specified relative magnetic permeability. In the saturated state, the relative permeability is  $\infty$ .

The equations for reluctance with single saturation point are

$$mmf = l_{eff} H$$

$$\varphi = s_{eff} B$$

$$mmf = R\phi$$

If  $B < B_{sat}$ .

$$B = \mu_0\mu_{r\_unsat}H$$

Otherwise,

$$B = B_{sat} + \mu_0\left(H - \frac{B_{sat}}{\mu_0\mu_{r\_unsat}}\right)$$

where:

- $mmf$  is the magnetomotive force (mmf) across the component.
- $l_{eff}$  is the effective length of the section being modeled.
- $H$  is the field strength.
- $\phi$  is magnetic flux.
- $s_{eff}$  is the effective cross-sectional area of the section being modeled.
- $B$  is the flux density.
- $B_{sat}$  is the flux density at saturation.
- $R_{sat}$  is the magnetic reluctance at saturation.
- $\mu_0$  is the permeability in a vacuum.
- $\mu_r$  is the relative magnetic permeability.
- $\mu_{r\_unsat}$  is the unstaured relative magnetic permeability.

### Reluctance (B-H Curve)

For the reluctance (B-H Curve) parameterization, specify the material property by B-H curve.

### Equations for Reluctance with Hysteresis Parameterization

The flux density and magnetomotive force equations are:

$$B = \phi/s_{eff}$$

$$mmf = l_{eff} \cdot H$$

where:

- $B$  is flux density.
- $\phi$  is magnetic flux.
- $s_{eff}$  is the effective cross-sectional area of the section being modeled.
- $mmf$  is magnetomotive force (mmf) across the component.
- $l_{eff}$  is the effective length of the section being modeled.
- $H$  is field strength.

The block then implements the relationship between  $B$  and  $H$  according to the Jiles-Atherton [1 on page 1-1160, 2 on page 1-1160] equations. The equation that relates  $B$  and  $H$  to the magnetization of the core is:



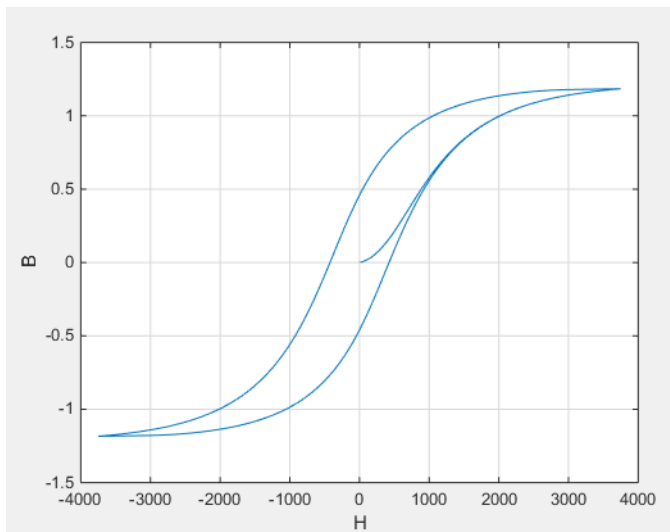
$$B = \mu_0(H + M)$$

where:

- $\mu_0$  is the magnetic permeability constant.
- $M$  is magnetization of the core.

The magnetization acts to increase the magnetic flux density, and its value depends on both the current value and the history of the field strength  $H$ . The block uses the Jiles-Atherton equations to determine  $M$  at any given time.

The figure below shows a typical plot of the resulting relationship between  $B$  and  $H$ .



In this case, the magnetization starts as zero, and hence the plot starts at  $B = H = 0$ . As the field strength increases, the plot tends to the positive-going hysteresis curve; then on reversal the rate of change of  $H$ , it follows the negative-going hysteresis curve. The difference between positive-going and negative-going curves is due to the dependence of  $M$  on the trajectory history. Physically the behavior corresponds to magnetic dipoles in the core aligning as the field strength increases, but not then fully recovering to their original position as field strength decreases.

The starting point for the Jiles-Atherton equation is to split the magnetization effect into two parts, one that is purely a function of effective field strength ( $H_{eff}$ ) and the other an irreversible part that depends on history:

$$M = cM_{an} + (1 - c)M_{irr}$$

The  $M_{an}$  term is called the anhysteretic magnetization because it exhibits no hysteresis. It is described by the following function of the current value of the effective field strength,  $H_{eff}$ :

$$M_{an} = M_s \left( \coth\left(\frac{H_{eff}}{a}\right) - \frac{a}{H_{eff}} \right)$$

This function defines a saturation curve with limiting values  $\pm M_s$  and point of saturation determined by the value of  $a$ , the anhysteretic shape factor. It can be approximately thought of as describing the average of the two hysteretic curves. In the block interface, you provide values for  $dM_{an}/dH_{eff}$  when

$H_{eff} = 0$  and a point  $[H_1, B_1]$  on the anhysteretic B-H curve, and these are used to determine values for  $\alpha$  and  $M_s$ .

The parameter  $c$  is the coefficient for reversible magnetization, and dictates how much of the behavior is defined by  $M_{an}$  and how much by the irreversible term  $M_{irr}$ . The Jiles-Atherton model defines the irreversible term by a partial derivative with respect to field strength:

$$\frac{dM_{irr}}{dH} = \frac{M_{an} - M_{irr}}{K\delta - \alpha(M_{an} - M_{irr})}$$

For  $H \geq 0$ ,  $\delta = 1$ .

For  $H < 0$ ,  $\delta = -1$ .

Comparison of this equation with a standard first order differential equation reveals that as increments in field strength,  $H$ , are made, the irreversible term  $M_{irr}$  attempts to track the reversible term  $M_{an}$ , but with a variable tracking gain of  $1/(K\delta - \alpha(M_{an} - M_{irr}))$ . The tracking error acts to create the hysteresis at the points where  $\delta$  changes sign. The main parameter that shapes the irreversible characteristic is  $K$ , which is called the bulk coupling coefficient. The parameter  $\alpha$  is called the inter-domain coupling factor, and is also used to define the effective field strength used when defining the anhysteretic curve:

$$H_{eff} = H + \alpha M$$

The value of  $\alpha$  affects the shape of the hysteresis curve, larger values acting to increase the B-axis intercepts. However, notice that for stability the term  $K\delta - \alpha(M_{an} - M_{irr})$  must be positive for  $\delta > 0$  and negative for  $\delta < 0$ . Therefore not all values of  $\alpha$  are permissible, a typical maximum value being of the order  $1e-3$ .

#### Procedure for Finding Approximate Values for Jiles-Atherton (JA) Equation Coefficients

You can determine representative parameters for the equation coefficients by using the following procedure:

- 1 Provide a value for the **Anhysteretic B-H gradient when H is zero** parameter ( $dM_{an}/dH_{eff}$  when  $H_{eff} = 0$ ) plus a data point  $[H_1, B_1]$  on the anhysteretic B-H curve. From these values, the block initialization determines values for  $\alpha$  and  $M_s$ .
- 2 Set the **Coefficient for reversible magnetization, c** parameter to achieve correct initial B-H gradient when starting a simulation from  $[H \ B] = [0 \ 0]$ . The value of  $c$  is approximately the ratio of this initial gradient to the **Anhysteretic B-H gradient when H is zero**. The value of  $c$  must be greater than 0 and less than 1.
- 3 Set the **Bulk coupling coefficient, K** parameter to the approximate magnitude of  $H$  when  $B = 0$  on the positive-going hysteresis curve.
- 4 Start with  $\alpha$  very small, and gradually increase to tune the value of  $B$  when crossing  $H = 0$  line. A typical value is in the range of  $1e-4$  to  $1e-3$ . Values that are too large cause the gradient of the B-H curve to tend to infinity, which is nonphysical and generates a run-time assertion error.

To get a good match against a predefined B-H curve, you may have to iterate on these four steps

#### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see "Set Priority and Initial Target for Block Variables".

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **N — North terminal**

magnetic

Magnetic conserving port associated with the block North terminal.

#### **S — South terminal**

magnetic

Magnetic conserving port associated with the block South terminal.

## Parameters

### Main

#### **Effective length — Effective length of the section being modeled**

0.032 m (default) | positive noninfinite scalar

Effective length of the section being modeled, that is, the average distance of the magnetic path.

The value must be positive and noninfinite.

#### **Effective cross-sectional area — Effective cross-sectional area of the section being modeled**

1.6e-5 m<sup>2</sup> (default) | positive noninfinite scalar

Effective cross-sectional area of the section being modeled, that is, the average area of the magnetic path.

The value must be positive and noninfinite.

#### **Averaging period for power logging — Excitation period used for averaging**

0 s (default) | scalar

Averaging period for the hysteresis losses calculation. These losses are proportional to the area enclosed by the B-H trajectory. If the block is excited at a known, fixed frequency, you can set this value to the corresponding excitation period to calculate the hysteresis loss. In this case, the block logs the hysteresis loss once per AC cycle to the variable `power_dissipated`. If you are using a fixed-step solver, this value must be an integer multiple of the simulation step size.

If the block is not excited at a known, fixed frequency, set this parameter to 0. In this case, the block sets `power_dissipated` to zero, and you can calculate the actual hysteresis loss by post-processing the logged variable `power_instantaneous`.

**Dependencies**

This parameter is visible only when, in the **B-H Curve** settings, the **Parameterized by** parameter is set to Nonlinear reluctance with hysteresis (JA model).

**B-H Curve****Parameterized by — B-H Curve parameterization method**

Reluctance (B-H curve) (default) | Linear reluctance | Reluctance with single saturation point | Nonlinear reluctance with hysteresis (JA model)

B-H Curve parameterization method.

**Dependencies**

Selecting a parameterization method makes related parameters in the **B-H Curve** settings. If the Nonlinear reluctance with hysteresis (JA model) is the selected parameterization method, in the **Main** settings, the **Averaging period for power logging** is visible.

**Relative magnetic permeability — Relative magnetic permeability**

5000 (default) | scalar

Relative magnetic permeability.

The value must be positive and noninfinite.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Linear reluctance.

**Unsaturated relative magnetic permeability — Relative magnetic permeability**

5000 (default) | scalar

Relative magnetic permeability for an unsaturated inductor.

**Dependencies**

This parameter is visible only when the **Parameterization method** parameter is set to Reluctance with single saturation point.

**Magnetic flux density at saturation — Magnetic flux density at saturation**

1.5 T (default) | positive noninfinite scalar

Magnetic flux density for a saturated inductor.

The value must be positive and noninfinite.

**Dependencies**

This parameter is visible only when the **Parameterization method** parameter is set to Reluctance with single saturation point.

**Magnetic field strength vector, H — Magnetic field strength vector**

[0, 200, 400, 600, 800, 1000] A/m (default) | noninfinite vector

Magnetic field strength,  $H$ , specified as a vector with the same number of elements as the magnetic flux density,  $B$ . The vector must start with zero and increase monotonically.

**Dependencies**

This parameter is visible only when the **Parameterization method** parameter is set to Reluctance (B-H curve).

**Magnetic flux density vector, B — Magnetic flux density vector**

[0, 1.25, 1.35, 1.44, 1.48, 1.49] T (default) | vector

Magnetic flux density,  $B$ , specified as a vector with the same number of elements as the magnetic field strength vector,  $H$ . The vector must start with zero and increase monotonically.

**Dependencies**

This parameter is visible only when the **Parameterization method** parameter is set to Reluctance (B-H curve).

**Anhyseretic B-H gradient when H is zero — Gradient of the anhyseretic B-H curve around zero field strength**

0.0063 m\*T/A (default) | scalar

The gradient of the anhyseretic (no hysteresis) B-H curve around zero field strength. Set it to the average gradient of the positive-going and negative-going hysteresis curves.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Flux density point on anhyseretic B-H curve — Flux density of the point for field strength measurement**

1.49 T (default) | scalar

Specify a point on the anhyseretic curve by providing its flux density value. Picking a point at high field strength where the positive-going and negative-going hysteresis curves align is the most accurate option.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Corresponding field strength — Field strength at measurement point**

1000 A/m (default) | scalar

The corresponding field strength for the point that you define by the **Flux density point on anhyseretic B-H curve** parameter.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Coefficient for reversible magnetization, c — Proportion of magnetization that is reversible**

0.1 (default) | scalar

The proportion of the magnetization that is reversible. The value must be greater than zero and less than one.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Bulk coupling coefficient, K — Jiles-Atherton equations parameter**

200 A/m (default) | scalar

The Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Inter-domain coupling factor, alpha — Jiles-Atherton equations parameter**

0.0001 (default) | scalar

The Jiles-Atherton parameter that primarily affects the points at which the B-H curves intersect the zero field strength line. Typical values are in the range of 1e-4 to 1e-3.

**Dependencies**

This parameter is visible only when the Parameterization method parameter is set to Nonlinear reluctance with hysteresis (JA model).

**Interpolation method — Interpolation method**

Linear (default) | Smooth

Select the method that the block uses to determine the output current values at intermediate time points that are not specified in the preceding vectors:

- **Linear** — Prioritize performance by using a linear function.
- **Smooth** — Prioritize accuracy by producing a continuous curve with continuous first-order derivatives.

## Version History

**Introduced in R2017b**

## References

[1] Jiles, D. C. and D. L. Atherton. "Theory of ferromagnetic hysteresis." *Journal of Magnetism and Magnetic Materials*. Vol. 61, 1986, pp. 48-60.

[2] Jiles, D. C. and D. L. Atherton. "Ferromagnetic hysteresis." *IEEE Transactions on Magnetics*. Vol. 19, No. 5, 1983, pp. 2183-2184.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Nonlinear Inductor | Nonlinear Transformer

# Nonlinear Transformer

Transformer with nonideal core

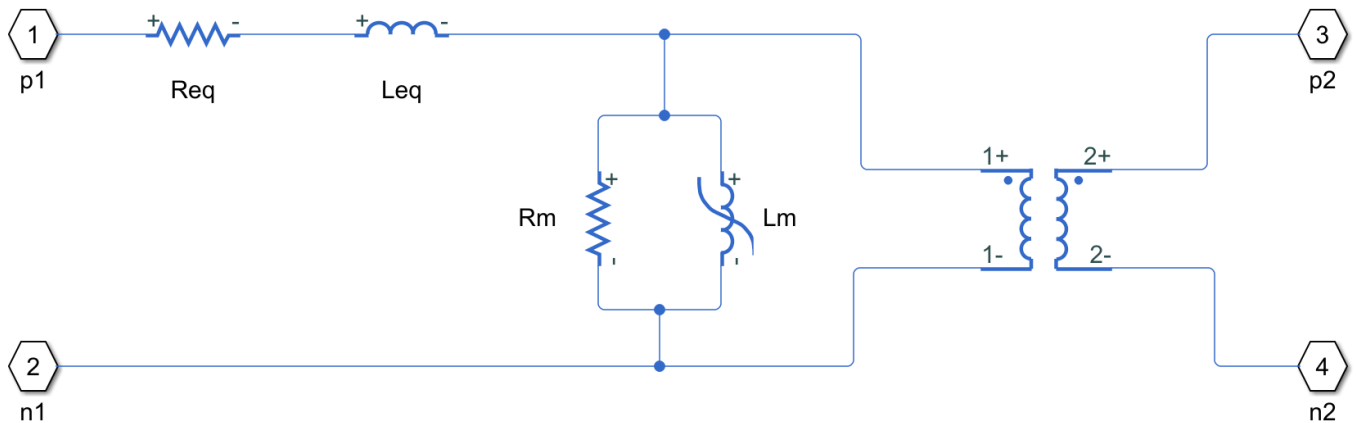
**Library:** Simscape / Electrical / Passive / Transformers



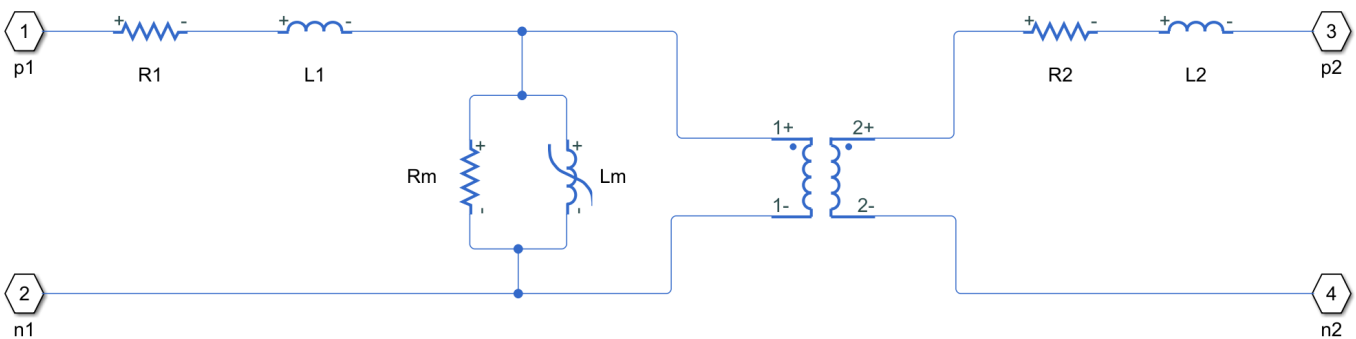
## Description

The Nonlinear Transformer block represents a transformer with a nonideal core. A core may be nonideal due to its magnetic properties and dimensions. The equivalent circuit topology depends upon which of the two winding leakage parameterization options you select:

- Combined primary and secondary values



- Separate primary and secondary values



where:



- $R_{eq}$  is the combined winding resistance.
- $L_{eq}$  is the combined leakage inductance.
- $R_1$  is the primary winding resistance.
- $L_1$  is the primary winding inductance.
- $R_2$  is the secondary winding resistance.
- $L_2$  is the secondary winding inductance.
- $R_m$  is the magnetization resistance.
- $L_m$  is the magnetization inductance.

The block provides the following parameterization options for the nonlinear magnetization inductance:

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

For more information, see the Nonlinear Inductor block reference page.

## Ports

### Conserving

#### **1+ — Primary winding positive polarity**

electrical

Electrical conserving port associated with the primary winding positive polarity.

#### **1- — Primary winding negative polarity**

electrical

Electrical conserving port associated with the primary winding negative polarity.

Electrical conserving port associated with the primary winding negative polarity.

#### **2+ — Secondary winding positive polarity**

electrical

Electrical conserving port associated with the secondary winding positive polarity.

#### **2- — Secondary winding negative polarity**

electrical

Electrical conserving port associated with the secondary winding negative polarity.

## Parameters

### Main

#### **Primary number of turns — Primary number of turns**

100 (default)

The number of turns of wire on the primary winding of the transformer.

#### **Secondary number of turns — Secondary number of turns**

200 (default)

The number of turns of wire on the secondary winding of the transformer.

#### **Winding parameterized by — Winding leakage parameterization**

Combined primary and secondary values (default) | Separate primary and secondary values

Select one of the following methods for the winding leakage parameterization:

- **Combined primary and secondary values** — Use the lumped resistance and inductance values representing the combined leakage in the primary and secondary windings. This is the default option.
- **Separate primary and secondary values** — Use separate resistances and inductances to represent leakages in the primary and secondary windings.

#### **Combined winding resistance — Combined winding resistance**

0.01 Ohm (default)

The lumped equivalent resistance  $R_{eq}$ , which represents the combined power loss of the primary and secondary windings.

#### **Dependencies**

This parameter is visible only when you select Combined primary and secondary values for the **Winding parameterized by** parameter.

#### **Combined leakage inductance — Combined leakage inductance**

0.0001 H (default)

The lumped equivalent inductance  $L_{eq}$ , which represents the combined magnetic flux loss of the primary and secondary windings.

#### **Dependencies**

This parameter is visible only when you select Combined primary and secondary values for the **Winding parameterized by** parameter.

#### **Primary winding resistance — Primary winding resistance**

0.01 Ohm (default)

The resistance  $R_1$ , which represents the power loss of the primary winding.

**Dependencies**

This parameter is visible only when you select Separate primary and secondary values for the **Winding parameterized by** parameter.

**Primary leakage inductance — Primary leakage inductance**

0.0001 H (default)

The inductance  $L1$ , which represents the magnetic flux loss of the primary winding.

**Dependencies**

This parameter is visible only when you select Separate primary and secondary values for the **Winding parameterized by** parameter.

**Secondary winding resistance — Secondary winding resistance**

0.01 Ohm (default)

The resistance  $R2$ , which represents the power loss of the secondary winding.

**Dependencies**

This parameter is visible only when you select Separate primary and secondary values for the **Winding parameterized by** parameter.

**Secondary leakage inductance — Secondary leakage inductance**

0.0001 H (default)

The inductance  $L2$ , which represents the magnetic flux loss of the secondary winding.

**Dependencies**

This parameter is visible only when you select Separate primary and secondary values for the **Winding parameterized by** parameter.

**Averaging period for power logging — Averaging period for power logging**

0 s (default)

Averaging period for power logging

**Magnetization****Magnetization resistance — Magnetization resistance**

100 Ohm (default)

The resistance  $Rm$ , which represents the magnetic losses in the transformer core.

**Magnetization inductance parameterized by — Nonlinear magnetization inductance parameterization**

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

Select one of the following methods for the nonlinear magnetization inductance parameterization:

- **Single inductance (linear)** — Provide the unsaturated inductance value.
- **Single saturation point** — Provide the values for the unsaturated and saturated inductances, as well as saturation magnetic flux. This is the default option.
- **Magnetic flux versus current characteristic** — Provide the current vector and the magnetic flux vector, to populate the magnetic flux versus current lookup table.
- **Magnetic flux density versus field strength characteristic** — Provide the values for effective core length and cross-sectional area, as well as the magnetic field strength vector and the magnetic flux density vector, to populate the magnetic flux density versus magnetic field strength lookup table.
- **Magnetic flux density versus field strength characteristic with hysteresis** — In addition to the number of turns and the effective core length and cross-sectional area, provide the values for the initial anhysteretic B-H curve gradient, the magnetic flux density and field strength at a certain point on the B-H curve, as well as the coefficient for the reversible magnetization, bulk coupling coefficient, and inter-domain coupling factor, to define magnetic flux density as a function of both the current value and the history of the field strength.

**Unsaturated inductance — Unsaturated inductance**

4e-2 H (default)

The value of inductance used when the magnetization inductance  $L_m$  is operating in its linear region.

**Dependencies**

This parameter is visible only when you select **Single inductance (linear)** or **Single saturation point** for the **Magnetization inductance parameterized by** parameter.

**Saturated inductance — Saturated inductance**

1e-2 H (default)

The value of inductance used when the magnetization inductance  $L_m$  is operating beyond its saturation point.

**Dependencies**

This parameter is visible only when you select **Single saturation point** for the **Magnetization inductance parameterized by** parameter.

**Saturation magnetic flux — Saturation magnetic flux**

1.6e-04 Wb (default)

The value of magnetic flux at which the magnetization inductance  $L_m$  saturates.

**Dependencies**

This parameter is visible only when you select **Single saturation point** for the **Magnetization inductance parameterized by** parameter.

**Current vector,  $i$  — Current data**

[0, .4, .8, 1.2, 1.6, 2] A (default)

The current data used to populate the magnetic flux versus current lookup table.

**Dependencies**

This parameter is visible only when you select Magnetic flux versus current characteristic for the **Magnetization inductance parameterized by** parameter.

**Magnetic flux vector, phi – Magnetic flux vector**

[0, .161, .25, .284, .295, .299] .\* 1e-3 Wb (default)

The magnetic flux data used to populate the magnetic flux versus current lookup table.

**Dependencies**

This parameter is visible only when you select Magnetic flux versus current characteristic for the **Magnetization inductance parameterized by** parameter.

**Magnetic field strength vector, H – Magnetic field strength vector**

[0, 200, 400, 600, 800, 1000] A/m (default)

The magnetic field strength data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic for the **Magnetization inductance parameterized by** parameter.

**Magnetic flux density vector, B – Magnetic flux density vector**

[0, .81, 1.25, 1.42, 1.48, 1.49] T (default)

The magnetic flux density data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic for the **Magnetization inductance parameterized by** parameter.

**Effective length – Effective length**

0.2 m (default)

The effective core length, that is, the average distance of the magnetic path around the transformer core.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic for the **Magnetization inductance parameterized by** parameter.

**Effective cross-sectional area – Effective cross-sectional area**

2e-4 m<sup>2</sup> (default)

The effective core cross-sectional area, that is, the average area of the magnetic path around the transformer core.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic for the **Magnetization inductance parameterized by** parameter.

**Anhyseretic B-H gradient when H is zero – Anhyseretic B-H gradient around zero field strength**

0.005 m\*T/A (default)

The gradient of the anhyseretic (no hysteresis) B-H curve around zero field strength. Set it to the average gradient of the positive-going and negative-going hysteresis curves.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Flux density point on anhyseretic B-H curve – Flux density point on anhyseretic B-H curve**

1.49 T (default)

Specify a point on the anhyseretic curve by providing its flux density value. Picking a point at high field strength where the positive-going and negative-going hysteresis curves align is the most accurate option.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Corresponding field strength – Corresponding field strength**

1000 A/m (default)

The corresponding field strength for the point that you define by the **Flux density point on anhyseretic B-H curve** parameter.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Coefficient for reversible magnetization, c – Coefficient for reversible magnetization**

0.1 (default)

The proportion of the magnetization that is reversible. The value should be greater than zero and less than one.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Bulk coupling coefficient, K – Bulk coupling coefficient**

200 A/m (default)

The Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Inter-domain coupling factor, alpha – Inter-domain coupling factor**

1e-4 (default)

The Jiles-Atherton parameter that primarily affects the points at which the B-H curves intersect the zero field strength line. Typical values are in the range of 1e-4 to 1e-3.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Interpolation option – Interpolation option**

Linear (default) | Smooth

The lookup table interpolation option. Select one of the following interpolation methods:

- **Linear** — Select this option to get the best performance.
- **Smooth** — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

**Dependencies**

This parameter is visible only when you select Magnetic flux versus current characteristic or Magnetic flux density versus field strength characteristic for the **Magnetization inductance parameterized by** parameter.

**Initial Conditions****Combined leakage inductance initial current – Combined leakage inductance initial current**

0 A (default)

The value of current through the combined leakage inductance  $L_{eq}$  at time zero.

**Dependencies**

This parameter is visible only when you select Combined primary and secondary values for the **Winding parameterized by** parameter on the **Main** tab.

**Primary leakage inductance initial current – Primary leakage inductance initial current**

0 A (default)

The value of current through the primary leakage inductance  $L1$  at time zero.

**Dependencies**

This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab.

**Secondary leakage inductance initial current — Secondary leakage inductance initial current**

0 A (default)

The value of current through the secondary leakage inductance  $L2$  at time zero.

**Dependencies**

This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab.

**Specify magnetization inductance initial state by — Initial state specification option**

Current (default) | Magnetic flux

Select the appropriate initial state specification option:

- **Current** — Specify the initial state of the magnetization inductance  $Lm$  by the initial current. This is the default option.
- **Magnetic flux** — Specify the initial state of the magnetization inductance  $Lm$  by the magnetic flux.

**Dependencies**

This parameter is not visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Magnetization inductance parameterized by** parameter on the **Magnetization** tab.

**Magnetization inductance initial current — Magnetization inductance initial current**

0 A (default)

The initial current value used to calculate the value of magnetic flux within the magnetization inductance  $Lm$  at time zero. This is the current passing through the magnetization inductance  $Lm$ . Total magnetization current consists of current passing through the magnetization resistance  $Rm$  and current passing through the magnetization inductance  $Lm$ .

**Dependencies**

This parameter is visible only when you select `Current` for the **Specify magnetization inductance initial state by** parameter.

**Magnetization inductance initial magnetic flux — Magnetization inductance initial magnetic flux**

0 Wb (default)

The value of the magnetic flux in the magnetization inductance  $Lm$  at time zero.

**Dependencies**

This parameter is visible only when you select `Magnetic flux` for the **Specify magnetization inductance initial state by** parameter.



**Magnetization inductance initial magnetic flux density – Magnetization inductance initial magnetic flux density**

0 T (default)

The value of magnetic flux density at time zero. This parameter is visible only when you select Magnetic flux density versus magnetic field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter on the **Magnetization** tab.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Magnetization inductance initial field strength – Magnetization inductance initial field strength**

0 A/m (default)

The value of magnetic field strength at time zero.

**Dependencies**

This parameter is visible only when you select Magnetic flux density versus field strength characteristic with hysteresis for the **Magnetization inductance parameterized by** parameter.

**Parasitics****Combined leakage inductance parasitic parallel conductance – Combined leakage inductance parasitic parallel conductance**

1e-9 1/0hm (default)

Use this parameter to represent small parasitic effects in parallel to the combined leakage inductance  $L_{eq}$ . A small parallel conductance may be required for the simulation of some circuit topologies.

**Dependencies**

This parameter is visible only when you select Combined primary and secondary values for the **Winding parameterized by** parameter on the **Main** tab.

**Primary leakage inductance parasitic parallel conductance – Primary leakage inductance parasitic parallel conductance**

1e-9 1/0hm (default)

Use this parameter to represent small parasitic effects in parallel to the primary leakage inductance  $L_1$ . A small parallel conductance may be required for the simulation of some circuit topologies.

**Dependencies**

This parameter is visible only when you select Separate primary and secondary values for the **Winding parameterized by** parameter on the **Main** tab.

**Secondary leakage inductance parasitic parallel conductance – Secondary leakage inductance parasitic parallel conductance**

1e-9 (default)

Use this parameter to represent small parasitic effects in parallel to the secondary leakage inductance  $L2$ . A small parallel conductance may be required for the simulation of some circuit topologies.

**Dependencies**

This parameter is visible only when you select **Separate primary and secondary values** for the **Winding parameterized by** parameter on the **Main** tab.

## Version History

**Introduced in R2012b**

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Earthing Transformer | Ideal Transformer | Tap-Changing Transformer | Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer

# NPN Bipolar Transistor

NPN bipolar transistor using enhanced Ebers-Moll equations

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The NPN Bipolar Transistor block uses a variant of the Ebers-Moll equations to represent an NPN bipolar transistor. The Ebers-Moll equations are based on two exponential diodes plus two current-controlled current sources. The NPN Bipolar Transistor block provides the following enhancements to that model:

- Early voltage effect
- Optional base, collector, and emitter resistances.
- Optional fixed base-emitter and base-collector capacitances.

The collector and base currents are:

$$I_C = IS \left[ \left( e^{qV_{BE}/(kT_{m1})} - e^{qV_{BC}/(kT_{m1})} \right) \left( 1 - \frac{V_{BC}}{V_A} \right) - \frac{1}{\beta_R} \left( e^{qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

$$I_B = IS \left[ \frac{1}{\beta_F} \left( e^{qV_{BE}/(kT_{m1})} - 1 \right) + \frac{1}{\beta_R} \left( e^{qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

Where:

- $I_B$  and  $I_C$  are base and collector currents, defined as positive into the device.
- $IS$  is the saturation current.
- $V_{BE}$  is the base-emitter voltage and  $V_{BC}$  is the base-collector voltage.
- $\beta_F$  is the ideal maximum forward current gain BF
- $\beta_R$  is the ideal maximum reverse current gain BR
- $V_A$  is the forward Early voltage VAF
- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $T_{m1}$  is the transistor temperature, as defined by the **Measurement temperature** parameter value.

You can specify the transistor behavior using datasheet parameters that the block uses to calculate the parameters for these equations, or you can specify the equation parameters directly.

If  $qV_{BC} / (kT_{m1}) > 40$  or  $qV_{BE} / (kT_{m1}) > 40$ , the corresponding exponential terms in the equations are replaced with  $(qV_{BC} / (kT_{m1}) - 39)e^{40}$  and  $(qV_{BE} / (kT_{m1}) - 39)e^{40}$ , respectively. This helps prevent numerical issues associated with the steep gradient of the exponential function  $e^x$  at large values of  $x$ .

Similarly, if  $qV_{BC} / (kT_{m1}) < -39$  or  $qV_{BE} / (kT_{m1}) < -39$  then the corresponding exponential terms in the equations are replaced with  $(qV_{BC} / (kT_{m1}) + 40)e^{-39}$  and  $(qV_{BE} / (kT_{m1}) + 40)e^{-39}$ , respectively.

Optionally, you can specify fixed capacitances across the base-emitter and base-collector junctions. You also have the option to specify base, collector, and emitter connection resistances.

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. You can optionally include modeling the dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the junction capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The measurement temperature value,  $T_{m1}$ , is replaced with the simulation temperature,  $T_s$ . The saturation current,  $IS$ , and the forward and reverse gains ( $\beta_F$  and  $\beta_R$ ) become a function of temperature according to the following equations:

$$IS_{T_s} = IS_{T_{m1}} \cdot (T_s/T_{m1})^{XTI} \cdot \exp\left(-\frac{EG}{kT_s}(1 - T_s/T_{m1})\right)$$

$$\beta_{F_s} = \beta_{F_{m1}} \left(\frac{T_s}{T_{m1}}\right)^{XTB}$$

$$\beta_{R_s} = \beta_{R_{m1}} \left(\frac{T_s}{T_{m1}}\right)^{XTB}$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $IS_{T_{m1}}$  is the saturation current at the measurement temperature.
- $IS_{T_s}$  is the saturation current at the simulation temperature. This is the saturation current value used in the bipolar transistor equations when temperature dependence is modeled.
- $\beta_{F_{m1}}$  and  $\beta_{R_{m1}}$  are the forward and reverse gains at the measurement temperature.
- $\beta_{F_s}$  and  $\beta_{R_s}$  are the forward and reverse gains at the simulation temperature. These are the values used in the bipolar transistor equations when temperature dependence is modeled.
- $EG$  is the energy gap for the semiconductor type measured in Joules. The value for silicon is usually taken to be 1.11 eV, where 1 eV is 1.602e-19 Joules.
- $XTI$  is the saturation current temperature exponent.
- $XTB$  is the forward and reverse gain temperature coefficient.
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).

Appropriate values for  $XTI$  and  $EG$  depend on the type of transistor and the semiconductor material used. In practice, the values of  $XTI$ ,  $EG$ , and  $XTB$  need tuning to model the exact behavior of a particular transistor. Some manufacturers quote these tuned values in a SPICE Netlist, and you can read off the appropriate values. Otherwise you can determine values for  $XTI$ ,  $EG$ , and  $XTB$  by using a datasheet-defined data at a higher temperature  $T_{m2}$ . The block provides a datasheet parameterization option for this.

You can also tune the values of  $X_{TI}$ ,  $EG$ , and  $X_{TB}$  yourself, to match lab data for your particular device. You can use Simulink Design Optimization software to help tune the values.

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- The block does not account for temperature-dependent effects on the junction capacitances.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.

### Ports

#### Conserving

##### **B — Base terminal**

electrical

Electrical conserving port associated with the transistor base terminal

##### **C — Collector terminal**

electrical

Electrical conserving port associated with the transistor collector terminal

##### **E — Emitter terminal**

electrical

Electrical conserving port associated with the transistor emitter terminal

**H — Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

**Main****Parameterization — Block parameterization**

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- **Specify from a datasheet** — Provide parameters that the block converts to equations that describe the transistor. The block calculates the forward Early voltage  $VAF$  as  $I_c/h_{oe}$ , where  $I_c$  is the **Collector current at which h-parameters are defined** parameter value, and  $h_{oe}$  is the **Output admittance  $h_{oe}$**  parameter value [1]. The block sets  $BF$  to the small-signal **Forward current transfer ratio  $h_{fe}$**  value. The block calculates the saturation current  $IS$  from the specified **Voltage  $V_{be}$**  value and the corresponding **Current  $I_b$  for voltage  $V_{be}$**  value when  $I_c$  is zero. This is the default method.
- **Specify using equation parameters directly** — Provide equation parameters  $IS$ ,  $BF$ , and  $VAF$ .

**Forward current transfer ratio  $h_{fe}$  — Forward current transfer ratio**

100 (default)

Small-signal current gain.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Output admittance  $h_{oe}$  — Output admittance**

50e-6 1/0hm (default)

Derivative of the collector current with respect to the collector-emitter voltage for a fixed base current.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Collector current at which h-parameters are defined – Collector current at which h-parameters are defined**

1 mA (default)

The h-parameters vary with operating point, and are defined for this value of the collector current.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Collector-emitter voltage at which h-parameters are defined – Collector-emitter voltage at which h-parameters are defined**

5 V (default)

The h-parameters vary with operating point, and are defined for this value of the collector-emitter voltage.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Voltage Vbe – Base-emitter voltage**

0.55 V (default)

Base-emitter voltage when the base current is  $I_b$ . The [  $V_{be}$   $I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Current  $I_b$  for voltage Vbe – Current  $I_b$  for voltage Vbe**

0.5 mA (default)

Base current when the base-emitter voltage is  $V_{be}$ . The [  $V_{be}$   $I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Forward current transfer ratio BF – Forward current transfer ratio**

100 (default)

Ideal maximum forward current gain.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Saturation current  $I_S$  – Saturation current**

1e-14 A (default)

Transistor saturation current.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Forward Early voltage VAF — Forward Early voltage**

200 V (default)

In the standard Ebers-Moll equations, the gradient of the  $I_c$  versus  $V_{ce}$  curve is zero in the normal active region. The additional forward Early voltage term increases this gradient. The intercept on the  $V_{ce}$ -axis is equal to  $-VAF$  when the linear region is extrapolated.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Reverse current transfer ratio BR — Reverse current transfer ratio**

1 (default)

Ideal maximum reverse current gain. This value is often not quoted in manufacturer datasheets, because it is not significant when the transistor is biased to operate in the normal active region. When the value is not known and the transistor is not to be operated on the inverse region, use the default value of 1.

**Measurement temperature — Measurement temperature**

25 degC (default)

Temperature  $T_{m1}$  at which  $V_{be}$  and  $I_b$ , or  $I_S$ , are measured. The default value is 25 °C.

**Ohmic Resistance****Collector resistance RC — Collector resistance**

0.01 Ohm (default)

Resistance at the collector.

**Emitter resistance RE — Emitter resistance**

1e-4 Ohm (default)

Resistance at the emitter.

**Zero bias base resistance RB — Zero bias base resistance RB**

1 Ohm (default)

Resistance at the base at zero bias.

**Capacitance****Base-collector junction capacitance — Base-collector junction capacitance**

5 pF (default)

Parasitic capacitance across the base-collector junction.



**Base-emitter junction capacitance — Base-emitter junction capacitance**

5 pF (default)

Parasitic capacitance across the base-emitter junction.

**Total forward transit time — Total forward transit time**

0 us (default)

Represents the mean time for the minority carriers to cross the base region from the emitter to the collector, and is often denoted by the parameter TF [1].

**Total reverse transit time — Total reverse transit time**

0 us (default)

Represents the mean time for the minority carriers to cross the base region from the collector to the emitter, and is often denoted by the parameter TR [1].

**Temperature Dependence****Parameterization — Temperature dependence parameterization**

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None — Simulate at parameter measurement temperature — Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Model temperature dependence — Provide a value for simulation temperature, to model temperature-dependent effects. You also have to provide a set of additional parameters depending on the block parameterization method. If you parameterize the block from a datasheet, you have to provide values for a second [ *Vbe Ib* ] data pair and  $h_{fe}$  at second measurement temperature. If you parameterize by directly specifying equation parameters, you have to provide the values for *XTI*, *EG*, and *XTB*.

**Forward current transfer ratio,  $h_{fe}$ , at second measurement temperature — Forward current transfer ratio,  $h_{fe}$ , at second measurement temperature**

125 (default)

Small-signal current gain at the second measurement temperature. It must be quoted at the same collector-emitter voltage and collector current as for the **Forward current transfer ratio  $h_{fe}$**  parameter on the **Main** tab.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Voltage  $V_{be}$  at second measurement temperature — Voltage  $V_{be}$  at second measurement temperature**

0.45 V (default)

Base-emitter voltage when the base current is  $I_b$  and the temperature is set to the second measurement temperature. The [ *Vbe Ib* ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Current  $I_b$  for voltage  $V_{be}$  at second measurement temperature – Current  $I_b$  for voltage  $V_{be}$  at second measurement temperature**

0.5 mA (default)

Base current when the base-emitter voltage is  $V_{be}$  and the temperature is set to the second measurement temperature. The [  $V_{be}$   $I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Second temperature  $T_{m2}$  at which  $h_{fe}$ ,  $V_{be}$ , and  $I_b$  are measured.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Current gain temperature coefficient, XTB – Current gain temperature coefficient**

0 (default)

Current gain temperature coefficient value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Energy gap, EG – Energy gap**

1.11 eV (default)

Energy gap value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Saturation current temperature coefficient value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

**Version History****Introduced in R2008a****Thermal network simlog paths***Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

**References**

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.
- [2] H. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition, Cambridge University Press, 1984.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Diode | PNP Bipolar Transistor

## On-Off Delay

Boolean-signal delay

**Library:** Simscape / Electrical / Control / General Control



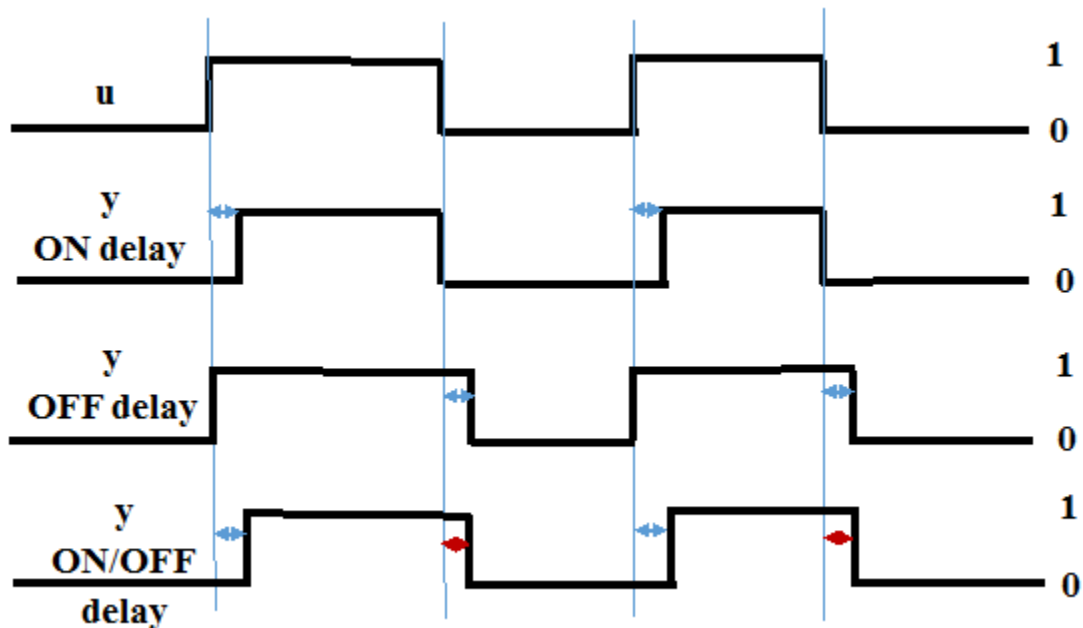
### Description

The On-Off Delay block applies a delay on the Boolean input signal.

A time delay is added when a transition is detected in the input signal. This block allows you to add a time delay to the input signal when:

- An ON transition (input change from 0 to 1) is detected,
- An OFF transition (input signal change from 1 to 0) is detected, or
- Either transition is detected.

The operation of the on-off delay is represented in the following figure:



### Ports

#### Input

**u** – Input signal

0 | 1

Input Boolean signal.

Data Types: Boolean

## Output

**y — Output signal**

0 | 1

Output signal with delays.

Data Types: Boolean

## Parameters

**ON delay time (s) — Input ON delay time**

0.01 (default) | 0 | positive scalar

Specify delay time when input is ON.

**OFF delay time (s) — Input OFF delay time**

0 (default) | positive scalar

Specify delay time when input is OFF.

**Initial state — Signal initial state**

0 (default) | 1

Specify initial state.

**Sample time — Block sample time**

0.001 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For discrete-time operation, set the sample time to a positive value. For continuous-time operation, set the sample time to 0.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2018b

## Extended Capabilities

### C/C++ Code Generation

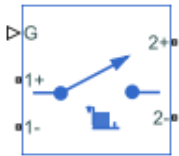
Generate C and C++ code using Simulink® Coder™.

## **See Also**

# One-Quadrant Chopper

Controller-driven one-quadrant chopper

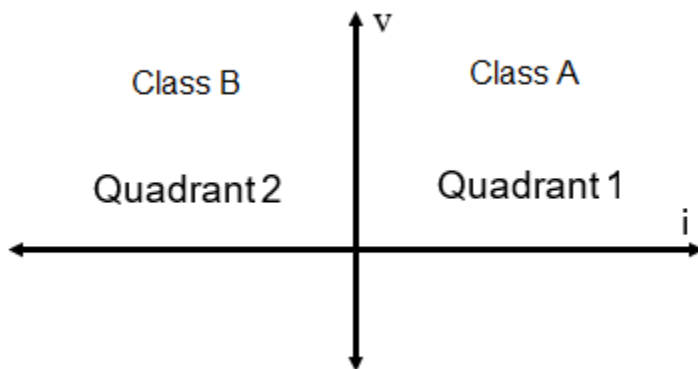
**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



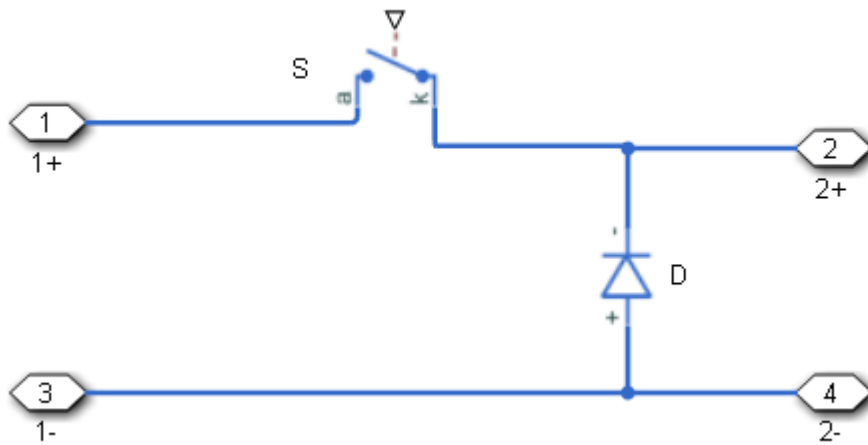
## Description

The One-Quadrant Chopper block represents a one-quadrant controlled chopper for converting a fixed DC input to a variable DC output.

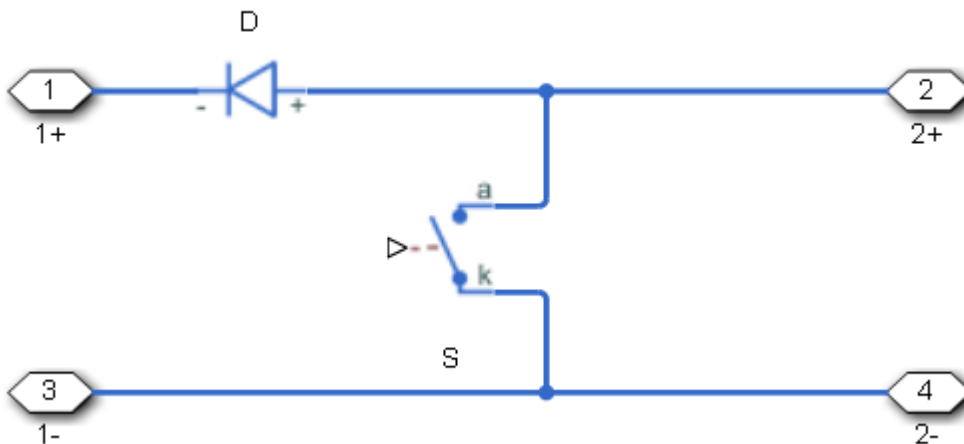
The circuit topology and quadrant depend on the class of chopper that you specify.



A first-quadrant or class A chopper contains a power switch and a diode.



A second-quadrant or class B chopper also contains a power switch and a diode.



For either topology, the switch S can be a fully controlled switching device (for example, an IGBT or GTO) or a partially controlled switching device (for example, a thyristor).

Options for the switching device type are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).



- **Averaged Switch** — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

## Protection

An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load. To protect the semiconductor device, an integral protection diode provides a conduction path for reverse current.

To include and configure the internal protection diode block for the *S* switching device, use the **Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.

Goals		Value to Select	Integral Protection Diode
Do not include protection.		None	None
Include protection.	Prioritize simulation speed.	Diode with no dynamics	The Diode block
	Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

## Gate Control

You can connect gate-control voltage signals to the gate ports of the switching devices.

- For the PS control port model:
  - 1 Convert a Simulink gate-control voltage signal to a physical signal using a Simulink-PS Converter block.
  - 2 Connect the Simulink-PS Converter block to the **G** port.
- For the electrical control ports model:
  - 1 Connect a Simscape electrical-domain positive DC voltage signal to the **G+** port.
  - 2 Connect the Simscape electrical-domain negative DC voltage signal to the **G-** port.

## Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to `Averaged switch` and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input  $\mathbf{G}$ . To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Ports

### Input

#### **G** — Switching device gate control

physical signal | scalar

Physical signal port associated with the gate terminals of the switching device.

#### Dependencies

To enable this port, set **Gate-control port** to PS.

Data Types: double

### Conserving

#### **G+** — Switching device gate control positive terminal

electrical | scalar

Positive electrical conserving port associated with the positive gate terminal of the switching device.

#### Dependencies

To enable this port, set **Gate-control port** to Electrical.

Data Types: double

#### **G-** — Switching device gate control negative terminal

electrical | scalar

Negative electrical conserving port associated with the negative gate terminal of the switching device.

#### Dependencies

To enable this port, set **Gate-control port** to Electrical.

Data Types: double

### **1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

### **1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

### **2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

### **2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

## **Parameters**

### **Switching Devices**

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

**Switching Devices Parameter Dependencies**

<b>Switching Device</b>					
<b>Chopper type</b> — Choose Class A - first quadrant or Class B - second quadrant.					
<b>Switching device</b> — Choose Ideal Semiconductor Switch, GTO, IGBT, MOSFET, Thyristor, or Averaged Switch.					
<b>Ideal Semiconductor Switch</b>	<b>GTO</b>	<b>IGBT</b>	<b>MOSFET</b>	<b>Thyristor</b>	<b>Averaged Switch</b>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

**Gate-control port — Whether to specify physical or electrical control ports**

PS (default) | Electrical

Whether to specify physical or electrical control ports for the switching device gate.

**Chopper type — Chopper class**

Class A - first quadrant (default) | Class B - second quadrant

Chopper class.

**Switching device — Switch type**

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Thyristor | Averaged Switch

Switching device type for the converter.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Forward voltage — Voltage**

0.8 Ohm (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### On-state resistance — Resistance

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on
- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Drain-source on resistance — Resistance

0.001 Ohm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Off-state conductance — Conductance

1e-5 1/Ohm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

#### Dependencies

See the Switching Devices Parameter Dependencies table.

**Threshold voltage — Voltage threshold**

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate trigger voltage,  $V_{gt}$  — Voltage threshold**

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate turn-off voltage,  $V_{gt\_off}$  — Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Holding current — Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

## Diode

The visibility of **Diode** parameters depends on how you configure the protection diode **Model dynamics** and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

### Diode Parameter Dependencies

Diode			
<b>Model dynamics</b> — Choose Diode with no dynamics or Diode with charge dynamics.			
<b>Diode with no dynamics</b>	<b>Diode with charge dynamics</b>		
Forward voltage	Forward voltage		
On resistance	On resistance		
Off conductance	Off conductance		
	Junction capacitance		
	Peak reverse current, $i_{RM}$		
	Initial forward current when measuring $i_{RM}$		
	Rate of change of current when measuring $i_{RM}$		
	<b>Reverse recovery time parameterization</b> — Choose Specify stretch factor, Specify reverse recovery time directly, or Specify reverse recovery charge.		
	<b>Specify stretch factor</b>	<b>Specify reverse recovery time directly</b>	<b>Specify reverse recovery charge</b>
	Reverse recovery time stretch factor	Reverse recovery time, $t_{rr}$	Reverse recovery charge, $Q_{rr}$

### Model dynamics — Diode model

Diode with no dynamics (default) | Diode with charge dynamics

Diode type. The options are:

- Diode with no dynamics — Select this option to prioritize simulation speed using the Diode block.
- Diode with charge dynamics — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation model of the Diode block.

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

### Dependencies

See the Diode Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**On resistance — Resistance**

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

**Off conductance — Conductance**

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

**Junction capacitance — Capacitance**

50 nF (default) | scalar

Diode junction capacitance.

**Dependencies**

See the Diode Parameter Dependencies table.

**Peak reverse current,  $i_{RM}$  — Current**

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

**Dependencies**

See the Diode Parameter Dependencies table.

**Initial forward current when measuring  $i_{RM}$  — Current**

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

See the Diode Parameter Dependencies table.

**Rate of change of current when measuring  $i_{RM}$  — Current change rate**

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time parameterization — Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select **Specify stretch factor** or **Specify reverse recovery charge**, you can specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.



**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time, trr**. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery time, trr – Time**

15 μs (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time, trr** parameter must be greater than the value of the **Peak reverse current, i<sub>RM</sub>** parameter divided by the value of the **Rate of change of current when measuring i<sub>RM</sub>** parameter.

**Dependencies**

See the Diode Parameter Dependencies table.

**Reverse recovery charge, Q<sub>rr</sub> – Charge**

1500 s\*μA (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, i<sub>RM</sub>**.
- $a$  is the value specified for **Rate of change of current when measuring i<sub>RM</sub>**.

**Dependencies**

See the Diode Parameter Dependencies table.

**Snubbers**

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.

## Snubbers Parameter Dependencies

Snubbers	
Snubber — Choose None or RC Snubber.	
None	RC Snubber
	Snubber resistance
	Snubber capacitance

### Snubber — Snubber model

None (default) | RC snubber

Switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber resistance — Resistance

0.1 Ohm (default) | scalar

Resistance of the switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber capacitance — Capacitance

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

#### Dependencies

See the Snubbers Parameter Dependencies table.

## Version History

Introduced in R2018b

## References

[1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics*. 2nd Ed. Hoboken, NJ: John Wiley & Sons Inc., 2010.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Two-Quadrant Chopper | Four-Quadrant Chopper | Average-Value Chopper

# Open Circuit (Three-Phase)

Three-phase terminator that draws no current

**Library:** Simscape / Electrical / Connectors & References



## Description

The Open Circuit (Three-Phase) block models a three-phase connection that draws no current on any of the three phases.

## Ports

### Conserving

~ — **Three-phase port**  
electrical

Expandable composite (**a,b, c**) three-phase port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Version History

### Introduced in R2013b

### Capping unconnected ports is no longer required

*Behavior changed in R2019b*

This block is no longer required. Starting in R2019b, the restriction that disallowed unconnected conserving ports in Simscape models has been lifted. Now, if you leave a conserving port unconnected, the physical network sets all the Through variables at this port to 0. However, you can still use terminator blocks to improve diagram readability.

There are no plans to remove the terminator blocks. All the existing models using these blocks work the same as in previous releases.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Floating Neutral (Three-Phase) | Grounded Neutral (Three-Phase) | Neutral Port (Three-Phase)

# Operational Transconductance Amplifier

Behavioral representation of operational transconductance amplifier

**Library:** Simscape / Electrical / Integrated Circuits



## Description

The Operational Transconductance Amplifier block 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 exploit the fact that the transconductance gain is a function of current flowing into the control current pin.

To support faster simulation, the behavioral representation does not model the detailed transistor implementation. Therefore, 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. The dynamics are approximated by a first-order lag, based on the value you specify for the block parameter **Bandwidth**.

## Control Current

The control current pin C is maintained at the voltage that you specify for the **Minimum output voltage**. In practice, the **Minimum output voltage** equals the negative supply voltage plus the transistor collector-emitter voltage drop. For example, if the **Minimum output voltage** for a supply voltage of +15V is -14.5, then to achieve a control current of 500 $\mu$ A, a resistor connected between the +15V rail and the control current pin must have a value of  $(15 - (-14.5)) / 500\text{e-}6 = 59\text{k}\Omega$ .

## Transconductance

The relationship between input voltage,  $v$ , and transconductance current,  $i_{gm}$ , is:

$$v = v_+ - v_-$$

$$i_{gm} = g_m \cdot v$$

$$g_m = \frac{g_{m0} \cdot i_c}{i_{c0}}$$

where:

- $v_+$  is the voltage presented at the block + pin.
- $v_-$  is the voltage presented at the block - pin.
- $g_m$  is the transconductance.
- $i_c$  is the control current flowing into the control current pin C.
- $i_{c0}$  is the reference control current, that is, the control current at which transconductance is quoted on the datasheet.

- $g_{m0}$  is the transconductance measured at the reference control current  $i_{c0}$ .

Therefore, increasing control current increases the transconductance.

### Output Resistance and Determining Output Current

The output resistance,  $R_{out}$ , is defined by:

$$i_{gm} + i_o = \frac{v_o}{R_{out}}$$

$$R_{out} = \frac{R_{out0} \cdot i_{c0}}{i_c}$$

where:

- $i_{gm}$  is the transconductance current.
- $i_o$  is the output current, defined as positive if flowing into the transconductance amplifier output pin.
- $i_c$  is the control current flowing into the control current pin C.
- $i_{c0}$  is the reference control current, that is, the control current at which output resistance is quoted on the datasheet.
- $R_{out0}$  is the output resistance measured at the reference control current  $i_{c0}$ .

Therefore, increasing control current reduces output resistance.

### Input Resistance

The relationship between input voltage,  $v$ , across the + and - pins and the current flowing,  $i$ , is:

$$\frac{v}{i} = R_{in}$$

$$R_{in} = \frac{R_{in0} \cdot i_{c0}}{i_c}$$

where:

- $i_c$  is the control current flowing into the control current pin C.
- $R_{in}$  is the input resistance for the current control current value,  $i_c$ .
- $i_{c0}$  is the reference control current, that is, the control current at which input resistance is quoted on the datasheet.
- $R_{in0}$  is the input resistance measured at the reference control current  $i_{c0}$ .

Therefore, increasing control current reduces input resistance.

### Limits

Because of the physical construction of an operational transconductance amplifier based on current mirrors, the transconductance current  $i_{gm}$  cannot exceed the control current. Hence the value of  $i_{gm}$  is limited by:

$$-i_c \leq i_{gm} \leq i_c \quad (1-46)$$

The output voltage is also limited by the supply voltage:

$$V_{\min} \leq v_o \leq V_{\max} \quad (1-47)$$

where  $V_{\min}$  is the **Minimum output voltage**, and  $V_{\max}$  is the **Maximum output voltage**. Output voltage limiting is implemented by adding a low resistance to the output when the voltage limit is exceeded. The value of this resistance is set by the **Additional output resistance at voltage swing limits** parameter.

The transconductance current is also slew-rate limited, a value for slew rate limiting typically being given on datasheets:

$$-\mu \leq \frac{di_{gm}}{dt} \leq \mu$$

where  $\mu$  is the **Maximum current slew rate**.

## Ports

### Conserving

#### + – Non-inverting input

electrical

Electrical conserving port associated with the op-amp non-inverting input.

#### - – Negative voltage

electrical

Electrical conserving port associated with the op-amp inverting input.

#### C – Control current

electrical

Electrical conserving port associated with the op-amp control current.

#### OUT – Output current

electrical

Electrical conserving port associated with the op-amp output. The port name is hidden on the block icon, but you can see it in simulation data logs.

## Parameters

### Nominal Measurements

#### Transconductance – Transconductance

9600 uS (default)

The transconductance,  $g_m$ , when the control current is equal to the **Reference control current**. This is the ratio of the transconductance current,  $i_{gm}$ , to the voltage difference,  $v$ , across the + and - pins.

#### Input resistance – Input resistance

25 kOhm (default)

The input resistance,  $R_{in}$ , when the control current is equal to the **Reference control current**. The input resistance is the ratio of the voltage difference,  $v$ , across the + and - pins to the current flowing from the + to the - pin.

### **Output resistance — Output resistance**

3 MOhm (default)

The output resistance,  $R_{out}$ , when the control current is equal to the **Reference control current**. See above for the equation defining output resistance.

### **Reference control current — Reference control current**

500  $\mu$ A (default)

The control current at which the **Transconductance**, **Input resistance**, and **Output resistance** are quoted.

### **Dynamics**

#### **Dynamics — Dynamics**

No lag (default) | Finite bandwidth with slew rate limiting

Select one of the following options:

- **No lag** — Do not model the dynamics of the relationship between output current and input voltage. This is the default.
- **Finite bandwidth with slew rate limiting** — Model the dynamics of the relationship between output current and input voltage using a first-order lag. If you select this option, the **Bandwidth**, **Maximum current slew rate**, and **Initial current** parameters appear on the **Dynamics** tab.

#### **Bandwidth — Bandwidth of the first-order lag**

2 MHz (default)

The bandwidth of the first-order lag used to model the dynamics of the relationship between output current and input voltage.

#### **Dependencies**

This parameter is visible only when the **Dynamics** parameter is set to **Finite bandwidth with slew rate limiting**.

#### **Maximum current slew rate — Maximum rate-of-charge of transconductance current**

2 A/ $\mu$ s (default)

The maximum rate-of-change of transconductance current when there is no feedback around the device. Note that datasheets sometimes quote slew rate as a maximum rate of change of voltage. In this case, the value depends on the particular test circuit. To get an accurate value for **Maximum current slew rate**, reproduce the test circuit in a Simscape Electrical model, and tune the parameter value to match the datasheet value. If the test circuit is open-loop, and the load resistance is quoted, you can obtain an approximate value for the **Maximum current slew rate** by dividing the voltage slew rate by the load resistance.

#### **Dependencies**

This parameter is visible only when the **Dynamics** parameter is set to **Finite bandwidth with slew rate limiting**.



**Initial current — Initial transconductance current**

0 A (default)

The initial transconductance current (note, not the initial output current). This is the transconductance current sinking to both the internal output resistance,  $R_{out}$ , and the output pin.

**Dependencies**

This parameter is visible only when the **Dynamics** parameter is set to Finite bandwidth with slew rate limiting.

**Limits****Minimum output voltage — Minimum output voltage**

-15 V (default)

The output voltage is limited to be greater than the value of this parameter.

**Maximum output voltage — Maximum output voltage**

15 V (default)

The output voltage is limited to be less than the value of this parameter.

**Additional output resistance at voltage swing limits — Additional output resistance at voltage swing limits**

1 Ohm (default)

To limit the output voltage swing, an additional output resistance is applied between output and the power rail when the output voltage exceeds the limit. The value of this resistance should be low compared to the output resistance and circuit load resistance.

**Minimum control current for simulation — Minimum control current for simulation**

0.001  $\mu$ A (default)

The control current measured at the control current pin C is limited to be greater than the value of this parameter. This prevents a potential divide-by-zero when calculating input and output resistance values based on the value of the control current.

## Version History

Introduced in R2011b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Op-Amp | Band-Limited Op-Amp | Finite-Gain Op-Amp

# Optocoupler

Behavioral model of optocoupler as LED, current sensor, and controlled current source

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

This block represents an optocoupler using a model that consists of the following components:

- An exponential light-emitting diode in series with a current sensor on the input side
- A controlled current source on the output side

The output-side current flows from the collector junction to the emitter junction. It has a value of  $CTR \cdot I_d$ , where  $CTR$  is the **Current transfer ratio** parameter value and  $I_d$  is the diode current.

Use the Optocoupler block to interface two electrical circuits without making a direct electrical connection. A common reason for doing this is that the two circuits work at very different voltage levels.

---

**Note** Each electrical circuit must have its own Electrical Reference block.

---

If the output circuit is a phototransistor, typical values for the **Current transfer ratio** parameter are 0.1 to 0.5. If the output stage consists of a Darlington pair, the parameter value can be much higher than this. The **Current transfer ratio** value also varies with the light-emitting diode current, but this effect is not modeled by the Photodiode block.

Some manufacturers provide a maximum data rate for optocouplers. In practice, the maximum data rate depends on the following factors:

- The capacitance of the photodiode and the type of the driving circuit
- The construction of the phototransistor and its associated capacitance

The Optocoupler block only lets you define the capacitance on the light-emitting diode. You can use the **Junction capacitance** parameter to add your own capacitance across the collector and emitter connections.

The Optocoupler block lets you model temperature dependence of the underlying diode. For details, see the Diode reference page.

## Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

- The output side is modeled as a controlled current source. As such, it only correctly approximates a bipolar transistor operating in its normal active region. To create a more detailed model, connect the Optocoupler output directly to the base of an NPN Bipolar Transistor block, and set the parameters to maintain a correct overall value for the current transfer ratio. If you need to connect optocouplers in series, use this approach to avoid the invalid topology of two current sources in series.
- The temperature dependence of the forward current transfer ratio is not modeled. Typically the temperature dependence of this parameter is much less than that of the optical diode I-V characteristic.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the diode positive terminal

#### - — Negative terminal

electrical

Electrical conserving port associated with the diode negative terminal

#### C — Transistor collector terminal

electrical

Electrical conserving port associated with the transistor collector terminal

**E – Transistor emitter terminal**

electrical

Electrical conserving port associated with the transistor emitter terminal

**H – Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

**Main****Current transfer ratio – Current transfer ratio**

0.2 (default)

The output current flowing from the transistor collector to emitter junctions is equal to the product of the current transfer ratio and the current flowing in the light-emitting diode.

**Diode parameterization – Model parameterization**

Use I-V curve data points (default) | Use parameters IS and N

Select one of the following methods for model parameterization:

- Use I-V curve data points – Specify measured data at two points on the diode I-V curve.
- Use parameters IS and N – Specify saturation current and emission coefficient.

**Currents [I1 I2] – Vector of current values at two points**

[.001, .015] A (default)

A vector of the current values at the two points on the diode I-V curve that the block uses to calculate IS and N.

**Dependencies**

This parameter is visible only when you select Use I-V curve data points for the **Diode parameterization** parameter.

**Voltages [V1 V2] – Vector of voltage values at two points**

[.9, 1.05] V (default)

A vector of the voltage values at the two points on the diode I-V curve that the block uses to calculate IS and N.

**Dependencies**

This parameter is visible only when you select Use I-V curve data points for the **Diode parameterization** parameter.

**Ohmic resistance, RS — Ohmic resistance**

0.1 Ohm (default)

The series diode connection resistance.

**Saturation current, IS — Saturation current**

1e-10 A (default)

The magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels.

**Dependencies**

This parameter is visible only when you select Use parameters IS and N for the **Diode parameterization** parameter.

**Measurement temperature — Measurement temperature**

25 °C (default)

The temperature at which IS or the I-V curve was measured. The default value is 25 °C.

**Emission coefficient, N — Diode emission coefficient**

2 (default)

The diode emission coefficient or ideality factor.

**Dependencies**

This parameter is visible only when you select Use parameters IS and N for the **Diode parameterization** parameter.

**Junction Capacitance****Capacitance — Modeling diode junction capacitance**

Fixed or zero junction capacitance (default) | Use C-V curve data points | Use parameters CJ0, VJ, M & FC

Select one of the following options for modeling the diode junction capacitance:

- Fixed or zero junction capacitance — Model the junction capacitance as a fixed value.
- Use C-V curve data points — Specify measured data at three points on the diode C-V curve.
- Use parameters CJ0, VJ, M & FC — Specify zero-bias junction capacitance, junction potential, grading coefficient, and forward-bias depletion capacitance coefficient.

**Junction capacitance — Diode junction capacitance**

5 pF (default)

Fixed junction capacitance value.

**Dependencies**

This parameter is visible only when you select **Fixed** or **zero junction capacitance** for the **Capacitance** parameter.

**Zero-bias junction capacitance, CJ0 — Zero-bias junction capacitance**

5 pF (default)

The value of the capacitance placed in parallel with the exponential diode term.

**Dependencies**

This parameter is visible only when you select **Use parameters CJ0, VJ, M & FC** for the **Capacitance** parameter.

**Junction potential, VJ — Junction potential**

1 V (default)

The junction potential.

**Dependencies**

This parameter is visible only when you select **Use parameters CJ0, VJ, M & FC** for the **Capacitance** parameter.

**Grading coefficient, M — Grading coefficient**

0.5 (default)

The coefficient that quantifies the grading of the junction.

**Dependencies**

This parameter is visible only when you select **Use parameters CJ0, VJ, M & FC** for the **Capacitance** parameter.

**Reverse bias voltages [VR1 VR2 VR3] — Vector of reverse bias voltages**

[.1, 10, 100] V (default)

A vector of the reverse bias voltage values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select **Use C-V curve data points** for the **Capacitance** parameter.

**Corresponding capacitances [C1 C2 C3] — Vector of corresponding capacitances**

[3.5, 1, .4] pF (default)

A vector of the capacitance values at the three points on the diode C-V curve that the block uses to calculate  $CJ0$ ,  $VJ$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select **Use C-V curve data points** for the **Capacitance** parameter.

**Capacitance coefficient, FC – Fitting capacitance coefficient**

0.5 (default)

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage.

**Dependencies**

This parameter is visible only when you select Use C-V curve data points or Use parameters CJ0, VJ, M & FC for the **Capacitance** parameter.

**Temperature Dependence****Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Use an I-V data point at second measurement temperature T2 | Specify saturation current at second measurement temperature T2 | Specify the energy gap EG

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Use an I-V data point at second measurement temperature T2 – If you select this option, you specify a second measurement temperature  $T_{m2}$ , and the current and voltage values at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify saturation current at second measurement temperature T2 – If you select this option, you specify a second measurement temperature  $T_{m2}$ , and saturation current value at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify the energy gap EG – Specify the energy gap value directly.

**Current I1 at second measurement temperature – Current I1 at second measurement temperature**

0.029 A (default)

Specify the diode current  $I1$  value when the voltage is  $V1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 for the **Parameterization** parameter.

**Voltage V1 at second measurement temperature – Voltage V1 at second measurement temperature**

1.05 V (default)

Specify the diode voltage  $V1$  value when the current is  $I1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 for the **Parameterization** parameter.

**Saturation current,  $I_S$ , at second measurement temperature – Saturation current,  $I_S$ , at second measurement temperature**

1.8e-8 A (default)

Specify the saturation current  $I_S$  value at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Second measurement temperature – Second measurement temperature**

125 °C (default)

Specify the value for the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 or Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Energy gap parameterization – Energy gap parameterization**

Use nominal value for silicon (EG=1.11eV) (default) | Use nominal value for 4H-SiC silicon carbide (EG=3.23eV) | Use nominal value for 6H-SiC silicon carbide (EG=3.00eV) | Use nominal value for germanium (EG=0.67eV) | Use nominal value for gallium arsenide (EG=1.43eV) | Use nominal value for selenium (EG=1.74eV) | Use nominal value for Schottky barrier diodes (EG=0.69eV) | Specify a custom value

Select a value for the energy gap from a list of predetermined options, or specify a custom value:

- Use nominal value for silicon (EG=1.11eV) — This is the default.
- Use nominal value for 4H-SiC silicon carbide (EG=3.23eV)
- Use nominal value for 6H-SiC silicon carbide (EG=3.00eV)
- Use nominal value for germanium (EG=0.67eV)
- Use nominal value for gallium arsenide (EG=1.43eV)
- Use nominal value for selenium (EG=1.74eV)
- Use nominal value for Schottky barrier diodes (EG=0.69eV)
- Specify a custom value — If you select this option, the **Energy gap, EG** parameter appears in the dialog box, to let you specify a custom value for  $EG$ .

**Dependencies**

This parameter is visible only when you select Specify the energy gap  $EG$  for the **Parameterization** parameter.

**Energy gap,  $EG$  – Energy gap**

1.11 eV (default)

Specify a custom value for the energy gap,  $EG$ .



**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Energy gap parameterization** parameter.

**Saturation current temperature exponent parameterization – Saturation current temperature exponent parameterization**

Use nominal value for pn-junction diode (XTI=3) (default) | Use nominal value for Schottky barrier diode (XTI=2) | Specify a custom value

Select one of the following options to specify the saturation current temperature exponent value:

- Use nominal value for pn-junction diode (XTI=3) – This is the default.
- Use nominal value for Schottky barrier diode (XTI=2)
- Specify a custom value – If you select this option, the **Saturation current temperature exponent, XTI** parameter appears in the dialog box, to let you specify a custom value for *XTI*.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Specify a custom value for the saturation current temperature exponent, *XTI*.

**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Saturation current temperature exponent parameterization** parameter.

**Device simulation temperature – Device simulation temperature**

25 °C (default)

Specify the value for the temperature  $T_s$ , at which the device is to be simulated.

**Version History**

Introduced in R2008a

**References**

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.
- [2] H. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition, Cambridge University Press, 1984.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Diode | NPN Bipolar Transistor | Controlled Current Source

# P-Channel JFET

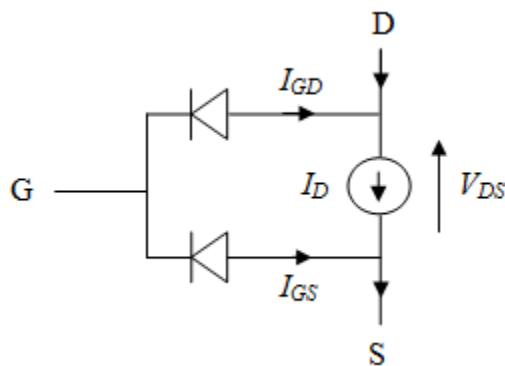
P-Channel junction field-effect transistor

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The P-Channel JFET block uses the Shichman and Hodges equations to represent a P-Channel JFET using a model with the following structure:



G is the transistor gate, D is the transistor drain and S is the transistor source. The drain current,  $I_D$ , depends on the region of operation and whether the transistor is operating in normal or inverse mode.

- In normal mode ( $-V_{DS} \geq 0$ ), the block provides the following relationship between the drain current  $I_D$  and the drain-source voltage  $V_{DS}$ .

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Off	$-V_{GS} \leq -V_{t0}$	$I_D = 0$
Linear	$0 < -V_{DS} < -V_{GS} + V_{t0}$	$I_D = \beta V_{DS}(2(-V_{GS} + V_{t0}) + V_{DS})(1 - \lambda V_{DS})$
Saturated	$0 < -V_{GS} + V_{t0} \leq -V_{DS}$	$I_D = -\beta (-V_{GS} + V_{t0})^2 (1 - \lambda V_{DS})$

- In inverse mode ( $-V_{DS} < 0$ ), the block provides the following relationship between the drain current  $I_D$  and the drain-source voltage  $V_{DS}$ .

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Off	$-V_{GD} \leq -V_{t0}$	$I_D = 0$

Region	Applicable Range of $V_{GS}$ and $V_{DS}$ Values	Corresponding $I_D$ Equation
Linear	$0 < V_{DS} < -V_{GD} + V_{t0}$	$I_D = \beta V_{DS}(2(-V_{GD} + V_{t0}) - V_{DS})(1 + \lambda V_{DS})$
Saturated	$0 < -V_{GD} + V_{t0} \leq V_{DS}$	$I_D = \beta (-V_{GD} + V_{t0})^2 (1 + \lambda V_{DS})$

In the preceding equations:

- $V_{GS}$  is the gate-source voltage.
- $V_{GD}$  is the gate-drain voltage.
- $V_{t0}$  is the threshold voltage. If you select Specify using equation parameters directly for the **Parameterization** parameter,  $V_{t0}$  is the **Threshold voltage** parameter value. Otherwise, the block calculates  $V_{t0}$  from the datasheet parameters you specify.
- $\beta$  is the transconductance parameter. If you select Specify using equation parameters directly for the **Parameterization** parameter,  $\beta$  is the **Transconductance parameter** parameter value. Otherwise, the block calculates  $\beta$  from the datasheet parameters you specify.
- $\lambda$  is the channel-length modulation parameter. If you select Specify using equation parameters directly for the **Parameterization** parameter,  $\lambda$  is the **Channel-length modulation** parameter value. Otherwise, the block calculates  $\lambda$  from the datasheet parameters you specify.

The currents in each of the diodes satisfy the exponential diode equation

$$I_{GD} = -IS \cdot \left( e^{-qV_{GD}/kT_{m1}} - 1 \right)$$

$$I_{GS} = -IS \cdot \left( e^{-qV_{GS}/kT_{m1}} - 1 \right)$$

where:

- $IS$  is the saturation current. If you select Specify using equation parameters directly for the **Parameterization** parameter,  $IS$  is the **Saturation current** parameter value. Otherwise, the block calculates  $IS$  from the datasheet parameters you specify.
- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $T_{m1}$  is the measurement temperature. The value comes from the **Measurement temperature** parameter.

The block models gate junction capacitance as a fixed gate-drain capacitance  $C_{GD}$  and a fixed gate-source capacitance  $C_{GS}$ . If you select Specify using equation parameters directly for the **Parameterization** parameter, you specify these values directly using the **Gate-drain junction capacitance** and **Gate-source junction capacitance** parameters. Otherwise, the block derives them from the **Input capacitance Ciss** and **Reverse transfer capacitance Crss** parameter values. The two parameterizations are related as follows:

- $C_{GD} = Crss$
- $C_{GS} = Ciss - Crss$

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. You can optionally include modeling the

dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the junction capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The measurement temperature value,  $T_{m1}$ , is replaced with the simulation temperature,  $T_s$ . The transconductance,  $\beta$ , and the threshold voltage,  $V_{t0}$ , become a function of temperature according to the following equations:

$$\beta_{T_s} = \beta_{T_{m1}} \left( \frac{T_s}{T_{m1}} \right)^{BEX}$$

$$V_{t0s} = V_{t01} + \alpha (T_s - T_{m1})$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $\beta_{T_{m1}}$  is JFET transconductance at the measurement temperature.
- $\beta_{T_s}$  is JFET transconductance at the simulation temperature. This is the transconductance value used in the JFET equations when temperature dependence is modeled.
- $V_{t01}$  is the threshold voltage at measurement temperature.
- $V_{t0s}$  is the threshold voltage at simulation temperature. This is the threshold voltage value used in the JFET equations when temperature dependence is modeled.
- $BEX$  is the mobility temperature exponent. A typical value of  $BEX$  is -1.5.
- $\alpha$  is the gate threshold voltage temperature coefficient,  $dV_{th}/dT$ .

For most JFETS, you can use the default value of -1.5 for  $BEX$ . Some datasheets quote the value for  $\alpha$ , but most typically they provide the temperature dependence for the saturated drain current,  $I_{dss}$ . Depending on the block parameterization method, you have two ways of specifying  $\alpha$ :

- If you parameterize the block from a datasheet, you have to provide  $I_{dss}$  at a second measurement temperature. The block then calculates the value for  $\alpha$  based on this data.
- If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

If you have more data comprising drain current as a function of gate-source voltage for fixed drain-source voltage plotted at more than one temperature, then you can also use Simulink Design Optimization software to help tune the values for  $\alpha$  and  $BEX$ .

In addition, the saturation current term,  $IS$ , in the gate-drain and gate-source current equations depends on temperature

$$IS_{T_s} = IS_{T_{m1}} \cdot (T_s/T_{m1})^{XTI} \cdot \exp\left(-\frac{EG}{kT_s}(1 - T_s/T_{m1})\right)$$

where:

- $IS_{T_{m1}}$  is the saturation current at the measurement temperature.
- $IS_{T_s}$  is the saturation current at the simulation temperature. This is the saturation current value used in the gate diode equations when temperature dependence is modeled..

- $EG$  is the energy gap.
- $k$  is the Boltzmann constant ( $1.3806503e-23$  J/K).
- $XTI$  is the saturation current temperature exponent.

Similar to  $\alpha$ , you have two ways of specifying  $EG$  and  $XTI$ :

- If you parameterize the block from a datasheet, you have to specify the gate reverse current,  $I_{gss}$ , at a second measurement temperature. The block then calculates the value for  $EG$  based on this data and assuming a p-n junction nominal value of 3 for  $XTI$ .
- If you parameterize by specifying equation parameters, you have to provide the values for  $EG$  and  $XTI$  directly. This option gives you most flexibility to match device behavior, for example, if you have a graph of  $I_{gss}$  as a function of temperature. With this data you can use Simulink Design Optimization software to help tune the values for  $EG$  and  $XTI$ .

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- This block does not allow you to specify initial conditions on the junction capacitances. If you select the **Start simulation from steady state** option in the Solver Configuration block, the block solves the initial voltages to be consistent with the calculated steady state. Otherwise, voltages are zero at the start of the simulation.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.
- The block does not account for temperature-dependent effects on the junction capacitances.
- When you specify  $I_{dss}$  at a second measurement temperature, it must be quoted for the same working point (that is, the same drain current and gate-source voltage) as for the  $I_{dss}$  value on the **Main** tab. Inconsistent values for  $I_{dss}$  at the higher temperature will result in unphysical values for  $\alpha$  and unrepresentative simulation results.

- You may need to tune the value of  $BEX$  to replicate the  $I_D$ - $V_{GS}$  relationship (if available) for a given device. The value of  $BEX$  affects whether the  $I_D$ - $V_{GS}$  curves for different temperatures cross each other, or not, for the ranges of  $I_D$  and  $V_{GS}$  considered.

## Ports

### Conserving

#### G – Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal

#### D – Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal

#### S – Source terminal

electrical

Electrical conserving port associated with the transistor source terminal

#### H – Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

#### Parameterization – Block parameterization

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- Specify from a datasheet – Provide parameters that the block converts to equations that describe the transistor. This is the default method.
- Specify using equation parameters directly – Provide equation parameters  $\beta$ ,  $IS$ ,  $V_{t0}$ , and  $\lambda$ .

#### Gate reverse current, $I_{gss}$ – Gate reverse current

5 nA (default)

The reverse current that flows in the diode when the drain and source are short-circuited and a large positive gate-source voltage is applied.

**Dependencies**

This parameter is visible only when you select `Specify` from a datasheet for the **Parameterization** parameter.

**Saturated drain current,  $I_{dss}$  — Saturated drain current**

-3 mA (default)

The current that flows when a large negative drain-source voltage is applied for a specified gate-source voltage. For a depletion-mode device, this gate-source voltage may be zero, in which case  $I_{dss}$  may be referred to as the zero-gate voltage drain current.

**Dependencies**

This parameter is visible only when you select `Specify` from a datasheet for the **Parameterization** parameter.

 **$I_{dss}$  measurement point,  $[V_{gs} V_{ds}]$  —  $I_{dss}$  measurement point**

[ 0 -15 ] V (default)

A vector of the values of  $V_{GS}$  and  $V_{DS}$  at which  $I_{dss}$  is measured. Normally  $V_{GS}$  is zero.  $V_{DS}$  should be less than zero.

**Dependencies**

This parameter is visible only when you select `Specify` from a datasheet for the **Parameterization** parameter.

**Small-signal parameters,  $[g_{fs} g_{os}]$  — Small-signal parameters**

[ 2.5e+3 75 ] uS (default)

A vector of the values of  $g_{fs}$  and  $g_{os}$ .  $g_{fs}$  is the forward transfer conductance, that is, the conductance for a fixed drain-source voltage.  $g_{os}$  is the output conductance, that is, the conductance for a fixed gate-source voltage.

**Dependencies**

This parameter is visible only when you select `Specify` from a datasheet for the **Parameterization** parameter.

**Small-signal measurement point  $[V_{gs} V_{ds}]$  — Small-signal measurement point**

[ 0 -15 ] V (default)

A vector of the values of  $V_{GS}$  and  $V_{DS}$  at which  $g_{fs}$  and  $g_{os}$  are measured.  $V_{DS}$  should be less than zero. For depletion-mode devices,  $V_{GS}$  is typically zero.

**Dependencies**

This parameter is visible only when you select `Specify` from a datasheet for the **Parameterization** parameter.

**Transconductance parameter — Transconductance parameter**

1e-04 A/V<sup>2</sup> (default)

The derivative of drain current with respect to gate voltage.



**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Saturation current — Saturation current**

1e-14 A (default)

The magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Threshold voltage — Threshold voltage**

2 V (default)

The gate-source voltage above which the transistor produces a nonzero drain current. For an enhancement device,  $V_{t0}$  should be negative. For a depletion mode device,  $V_{t0}$  should be positive.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Channel-length modulation — Channel-length modulation**

0 1/V (default)

The channel-length modulation.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Measurement temperature — Measurement temperature**

25 degC (default)

The temperature for which the datasheet parameters are quoted.

**Ohmic Resistance****Source ohmic resistance — Source ohmic resistance**

0.0001 Ohm (default) | nonnegative scalar

The transistor source resistance.

**Drain ohmic resistance — Drain ohmic resistance**

0.01 Ohm (default) | nonnegative scalar

The transistor drain resistance.

## Junction Capacitance

### Parameterization — Junction capacitance parameterization

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- Specify from a datasheet — Provide parameters that the block converts to junction capacitance values. This is the default method.
- Specify using equation parameters directly — Provide junction capacitance parameters directly.

### Input capacitance, Ciss — Input capacitance

5 pF (default)

The gate-source capacitance with the drain shorted to the source.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Reverse transfer capacitance, Crss — Reverse transfer capacitance

1 pF (default)

The drain-gate capacitance with the source connected to ground.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Gate-source junction capacitance — Gate-source junction capacitance

4 pF (default)

The value of the capacitance placed between the gate and the source.

#### Dependencies

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter in the **Junction Capacitance** tab.

### Gate-drain junction capacitance — Gate-drain junction capacitance

1 pF (default)

The value of the capacitance placed between the gate and the drain.

#### Dependencies

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter in the **Junction Capacitance** tab.

## Temperature Dependence

### Parameterization — Temperature dependence parameterization

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- **None** – Simulate at parameter measurement temperature – Temperature dependence is not modeled. This is the default method.
- **Model temperature dependence** – Model temperature-dependent effects. You also have to provide a set of additional parameters depending on the block parameterization method. If you parameterize the block from a datasheet, you have to provide values for  $I_{gss}$  and  $I_{dss}$  at second measurement temperature. If you parameterize by directly specifying equation parameters, you have to provide the values for  $EG$ ,  $XTI$ , and the gate threshold voltage temperature coefficient,  $dV_{t0}/dT$ . Regardless of the block parameterization method, you also have to provide values for  $BEX$  and for the simulation temperature,  $T_s$ .

**Gate reverse current,  $I_{gss}$ , at second measurement temperature – Gate reverse current,  $I_{gss}$ , at second measurement temperature**

950 nA (default)

The value of the gate reverse current,  $I_{gss}$ , at the second measurement temperature and when the  $I_{gss}$  measurement point is the same as defined by the **Gate reverse current,  $I_{gss}$**  parameter on the **Main** tab.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Saturated drain current,  $I_{dss}$ , at second measurement temperature – Saturated drain current,  $I_{dss}$ , at second measurement temperature**

-2.3 mA (default)

The value of the saturated drain current,  $I_{dss}$ , at the second measurement temperature, and when the  $I_{dss}$  measurement point is the same as defined by the  **$I_{dss}$  measurement point, [V<sub>gs</sub> V<sub>ds</sub>]** parameter on the **Main** tab.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Second temperature  $T_{m2}$  at which **Gate reverse current,  $I_{gss}$ , at second measurement temperature** and **Saturated drain current,  $I_{dss}$ , at second measurement temperature** are measured.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Energy gap,  $EG$  – Energy gap**

1.11 eV (default)

Energy gap value.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Saturation current temperature coefficient value.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Gate threshold voltage temperature coefficient, dVt0/dT – Gate threshold voltage temperature coefficient**

1 mV/K (default)

The rate of change of gate threshold voltage with temperature.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Mobility temperature exponent, BEX – Mobility temperature exponent**

-1.5 (default)

Mobility temperature coefficient value. You can use the default value for most JFETs. See the “Assumptions and Limitations” on page 1-1216 section for additional considerations.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

## Version History

**Introduced in R2008a****Thermal network simlog paths**

*Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

**References**

- [1] H. Shichman and D. A. Hodges, *Modeling and simulation of insulated-gate field-effect transistor switching circuits*. IEEE J. Solid State Circuits, SC-3, 1968.

[2] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993. Chapter 2.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

N-Channel JFET

## P-Channel LDMOS FET

P-Channel laterally diffused metal oxide semiconductor or vertically diffused metal oxide semiconductor transistors suitable for high voltage

**Library:** Simscape / Electrical / Semiconductors & Converters



### Description

The P-Channel LDMOS FET block lets you model LDMOS (or VDMOS) transistors suitable for high voltage. The model is based on surface potential and includes 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 option with exposed thermal port (see “Thermal Port” on page 1-1225), dynamic self-heating

For information on physical background and defining equations, see the N-Channel LDMOS FET block reference page. Both the p-type and n-type versions of the LDMOS model use the same underlying code with appropriate voltage transformations, to account for the different device types.

The charge model is similar to that of the surface-potential-based MOSFET model, with additional expressions to account for the charge in the drift region. The block uses the derived equations as described in [1], which include both inversion and accumulation in the drift region.

### Modeling Body Diode

The block models the body diode as an ideal, exponential diode with both junction and diffusion capacitances:

$$I_{dio} = I_s \left[ \exp\left(-\frac{V_{BD}}{n\phi_T}\right) - 1 \right]$$

$$C_j = \frac{C_{j0}}{\sqrt{1 + \frac{V_{BD}}{V_{bi}}}}$$

$$C_{diff} = \frac{\tau I_s}{n\phi_T} \exp\left(-\frac{V_{BD}}{n\phi_T}\right)$$

where:

- $I_{dio}$  is the current through the diode.
- $I_s$  is the reverse saturation current.
- $V_{BD}$  is the body-drain voltage.
- $n$  is the ideality factor.
- $\phi_T$  is the thermal voltage.
- $C_j$  is the junction capacitance of the diode.
- $C_{j0}$  is the zero-bias junction capacitance.
- $V_{bi}$  is the built-in voltage.
- $C_{diff}$  is the diffusion capacitance of the diode.
- $\tau$  is the transit time.

The capacitances are defined through an explicit calculation of charges, which are then differentiated to give the capacitive expressions above. The block computes the capacitive diode currents as time derivatives of the relevant charges, similar to the computation in the surface-potential-based MOSFET model.

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. To model the dependence on temperature during simulation, select **Model temperature dependence** for the **Parameterization** parameter on the **Temperature Dependence** tab.

The model includes temperature effects on the capacitance characteristics, as well as modeling the dependence of the transistor static behavior on temperature during simulation.

The **Measurement temperature** parameter on the **Main** tab specifies temperature  $T_{m1}$  at which the other device parameters have been extracted. The **Temperature Dependence** tab provides the simulation temperature,  $T_s$ , and the temperature-scaling coefficients for the other device parameters. For more information, see “Temperature Dependence” on page 1-1230.

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

If you expose the thermal port, the block includes dynamic self-heating. This lets you simulate the effect of self-heating on the electrical characteristics of the device.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **G – Gate terminal**

electrical

Electrical conserving port associated with the transistor gate terminal

#### **D – Drain terminal**

electrical

Electrical conserving port associated with the transistor drain terminal

#### **S – Source terminal**

electrical

Electrical conserving port associated with the transistor source terminal

#### **H – Thermal port**

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### **Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

#### **Gain, [channel drift\_region] – Gain**

[11.6, .01] A/V<sup>2</sup> (default)

The gain,  $\beta$ , of the MOSFET regions. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. This parameter primarily



defines the linear region of operation on an  $I_D$ - $V_{DS}$  characteristic. The values of both elements must be greater than 0.

### Flatband voltage, [channel drift\_region] – Flatband voltage

[-1.05, -0.1] V (default)

The flatband voltage,  $V_{FB}$ , defines the gate bias that must be applied in order to achieve the flatband condition at the surface of the silicon. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. You can also use this parameter to arbitrarily shift the threshold voltage due to material work function differences, and to trapped interface or oxide charges. In practice, however, it is usually recommended to modify the threshold voltage by using the **Body factor, [channel drift\_region]** and **Surface potential at strong inversion, [channel drift\_region]** parameters first, and only use this parameter for fine-tuning.

The threshold voltage for the channel region, for a short-circuited source-bulk connection, is approximately

$$-V_T = V_{FB} + 2\phi_B + 2\phi_T + \gamma\sqrt{2\phi_B + 2\phi_T}$$

where  $2\phi_B$  is the surface potential at strong inversion and  $\gamma$  is the body factor, both at the channel region.

### Body factor, [channel drift\_region] – Body factor

[3.4, 2.5]  $V^{1/2}$  (default)

Body factor,  $\gamma$ , in the surface-potential equation. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.

For the channel region, the body factor is

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_D}}{C_{ox}}$$

See the N-Channel MOSFET block reference page for details on this equation. The drift region equation is similar, except that  $N_D$  is replaced by the doping density,  $N_A$ . The channel-region parameter value primarily impacts the threshold voltage. For the drift region, this parameter primarily affects the charge model, and also has a minor effect on the pinch-off behavior of the bulk current through the drift region.

### Surface potential at strong inversion, [channel drift\_region] – Surface potential at strong inversion

[0.95, 0.95] V (default)

The  $2\phi_B$  term in the surface-potential equation. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.

The channel-region parameter value also primarily impacts the threshold voltage. For the drift region, this parameter affects the charge model only.

### Velocity saturation factor, [channel drift\_region] – Velocity saturation factor

[0.0, 0.1] 1/V (default)

Velocity saturation,  $\theta_{sat}$ , in the drain-current equation. Use this parameter in cases where a good fit to linear operation leads to a saturation current that is too high. By increasing this parameter value, you

reduce the saturation current. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The default value is  $[0.0, 0.1]$   $1/V$ , which means that velocity saturation in the channel region is off by default.

#### **Drift region surface scattering factor — Drift region surface scattering factor**

0  $1/V$  (default)

Surface scattering factor,  $\theta_{surf}$ , in the drain-current equation. This parameter applies to the drift region only and accounts for scattering in the accumulation layer due to the vertical electric field.

#### **Channel-length modulation factor — Channel-length modulation factor**

0 (default)

The factor,  $\alpha$ , multiplying the logarithmic term in the  $G_{AL}$  equation. See the N-Channel MOSFET block reference page for details on this equation. This parameter describes the onset of channel-length modulation. For device characteristics that exhibit a positive conductance in saturation, increase the parameter value to fit this behavior. This parameter applies to the channel region only. The default value is 0, which means that channel-length modulation is off by default.

#### **Channel-length modulation voltage — Channel-length modulation voltage**

5e-2 V (default)

The voltage  $V_p$  in the  $G_{AL}$  equation. See the N-Channel MOSFET block reference page for details on this equation. This parameter controls the drain-voltage at which channel-length modulation starts to become active. This parameter applies to the channel region only.

#### **Linear-to-saturation transition coefficient — Linear-to-saturation transition coefficient**

8 (default)

This parameter controls how smoothly the MOSFET transitions from linear into saturation, particularly when velocity saturation is enabled. This parameter can usually be left at its default value, but you can use it to fine-tune the knee of the  $I_D$ - $V_{DS}$  characteristic. This parameter applies both to the channel and drift regions. The expected range for this parameter value is between 2 and 8.

#### **Measurement temperature — Measurement temperature**

25 degC (default)

Temperature  $T_{m1}$  at which the block parameters are measured. If the **Device simulation temperature** parameter on the **Temperature Dependence** tab differs from this value, then device parameters will be scaled from their defined values according to the simulation and reference temperatures. For more information, see “Temperature Dependence” on page 1-1230.

#### **Ohmic Resistance**

##### **Source ohmic resistance — Source ohmic resistance**

1e-4 Ohm (default) | nonnegative scalar

The transistor source resistance, that is, the series resistance associated with the source contact. The value must be greater than or equal to 0.

##### **Drain ohmic resistance — Drain ohmic resistance**

0.07 Ohm (default) | nonnegative scalar

The transistor drain resistance, that is, the series resistance associated with the drain contact and with the LOCOS part of the drift region, which is not heavily impacted by the applied gate voltage. The value must be greater than or equal to  $\theta$ .

**Gate ohmic resistance — Gate ohmic resistance**

8.4 Ohm (default) | nonnegative scalar

The transistor gate resistance, that is, the series resistance associated with the gate contact. The value must be greater than or equal to  $\theta$ .

**Drift region low-bias resistance for gated region — Drift region low-bias resistance for gated region**

0.1 Ohm (default) | nonnegative scalar

Resistance  $R_D$  in the drain-current equation. It represents the resistance of the bulk part of the drift region in the absence of depletion from the top and bottom interfaces. The value must be greater than or equal to  $\theta$ .

**Drift region depletion layer thickness factor — Drift region depletion layer thickness factor**

0.2 (default)

Parameter  $\lambda_D$  in the drain-current equation. It is the ratio of vertical depths  $y_1$  and  $y_2$  at zero bias, where  $y_1$  represents the space-charge region and  $y_2$  represents the undepleted part of the drift region. See the N-Channel LDMOS FET block reference page for an illustration.

**Capacitances**

**Oxide capacitance, [channel drift\_region] — Oxide capacitance**

[1600.0, 1000.0] pF (default)

The parallel plate gate-channel and gate-drift-region capacitance. The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region.

**Gate-source overlap capacitance — Gate-source overlap capacitance**

15 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the source well.

**Gate-drain overlap capacitance — Gate-drain overlap capacitance**

15 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the drain well.

**Body Diode**

**Reverse saturation current — Reverse saturation current**

1e-13 A (default)

The current designated by the  $I_s$  symbol in the body-diode equations.

**Built-in voltage — Built-in voltage**

0.6 V (default)

The built-in voltage of the diode, designated by the  $V_{bi}$  symbol in the body-diode equations.

**Ideality factor – Ideality factor**

1 (default)

The factor designated by the  $n$  symbol in the body-diode equations.

**Zero-bias junction capacitance – Zero-bias junction capacitance**

1800 pF (default)

The capacitance between the drain and bulk contacts at zero-bias due to the body diode alone. It is designated by the  $C_{j0}$  symbol in the body-diode equations.

**Transit time – Transit time**

50 ns (default)

The time designated by the  $\tau$  symbol in the body-diode equations.

**Temperature Dependence****Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled. This is the default method.
- Model temperature dependence – Model temperature-dependent effects. Provide a value for the device simulation temperature,  $T_s$ , and the temperature-scaling coefficients for other block parameters.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

**Gain temperature exponent, [channel drift\_region] – Gain temperature exponent**

[1.3, 1.3] (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. Both in the channel and the drift region, the MOSFET gain,  $\beta$ , is assumed to scale exponentially with temperature,  $\beta = \beta_{m1} (T_{m1}/T_s)^{\eta_\beta}$ .  $\beta_{m1}$  is the value of the channel or drift region gain, as specified by the **Gain, [channel drift\_region]** parameter from the **Main** tab.  $\eta_\beta$  is the corresponding element of the **Gain temperature exponent, [channel drift\_region]** parameter.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

### Flatband voltage temperature coefficient, [channel drift\_region] – Flatband voltage temperature coefficient

[0.0005, 0.0005] V/K (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The flatband voltage,  $V_{FB}$ , is assumed to scale linearly with temperature,  $V_{FB} = V_{FBm1} + (T_s - T_{m1})S_{TV_{FB}}$ .  $V_{FBm1}$  is the value of the channel or drift region flatband voltage, as specified by the **Flatband voltage, [channel drift\_region]** parameter from the **Main** tab.  $S_{TV_{FB}}$  is the corresponding element of the **Flatband voltage temperature coefficient, [channel drift\_region]** parameter.

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

### Surface potential at strong inversion temperature coefficient – Surface potential at strong inversion temperature coefficient

-8.5e-4 V/K (default)

The surface potential at strong inversion,  $2\phi_B$ , is assumed to scale linearly with temperature,  $2\phi_B = 2\phi_{Bm1} + (T_s - T_{m1})S_{T\phi_B}$ .  $2\phi_{Bm1}$  is the value of the **Surface potential at strong inversion** parameter from the **Main** tab and  $S_{T\phi_B}$  is the **Surface potential at strong inversion temperature coefficient**.

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

### Velocity saturation temperature exponent, [channel drift\_region] – Velocity saturation temperature exponent

[1.04, 1.04] (default)

The parameter value is a two-element vector, with the first element corresponding to the channel, and the second — to the drift region. The velocity saturation,  $\theta_{sat}$ , is assumed to scale exponentially with temperature,  $\theta_{sat} = \theta_{sat,m1} (T_{m1}/T_s)^{\eta_\theta}$ .  $\theta_{sat,m1}$  is the value of the channel or drift region velocity saturation factor, as specified by the **Velocity saturation factor, [channel drift\_region]** parameter from the **Main** tab.  $\eta_\theta$  is the corresponding element of the **Velocity saturation temperature exponent, [channel drift\_region]** parameter.

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

### Ohmic resistance temperature exponent – Ohmic resistance temperature exponent

0.95 (default)

The series resistances are assumed to correspond to semiconductor resistances. Therefore, they decrease exponentially with increasing temperature.  $R_i = R_{i,m1} (T_{m1}/T_s)^{\eta_R}$ , where  $i$  is S, D, or G, for the source, drain, or gate series resistance, respectively.  $R_{i,m1}$  is the value of the corresponding series resistance parameter from the **Ohmic Resistance** tab and  $\eta_R$  is the **Ohmic resistance temperature exponent**.

**Dependencies**

This parameter is visible only when you select `Model temperature dependence` for the **Parameterization** parameter.

**Drift region low-bias resistance temperature exponent for gated portion – Drift region low-bias resistance temperature exponent for gated portion**

0.95 (default)

Resistance  $R_D$ , the low-bias resistance of the bulk part of the drift region, scales similarly to the other series resistances. A separate value of the temperature exponent for this resistance provides an extra degree of freedom.

**Dependencies**

This parameter is visible only when you select `Model temperature dependence` for the **Parameterization** parameter.

**Body diode reverse saturation current temperature exponent – Body diode reverse saturation current temperature exponent**

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$

$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** tab,  $k_B$  is Boltzmann's constant (8.617x10-5eV/K), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

**Dependencies**

This parameter is visible only when you select `Model temperature dependence` for the **Parameterization** parameter.

## Version History

**Introduced in R2016b****Thermal network simlog paths**

*Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

## References

- [1] Aarts, A., N. D'Halleweyn, and R. Van Langevelde. "A Surface-Potential-Based High-Voltage Compact LDMOS Transistor Model." *IEEE Transactions on Electron Devices*. 52(5):999 - 1007. June 2005.
- [2] Van Langevelde, R., A. J. Scholten, and D. B. M. Klaassen. "Physical Background of MOS Model 11. Level 1101." *Nat.Lab. Unclassified Report 2003/00239*. April 2003.
- [3] Oh, S-Y., D. E. Ward, and R. W. Dutton. "Transient analysis of MOS transistors." *IEEE J. Solid State Circuits*. SC-15, pp. 636-643, 1980.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

N-Channel LDMOS FET | N-Channel MOSFET

## P-Channel MOSFET

P-Channel metal oxide semiconductor field-effect transistor using either Shichman-Hodges equation or surface-potential-based model

**Library:** Simscape / Electrical / Semiconductors & Converters



### Description

The P-Channel MOSFET block provides two main modeling options:

- Based on threshold voltage — Uses 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. For details, see “Threshold-Based Model” on page 1-1235.

This modeling option provides four ways of parameterizing an N-Channel MOSFET:

- By specifying parameters from a datasheet.
- By specifying equation parameters directly.
- By a 2-D lookup table approximation to the I-V (current-voltage) curve. For details, see “Representation by 2-D Lookup Table” on page 1-1239.
- By a 3-D lookup table approximation to the I-V (current-voltage) curve that includes temperature data. For details, see “Representation by 3-D Lookup Table” on page 1-1241.
- Based on surface potential — Uses the surface-potential equation to represent the device. This modeling approach provides a greater level of model fidelity than the simple square-law (threshold-voltage-based) models can provide. The trade-off is that there are more parameters that require extraction. For details, see “Surface-Potential-Based Model” on page 1-1241.

Together with the thermal port variants (see “Thermal Port” on page 1-1247), the block therefore provides you with four choices. To select the desired modeling option, set the **Modeling option** parameter to either:

- **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.
- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port.
- **Surface-potential-based** — Model based on surface potential. This model does not simulate thermal effects.
- **Surface-potential-based with thermal** — Thermal modeling option of the model based on surface potential.



### Threshold-Based Model

The threshold-based modeling option of the block uses the Shichman and Hodges equations [1] for an insulated-gate field-effect transistor to represent a P-Channel MOSFET.

The drain-source current,  $I_{DS}$ , depends on the region of operation:

- In the off region ( $-V_{GS} < -V_{th}$ ) the drain-source current is:

$$I_{DS} = 0$$

- In the linear region ( $0 < -V_{DS} < -V_{GS} + V_{th}$ ) the drain-source current is:

$$I_{DS} = -K((V_{GS} - V_{th})V_{DS} - V_{DS}^2/2)(1 + \lambda |V_{DS}|)$$

- In the saturated region ( $0 < -V_{GS} + V_{th} < -V_{DS}$ ) the drain-source current is:

$$I_{DS} = -(K/2)(V_{GS} - V_{th})^2(1 + \lambda |V_{DS}|)$$

In the preceding equations:

- $K$  is the transistor gain.
- $V_{DS}$  is the negative drain-source voltage.
- $V_{GS}$  is the gate-source voltage.
- $V_{th}$  is the threshold voltage. For the four terminal parameterization,  $V_{th}$  is obtained using these equations:

$V_{BS}$ Range	$V_{th}$ Equation
$V_{BS} \leq 0$	$V_{th} = V_{T0} + \gamma(\sqrt{2\phi_B}) - \gamma(\sqrt{2\phi_B + V_{BS}})$
$0 < V_{BS} \leq 4\phi_B$	$V_{th} = V_{T0} - \frac{\gamma V_{BS}}{\sqrt{2\phi_B}}$
$V_{BS} > 4\phi_B$	$V_{th} = V_{T0} + \gamma(\sqrt{2\phi_B})$

- $\lambda$  is the channel modulation.

### Charge Model for Threshold-Based Modeling Option

The block models capacitances either by fixed capacitance values, or by tabulated values as a function of the drain-source voltage. In either case, you can either directly specify the gate-source and gate-drain capacitance values, or let the block derive them from the input and reverse transfer capacitance values. Therefore, the **Parameterization** options for charge model on the **Capacitance** setting are:

- Specify fixed input, reverse transfer and output capacitance — Provide fixed parameter values from datasheet and let the block convert the input and reverse transfer capacitance values to capacitance values, as described below. This is the default method.
- Specify fixed gate-source, gate-drain and drain-source capacitance — Provide fixed values for capacitance parameters directly.
- Specify tabulated input, reverse transfer and output capacitance — Provide tabulated capacitance and drain-source voltage values based on datasheet plots. The block converts the input and reverse transfer capacitance values to capacitance values, as described below.

- Specify tabulated gate-source, gate-drain and drain-source capacitance — Provide tabulated values for capacitances and drain-source voltage.

Use one of the tabulated capacitance options (Specify tabulated input, reverse transfer and output capacitance or Specify tabulated gate-source, gate-drain and drain-source capacitance) when the datasheet provides a plot of capacitances as a function of drain-source voltage. Using tabulated capacitance values gives more accurate dynamic characteristics and avoids the need for iterative tuning of parameters to fit the dynamics.

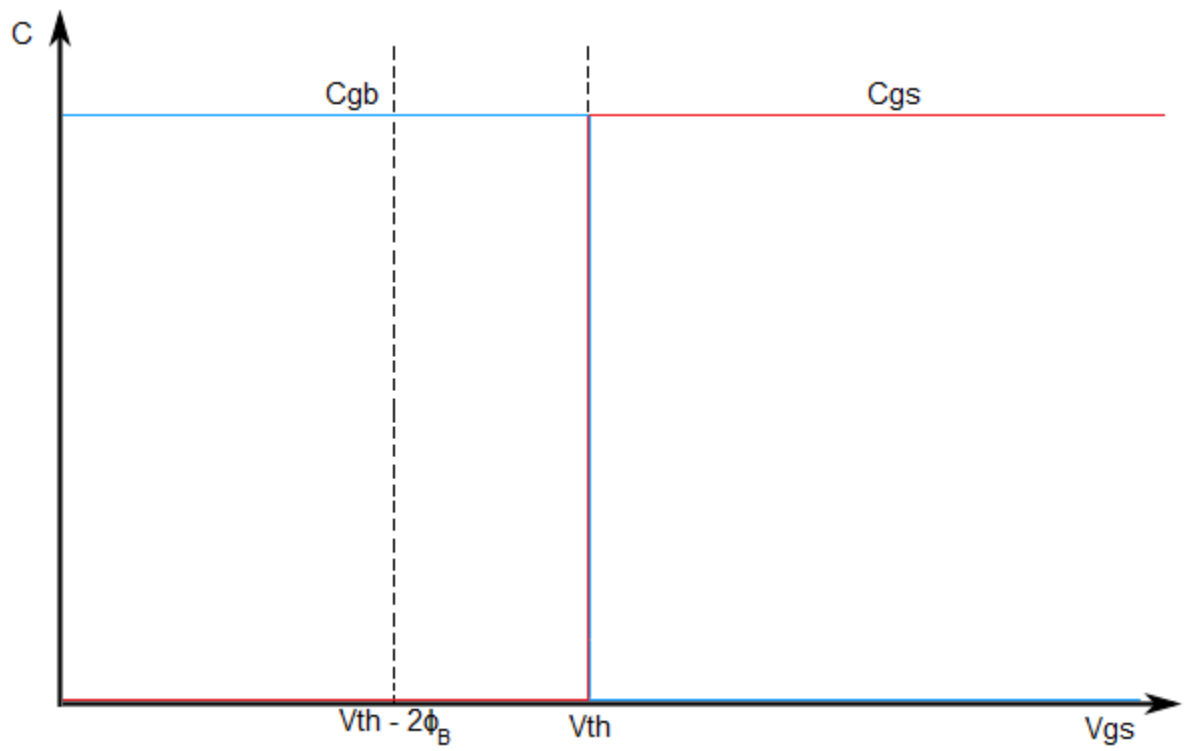
If you use the Specify fixed gate-source, gate-drain and drain-source capacitance or Specify tabulated gate-source, gate-drain and drain-source capacitance option, the **Capacitance** setting lets you specify the **Gate-drain capacitance, Cgd**, **Gate-source capacitance, Cgs**, and **Drain-source capacitance, Cds** parameter values (fixed or tabulated) directly. Otherwise, the block derives them from the **Input capacitance, Ciss**, **Reverse transfer capacitance, Crss**, and **Output capacitance, Coss** parameter values. These two parameterization methods are related as follows:

- $C_{GD} = Crss$
- $C_{GS} = Ciss - Crss$
- $C_{DS} = Coss - Crss$

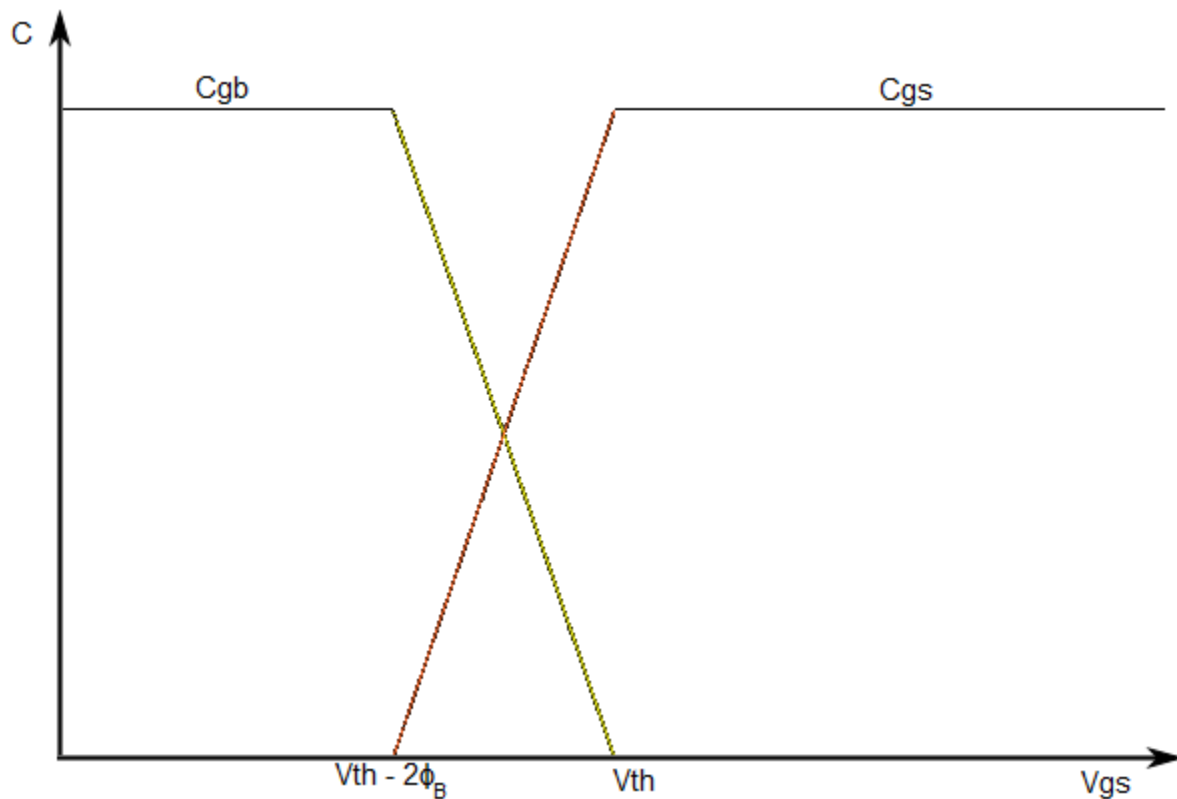
For the four terminals parameterization, the **Input capacitance, Ciss**, **Reverse transfer capacitance, Crss**, and **Output capacitance, Coss** are obtained using these equations:

- $C_{GD} = Crss$
- $C_{GS} + C_{GB} = Ciss - Crss$
- $C_{DB} = Coss - Crss$

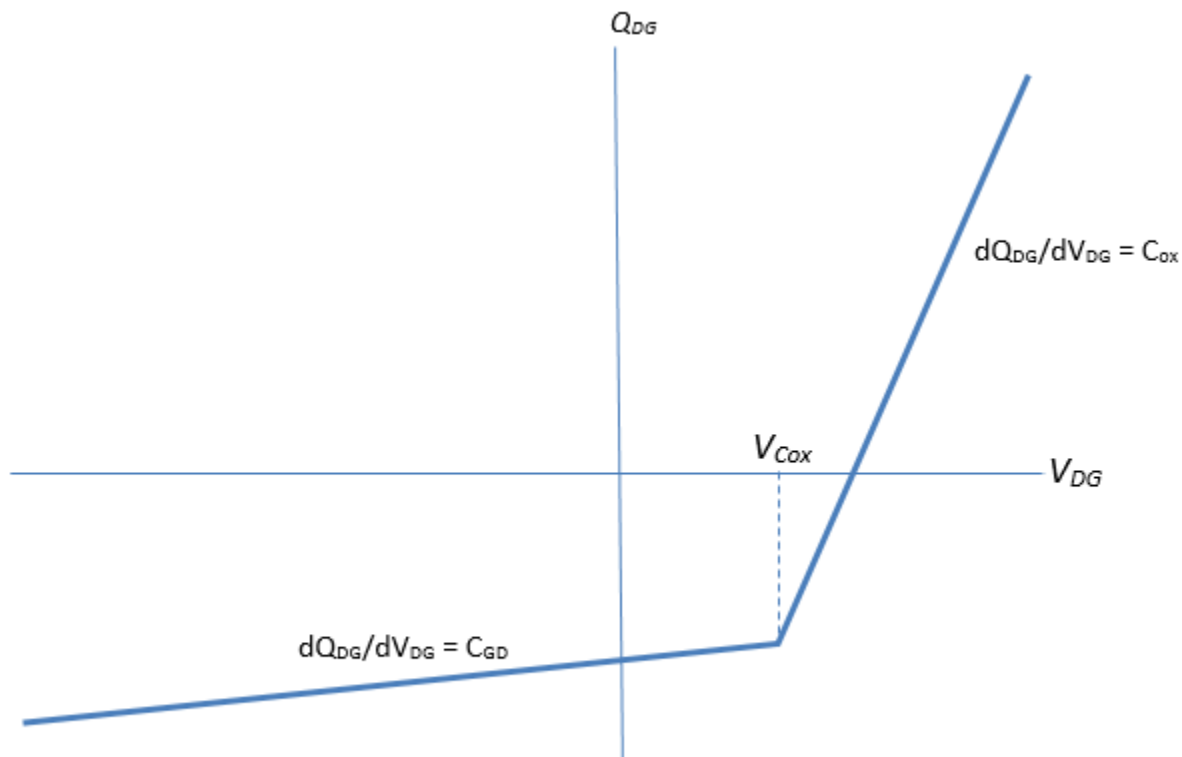
A simplified Meyer's capacitance model is used to describe the gate-source capacitance,  $C_{GS}$ , the gate-bulk capacitance,  $C_{GB}$ , and the gate-drain capacitance,  $C_{GD}$ . These figures show how the gate-bulk and gate-source capacitances change instantaneously, while the



Gate-bulk and gate-source capacitance change instantaneously.



The two fixed capacitance options (Specify fixed input, reverse transfer and output capacitance or Specify fixed gate-source, gate-drain and drain-source capacitance) let you model gate capacitance as a fixed gate-source capacitance  $C_{GS}$  and either a fixed or a nonlinear gate-drain capacitance  $C_{GD}$ . If you select the Gate-drain charge function is nonlinear option for the **Gate-drain charge-voltage linearity** parameter, then the gate-drain charge relationship is defined by the piecewise-linear function shown in the following figure.



**Nonlinear gate-drain charge function**

For instructions on how to map a time response to device capacitance values, see the N-Channel IGBT block reference page. However, this mapping is only approximate because the Miller voltage typically varies more from the threshold voltage than in the case for the IGBT.

---

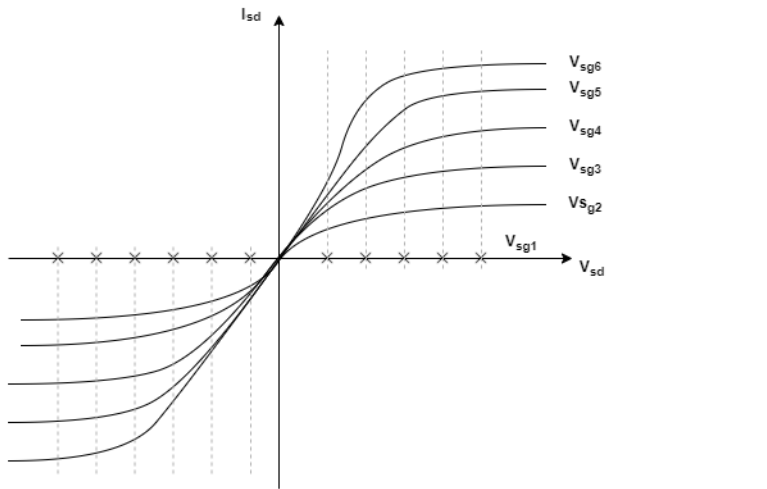
**Note** Because this block implementation includes a charge model, you must model the impedance of the circuit driving the gate to obtain representative turn-on and turn-off dynamics. Therefore, if you are simplifying the gate drive circuit by representing it as a controlled voltage source, you must include a suitable series resistor between the voltage source and the gate.

---

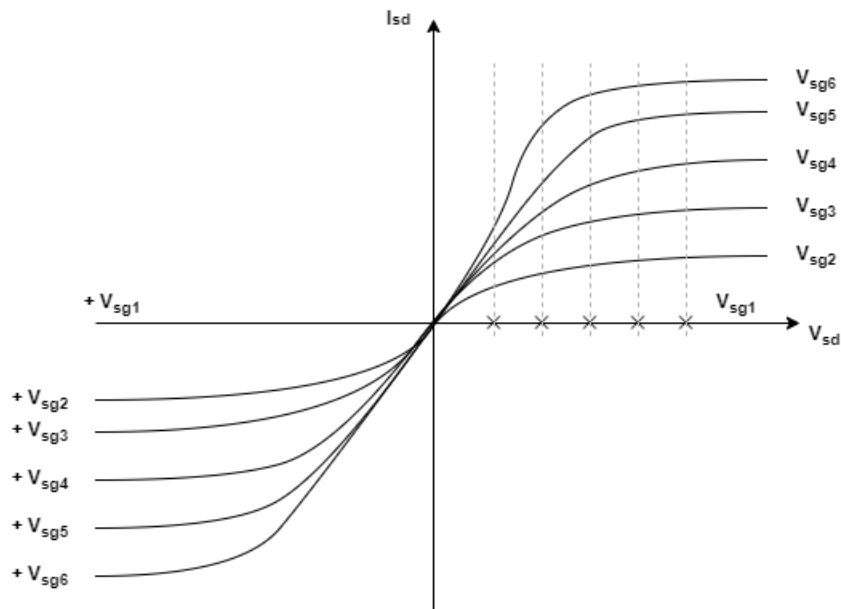
### Representation by 2-D Lookup Table

For the lookup table representation of the detailed block modeling option, you provide tabulated values for source-drain currents as a function of source-gate voltage and source-gate voltage. The main advantage of using this option is simulation speed. It also lets you parameterize the device from either measured data or from data obtained from another simulation environment.

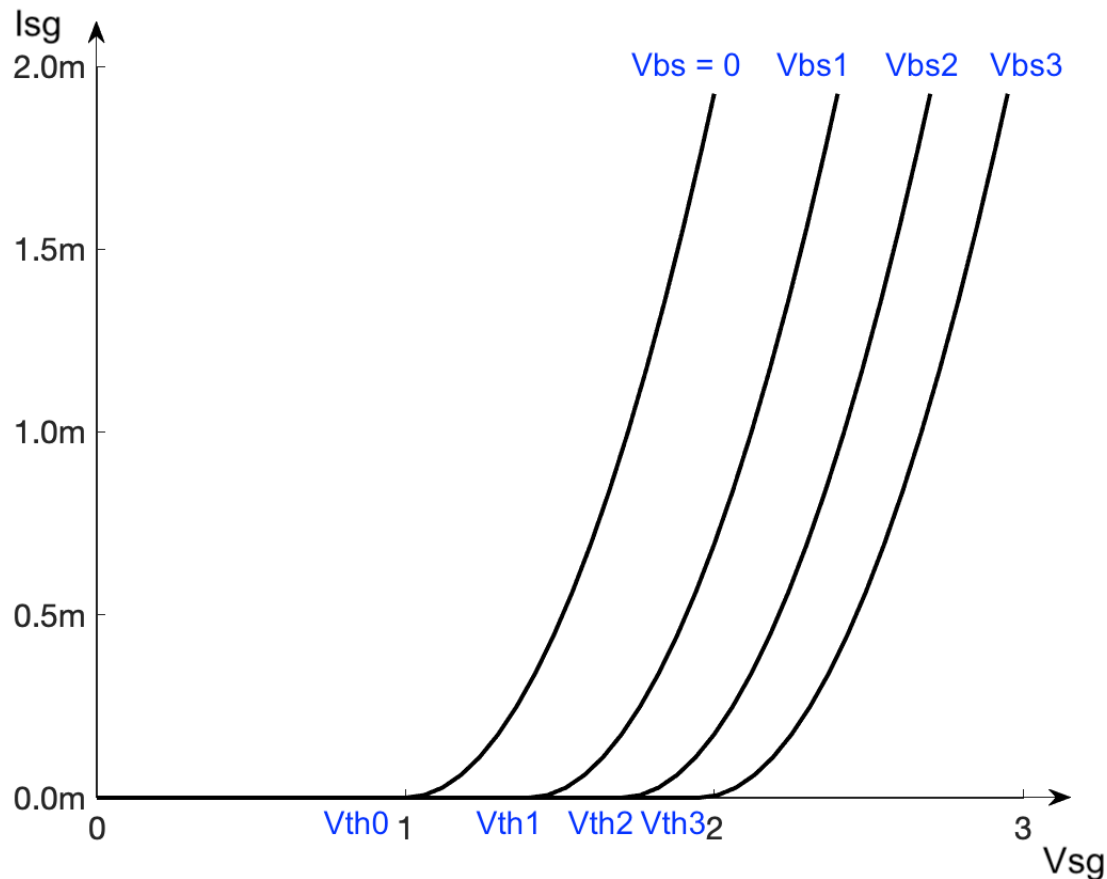
This figure shows the implementation of the 2-D lookup table option when you set **Isd-Vsd parameterization** to Provide negative and positive Vsd data:



This figure shows the implementation of the 2-D lookup table option when you set **I<sub>sd</sub>-V<sub>sd</sub> parameterization** to Provide positive V<sub>sd</sub> data only:



For the four terminal MOSFET, the surface potential and body factor values are calculated based on the nearest threshold voltage as shown in this picture:



### Representation by 3-D Lookup Table

For the temperature-dependent lookup table representation of the detailed block modeling option, you provide tabulated values for source-drain currents as a function of source-gate voltage, source-gate voltage, and temperature.

### Surface-Potential-Based Model

The surface-potential-based modeling option of the block provides a greater level of model fidelity than the simple square-law (threshold-voltage-based) model. The surface-potential-based block modeling option includes the following effects:

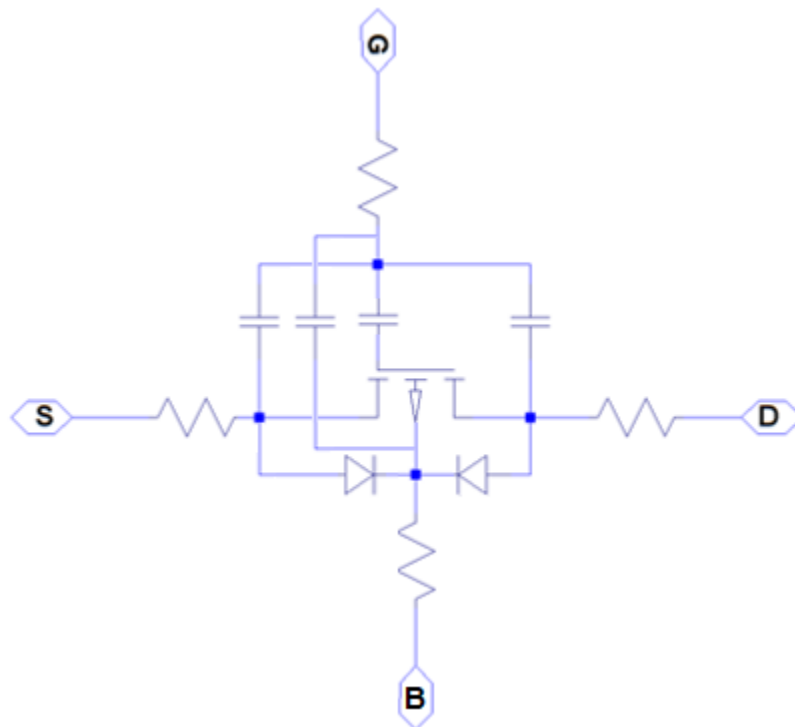
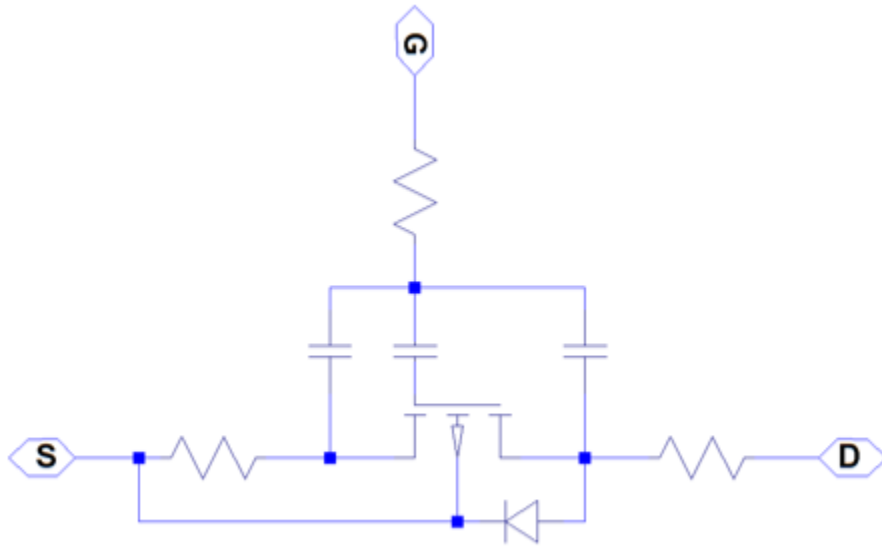
- Fully nonlinear capacitance model (including the nonlinear Miller capacitance)
- Charge conservation inside the model, so you can use the model for charge sensitive simulations
- Velocity saturation and channel-length modulation
- The intrinsic body diode
- Reverse recovery in the body diode model
- Temperature scaling of physical parameters
- For the thermal modeling option, dynamic self-heating (that is, you can simulate the effect of self-heating on the electrical characteristics of the device)

This model is a minimal version of the world-standard PSP model (see <https://briefs.techconnect.org/papers/introduction-to-psp-mosfet-model/>), including only certain effects from the PSP model to strike

a balance between model fidelity and complexity. For details of the physical background to the phenomena included in this model, see [2].

The surface-potential equation is derived similar to the way described on the N-Channel MOSFET block reference page, with all voltages, charges, and currents multiplied by -1.

The overall model consists of an intrinsic MOSFET defined by the surface-potential formulation, a body diode, series resistances, and fixed overlap capacitances, as shown in the schematic.





## Modeling Body Diode

The block models the body diode either as an ideal, exponential diode or as a tabulated diode.

### Exponential Diode

The block models the body diode either as an ideal, exponential diode or as a tabulated diode.

When you set **Model body diode** to `Exponential`, the junction and diffusion capacitances are:

$$I_{dio} = I_s \left[ \exp\left(-\frac{V_{BD}}{n\phi_T}\right) - 1 \right]$$

$$C_j = \frac{C_{j0}}{\sqrt{1 + \frac{V_{BD}}{V_{bi}}}}$$

$$C_{diff} = \frac{\tau I_s}{n\phi_T} \exp\left(-\frac{V_{BD}}{n\phi_T}\right)$$

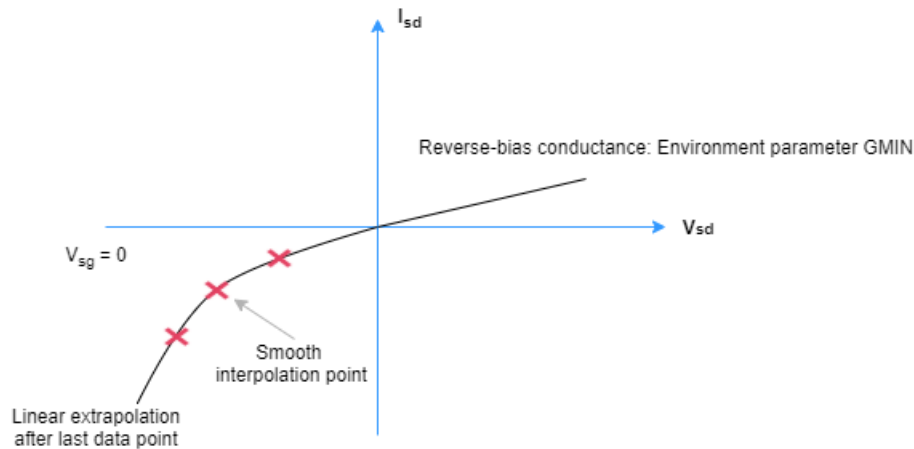
where:

- $I_{dio}$  is the current through the diode.
- $I_s$  is the reverse saturation current.
- $V_{BD}$  is the body-drain voltage.
- $n$  is the ideality factor.
- $\phi_T$  is the thermal voltage.
- $C_j$  is the junction capacitance of the diode.
- $C_{j0}$  is the zero-bias junction capacitance.
- $V_{bi}$  is the built-in voltage.
- $C_{diff}$  is the diffusion capacitance of the diode.
- $\tau$  is the transit time.

The capacitances are defined through an explicit calculation of charges, which are then differentiated to give the capacitive expressions above. The block computes the capacitive diode currents as time derivatives of the relevant charges, similar to the computation in the surface-potential-based MOSFET model.

### Tabulated Diode

To model a tabulated diode, set the **Model body diode** parameter to `Tabulated I-V curve`. This figure shows the implementation of the tabulated diode option:



When choosing this parameterization, you must provide the data for the forward bias only.

The block implements the diode using a smooth interpolation option. If the diode exceeds the provided tabulated data range, the block uses a linear extrapolation technique at the last voltage-current data point.

---

**Note** The tabulated diode does not model the reverse breakdown.

---

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. To model the dependence on temperature during simulation, select **Model temperature dependence** for the **Parameterization** parameter in the **Temperature Dependence** setting.

### Threshold-Based Model

For threshold-based modeling option, you can include modeling the dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the junction capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The gain,  $K$ , and the threshold voltage,  $V_{th}$ , become a function of temperature according to the following equations:

$$K_{T_s} = K_{T_{m1}} \left( \frac{T_s}{T_{m1}} \right)^{BEX}$$

$$V_{th_s} = V_{th1} + \alpha (T_s - T_{m1})$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $K_{T_{m1}}$  is the transistor gain at the measurement temperature.

- $K_{Ts}$  is the transistor gain at the simulation temperature. This is the transistor gain value used in the MOSFET equations when temperature dependence is modeled.
- $V_{thI}$  is the threshold voltage at the measurement temperature.
- $V_{ths}$  is the threshold voltage at the simulation temperature. This is the threshold voltage value used in the MOSFET equations when temperature dependence is modeled.
- $BEX$  is the mobility temperature exponent. A typical value of  $BEX$  is -1.5.
- $\alpha$  is the gate threshold voltage temperature coefficient,  $dV_{th}/dT$ .

For the four terminals parameterization,  $V_{th}$  is obtained using these equations:

$V_{BS}$ Range	$V_{th}$ Equation
$V_{BS} \leq 0$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma}{2\sqrt{2\phi_B}} \frac{d2\phi_B}{dT} + \frac{\gamma}{2\sqrt{2\phi_B - V_{BS}}} \frac{d2\phi_B}{dT}$
$0 < V_{BS} \leq 4\phi_B$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma V_{BS}}{4} (2\phi_B)^{-\frac{3}{2}} \frac{d2\phi_B}{dT}$
$V_{BS} > 4\phi_B$	$\frac{dV_{th}}{dT} = \frac{dV_{T0}}{dT} - \frac{\gamma}{2\sqrt{2\phi_B}} \frac{d2\phi_B}{dT}$

Where:

- $\phi_B = \frac{kT}{q} \ln\left(\frac{N_B}{n_i}\right)$  is the surface potential and  $\frac{d2\phi_B}{dT} = -\frac{1}{T} \left[ 2\phi_B - \left( \frac{E_g(0)}{q} + \frac{3kT}{q} \right) \right]$ .
- $E_g(0)$  is the extrapolated zero degree band-gap, which is equal to 1.16 eV for silicon.
- $V_{BS}$  is the bulk-source voltage.

For most MOSFETS, you can use the default value of -1.5 for  $BEX$ . Some datasheets quote the value for  $\alpha$ , but most typically they provide the temperature dependence for drain-source on resistance,  $R_{DS(on)}$ . Depending on the block parameterization method, you have two ways of specifying  $\alpha$ :

- If you parameterize the block from a datasheet, you have to provide  $R_{DS(on)}$  at a second measurement temperature. The block then calculates the value for  $\alpha$  based on this data.
- If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

If you have more data comprising drain current as a function of gate-source voltage for more than one temperature, then you can also use Simulink Design Optimization software to help tune the values for  $\alpha$  and  $BEX$ .

### Surface-Potential-Based Model

The surface-potential-based model includes temperature effects on the capacitance characteristics, as well as modeling the dependence of the transistor static behavior on temperature during simulation.

The **Measurement temperature** parameter on the **Main** setting specifies temperature  $T_{m1}$  at which the other device parameters have been extracted. The **Temperature Dependence** setting provides the simulation temperature,  $T_s$ , and the temperature-scaling coefficients for the other device parameters. For more information, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1268.

## Faults

The P-Channel MOSFET block models five types of fault:

- Open circuit — Failure due to metallization burnout
- Drain-source short — Failure due to avalanche breakdown on drain-source channel
- Drain-bulk short or source-bulk short — Failure due to avalanche breakdown on drain-bulk or source-bulk channels
- Gate oxide short — Failure of the gate oxide dielectric layer
- Parameter shift — Failure due to aging

The block can trigger fault events at a specific time or when the current or voltage exceed the limit for longer than a specific time interval.

To enable these trigger mechanisms, set the **Fault trigger** parameter to either Temporal or Behavioral.

### Parameter Shift Fault

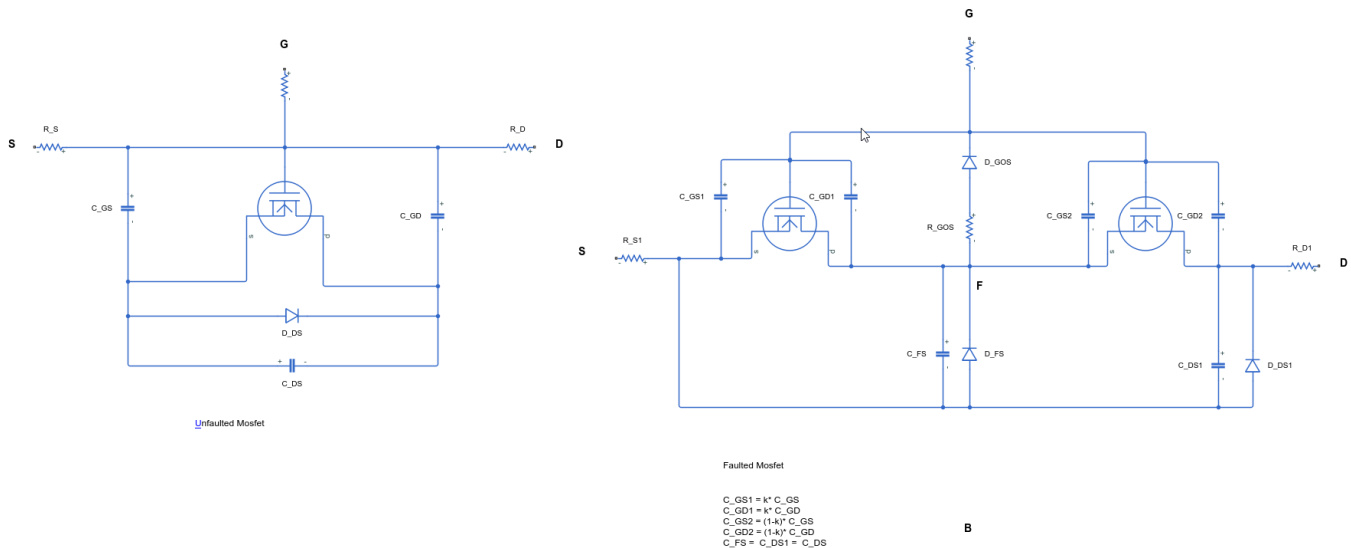
If, in the **Faults** settings, you set the **Failure mode** parameter to Parameter shift, the MOSFET fails due to the aging of its components. This equation defines the value of the shifted parameters:

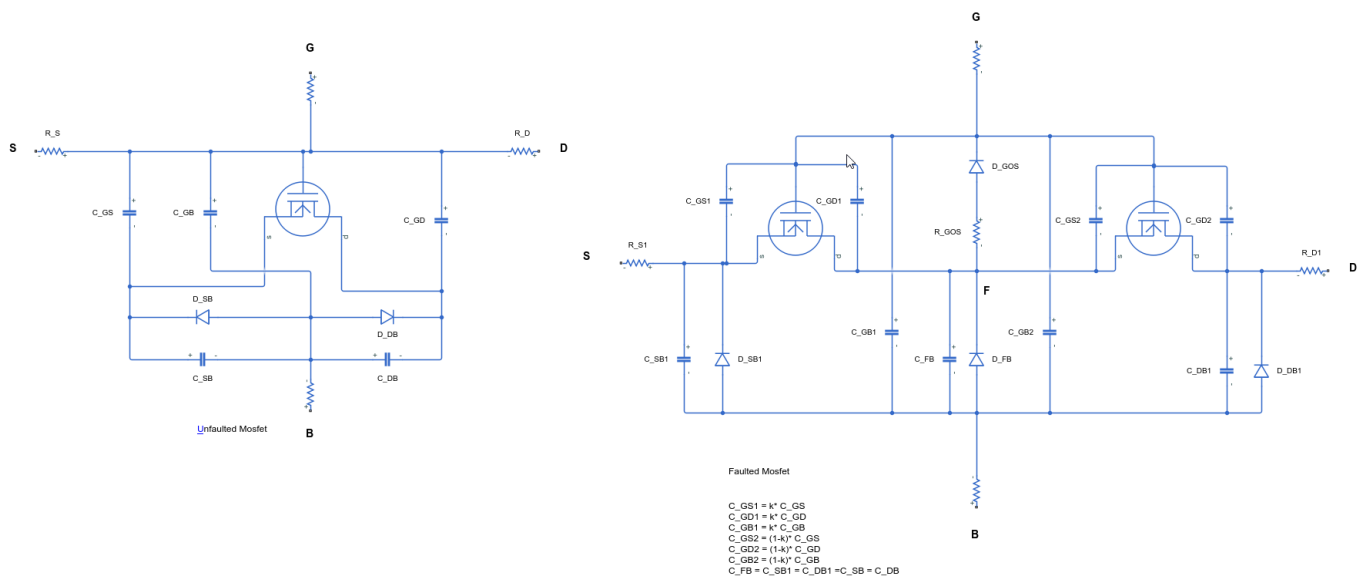
$$Parameter(t) = Parameter_{faulted} - (Parameter_{faulted} - Parameter_{unfaulted}) \operatorname{sech}\left(\frac{t - t_{th}}{\tau}\right),$$

where  $t_{th}$  is the time threshold when the fault is triggered and  $\tau$  is the value of the **Fault transition time constant, tau** parameter.

### Gate Oxide Short Fault

If, in the **Faults** settings, you set the **Failure mode** parameter to Gate oxide short, the gate oxide dielectric layer fails. These figures show the equivalent circuits for the three terminal or four terminal MOSFET in the unfaulted and faulted state:





## Thermal Port

The block has an optional thermal port, hidden by default. To expose the thermal port, set the **Modeling option** parameter to:

- Threshold-based with thermal — Model based on threshold voltage and with exposed thermal port
- Surface-potential-based with thermal — Model based on surface potential and with exposed thermal port

This action displays the thermal port **H** on the block icon, and exposes the **Thermal Port** parameters.

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

This table shows the relationship between the capacitances of the block and the initial targets:

Defined Capacitance	Initial Targets
Gate-emitter capacitance, $C_{ge}$	Set the initial target for the gate-emitter capacitance voltage only. Set the initial target of the collector-emitter capacitance voltage to 0 or set its priority to None.
Collector-emitter capacitance, $C_{ce}$	Set the initial target for the collector-emitter capacitance voltage only. Set the initial target of the gate-emitter capacitance voltage to 0 or set its priority to None.
Gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and gate-collector capacitance, $C_{gc}$	Set the initial targets for the gate-emitter and gate-collector voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial target for the gate-emitter and the collector-emitter capacitance.
Gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-collector and collector-emitter voltages by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.
Gate-emitter capacitance, $C_{ge}$ , gate-collector capacitance, $C_{gc}$ , and collector-emitter capacitance, $C_{ce}$	Set the initial targets for the gate-emitter, gate collector and collector-emitter capacitances by applying constraints on the gate-emitter and collector-emitter voltages. The initial condition of the gate-collector capacitance voltage is equal to the voltage between the gate-emitter and the collector-emitter.

**Note** Inside your model, the number of initial targets with **Priority** equal to Low or High must match the number of differential variables. The differential variables come from the inductors and the capacitances in the model.

## Assumptions and Limitations

When modeling temperature dependence for threshold-based block modeling option, consider the following:

- The block does not account for temperature-dependent effects on the capacitances.
- When you specify  $R_{DS(on)}$  at a second measurement temperature, it must be quoted for the same working point (that is, the same drain current and gate-source voltage) as for the other  $R_{DS(on)}$  value. Inconsistent values for  $R_{DS(on)}$  at the higher temperature will result in unphysical values for  $\alpha$  and unrepresentative simulation results. Typically  $R_{DS(on)}$  increases by a factor of about 1.5 for a hundred degree increase in temperature.
- You may need to tune the values of  $BEX$  and threshold voltage,  $V_{th}$ , to replicate the  $I_{DS}-V_{GS}$  relationship (if available) for a given device. Increasing  $V_{th}$  moves the  $I_{DS}-V_{GS}$  plots to the right. The value of  $BEX$  affects whether the  $I_{DS}-V_{GS}$  curves for different temperatures cross each other, or not, for the ranges of  $V_{DS}$  and  $V_{GS}$  considered. Therefore, an inappropriate value can result in the different temperature curves appearing to be reordered. Quoting  $R_{DS(on)}$  values for higher currents, preferably close to the current at which it will operate in your circuit, will reduce sensitivity to the precise value of  $BEX$ .

## Ports

### Conserving

#### G — Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal

#### D — Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal

#### S — Source terminal

electrical

Electrical conserving port associated with the transistor source terminal

#### B — Body terminal

electrical

Electrical conserving port associated with the transistor body terminal

### Dependencies

To enable this port, set **Number of terminals** to Four.

## Parameters

### Modeling Option — MOSFET representation

Threshold-based (default) | Threshold-based with thermal | Surface-potential-based | Surface-potential-based with thermal

Select the MOSFET representation:

- **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.

- **Threshold-based with thermal** — Model based on threshold voltage and with exposed thermal port.
- **Surface-potential-based** — Model based on surface potential. This model does not simulate thermal effects.
- **Surface-potential-based with thermal** — Thermal modeling option of the model based on surface potential.

### **Main (Threshold-Based Modeling Option)**

This configuration of the **Main** setting corresponds to the threshold-based block modeling option, which is the default. If you are using the surface-potential-based modeling option of the block, see “Main (Surface-Potential-Based Modeling Option)” on page 1-1255.

### **Number of terminals — Terminal parameterization**

Three (default) | Four

Number of terminals of the block.

### **Parameterization — Block parameterization**

Specify from a datasheet (default) | Specify using equation parameters directly | Lookup table (2-D, temperature independent) | Lookup table (3-D, temperature dependent)

Select one of the following methods for block parameterization:

- **Specify from a datasheet** — Provide the drain-source on resistance and the corresponding drain current and gate-source voltage. The block calculates the transistor gain for the Shichman and Hodges equations from this information.
- **Specify using equation parameters directly** — Provide the transistor gain.
- **Lookup table (2-D, temperature independent)** — Use 2-D table lookup for drain-source current as a function of gate-source voltage and drain-source voltage.
- **Lookup table (3-D, temperature dependent)** — Use 3-D table lookup for drain-source current as a function of gate-source voltage, drain-source voltage, and temperature.

### **Drain-source on resistance, $R_{DS(on)}$ — Drain-source on resistance**

0.167 Ohm (default)

The ratio of the drain-source voltage to the drain current for specified values of drain current and gate-source voltage.  $R_{DS(on)}$  should have a positive value.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.

### **Drain current, $I_{DS}$ , for $R_{DS(on)}$ — Drain current, $I_{DS}$ , for $R_{DS(on)}$**

-2.5 A (default)

The drain current the block uses to calculate the value of the drain-source resistance.  $I_{DS}$  should have a negative value.

#### **Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.



**Gate-source voltage, Vgs, for R\_DS(on) – Gate-source voltage, Vgs, for R\_DS(on)**  
-4.5 V (default)

The gate-source voltage the block uses to calculate the value of the drain-source resistance.  $V_{GS}$  should have a negative value.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet.

**Gain, K – Gain, K**  
2 A/V<sup>2</sup> (default)

Positive constant gain coefficient for the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify using equation parameters directly.

**Gate-source threshold voltage, Vth – Gate-source threshold voltage**  
-1.4 V (default)

Gate-source threshold voltage  $V_{th}$  in the Shichman and Hodges equations. For an enhancement device,  $V_{th}$  should be negative. For a depletion mode device,  $V_{th}$  should be positive.

**Dependencies**

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Gate-source threshold voltage at zero bulk-source voltage, Vth0 – Gate-source threshold voltage at zero bulk-source voltage**  
-1.4 V (default)

Gate-source threshold voltage at zero bulk-source voltage  $V_{th0}$  in the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Channel modulation, L – Channel modulation**  
0 1/V (default)

The channel-length modulation, usually denoted by the mathematical symbol  $\lambda$ . When in the saturated region, it is minus the rate of change of drain current with drain-source voltage. The effect on drain current is typically small, and the effect is neglected if calculating transistor gain  $K$  from drain-source on-resistance,  $R_{DS(on)}$ . A typical value is 0.02, but the effect can be ignored in most circuit simulations. However, in some circuits a small nonzero value may help numerical convergence.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Gate-source threshold voltage at first non-zero bulk-source voltage, Vth1 — Gate-source threshold voltage at first non-zero bulk-source voltage**

-1.6071 V (default)

Gate-source threshold voltage at first non-zero bulk-source voltage  $V_{th1}$  in the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**First bulk-source voltage, Vbs1 — First bulk-source voltage**

1 V (default)

First bulk-source voltage,  $V_{bs1}$

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Gate-source threshold voltage at second non-zero bulk-source voltage, Vth2 — Gate-source threshold voltage at second non-zero bulk-source voltage**

-1.7660 V (default)

Gate-source threshold voltage at second non-zero bulk-source voltage  $V_{th2}$  in the Shichman and Hodges equations.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Second bulk-source voltage, Vbs2 — Second bulk-source voltage**

2 V (default)

Second bulk-source voltage,  $V_{bs2}$

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify from a datasheet.

**Body factor — Body factor**

$0.5 V^{0.50000}$  (default)

Body factor,  $\gamma$ .

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify using equation parameters directly.

**Surface potential — Surface potential**

0.5 V (default)

Surface potential

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Specify using equation parameters directly.

**Measurement temperature — Measurement temperature**

25 degC (default)

Temperature  $T_{m1}$  at which **Drain-source on resistance, R\_DS(on)** is measured.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

**Vector of source-gate voltages, Vsg — Vector of source-gate voltages**

[-.5, 2, 3, 4, 5] V (default)

Vector of source-gate voltages.

**Dependencies**

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

**Vector of source-drain voltages, Vsd — Vector of source-drain voltages**

[0, 2, 4, 8, 12, 16, 20] V (default)

Vector of source-drain voltages, to be used for table lookup. The vector values must be strictly increasing.

**Dependencies**

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

**Vector of source-bulk voltages, Vsb — Vector of source-bulk voltages**

[0, -4, -6, -12] V (default)

Vector of source-bulk voltages, to be used for table lookup.

**Dependencies**

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

**Tabulated source-drain currents, Isd(Vsg,Vsd) — Tabulated source-drain currents temperature independent**

[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;  
0, .0115, .016, .018, .019, .0195, .02;  
0, .0152, .0225, .0265, .028, .0295, .03] A (default)

Tabulated values for source-drain currents as a function of source-gate voltage and source-drain voltage, to be used for 2-D table lookup. Each value in the matrix specifies the source-drain current

for a specific combination of source-gate voltage and source-drain voltage. The matrix size must match the dimensions defined by the source-gate voltage and source-drain voltage vectors.

#### Dependencies

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Lookup table (2-D, temperature independent).

#### Vector of temperatures, T — Vector of temperatures

[25 125] degC (default)

Vector of temperatures, to be used for table lookup. The vector values must be strictly increasing. The values can be nonuniformly spaced.

#### Dependencies

To enable this parameter, set **Parameterization** to Lookup table (3-D, temperature dependent).

#### Tabulated source-drain currents, Isd(Vsg,Vsd,T) — Tabulated source-drain currents temperature dependent

zeros(5, 7, 2) A (default)

Tabulated values for source-drain currents as a function of source-gate voltage, source-drain voltage, and temperature, to be used for 2-D table lookup. Each value in the matrix specifies the source-drain current for a specific combination of source-gate voltage, source-drain voltage, and temperature. The matrix size must match the dimensions defined by the source-gate voltage, source-drain voltage, and temperature vectors.

#### Dependencies

To enable this parameter, set **Number of terminals** to Three and **Parameterization** to Lookup table (3-D, temperature dependent).

#### Tabulated gate-source threshold voltage, Vth(Vsb) — Vector of gate-source threshold voltage temperature independent

[.4, 1.2, 1.4, 1.45] V (default)

Tabulated values for gate-source threshold voltage as a function of source-bulk voltage, to be used for 2-D table lookup. The vector values must be strictly increasing.

#### Dependencies

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (2-D, temperature independent).

#### Tabulated source-drain currents at zero source-bulk voltage, Isd(Vsg,Vsd) — Tabulated source-drain currents at zero source-bulk voltage, temperature independent

[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;  
0, .0115, .016, .018, .019, .0195, .02;  
0, .0152, .0225, .0265, .028, .0295, .03] A (default)

Tabulated values for source-drain currents at zero source-bulk voltage, as a function of source-gate voltage and source-drain voltage, to be used for 2-D table lookup. Each value in the matrix specifies the source-drain current for a specific combination of source-gate voltage and source-drain voltage.

The matrix size must match the dimensions defined by the source-gate voltage and source-drain voltage vectors.

#### Dependencies

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (2-D, temperature independent).

#### Tabulated gate-source threshold voltage, $V_{th}(V_{sb}, T)$ — Vector of gate-source threshold voltage temperature dependent

[.4, .35; 1.2, 1.15; 1.4, 1.35; 1.45, 1.4] V (default)

Tabulated values for gate-source threshold voltage as a function of bulk-source voltage and temperature, to be used for 3-D table lookup. The vector values must be strictly increasing.

#### Dependencies

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (3-D, temperature dependent).

#### Tabulated source-drain currents at zero source-bulk voltage, $I_{sd}(V_{sg}, V_{sd}, T)$ — Tabulated source-drain currents at zero source-bulk voltage, temperature dependent

zeros(5, 7, 2) A (default)

Tabulated values for source-drain currents at zero source-bulk voltage, as a function of source-gate voltage, source-drain voltage, and temperature, to be used for 3-D table lookup. Each value in the matrix specifies the source-drain current for a specific combination of source-gate voltage, source-drain voltage, and temperature. The matrix size must match the dimensions defined by the source-gate voltage, source-drain voltage, and temperature vectors.

#### Dependencies

To enable this parameter, set **Number of terminals** to Four and **Parameterization** to Lookup table (3-D, temperature dependent).

#### Isd-Vsd parameterization — Isd-Vsd table data as symmetric data

Provide negative and positive Vsd data (default) | Provide positive Vsd data only

Whether to provide Isd-Vsd table data as symmetric data. If you choose Provide negative and positive Vsd data, the data is not symmetric. If you choose Provide positive Vsd data only, the block rotates and flips the positive data to obtain the negative data.

#### Dependencies

To enable this parameter, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).

#### Main (Surface-Potential-Based Modeling Option)

This configuration of the **Main** setting corresponds to the surface-potential-based block modeling option. If you are using the threshold-based modeling option of the block, based on the Shichman and Hodges equations, see “Main (Threshold-Based Modeling Option)” on page 1-1250.

#### Number of terminals — Terminal parameterization

Three (default) | Four

Number of terminals of the block.

**Gain – Gain**18 A/V<sup>2</sup> (default)

The MOSFET gain,  $\beta$ . This parameter primarily defines the linear region of operation on an  $I_D$ - $V_{DS}$  characteristic.

**Flatband voltage – Flatband voltage**

-1.1 V (default)

The flatband voltage,  $V_{FB}$ , defines the gate bias that must be applied in order to achieve the flatband condition at the surface of the silicon. You can also use this parameter to arbitrarily shift the threshold voltage due to material work function differences, and to trapped interface or oxide charges. In practice, however, it is usually recommended to modify the threshold voltage by using the **Body factor** and **Surface potential at strong inversion** parameters first, and only use this parameter for fine-tuning.

**Body factor – Body factor**3.5 V<sup>1/2</sup> (default)

Body factor,  $\gamma$ , in the surface-potential equation. This parameter primarily impacts the threshold voltage.

**Surface potential at strong inversion – Surface potential at strong inversion**

1 V (default)

The  $2\phi_B$  term in the surface-potential equation. This parameter also primarily impacts the threshold voltage.

**Velocity saturation factor – Velocity saturation factor**

0.4 1/V (default)

Velocity saturation,  $\theta_{sat}$ , in the drain-current equation. Use this parameter in cases where a good fit to linear operation leads to a saturation current that is too high. By increasing this parameter value, you reduce the saturation current. For high-voltage devices, it is often the case that a good fit to linear operation leads to a saturation current that is too low. In such a case, either increase both the gain and the drain ohmic resistance or use a P-Channel LDMOS FET block instead.

**Channel-length modulation factor – Channel-length modulation factor**

0 (default)

The factor,  $\alpha$ , multiplying the logarithmic term in the  $G_{\Delta L}$  equation. This parameter describes the onset of channel-length modulation. For device characteristics that exhibit a positive conductance in saturation, increase the parameter value to fit this behavior. The default value is 0, which means that channel-length modulation is off by default.

**Channel-length modulation voltage – Channel-length modulation voltage**

5e-2 V (default)

The voltage  $V_p$  in the  $G_{\Delta L}$  equation. This parameter controls the drain-voltage at which channel-length modulation starts to become active.

**Surface roughness scattering factor – Surface roughness scattering factor**

0 1/V (default)

Indicates the strength of the mobility reduction. The mobility is  $\mu = \mu/G_{mob}$ , where  $\mu_0$  is the low-field mobility without the effect of surface scattering. The mobility reduction factor,  $G_{mob}$ , is given by

$G_{mob} = \sqrt{1 + (\theta_{sr} V_{eff})^2}$ , where  $\theta_{sr}$  is the surface roughness scattering factor and  $V_{eff}$  is a voltage that is indicative of the effective vertical electric field strength in the channel,  $E_{eff}$ . For high vertical electric fields, the mobility is roughly proportional to  $E_{eff}$  for holes.

### Linear-to-saturation transition coefficient – Linear-to-saturation transition coefficient

8 (default)

This parameter controls how smoothly the MOSFET transitions from linear into saturation, particularly when velocity saturation is enabled. This parameter can usually be left at its default value, but you can use it to fine-tune the knee of the  $I_D$ - $V_{DS}$  characteristic. The expected range for this parameter value is between 2 and 8.

### Measurement temperature – Measurement temperature

25 degC (default)

Temperature  $T_{m1}$  at which the block parameters are measured. If the **Device simulation temperature** parameter on the **Temperature Dependence** setting differs from this value, then device parameters will be scaled from their defined values according to the simulation and reference temperatures. For more information, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1268.

### Ohmic Resistance

If you set **Modeling option** to Threshold-based or Threshold-based with thermal, to enable this setting set **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.

### Source ohmic resistance – Source ohmic resistance

1e-4 Ohm (default) | nonnegative scalar

Transistor source resistance, that is, the series resistance associated with the source contact. The value must be greater than or equal to 0. The default value for threshold-based variants is 1e-4 Ohm. The default value for surface-potential-based variants is 2e-3 Ohm.

### Drain ohmic resistance – Drain ohmic resistance

0.01 Ohm (default) | nonnegative scalar

Transistor drain resistance, that is, the series resistance associated with the drain contact. The value must be greater than or equal to 0. The default value for threshold-based variants is 0.01 Ohm. The default value for surface-potential-based variants is 0.17 Ohm.

### Gate ohmic resistance – Gate ohmic resistance

8.4 Ohm (default) | nonnegative scalar

Transistor gate resistance, that is, the series resistance associated with the gate contact. The value must be greater than or equal to 0.

### Dependencies

To enable this parameter, set **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

### Body ohmic resistance – Transistor body resistance

0.001 Ohm (default) | nonnegative scalar

Transistor body resistance, that is, the series resistance associated with the body contact.

**Dependencies**

To enable this parameter, set:

- **Number of terminals** to Four.
- **Modeling option** to Threshold-based or Threshold-based with thermal.

**Bulk ohmic resistance — Transistor bulk resistance**

2e-3 Ohm (default) | nonnegative scalar

Transistor body resistance, that is, the series resistance associated with the bulk contact.

**Dependencies**

To enable this parameter, set:

- **Number of terminals** to Four.
- **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

**Capacitance**

To enable this setting, set **Modeling option** to Threshold-based or Threshold-based with thermal.

**Parameterization — Capacitance parameterization**

Specify fixed input, reverse transfer, and output capacitance (default) | Specify fixed gate-source, gate-drain, and drain-source capacitance | Specify tabulated input, reverse transfer, and output capacitance | Specify tabulated gate-source, gate-drain, and drain-source capacitance

Select one of the following methods for capacitance parameterization:

- **Specify fixed input, reverse transfer, and output capacitance** — Provide fixed parameter values from datasheet and let the block convert the input, output, and reverse transfer capacitance values to capacitance values, as described in “Charge Model for Threshold-Based Modeling Option” on page 1-1235. This is the default method.
- **Specify fixed gate-source, gate-drain, and drain-source capacitance** — Provide fixed values for capacitance parameters directly.
- **Specify tabulated input, reverse transfer, and output capacitance** — Provide tabulated capacitance and source-drain voltage values based on datasheet plots. The block converts the input, output, and reverse transfer capacitance values to capacitance values, as described in “Charge Model for Threshold-Based Modeling Option” on page 1-1235.
- **Specify tabulated gate-source, gate-drain, and drain-source capacitance** — Provide tabulated values for capacitances and source-drain voltage.

**Input capacitance, Ciss — Input gate-source capacitance**

182 pF (default)

Gate-source capacitance with the drain shorted to the source.



**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Input capacitance,  $C_{iss}(V_{sd})$  — Input gate-source capacitance with tabulated input**  
[225 210 200 185 175 170] pF (default)

Gate-source capacitance with the drain shorted to the source, tabulated with source-drain voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Reverse transfer capacitance,  $C_{rss}$  — Reverse transfer capacitance**  
24 pF (default)

Drain-gate capacitance with the source connected to ground, also known as the Miller capacitance.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Reverse transfer capacitance,  $C_{rss}(V_{sd})$  — Reverse transfer capacitance with tabulated inputs**  
[75 60 50 35 25 20] pF (default)

Drain-gate capacitance with the source connected to ground, also known as the Miller capacitance, tabulated with source-drain voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Output capacitance,  $C_{oss}$  — Output capacitance**  
0 pF (default)

Drain-source capacitance with the gate and source shorted.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance.

**Output capacitance,  $C_{oss}(V_{sd})$  — Output capacitance with tabulated inputs**  
[180 160 125 80 60 45] pF (default)

Drain-source capacitance with the gate and source shorted, tabulated with source-drain voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

**Gate-source capacitance, Cgs — Gate-source capacitance**

158 pF (default)

Capacitance placed between the gate and the source.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

**Gate-source capacitance, Cgs(Vsg, Vsd) — Gate-source capacitance with tabulated values**

[150 150 150 150 150 150] pF (default)

Capacitance placed between the gate and the source, tabulated with source-gate voltage and source-drain voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

**Gate-drain capacitance, Cgd — Gate-drain capacitance**

24 pF (default)

Capacitance placed between the gate and the drain.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

**Gate-drain capacitance, Cgd(Vsg, Vsd) — Gate-drain capacitance with tabulated values**

[75 60 50 35 25 20] pF (default)

Capacitance placed between the gate and the drain, tabulated with source-gate voltage and source-drain voltage.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

**Gate-bulk capacitance, Cgb — Gate-bulk capacitance**

0 pF (default)

Capacitance placed between the gate and the bulk.

**Dependencies**

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance and **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.

**Gate-bulk capacitance, Cgb(Vsg, Vsd) — Gate-bulk capacitance with tabulated values**

[450 400 300 190 95 55] pF (default)

Capacitance placed between the gate and the bulk, tabulated with source-gate voltage and source-drain voltage.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance and **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.

#### Corresponding source-gate voltages, $V_{sg}$ — Corresponding source-gate voltages

0 V (default)

Source-gate voltages corresponding to the tabulated capacitance values.

If this value is a scalar, then **Gate-source capacitance,  $C_{gs}(V_{sg}, V_{sd})$**  and **Gate-drain capacitance,  $C_{gd}(V_{sg}, V_{sd})$**  are vectors of scalars. If this value is a vector, then **Gate-source capacitance,  $C_{gs}(V_{sg}, V_{sd})$**  and **Gate-drain capacitance,  $C_{gd}(V_{sg}, V_{sd})$**  are matrices of scalars.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

#### Drain-source capacitance, $C_{ds}$ — Drain-source capacitance

0 pF (default)

Capacitance placed between the drain and the source.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.

#### Drain-source capacitance, $C_{ds}(v_{sd})$ — Drain-source capacitance with tabulated values

[105 100 75 45 35 25] pF (default)

Capacitance placed between the drain and the source, tabulated with the source-drain voltage.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

#### Corresponding source-drain voltages, $V_{sd}$ — Corresponding source-drain voltages

[0.1 0.3 1 3 10 30] V (default)

Source-drain voltages corresponding to the tabulated capacitance values.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

#### Source-gate voltage, $V_{sg}$ , for tabulated capacitances — Source-gate voltage, $V_{sg}$ , for tabulated capacitances

0 V (default)

For tabulated capacitance models, this parameter controls the voltage dependence of the **Reverse transfer capacitance, Crss** or the **Gate-drain capacitance, Cgd** parameter (depending on the selected parameterization option). These capacitances are a function of the drain-gate voltage. The block calculates drain-gate voltages by subtracting this gate-source voltage value from the negative of the values specified for the **Corresponding source-drain voltages** parameter ( $V_{dg} = -V_{sd} - V_{gs}$ ).

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.

#### Gate-drain charge-voltage linearity — Gate-drain charge-voltage linearity

Gate-drain capacitance is constant (default) | Gate-drain charge function is nonlinear

The two fixed capacitance options let you model gate capacitance as a fixed gate-source capacitance  $C_{GS}$  and either a fixed or a nonlinear gate-drain capacitance  $C_{GD}$ . Select whether the gate-drain capacitance is fixed or nonlinear:

- **Gate-drain capacitance is constant** — The capacitance value is constant and defined according to the selected parameterization option, either directly or derived from a datasheet. This is the default method.
- **Gate-drain charge function is nonlinear** — The gate-drain charge relationship is defined according to the piecewise-nonlinear function described in “Charge Model for Threshold-Based Modeling Option” on page 1-1235. Two additional parameters appear to let you define the gate-drain charge function.

#### Dependencies

To enable this parameter, in the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer and output capacitance or to Specify fixed gate-source, gate-drain and drain-source capacitance.

#### Gate-drain oxide capacitance — Gate-drain oxide capacitance

200 pF (default)

The gate-drain capacitance when the drain-gate voltage is less than the **Drain-gate voltage at which oxide capacitance becomes active** parameter value.

#### Dependencies

To enable this parameter, set **Gate-drain charge-voltage linearity** to Gate-drain charge function is nonlinear.

#### Drain-gate voltage at which oxide capacitance becomes active — Drain-gate voltage at which oxide capacitance becomes active

0.5 V (default)

The drain-gate voltage at which the drain-gate capacitance switches between off-state ( $C_{GD}$ ) and on-state ( $C_{ox}$ ) capacitance values.

#### Dependencies

To enable this parameter, set **Gate-drain charge-voltage linearity** to Gate-drain charge function is nonlinear.

**Gate-bulk and gate-source charge-voltage linearity – Gate-bulk and gate-source charge-voltage linearity**

Gate-bulk and gate-source capacitance change instantly (default) | Gate-bulk and gate-source capacitance change gradually | Separate gate-bulk and gate-source capacitance

Gate-bulk and gate-source charge-voltage linearity.

**Dependencies**

To enable this parameter, in the **Main** setting, set **Number of terminals** to Four.

To enable the Separate gate-bulk and gate-source capacitance option, in the **Capacitance** setting, set **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance

**Channel Capacitances**

To enable this setting, set **Modeling option** to Surface-potential-based or Surface-potential-based with thermal.

**Oxide capacitance – Oxide capacitance**

1500 pF (default)

The parallel plate gate-channel capacitance.

**Gate-source overlap capacitance – Gate-source overlap capacitance**

100 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the source well.

**Gate-drain overlap capacitance – Gate-drain overlap capacitance**

14 pF (default)

The fixed, linear capacitance associated with the overlap of the gate electrode with the drain well.

**Body Diode****Model body diode – Body diode modeling option**

No (default) | Exponential | Tabulated I-V curve

Whether to model the body diode.

**Table type – Tabulated function**

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

To enable this parameter, set **Model body diode** to Tabulated I-V curve.

**Reverse saturation current,  $I_s$  – Reverse saturation current**

0 A (default)

The current designated by the  $I_s$  symbol in the body-diode equations. The default value for threshold-based modeling option is 0 A. The default value for surface-potential-based modeling option is  $5.2e-13$  A.

To enable conduction through the body diode, for applications where the MOSFET current changes sign during the simulation, such as when the MOSFET is driving an inductive load, set this parameter to a non-zero value.

For applications where the MOSFET current never changes sign, such as in a small-signal amplifier, set this parameter to 0 to improve simulation speed.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Built-in voltage, Vbi — Built-in voltage**

0.6 V (default)

The built-in voltage of the diode, designated by the  $V_{bi}$  symbol in the body-diode equations. Built-in voltage has an impact only on the junction capacitance equation. It does not affect the conduction current.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Ideality factor, n — Ideality factor**

1 (default)

The factor designated by the  $n$  symbol in the body-diode equations.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Zero-bias junction capacitance, Cj0 — Zero-bias junction capacitance**

0 pF (default)

The capacitance between the drain and bulk contacts at zero-bias due to the body diode alone. It is designated by the  $C_{j0}$  symbol in the body-diode equations. The default value for threshold-based modeling option is 0 pF. The default value for surface-potential-based modeling option is 480 pF.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Transit time, TT — Transit time**

$50e-9$  s (default)

The time designated by the  $\tau$  symbol in the body-diode equations.

When the **Reverse saturation current** and **Transit time** parameters are both non-zero, this block includes the reverse recovery inside the body diode model.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential.

**Measurement temperature – Measurement temperature**

25 °C (default)

Temperature at which the block parameters are measured.

**Dependencies**

To enable this parameter, set **Model body diode** to Exponential and, in the **Main** setting, set **Parameterization** to either Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent)

**Diode forward voltages, Vf(If) – Diode forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85] V (default)

Tabulated values of the minimum voltage that needs to be applied for the diode to become forward-biased.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify from equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in Vf(If) form.

**Diode forward currents, If – Diode forward currents temperature independent**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Tabulated values of the forward currents.

**Dependencies**

To enable this parameter, in the **Body diode** setting, set **Model body diode** to Tabulated I-V curve and **Table type** to Table in Vf(If) form or Table in Vf(Tj,If) form.

**Diode forward currents, If(Vf) – Diode forward currents tabulated with forward voltages**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2] A (default)

Tabulated values of the forward current, as a function of the forward voltages.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify from equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in If(Vf) form.

**Diode forward voltages, Vf – Diode forward voltages temperature independent**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Tabulated values of the forward voltages.

**Dependencies**

To enable this parameter, in the **Body diode** setting, set **Model body diode** to Tabulated I-V curve and **Table type** to Table in If(Vf) form or Table in If(Tj,Vf) form.

**Diode junction temperatures, Tj – Diode junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures.

**Dependencies**

To enable this parameter, set **Model body diode** to Tabulated I-V curve and, in the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).

**Diode forward currents, If(Tj,Vf) – Diode forward currents temperature dependent**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default)

Tabulated values of the forward current, as a function of the forward voltages and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in If(Tj,Vf) form.

**Diode forward voltages, Vf(Tj,If) – Diode forward voltages temperature dependent**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default)

Tabulated values of the forward voltage, as a function of the forward currents and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.

**Table type** to Table in Vf(Tj,If) form.



### Temperature Dependence (Threshold-Based Modeling Option)

This configuration of the **Temperature Dependence** setting corresponds to the threshold-based block modeling option, which is the default. If you are using the surface-potential-based modeling option of the block, see “Temperature Dependence (Surface-Potential-Based Modeling Option)” on page 1-1268

#### Parameterization — Temperature dependence parameterization

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None — Simulate at parameter measurement temperature — Temperature dependence is not modeled. This is the default method.
- Model temperature dependence — Model temperature-dependent effects. Provide a value for simulation temperature,  $T_s$ , a value for  $BEX$ , and a value for the measurement temperature  $T_{m1}$  (using the **Measurement temperature** parameter on the **Main** setting). You also have to provide a value for  $\alpha$  using one of two methods, depending on the value of the **Parameterization** parameter on the **Main** setting. If you parameterize the block from a datasheet, you have to provide  $R_{DS(on)}$  at a second measurement temperature, and the block will calculate  $\alpha$  based on that. If you parameterize by specifying equation parameters, you have to provide the value for  $\alpha$  directly.

#### Drain-source on resistance, $R_{DS(on)}$ , at second measurement temperature — Drain-source on resistance, $R_{DS(on)}$ , at second measurement temperature

0.25 Ohm (default)

The ratio of the drain-source voltage to the drain current for specified values of drain current and gate-source voltage at second measurement temperature. It must be quoted for the same working point (drain current and gate-source voltage) as the **Drain-source on resistance,  $R_{DS(on)}$**  parameter on the **Main** setting.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** setting.

#### Second measurement temperature — Second measurement temperature

125 degC (default)

Second temperature  $T_{m2}$  at which **Drain-source on resistance,  $R_{DS(on)}$ , at second measurement temperature** is measured.

#### Dependencies

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** setting.

#### Gate threshold voltage temperature coefficient, $dV_{th}/dT$ — Gate threshold voltage temperature coefficient, $dV_{th}/dT$

2 mV/K (default)

The rate of change of gate threshold voltage with temperature.

## Dependencies

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter on the **Main** setting.

## Mobility temperature exponent, BEX — Mobility temperature exponent

-1.5 (default)

Mobility temperature coefficient value. You can use the default value for most MOSFETs. See the “Assumptions and Limitations” on page 1-1248 section for additional considerations.

## Body diode reverse saturation current temperature exponent — Body diode reverse saturation current temperature exponent

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$

$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** setting,  $k_B$  is Boltzmann’s constant ( $8.617 \times 10^{-5} \text{ eV/K}$ ), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

## Device simulation temperature — Device simulation temperature

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

## Temperature Dependence (Surface-Potential-Based Modeling Option)

This configuration of the **Temperature Dependence** setting corresponds to the surface-potential-based block modeling option. If you are using the threshold-based modeling option of the block, see “Temperature Dependence (Threshold-Based Modeling Option)” on page 1-1267

## Parameterization — Temperature dependence parameterization

None — Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None — Simulate at parameter measurement temperature — Temperature dependence is not modeled. This is the default method.
- Model temperature dependence — Model temperature-dependent effects. Provide a value for the device simulation temperature,  $T_s$ , and the temperature-scaling coefficients for other block parameters.

**Gain temperature exponent – Gain temperature exponent**

1.3 (default)

The MOSFET gain,  $\beta$ , is assumed to scale exponentially with temperature,  $\beta = \beta_{m1} (T_{m1}/T_s)^{\eta_\beta}$ .  $\beta_{m1}$  is the value of the **Gain** parameter from the **Main** setting and  $\eta_\beta$  is the **Gain temperature exponent**.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

**Flatband voltage temperature coefficient – Flatband voltage temperature coefficient**

5e-4 V/K (default)

The flatband voltage,  $V_{FB}$ , is assumed to scale linearly with temperature,  $V_{FB} = V_{FBm1} + (T_s - T_{m1})S_{TV_{FB}}$ .  $V_{FBm1}$  is the value of the **Flatband voltage** parameter from the **Main** setting and  $S_{TV_{FB}}$  is the **Flatband voltage temperature coefficient**.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

**Surface potential at strong inversion temperature coefficient – Surface potential at strong inversion temperature coefficient**

-8.5e-4 V/K (default)

The surface potential at strong inversion,  $2\phi_B$ , is assumed to scale linearly with temperature,  $2\phi_B = 2\phi_{Bm1} + (T_s - T_{m1})S_{T\phi_B}$ .  $2\phi_{Bm1}$  is the value of the **Surface potential at strong inversion** parameter from the **Main** setting and  $S_{T\phi_B}$  is the **Surface potential at strong inversion temperature coefficient**.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

**Velocity saturation temperature exponent – Velocity saturation temperature exponent**

1.04 (default)

The velocity saturation,  $\theta_{sat}$ , is assumed to scale exponentially with temperature,  $\theta_{sat} = \theta_{sat,m1} (T_{m1}/T_s)^{\eta_\theta}$ .  $\theta_{sat,m1}$  is the value of the **Velocity saturation factor** parameter from the **Main** setting and  $\eta_\theta$  is the **Velocity saturation temperature exponent**.

**Dependencies**

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

**Surface roughness scattering temperature exponent – Surface roughness scattering temperature exponent**

0.65 (default)

This parameter leads to a temperature-dependent reduction in the MOSFET transconductance at high gate voltage. The surface roughness scattering,  $\theta_{sr}$ , is assumed to scale exponentially with

temperature,  $\theta_{sr} = \theta_{sr,m1} (T_{m1}/T_s)^{\eta_{sr}}$ .  $\theta_{sr,m1}$  is the value of the **Surface roughness scattering factor** parameter from the **Main** setting and  $\eta_{sr}$  is the **Surface roughness scattering temperature exponent**.

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

#### Resistance temperature exponent — Resistance temperature exponent

0.95 (default)

The series resistances are assumed to correspond to semiconductor resistances. Therefore, they decrease exponentially with increasing temperature.  $R_i = R_{i,m1} (T_{m1}/T_s)^{\eta_R}$ , where  $i$  is S, D, or G, for the source, drain, or gate series resistance, respectively.  $R_{i,m1}$  is the value of the corresponding series resistance parameter from the **Ohmic Resistance** setting and  $\eta_R$  is the **Resistance temperature exponent**.

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

#### Body diode reverse saturation current temperature exponent — Body diode reverse saturation current temperature exponent

3 (default)

The reverse saturation current for the body diode is assumed to be proportional to the square of the intrinsic carrier concentration,  $n_i = N_C \exp(-E_G/2k_B T)$ .  $N_C$  is the temperature-dependent effective density of states and  $E_G$  is the temperature-dependent bandgap for the semiconductor material. To avoid introducing another temperature-scaling parameter, the block neglects the temperature dependence of the bandgap and uses the bandgap of silicon at 300K (1.12eV) for all device types. Therefore, the temperature-scaled reverse saturation current is given by

$$I_s = I_{s,m1} \left( \frac{T_s}{T_{m1}} \right)^{\eta_{Is}} \cdot \exp \left( \frac{E_G}{k_B} \cdot \left( \frac{1}{T_{m1}} - \frac{1}{T_s} \right) \right).$$

$I_{s,m1}$  is the value of the **Reverse saturation current** parameter from the **Body Diode** setting,  $k_B$  is Boltzmann's constant (8.617x10<sup>-5</sup>eV/K), and  $\eta_{Is}$  is the **Body diode reverse saturation current temperature exponent**. The default value is 3, because  $N_C$  for silicon is roughly proportional to  $T^{3/2}$ . You can remedy the effect of neglecting the temperature-dependence of the bandgap by a pragmatic choice of  $\eta_{Is}$ .

#### Dependencies

This parameter is visible only when you select Model temperature dependence for the **Parameterization** parameter.

#### Device simulation temperature — Device simulation temperature

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

**Dependencies**

This parameter is visible only when you select `Model temperature dependence` for the **Parameterization** parameter.

**Faults****Enable faults — Faults modeling option**

`Off` (default) | `On`

Whether to enable faults in MOSFET.

**Fault trigger — Fault trigger option**

`Temporal` (default) | `Behavioral`

Whether to trigger faults on temporal or behavioral conditions. Set this parameter to:

- `Temporal` — Trigger fault at a specific time.
- `Behavioral` — Trigger fault when a current limit or a voltage limit are exceeded for longer than a specific time interval.

**Simulation time for fault event — Time before entering faulted state**

`inf s` (default)

Simulation time at which you want the block to enter the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to `On` and **Fault trigger** to `Temporal`.

**Maximum permissible source-gate voltage — Source-gate voltage threshold to fault transition**

`inf V` (default) | positive scalar

Maximum permissible source-gate voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to `On` and **Fault trigger** to `Behavioral`.

**Maximum permissible source-drain voltage — Source-drain voltage threshold to fault transition**

`inf V` (default) | positive scalar

Maximum permissible source-drain voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to `On` and **Fault trigger** to `Behavioral`.

**Maximum permissible bulk-drain voltage — Bulk-drain voltage threshold to fault transition**

`inf V` (default) | positive scalar

Maximum permissible bulk-drain voltage value, above which a fault might be triggered.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Fault trigger** to Behavioral.

**Maximum permissible source-drain current — Current threshold to fault transition**

inf A (default) | positive scalar

Maximum permissible source-drain current value. If the current exceeds this value for longer than the value of the **Time to fail when exceeding drain-source current** parameter, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Time to fail when exceeding source-drain current — Time to fail when exceeding source-drain current**

1 s (default) | positive scalar

Amount of time that the source-drain current must continuously exceed the maximum permissible drain-source current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Fault trigger** to Behavioral.

**Maximum permissible body diode current — Body diode current threshold to fault transition**

inf A (default) | positive scalar

Maximum permissible body diode current value. If the current exceeds this value for longer than the value of the **Time to fail when exceeding body diode current** parameter, then the block enters the faulted state.

**Dependencies**

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

**Time to fail when exceeding body diode current — Time to fail when exceeding body diode current**

1 s (default) | positive scalar

Amount of time that the body diode current must continuously exceed the maximum permissible body diode current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

**Maximum diode recovery, dV/dt — Maximum diode recovery**

inf V/s (default) | positive scalar

Maximum amount of diode recovery per second.

#### Dependencies

To enable this parameter, set **Model body diode** to either Exponential or Tabulated I-V curve, **Enable faults** to On, and **Fault trigger** to Behavioral.

#### Maximum permissible temperature — Temperature threshold to fault transition

inf K (default) | positive scalar

Maximum permissible temperature value. If the temperature exceeds this value for longer than the value of the **Time to fail when exceeding temperature** parameter, then the block enters the faulted state.

#### Dependencies

To enable this parameter, set **Modeling option** to Threshold-based with thermal, **Enable faults** to On, and **Fault trigger** to Behavioral.

#### Time to fail when exceeding temperature — Time to fail when exceeding temperature

1 s (default) | positive scalar

Amount of time that the temperature must continuously exceed the maximum permissible temperature before a behavioral fault is triggered.

#### Dependencies

To enable this parameter, set **Modeling option** to Threshold-based with thermal, **Enable faults** to On, and **Fault trigger** to Behavioral.

#### Fault transition time constant, tau — Fault transition time constant

1e-3 s (default)

Time constant for transitioning to faulted state.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

#### Failure mode — Failure category

Open circuit (default) | Drain-source short | Drain-bulk short or source-bulk short  
| Gate oxide short | Parameter shift

Type of failure. You can set this parameter to:

- `Open circuit` — Failure due to metallization burnout.
- `Drain-source short` — Failure due to avalanche breakout on drain-source channel.
- `Drain-bulk short or source-bulk short` — Failure due to avalanche breakout on drain-bulk or source-bulk channels.
- `Gate oxide short` — Failure of the gate oxide dielectric layer.
- `Parameter shift` — Failure due to aging.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

To enable the Drain-source short option, in the **Main** setting, set **Number of terminals** to Three.

To enable the Drain-bulk short or source-bulk short option, in the **Main** setting, set **Number of terminals** to Four.

**Faulted series gate resistance, Rg — Faulted series gate resistance**

1e6 Ohm (default)

Series gate resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series drain resistance, Rd — Faulted series drain resistance**

1000 Ohm (default)

Series drain resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series source resistance, Rs — Faulted series source resistance**

1000 Ohm (default)

Series source resistance in faulted state.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Open circuit.

**Faulted series bulk resistance, Rb — Faulted series source resistance**

1000 Ohm (default)

Series source resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Open circuit.

**Faulted parallel drain-source conductance, Gds — Faulted parallel drain-source conductance**

1000 1/Ohm (default)

Parallel drain-source conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.



- **Enable faults** to On.
- **Failure mode** to Drain-source short.

**Faulted parallel drain-bulk conductance, Gdb — Faulted parallel drain-bulk conductance**

1000 1/Ohm (default)

Parallel drain-bulk conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Drain-bulk short or source-bulk short.

**Faulted parallel source-bulk conductance, Gsb — Faulted parallel source-bulk conductance**

1000 1/Ohm (default)

Parallel source-bulk conductance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- **Enable faults** to On.
- **Failure mode** to Drain-bulk short or source-bulk short.

**Location of fault node (percentage of total channel length from source) — Fault node location**

0 (default)

Location of the fault node, as a percentage of the total channel length from the source.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Gate oxide short conductance, G — Gate oxide short conductance**

1000 1/Ohm (default)

Gate oxide short conductance.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Model gate oxide short diode — Model option for gate oxide short diode**

No (default) | Yes

Whether to model the gate oxide short diode.

**Dependencies**

To enable this parameter, set **Enable faults** to On and **Failure mode** to Gate oxide short.

**Gate oxide short diode reverse saturation current, Is — Reverse saturation current of gate oxide short diode**

0 A (default)

Reverse saturation current of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode built-in voltage, Vbi — Built-in voltage of gate oxide short diode**

0.6 V (default)

Built-in voltage of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode ideality factor, n — Ideality factor of gate oxide short diode**

1 (default)

Ideality factor of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode zero-bias junction capacitance, Cj0 — Zero-bias junction capacitance of gate oxide short diode**

0 pF (default)

Zero-bias junction capacitance of the gate oxide short diode.

**Dependencies**

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Gate oxide short diode transit time, TT** — Transit time of gate oxide short diode  
50e-9 s (default)

Transit time of the gate oxide short diode.

#### Dependencies

To enable this parameter, set:

- **Enable faults** to On.
- **Failure mode** to Gate oxide short.
- **Model gate oxide short diode** to Yes.

**Shifted gain as percentage of unshifted** — Shifted gain as percentage of unshifted  
100 (default)

Shifted gain as percentage of unshifted.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Threshold voltage shift value, dVth** — Threshold voltage shift value  
0 V (default)

Shift value of Threshold voltage.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted channel modulation, L** — Faulted channel modulation  
0 1/V (default)

Channel modulation in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Shifted body factor as percentage of unshifted — Shifted body factor as percentage of unshifted**

100 (default)

Shifted body factor as percentage of unshifted.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Surface potential shift value, dPhib — Surface potential shift value**

0 V (default)

Shift value of surface potential.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted source ohmic resistance, Rs — Faulted source ohmic resistance**

0.0001 Ohm (default)

Transistor source resistance in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted drain ohmic resistance, Rd — Faulted drain ohmic resistance**

0.01 Ohm (default)

Transistor drain resistance in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

#### Faulted body ohmic resistance, $R_b$ — Faulted body ohmic resistance

0.001 Ohm (default)

Transistor body resistance in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Specify from a datasheet or Specify using equation parameters directly.
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

#### Faulted vector of source-gate voltages, $V_{sg}$ — Faulted vector of source-gate voltages

[-.5, 2, 3, 4, 5] V (default)

Source-gate voltages in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

#### Faulted vector of source-drain voltages, $V_{sd}$ — Faulted vector of source-drain voltages

[0, 2, 4, 8, 12, 16, 20] V (default)

Source-drain voltages in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.

- **Failure mode** to Parameter shift.

**Faulted vector of source-bulk voltages, Vsb — Faulted vector of source-bulk voltages**

[0, -4, -6, -12] V (default)

Source-bulk voltages in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent) or Lookup table (3-D, temperature dependent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated gate-source threshold voltage, Vth(Vsb) — Faulted tabulated gate-source threshold voltage**

[.4, 1.2, 1.4, 1.45] V (default)

Tabulated gate-source threshold voltage in faulted state.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated gate-source threshold voltage, Vth(Vsb,T) — Faulted tabulated gate-source threshold voltage dependent on temperature**

[.4, .35; 1.2, 1.15; 1.4, 1.35; 1.45, 1.4] V (default)

Gate-source threshold voltage in faulted state, tabulated on bulk-source voltage and temperature.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated source-drain currents, Isd(Vsg,Vsd) — Faulted tabulated source-drain currents**

[0, 0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;  
0, .0055, .008, .009, .0095, .01, .0101;

```
0, .0115, .016, .018, .019, .0195, .02;
0, .0152, .0225, .0265, .028, .0295, .03] A (default)
```

Tabulated source-drain currents in faulted state.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

#### Faulted tabulated source-drain currents, $I_{sd}(V_{sg}, V_{sd}, T)$ — Faulted tabulated source-drain currents dependent on temperature

```
zeros(5, 7, 2) A (default)
```

Source-drain currents in faulted state, tabulated on source-gate voltage, source-drain voltage, and temperature.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Three.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

#### Faulted tabulated source-drain currents at zero source-bulk voltage, $I_{sd}(V_{sg}, V_{sd})$ — Faulted tabulated source-drain currents at zero source-bulk voltage

```
[0, 0, 0, 0, 0, 0, 0; 0, .003, .0038, .0043, .0046, .0048, .0049;
0, .0055, .008, .009, .0095, .01, .0101;
0, .0115, .016, .018, .019, .0195, .02;
0, .0152, .0225, .0265, .028, .0295, .03] A (default)
```

Tabulated source-drain currents in faulted state, when the source-bulk voltage is zero.

#### Dependencies

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (2-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted tabulated source-drain currents at zero source-bulk voltage,  $I_{sd}(V_{sg}, V_{sd}, T)$  — Faulted tabulated source-drain currents at zero source-bulk voltage, dependent on temperature**

zeros(5, 7, 2) A (default)

Tabulated source-drain currents in faulted state, when the source-bulk voltage is zero, tabulated on source-gate voltage, source-drain voltage, and temperature.

**Dependencies**

To enable this parameter, set:

- In the **Main** setting, **Number of terminals** to Four.
- In the **Main** setting, **Parameterization** to Lookup table (3-D, temperature independent).
- **Enable faults** to On.
- **Failure mode** to Parameter shift.

**Faulted gate-source capacitance as percentage of unfaulted — Faulted gate-source capacitance as percentage of unfaulted**

100 (default)

Gate-source capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-drain capacitance as percentage of unfaulted — Faulted gate-drain capacitance as percentage of unfaulted**

100 (default)

Gate-drain capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:



**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-bulk capacitance as percentage of unfaulted — Faulted gate-bulk capacitance as percentage of unfaulted**

100 (default)

Gate-bulk capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify fixed gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted drain-source capacitance as percentage of unfaulted — Faulted drain-source capacitance as percentage of unfaulted**

100 (default)

Drain-source capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set **Enable faults** to On.

**Condition Set Two**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Open circuit, Drain-source short, Drain-bulk short or source-bulk short, or Gate oxide short.

**Faulted gate-drain oxide capacitance as percentage of unfaulted — Faulted gate-drain oxide capacitance as percentage of unfaulted**

100 (default)

Gate-drain oxide capacitance in faulted state as percentage of unfaulted.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:
  - **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
  - **Gate-drain charge-voltage linearity** to Gate-drain charge function is non linear.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted drain-gate voltage at which oxide capacitance becomes active — Faulted drain-gate voltage at which oxide capacitance becomes active**

-0.5 V (default)

Drain-gate voltage in faulted state at which oxide capacitance becomes active.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:
  - **Parameterization** to Specify fixed input, reverse transfer, and output capacitance or Specify fixed gate-source, gate-drain, and drain-source capacitance.
  - **Gate-drain charge-voltage linearity** to Gate-drain charge function is non linear.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted input capacitance, Ciss(Vds) – Faulted input capacitance**

[720, 700, 590, 470, 390, 310] pF (default)

Input capacitance in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted reverse transfer capacitance, Crss(Vds) – Faulted reverse transfer capacitance**

[450, 400, 300, 190, 95, 55] pF (default)

Reverse transfer capacitance in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted output capacitance, Coss(Vds) – Faulted output capacitance**

[900, 810, 690, 420, 270, 170] pF (default)

Output capacitance in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted corresponding source-drain voltages, Vsd — Faulted corresponding source-drain voltages**

[.1, .3, 1, 3, 10, 30] V (default)

Corresponding drain-source voltages in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance or Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted source-gate voltages, Vsg, for tabulated capacitances — Faulted source-gate voltages for tabulated capacitances**

0 V (default)

Source-gate voltages for tabulated capacitances, in faulted state.

**Dependencies**

To enable this parameter, meet one of these sets of conditions:

**Condition Set One**

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated input, reverse transfer, and output capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Condition Set Two**

- In the **Main** setting, **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

- **Gate-bulk and gate-source charge-voltage linearity** to Gate-bulk and gate-source capacitance change instantly or Gate-bulk and gate-source capacitance change gradually.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-source capacitance,  $C_{gs}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[270, 300, 290, 280, 295, 255] pF (default)

Gate-source capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-drain capacitance,  $C_{gd}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[450, 400, 300, 190, 95, 55] pF (default)

Gate-drain capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted gate-bulk capacitance,  $C_{gb}(V_{gs}, V_{ds})$  — Faulted gate-source capacitance**  
[270, 300, 290, 280, 295, 255] pF (default)

Gate-bulk capacitance in faulted state, tabulated with gate-source voltage and drain-source voltage.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Number of terminals** to Four.
- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.

- **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted corresponding source-gate voltages,  $V_{sg}$  — Faulted corresponding source-gate voltages**

0 V (default)

Corresponding source-gate voltages in faulted state.

**Dependencies**

To enable this parameter,

- In the **Capacitance** setting, set:
  - **Parameterization** to Specify tabulated gate-source, gate-drain, and drain-source capacitance.
  - **Gate-bulk and gate-source charge-voltage linearity** to Separate gate-bulk and gate-source capacitance.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode reverse saturation current,  $I_s$  — Faulted diode reverse saturation current**

0 A (default)

Reverse saturation current of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode built-in voltage,  $V_{bi}$  — Faulted diode built-in voltage**

0.6 V (default)

Built-in voltage of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.

- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted diode ideality factor, n – Faulted diode ideality factor**

1 (default)

Ideality factor of the diode in faulted state.

#### **Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted diode zero-bias junction capacitance as percentage of unfaulted – Faulted diode zero-bias junction capacitance as percentage of unfaulted**

100 (default)

Faulted diode zero-bias junction capacitance as percentage of unfaulted.

#### **Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted diode transit time, TT – Faulted diode transit time**

50e-9 s (default)

Transit time of the diode in faulted state.

#### **Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set **Model body diode** to Exponential.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

#### **Faulted diode forward currents, If(Vf) – Faulted diode forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2] A (default)

Forward currents of the diode in faulted state, tabulated with the forward voltages.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify using equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(V_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to 0n.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f$  – Faulted diode forward voltages**  
[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Forward voltages of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(V_f)$  form or Table in  $I_f(T_j, V_f)$  form
- In the **Faults** setting, set:
  - **Enable faults** to 0n.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f(I_f)$  – Faulted diode forward voltages**  
[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85] V (default)

Forward voltages of the diode in faulted state, tabulated with the forward currents.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Specify from a datasheet, Specify using equation parameters directly, or Lookup table (2-D, temperature independent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $V_f(I_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to 0n.
  - **Failure mode** to Parameter shift.



**Faulted diode forward currents,  $I_f$  – Faulted diode forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Forward currents of the diode in faulted state.

**Dependencies**

To enable this parameter,

- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $V_f(I_f)$  form and Table in  $V_f(T_j, I_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward currents,  $I_f(T_j, V_f)$  – Faulted diode forward currents with tabulated inputs**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default)

Forward currents of the diode in faulted state, tabulated with the forward voltages and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).
- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $I_f(T_j, V_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

**Faulted diode forward voltages,  $V_f(T_j, I_f)$  – Faulted diode forward voltages with tabulated inputs**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default)

Forward voltages of the diode in faulted state, tabulated with the forward currents and junction temperatures.

**Dependencies**

To enable this parameter,

- In the **Main** setting, set **Parameterization** to Lookup table (3-D, temperature dependent).

- In the **Body Diode** setting, set:
  - **Model body diode** to Tabulated I-V curve.
  - **Table type** to Table in  $V_f(T_j, I_f)$  form.
- In the **Faults** setting, set:
  - **Enable faults** to On.
  - **Failure mode** to Parameter shift.

## Version History

Introduced in R2008a

## References

- [1] Shichman, H. and D. A. Hodges. "Modeling and simulation of insulated-gate field-effect transistor switching circuits." *IEEE J. Solid State Circuits*. SC-3, 1968.
- [2] Van Langevelde, R., A. J. Scholten, and D. B. M. Klaassen. "Physical Background of MOS Model 11. Level 1101." *Nat.Lab. Unclassified Report 2003/00239*. April 2003.
- [3] Oh, S-Y., D. E. Ward, and R. W. Dutton. "Transient analysis of MOS transistors." *IEEE J. Solid State Circuits*. SC-15, pp. 636-643, 1980.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

N-Channel MOSFET

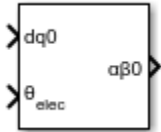
### Functions

semiconductorSubcircuit2lookup

# Park to Clarke Angle Transform

Implement  $dq0$  to  $\alpha\beta0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

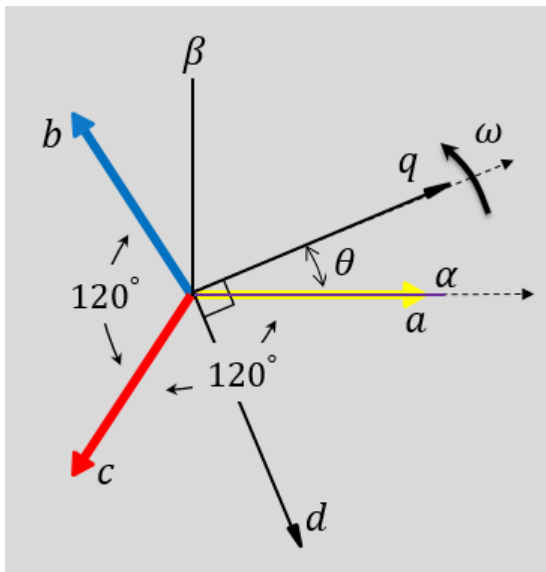


## Description

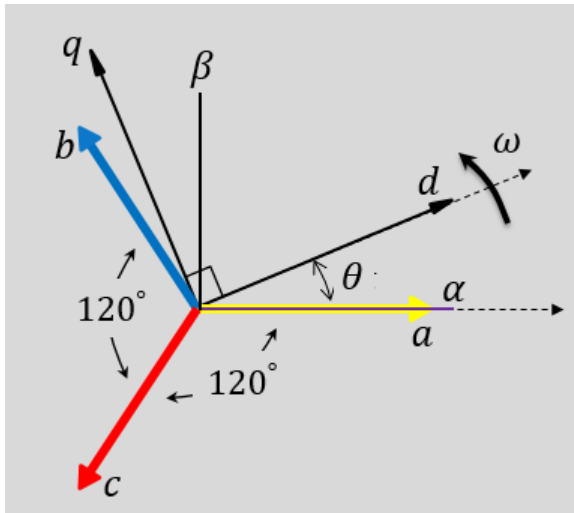
The Park to Clarke Angle Transform block converts the direct, quadrature, and zero components in a rotating reference frame to alpha, beta, and zero components in a stationary reference frame. For balanced systems, the zero components are equal to zero.

You can configure the block to align the phase  $a$ -axis of the three-phase system to either the  $q$ - or  $d$ -axis of the rotating reference frame at time,  $t = 0$ . The figures show the direction of the magnetic axes of the stator windings in the three-phase system, a stationary  $\alpha\beta0$  reference frame, and a rotating  $dq0$  reference frame where:

- The  $a$ -axis and the  $q$ -axis are initially aligned.



- The  $a$ -axis and the  $d$ -axis are initially aligned.

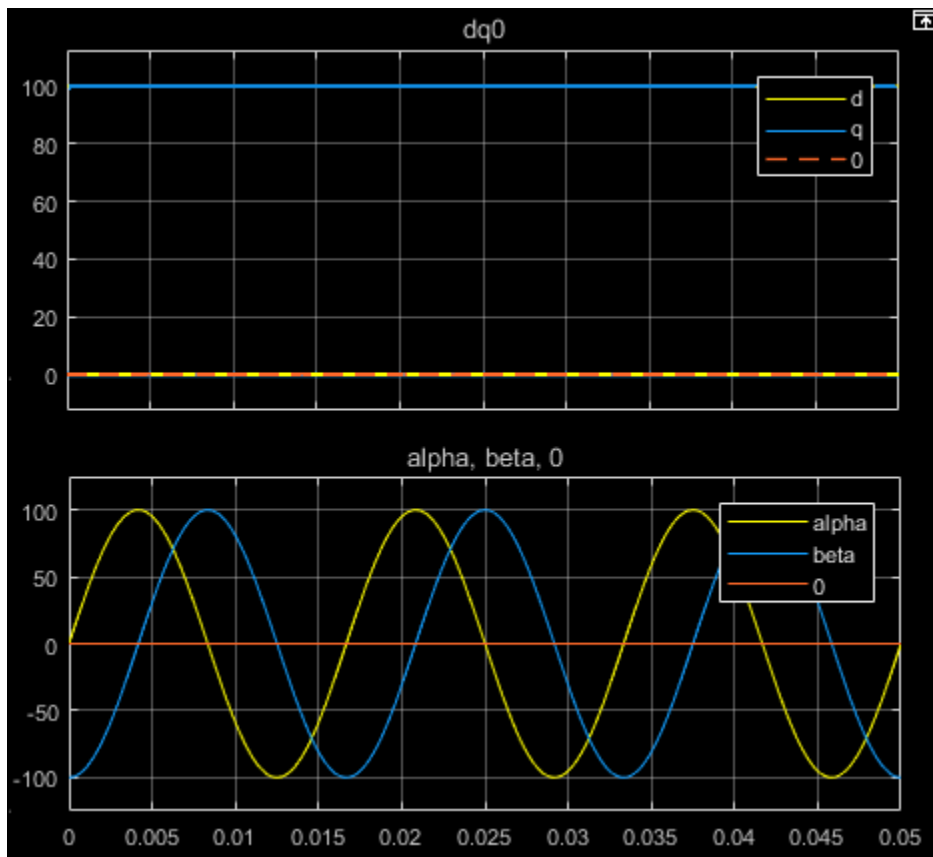


In both cases, the angle  $\theta = \omega t$ , where

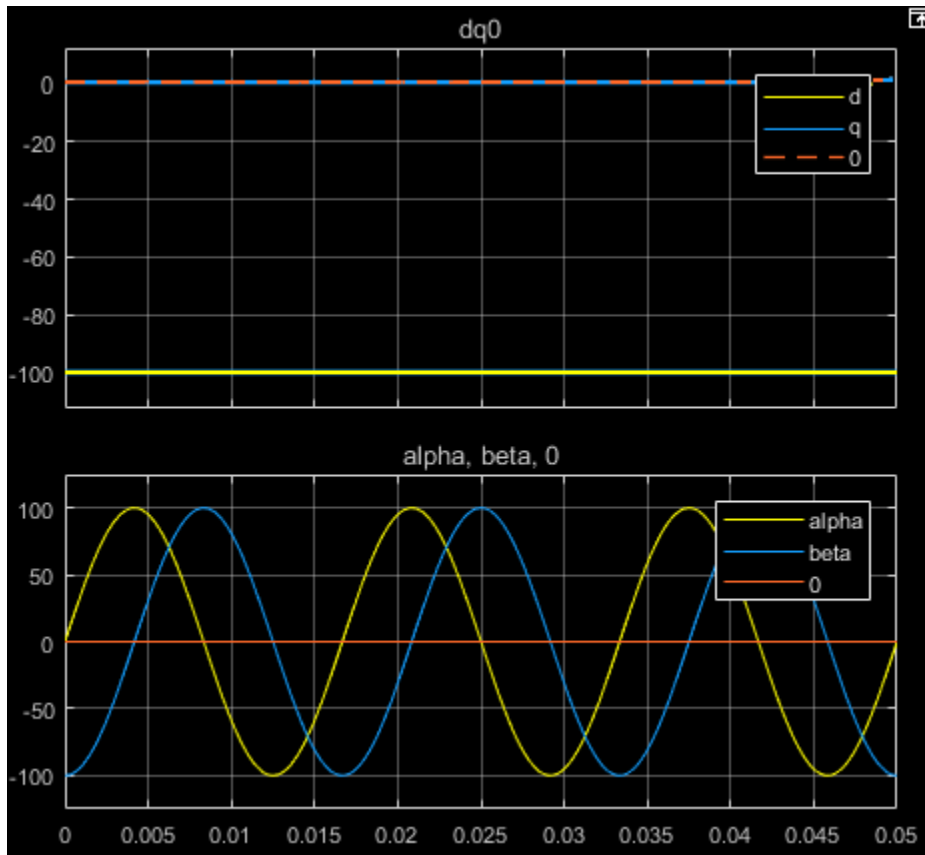
- $\theta$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment.
- $\omega$  is the rotational speed of the  $d$ - $q$  reference frame.
- $t$  is the time, in s, from the initial alignment.

The figures show the time-response of the individual components of equivalent balanced  $dq0$  and  $\alpha\beta0$  for an:

- Alignment of the  $a$ -phase vector to the  $q$ -axis



- Alignment of the  $a$ -phase vector to the  $d$ -axis



### Equations

The Park to Clarke Angle Transform block implements the transform for an  $a$ -phase to  $q$ -axis alignment as

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 0 \\ -\cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix}$$

where:

- $d$  and  $q$  are the direct-axis and quadrature-axis components of the two-axis system in the rotating reference frame.
- $0$  is the zero component.
- $\alpha$  and  $\beta$  are the alpha-axis and beta-axis components of the two-phase system in the stationary reference frame.

For an  $a$ -phase to  $d$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ q \\ 0 \end{bmatrix}$$

## Ports

### Input

#### **$dq0$ — $d$ - $q$ axis and zero components**

vector

Direct-axis and quadrature-axis components and the zero component of the system in the rotating reference frame.

Data Types: `single` | `double`

#### **$\theta_{abc}$ — Rotational angle**

scalar | in radians

Angular position of the rotating reference frame. The value of this parameter is equal to the polar distance from the vector of the a-phase in the  $abc$  reference frame to the initially aligned axis of the  $dq0$  reference frame.

Data Types: `single` | `double`

### Output

#### **$\alpha\beta0$ — $\alpha$ - $\beta$ axis and zero components**

vector

Alpha-axis component,  $\alpha$ , beta-axis component,  $\beta$ , and zero component of the two-phase system in the stationary reference frame.

Data Types: `single` | `double`

## Parameters

#### **Phase-a axis alignment — $dq0$ reference frame alignment**

Q-axis (default) | D-axis

Align the  $a$ -phase vector of the  $abc$  reference frame to the  $d$ - or  $q$ -axis of the rotating reference frame.

## Version History

Introduced in R2017b

## References

- [1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscatawy, NJ: Wiley-IEEE Press, 2013.

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform



# Passive Harmonic Filter (Three-Phase)

Harmonic current filter

**Library:** Simscape / Electrical / Passive



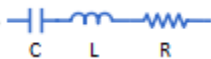
## Description

The Passive Harmonic Filter (Three-Phase) block suppresses system harmonic currents and decreases voltage distortion by providing low-impedance paths for the harmonics. At the rated frequency, the passive shunt filters are capacitive and provide reactive power, which can improve electrical power factor.

Each of the four models for the block corresponds to an option for the **Filter type** parameter:

- “Band-Pass Filter, Single Tuned” on page 1-1299
- “Band-Pass Filter, Double Tuned” on page 1-1300
- “High-Pass Filter, Second-Order” on page 1-1300
- “High-Pass Filter, C-type” on page 1-1301

### Band-Pass Filter, Single Tuned



At the tuned frequency, LC resonance occurs and the impedance of the filter reaches its minimum, which equals the value of the resistance.

The filter tuned frequency is defined by this equation:

$$f_1 = nf_0 = \frac{1}{2\pi\sqrt{LC}},$$

where  $f_1$  and  $f_0$  are the tuned and fundamental frequency,  $n$  is the harmonic order of the tuned frequency,  $L$  is the inductance and  $C$  is the capacitance.

The quality factor is defined as the ratio between the inductive or capacitive reactance at the tuned frequency and the resistance, as described by this equation:

$$Q_f = \frac{X_{LN}}{R} = \frac{X_{CN}}{R},$$

where:

- $R$  is the resistance.

- $X_{LN}$  is the impedance of the inductor at the tuned frequency  $X_{LN} = 2\pi f_1 L$ .
- $X_{CN}$  is the impedance of the capacitor at the tuned frequency  $X_{CN} = \frac{1}{2\pi f_1 C}$ .

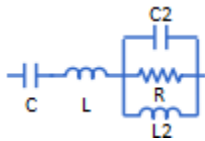
Higher quality factor values result in sharper frequencies. However, this produces high-power dissipation at the base frequency due to a relative low resistance.

The rated reactive power is given by:

$$Q_r = \frac{V^2}{X_{L0} - X_{C0}} = \frac{V^2}{\frac{X_{C0}}{n^2} - X_{C0}} = \frac{V^2}{X_{C0}} \cdot \frac{n^2}{n^2 - 1},$$

where  $Q_r$  is the rated reactive power for one phase and  $V$  is the branch voltage in root mean square.

### Band-Pass Filter, Double Tuned



A double-tuned filter has two tuned frequencies,  $f_1$  and  $f_2$ , where  $f_1 = n_1 f_0$  and  $f_2 = n_2 f_0$ .

The double-tuned filter comprises a series LC and a parallel RCL circuit, each tuned at frequencies  $f_s$  and  $f_p$ , close to the mean geometric frequency of  $f_1$  and  $f_2$ , which are represented by the equation:

$$f_m = \sqrt{f_1 f_2} = \sqrt{f_s f_p},$$

where

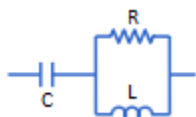
$$f_s = \frac{1}{2\pi\sqrt{LC}}$$

$$f_p = \frac{1}{2\pi\sqrt{L_2 C_2}}.$$

The quality factor of this filter is defined as the quality factor of the parallel R and L elements at the mean geometric frequency,  $f_m$ :

$$Q_f = \frac{R}{2\pi f_m L_2}.$$

### High-Pass Filter, Second-Order



The second order high-pass filter shunts a large percentage of the harmonics at and above the tuned frequency. The filter is designed to have a flat impedance for high-order harmonics.

The tuned frequency is described by this equation:

$$f_1 = nf_0 = \frac{1}{2\pi\sqrt{LC}}.$$

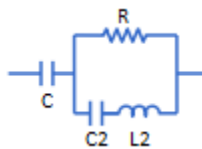
The quality factor is the reciprocal of the band-pass, single-tuned filter:

$$Q_f = \frac{R}{X_{LN}} = \frac{R}{X_{CN}}.$$

The rated reactive power is the same of the band-pass, single-tuned filter:

$$Q_r = \frac{V^2}{X_{L0} - X_{C0}} = \frac{V^2}{\frac{X_{C0}}{n^2} - X_{C0}} = \frac{V^2}{X_{C0}} \cdot \frac{n^2}{n^2 - 1}.$$

### High-Pass Filter, C-type



Compared to the single-tuned version, the C-type, high-pass filter has lower losses at the fundamental frequency, because the capacitor and inductor are parallel with the resistor.

To prevent fundamental currents from passing through the resistor, the resonance frequency of  $L_2$  and  $C_2$  is tuned to the fundamental frequency:

$$f_0 = \frac{1}{2\pi\sqrt{L_2 C_2}}.$$

The quality factor is calculated using this equation:

$$Q_f = \frac{R}{2\pi f_0 n L_2}.$$

### Values of RCL Filter Components

	Single-Tuned	Double-Tuned	Second-Order, High-Pass	C-type, High-Pass
R	$\frac{1}{2\pi f_0 C Q_f}$	$2\pi f_m L_2 Q_f$	$2\pi n f_0 L Q_f$	$Q_f (2\pi n f_0 L_2)$
L	$\frac{1}{C(2\pi n f_0)^2}$	$\frac{V^2}{Q_r \pi f_0 (n_1^2 + n_2^2 + 2)}$	$\frac{1}{C(2\pi n f_0)^2}$	None

	Single-Tuned	Double-Tuned	Second-Order, High-Pass	C-type, High-Pass
C	$\frac{Q_r}{2\pi f_0 V^2} \cdot \frac{n^2 - 1}{n^2}$	$\frac{Q_r}{4\pi f_0 V^2} \left( \frac{n_1^2 - 1}{n_1^2} + \frac{n_2^2 - 1}{n_2^2} \right)$	$\frac{Q_r}{2\pi f_0 V^2} \cdot \frac{n^2 - 1}{n^2}$	$\frac{Q_r}{2\pi f_0 V^2}$
L <sub>2</sub>	None	$\frac{\left(1 - \frac{f_1^2}{f_s^2}\right) \left(1 - \frac{f_1^2}{f_p^2}\right)}{C(2\pi f_1)^2}$ where $f_s = \frac{1}{2\pi\sqrt{LC}}$ , $f_p = \frac{f_1 f_2}{f_s}$	None	$\frac{1}{C_2(2\pi f_0)^2}$
C <sub>2</sub>	None	$\frac{1}{L_2(2\pi f_p)^2}$	None	$C(n^2 - 1)$

## Ports

### Conserving

~ – **Three-phase voltage**  
electrical

Expandable three-phase port associated with the voltage.

## Parameters

### Modeling option – Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Rated reactive power – Reactive power

100e6 V\*A (default)

Reactive power at rated frequency and voltage, in V\*A.

### Rated voltage (phase-to-phase RMS) – Phase-to-phase RMS Voltage

4160 V (default)

Rated phase-to-phase root mean squared (RMS) voltage, in V.

**Rated frequency — Rated frequency**

60 Hz (default)

Fundamental frequency, in Hz.

**Quality factor — Quality factor**

10 (default)

Index of the sharpness of the tuned frequency.

**Filter type — Filter model**

Band-pass filter, single-tuned (default) | Band-pass filter, double-tuned | High-pass filter, second-order | High-pass filter, C-type

Type of harmonic filter.

**Dependencies**

The **Tuned frequency** parameter becomes visible if you select Band-pass filter, single-tuned, High-pass filter, second-order, or High-pass filter, C-type for the **Filter type** parameter.

If you select Band-pass filter, double-tuned for the **Filter type** parameter, the **Tuned frequency 1** and **Tuned frequency 2** parameters become visible.

**Filter connection — Connection type**

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta

Type of connection

**Tuned frequency — Tuned frequency**

5 \* 60 Hz (default)

Tuned frequency for the filter, in Hz.

**Dependencies**

This parameter is visible only when you select Band-pass filter, single-tuned, High-pass filter, second-order, or High-pass filter, C-type for the **Filter type** parameter.

**Tuned frequency 1 — First tuned frequency**

7 \* 60 Hz (default)

First tuned frequency, in Hz.

**Dependencies**

This parameter is visible only when you select Band-pass filter, double-tuned for the **Filter type** parameter.

**Tuned frequency 2 — Second tuned frequency**

11 \* 60 Hz (default)

Second tuned frequency, in Hz.

**Dependencies**

This parameter is visible only when you select Band-pass filter, double-tuned for the **Filter type** parameter.

**Version History**

**Introduced in R2019b**

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

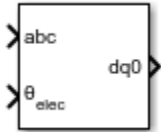
**See Also**

Floating Neutral (Three-Phase) | Grounded Neutral (Three-Phase) | Neutral Port (Three-Phase) | Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase) | Wye-Connected Load

## Park Transform

Implement  $abc$  to  $dq0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms

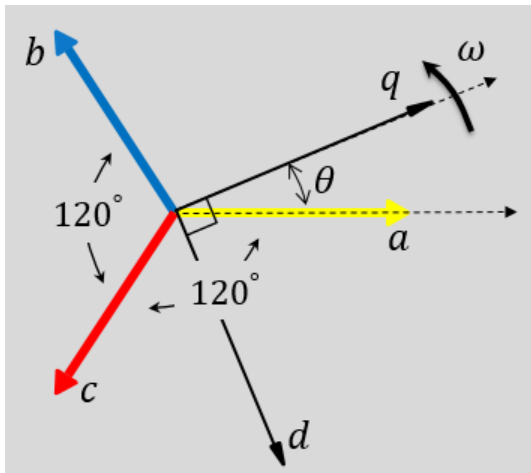


### Description

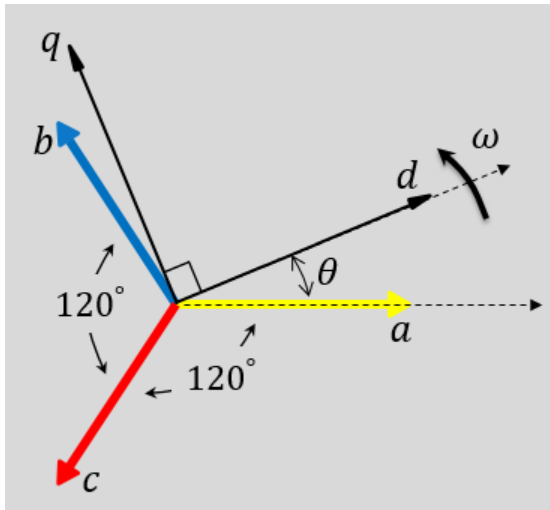
The Park Transform block converts the time-domain components of a three-phase system in an  $abc$  reference frame to direct, quadrature, and zero components in a rotating reference frame. The block can preserve the active and reactive powers with the powers of the system in the  $abc$  reference frame by implementing an invariant version of the Park transform. For a balanced system, the zero component is equal to zero.

You can configure the block to align the  $a$ -axis of the three-phase system to either the  $d$ - or  $q$ -axis of the rotating reference frame at time,  $t = 0$ . The figures show the direction of the magnetic axes of the stator windings in an  $abc$  reference frame and a rotating  $dq0$  reference frame where:

- The  $a$ -axis and the  $q$ -axis are initially aligned.



- The  $a$ -axis and the  $d$ -axis are initially aligned.



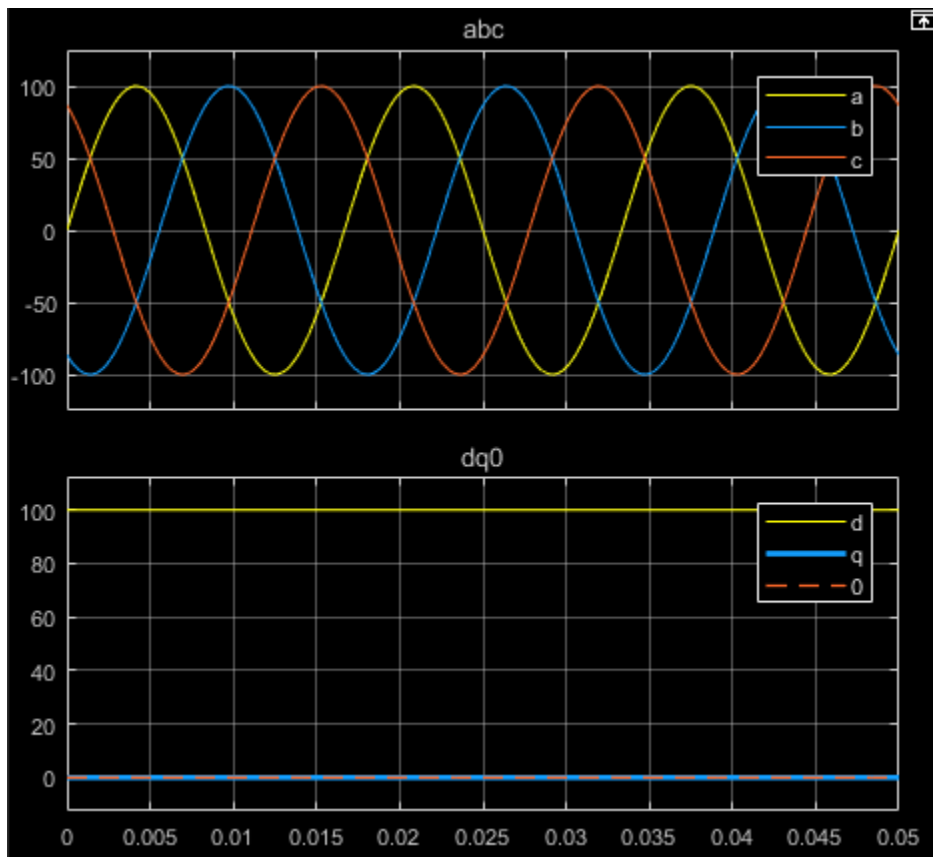
In both cases, the angle  $\theta = \omega t$ , where:

- $\theta$  is the angle between the  $a$  and  $q$  axes for the  $q$ -axis alignment or the angle between the  $a$  and  $d$  axes for the  $d$ -axis alignment.
- $\omega$  is the rotational speed of the  $d$ - $q$  reference frame.
- $t$  is the time, in s, from the initial alignment.

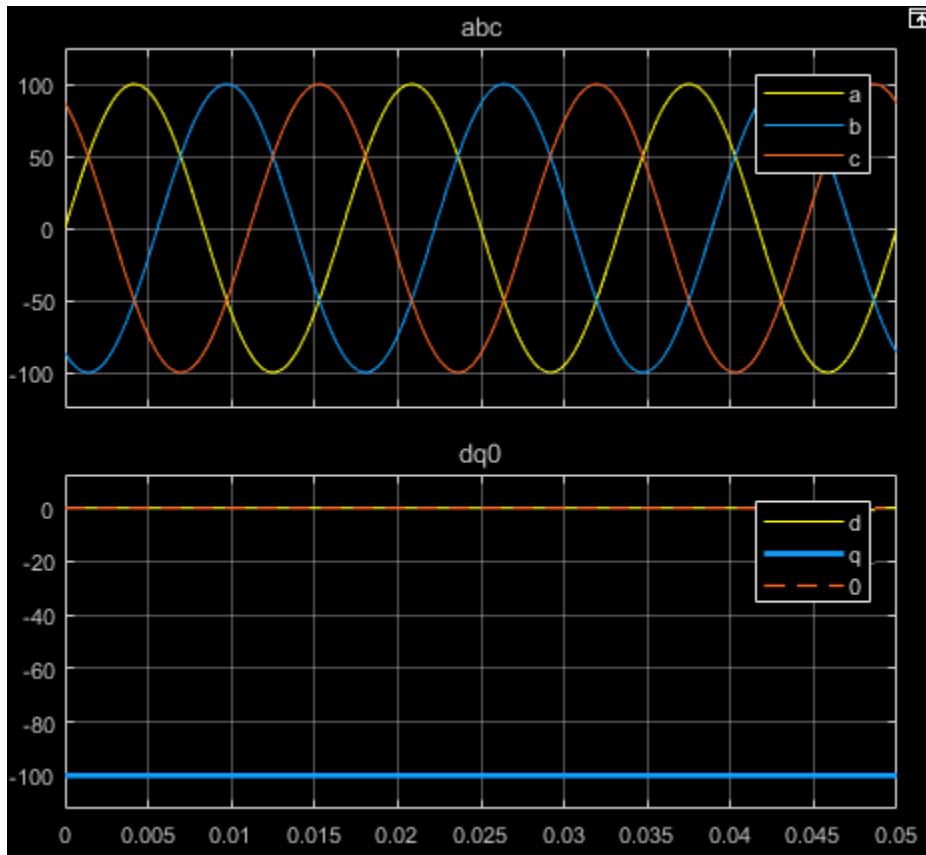
The figures show the time-response of the individual components of equivalent balanced  $abc$  and  $dq0$  for an:

- Alignment of the  $a$ -phase vector to the  $q$ -axis





- Alignment of the  $a$ -phase vector to the  $d$ -axis



### Equations

The Park Transform block implements the transform for an  $a$ -phase to  $q$ -axis alignment as

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

where:

- $a$ ,  $b$ , and  $c$  are the components of the three-phase system in the  $abc$  reference frame.
- $d$  and  $q$  are the components of the two-axis system in the rotating reference frame.
- $0$  is the zero component of the two-axis system in the stationary reference frame.

For a power invariant  $a$ -phase to  $q$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \frac{\sqrt{1}}{2} & \frac{\sqrt{1}}{2} & \frac{\sqrt{1}}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

For an  $a$ -phase to  $d$ -axis alignment, the block implements the transform using this equation:

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

The block implements a power invariant  $a$ -phase to  $d$ -axis alignment as

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

## Ports

### Input

#### **abc — a-, b-, and c-phase components**

vector

Components of the three-phase system in the  $abc$  reference frame.

Data Types: single | double

#### **$\theta_{abc}$ — Rotational angle**

scalar | in radians

Angular position of the rotating reference frame. The value of this parameter is equal to the polar distance from the vector of the a-phase in the  $abc$  reference frame to the initially aligned axis of the  $dq0$  reference frame.

Data Types: single | double

### Output

#### **dq0 — d-q axis and zero components**

vector

Direct-axis and quadrature-axis components and the zero component of the system in the rotating reference frame.

Data Types: single | double

## Parameters

#### **Power Invariant — Power invariant transform**

off (default) | on

Option to preserve the active and reactive power of the  $abc$  reference frame.

#### **Phase-a axis alignment — dq0 reference frame alignment**

Q-axis (default) | D-axis

Align the  $a$ -phase vector of the  $abc$  reference frame to the  $d$ - or  $q$ -axis of the rotating reference frame.

## **Version History**

**Introduced in R2017b**

## **References**

[1] Krause, P., O. Wasynczuk, S. D. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ: Wiley-IEEE Press, 2013.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform

# Peltier Device

Electrothermal converter

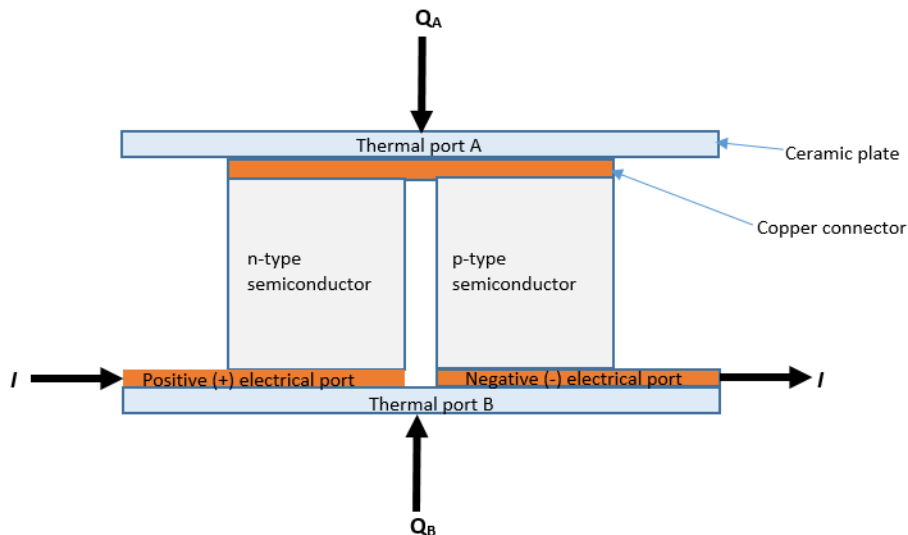
**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Peltier Device block represents a converter between the electrical and thermal energy:

- With no current flowing, if the temperature presented at thermal port **B** is greater than the temperature presented at thermal port **A**, then there is a positive potential difference measured from the positive to the negative electrical port.
- When the block acts as a cooling device, a positive current causes heat to flow from port **A** to port **B**, cooling port **A** relative to port **B**.



The defining equations are:

$$Q_A = \alpha T_A I - \frac{1}{2} I^2 R + K(T_A - T_B)$$

$$Q_B = -\alpha T_B I - \frac{1}{2} I^2 R + K(T_B - T_A)$$

$$W = VI$$

$$W + Q_A + Q_B = 0$$

where:

- $Q_A$  is heat flow into port **A**.
- $Q_B$  is heat flow into port **B**.
- $T_A$  is port **A** temperature.
- $T_B$  is port **B** temperature.
- $W$  is electrical power (positive when flowing into the block).
- $V$  is potential difference across the + and - ports.
- $I$  is electrical current, positive from + to - port.
- $R$  is total electrical resistance.
- $\alpha$  is Seebeck coefficient.
- $K$  is thermal conductance.

Substituting for powers and dividing all terms by current gives the electrical equation:

$$V = \alpha(T_B - T_A) + IR$$

The block has a logging variable `power_dissipated` (**Dissipated power**). This variable reports the DC electrical power, that is, the average electrical power over one AC cycle if you drive the device with an AC source. In terms of equations it is equal to the instantaneous value of  $I^2R$  and is useful in a cooling application to indicate the nonproductive portion of the heat flow. In a heating application, the interpretation of dissipated power is less useful.

## Ports

### Conserving

#### + – Positive electrical port

electrical

Electrical conserving port associated with the positive terminal.

#### - – Negative electrical port

electrical

Electrical conserving port associated with the negative terminal.

#### A – Thermal port

thermal

Thermal conserving port. When the block acts as a cooling device, a positive current causes heat to flow from port **A** to port **B**, cooling port **A** relative to port **B**.

#### B – Thermal port

thermal

Thermal conserving port.

## Parameters

### Electrical

**Seebeck coefficient – Thermoelectric sensitivity coefficient**

220e-6 V/K (default)

Measure of the magnitude of an induced thermoelectric voltage in response to a temperature difference across that material.

**Total electrical resistance – Resistance between the electrical ports**

0.02  $\Omega$  (default)

Total electrical resistance between the + and - ports.

### Thermal

**Thermal conductance – Conductance between the thermal ports**

1.5e-3 W/K (default)

Thermal conductance between the A and B ports.

## Version History

Introduced in R2018b

## Extended Capabilities

**C/C++ Code Generation**

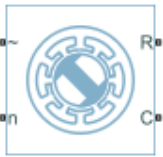
Generate C and C++ code using Simulink® Coder™.

## See Also

# PMSM

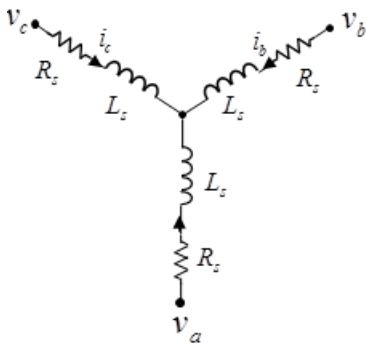
Permanent magnet synchronous motor with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



## Description

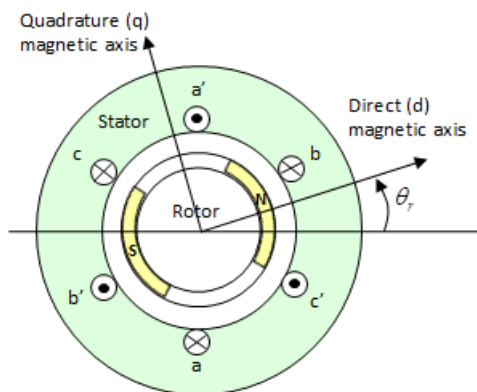
The PMSM block models a permanent magnet synchronous motor with a three-phase wye-wound stator. The figure shows the equivalent electrical circuit for the stator windings.



You can choose different built-in parameterizations for this block. For more information, see the “Predefined Parameterization” on page 1-1321 section.

## Motor Construction

This figure shows the motor construction with a single pole-pair on the rotor.





Permanent magnets generate a rotor magnetic field that creates a sinusoidal rate of change of flux based on the rotor angle.

For the axes convention, when you set the **Rotor angle definition** parameter to **Angle between the a-phase magnetic axis and the d-axis**, the *a*-phase and permanent magnet fluxes align when the rotor mechanical angle,  $\theta_r$ , is zero. When you set the **Rotor angle definition** parameter to **Angle between the a-phase magnetic axis and the q-axis**, the rotor mechanical angle is the angle between the *a*-phase magnetic axis and the rotor *q*-axis.

## Equations

Voltages across the stator windings are

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ , and  $\frac{d\psi_c}{dt}$  are the rates of change for the magnetic flux in each stator winding.

The permanent magnet and the three windings contribute to the total flux linking each winding. The total flux is

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix},$$

where:

- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the total fluxes linking each stator winding.
- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ , and so on, are the mutual inductances of the stator windings.
- $\psi_{am}$ ,  $\psi_{bm}$ , and  $\psi_{cm}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of the rotor electrical angle and defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3))$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_e + \pi/6))$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/6 - 2\pi/3))$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_e + \pi/6 + 2\pi/3))$$

where:

- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the  $d$ -axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the  $q$ -axis.
- $L_s$  is the stator self-inductance per phase. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the amount the self-inductance and mutual inductance fluctuate with the changing of the rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The permanent magnet flux linking winding  $a$  is a maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore, the linked motor flux is defined by:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_e - 2\pi/3) \\ \psi_m \cos(\theta_e + 2\pi/3) \end{bmatrix}$$

where  $\psi_m$  is the permanent magnet flux linkage.

### Simplified Electrical Equations

Applying Park's transformation to the block electrical equations produces an expression for torque that is independent of the rotor angle.

Park's transformation is defined by:

$$P = \frac{2}{3} \begin{bmatrix} \cos \theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin \theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

where  $\theta_e$  is the electrical angle defined as  $N\theta_r$ .  $N$  is the number of pole pairs.

Using Park's transformation on the stator winding voltages and currents transforms them to the dq0 frame, which is independent of the rotor angle:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

Applying Park's transformation to the first two electrical equations produces these equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N \omega i_q L_q$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N \omega (i_d L_d + \psi_m)$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

$$T = \frac{3}{2} N (i_q (i_d L_d + \psi_m) - i_d i_q L_q)$$

where:

- $L_d = L_s + M_s + 3/2 L_m$ .  $L_d$  is the stator  $d$ -axis inductance.

- $L_q = L_s + M_s - 3/2 L_m$ .  $L_q$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 2M_s$ .  $L_0$  is the stator zero-sequence inductance.
- $\omega$  is the rotor mechanical rotational speed.
- $N$  is the number of rotor permanent magnet pole pairs.
- $T$  is the rotor torque. Torque flows from the motor case (block physical port C) to the motor rotor (block physical port R).

The PMSM block uses the original, non-orthogonal implementation of the Park transform. If you try to apply the alternative implementation, you get different results for the dq0 voltage and currents.

### Alternative Flux Linkage Parameterization

You can parameterize the motor using the back EMF or torque constants which are more commonly given on motor datasheets by using the **Permanent magnet flux linkage** parameter.

The back EMF constant is the peak voltage induced by the permanent magnet in the per-unit rotational speed of each of the phases. The relationship between the peak permanent magnet flux linkage and the back EMF is:

$$k_e = N \psi_m .$$

The back EMF,  $e_{ph}$ , for one phase is:

$$e_{ph} = k_e \omega .$$

The torque constant is the peak torque induced by the per-unit current of each of the phases. It is numerically identical in value to the back EMF constant when both are expressed in SI units:

$$k_t = N \psi_m .$$

When  $L_d=L_q$  and the currents in all three phases are balanced, the combined torque  $T$  is:

$$T = \frac{3}{2}k_t i_q = \frac{3}{2}k_t I_{pk},$$

where  $I_{pk}$  is the peak current in any of the three windings.

The factor 3/2 follows from this being the steady-state sum of the torques from all phases. Therefore, the torque constant  $k_t$  can also be:

$$k_t = \frac{2}{3} \left( \frac{T}{I_{pk}} \right),$$

where  $T$  is the measured total torque when testing with a balanced three-phase current with peak line voltage  $I_{pk}$ . The RMS line current is:

$$k_t = \frac{\sqrt{2}}{3} \frac{T}{i_{line, rms}}.$$

### Calculating Iron Losses

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. The iron losses model, which is based on the work of Mellor [3].

The term representing the main magnetizing path depends on the induced RMS line-to-neutral stator voltage,  $V_{m_{rms}}$ :

$$P_{OC}(V_{m_{rms}}) = \frac{a_h}{k} V_{m_{rms}} + \frac{a_j}{k^2} V_{m_{rms}}^2 + \frac{a_{ex}}{k^{1.5}} V_{m_{rms}}^{1.5}$$

This is the dominant term during no-load operation.  $k$  is the back emf constant relating RMS volts per Hz. It is defined as  $k = V_{m_{rms}}/f$ , where  $f$  is the electrical frequency. The first term on the right-hand

side is the magnetic hysteresis loss, the second is the eddy current loss and the third is the excess loss. The three coefficients appearing on the numerators are derived from the values that you provide for the open-circuit hysteresis, eddy, and excess losses.

The term representing the cross-tooth tip path becomes important when a demagnetizing field is set up and can be determined from a finite element analysis short-circuit test. It depends on the RMS emf associated with the cross-tooth tip flux,  $V_{d_{rms}}^*$ :

$$P_{SC}(V_{d_{rms}}^*) = \frac{b_h}{k} V_{d_{rms}}^* + \frac{b_j}{k^2} V_{d_{rms}}^{*2} + \frac{b_{ex}}{k^{1.5}} V_{d_{rms}}^{*1.5}$$

The three numerator terms are derived from the values you provide for the short-circuit hysteresis, eddy, and excess losses.

### Predefined Parameterization

There are multiple available built-in parameterizations for the PMSM block.

This pre-parameterization data allows you to set up the block to represent components by specific suppliers. The parameterizations of these permanent magnet synchronous motors match the manufacturer data sheets. To load a predefined parameterization, double-click the PMSM block, click the **<click to select>** hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### ~ — Three-phase port

electrical

Expandable three-phase port.

#### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### a — a-phase

electrical

Electrical conserving port associated with *a*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### b — b-phase

electrical

Electrical conserving port associated with *b*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### c — c-phase

electrical

Electrical conserving port associated with *c*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### n — Neutral phase

electrical

Electrical conserving port associated with the neutral phase.

#### Dependencies

To enable this port, set **Winding Type** to Wye-wound and **Zero sequence** to Include.

#### R — Motor rotor

mechanical

Mechanical rotational conserving port associated with the motor rotor.



**C – Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

**HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC – Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR – Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Selected part – Predefined parameterization option**

&lt;click to select&gt; (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see “Load Predefined Parameterizations”.

**Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Electrical connection – Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Winding type – Windings configuration**

Wye-wound (default) | Delta-wound

Select the configuration for the windings:

- Wye-wound — The windings are wye-wound.
- Delta-wound — The windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

**Modeling fidelity – Modeling fidelity**Constant *L<sub>d</sub>*, *L<sub>q</sub>* and *PM* (default) | Tabulated *L<sub>d</sub>*, *L<sub>q</sub>* and *PM*

Select the modeling fidelity:

- Constant *L<sub>d</sub>*, *L<sub>q</sub>* and *PM* — *L<sub>d</sub>*, *L<sub>q</sub>* and *PM* values are constant and defined by their respective parameters.
- Tabulated *L<sub>d</sub>*, *L<sub>q</sub>* and *PM* — *L<sub>d</sub>*, *L<sub>q</sub>* and *PM* values are computed online from DQ currents look-up tables as follows:

$$L_d = f_1(i_d, i_q)$$

$$L_q = f_2(i_d, i_q)$$

$$\lambda_{PM} = f_3(i_d, i_q)$$

**Dependencies**To enable this parameter, set **Stator parameterization** to Specify *L<sub>d</sub>*, *L<sub>q</sub>*, and *L<sub>0</sub>*.**Number of pole pairs – Number of pole pairs**

6 (default)

Number of permanent magnet pole pairs on the rotor.

**Permanent magnet flux linkage parameterization – Permanent magnet flux linkage parameterization**

Specify flux linkage (default) | Specify torque constant | Specify back EMF constant

Choose Specify flux linkage, Specify torque constant, or Specify back EMF constant.

**Permanent magnet flux linkage – Permanent magnet flux linkage**

0.03 Wb (default)

Peak permanent magnet flux linkage with any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify flux linkage and **Modeling fidelity** to Constant Ld, Lq and PM.

**Torque constant — Torque constant**

0.18 N\*m/A (default)

Torque constant with any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify torque constant and **Modeling fidelity** to Constant Ld, Lq and PM.

**Back EMF constant — Back EMF constant**

0.18 V\*s/rad (default)

Back EMF constant with any of the stator windings.

---

**Note** The back EMF constant is the peak voltage induced by the permanent magnet in the per-unit rotational speed of each of the phases.

---

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify back EMF constant and **Modeling fidelity** to Constant Ld, Lq and PM.

**Stator parameterization — Stator parameterization**

Specify Ld, Lq, and L $\theta$  (default) | Specify Ls, Lm, and Ms

Choose Specify Ld, Lq, and L $\theta$  or Specify Ls, Lm, and Ms.

**Stator d-axis inductance, Ld — Stator d-axis inductance**

0.00022 H (default)

D-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L $\theta$  and **Modeling fidelity** to Constant Ld, Lq and PM.

**Stator q-axis inductance, Lq — Stator q-axis inductance**

0.00022 H (default)

Q-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L $\theta$  and **Modeling fidelity** to Constant Ld, Lq and PM.

**Direct-axis current vector, iD — Direct-axis current vector**

[-200, 0, 200] A (default)

Direct-axis current vector,  $iD$ .

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $Ld$ ,  $Lq$ , and  $L0$  and **Modeling fidelity** to Tabulated  $Ld$ ,  $Lq$  and  $PM$ .

**Quadrature-axis current vector,  $iQ$  – Quadrature-axis current vector**

$[-200, 0, 200]$  A (default)

Quadrature-axis current vector,  $iQ$ .

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $Ld$ ,  $Lq$ , and  $L0$  and **Modeling fidelity** to Tabulated  $Ld$ ,  $Lq$  and  $PM$ .

 **$Ld$  matrix,  $Ld(iD,iQ)$  –  $Ld$  matrix**

$0.0002 * \text{ones}(3, 3)$  H (default)

$Ld$  matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $Ld$ ,  $Lq$ , and  $L0$  and **Modeling fidelity** to Tabulated  $Ld$ ,  $Lq$  and  $PM$ .

 **$Lq$  matrix,  $Lq(iD,iQ)$  –  $Lq$  matrix**

$0.0002 * \text{ones}(3, 3)$  H (default)

$Lq$  matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $Ld$ ,  $Lq$ , and  $L0$  and **Modeling fidelity** to Tabulated  $Ld$ ,  $Lq$  and  $PM$ .

**Permanent magnet flux linkage,  $PM(iD,iQ)$  – Permanent magnet flux linkage**

$0.1 * \text{ones}(3, 3)$  Wb (default)

Permanent magnet flux linkage.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify  $Ld$ ,  $Lq$ , and  $L0$
- **Modeling fidelity** to Tabulated  $Ld$ ,  $Lq$  and  $PM$
- **Permanent magnet flux linkage parameterization** to Specify flux linkage

**Torque constant matrix,  $kt(iD,iQ)$  – Force constant matrix**

$0.18 * \text{ones}(3, 3)$  N\*m/A (default)

Torque constant matrix.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$
- **Modeling fidelity** to Tabulated  $L_d$ ,  $L_q$  and PM
- **Permanent magnet flux linkage parameterization** to Specify torque constant

**Back EMF constant matrix,  $k_e(i_d, i_q)$  — Permanent magnet flux linkage**

0.18 \* ones(3, 3) V/(rad/s) (default)

Back EMF constant matrix.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$
- **Modeling fidelity** to Tabulated  $L_d$ ,  $L_q$  and PM
- **Permanent magnet flux linkage parameterization** to Specify back EMF constant

**Stator zero-sequence inductance,  $L_0$  — Stator zero-sequence inductance**

0.00016 H (default)

Zero-sequence inductance.

**Dependencies**

To enable this parameter either:

- Set **Winding Type** to Wye-wound, **Zero sequence** to Include, and **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .
- Set **Winding Type** to Delta-wound and **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .

**Stator self-inductance per phase,  $L_s$  — Stator self-inductance per phase**

0.0002 H (default)

Average self-inductance of each of the three stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator inductance fluctuation,  $L_m$  — Stator inductance fluctuation**

-0.0002 H (default)

Amount that the self-inductance and mutual inductance of the stator windings fluctuate with the rotor angle.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator mutual inductance,  $M_s$  — Stator mutual inductance**

0.00002 H (default)

Average mutual inductance between the stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator resistance per phase, Rs — Stator resistance per phase**

0.013 Ohm (default)

Resistance of each of the stator windings.

**Zero sequence — Zero sequence option**

Include (default) | Exclude

Option to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. To prioritize model fidelity, use this default setting. Using this option:
  - Results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
  - Exposes a zero-sequence parameter in the **Impedances** settings.
- **Exclude** — Exclude zero-sequence terms. To prioritize simulation speed for desktop simulation or real-time deployment, select this option.

**Dependencies**

This parameter is visible only when you set the **Winding Type** parameter to Wye-wound.

**Rotor angle definition — Reference point for the rotor angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. “Motor Construction” on page 1-1314 figure shows the default value, Angle between the a-phase magnetic axis and the d-axis. When you select this value, the rotor and a-phase fluxes align when the rotor angle is zero.

When you select Angle between the a-phase magnetic axis and the q-axis, the a-phase current generates maximum torque when the rotor angle is zero.

**Iron Losses****Model — Enable iron losses computation**

None (default) | Empirical

Specify the iron losses computational model.

**Open-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] — Open-circuit iron losses**

[0, 0, 0] W (default)

Row vector, of length 3, of the open-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

**Short-circuit iron losses, [P\_hysteresis P\_eddy P\_excess] – Short-circuit iron losses**

[0, 0, 0] W (default)

Row vector, of length 3, of the short-circuit iron losses due to hysteresis, Eddy, and excess losses, respectively, at the frequency specified by **Electrical frequency at which losses determined**.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

**Electrical frequency at which losses determined – Electrical frequency at which losses determined**

60 Hz (default)

Electrical frequency at which the open-circuit and short-circuit iron losses were measured.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

**Short-circuit RMS current for short-circuit iron losses – Short-circuit RMS current for short-circuit iron losses**

95 A (default)

The resulting short-circuit RMS phase current when measuring the short-circuit losses.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

**Mechanical****Rotor inertia – Rotor inertia**0.01 kg\*m<sup>2</sup> (default) | nonnegative scalar

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

**Rotor damping – Rotor damping**

0 N\*m/(rad/s) (default) | nonnegative scalar

Rotary damping.

**Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature – Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 1/K (default) | scalar

Fractional rate at which the permanent magnet flux density changes with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

**Thermal mass for each stator winding – Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Percentage of main flux path iron losses associated with the rotor – Percentage of main flux path iron losses associated with the rotor**

90 (default)

The percentage of the main flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

**Percentage of cross-tooth flux path iron losses associated with the rotor – Percentage of cross-tooth flux path iron losses associated with the rotor**

30 (default)

The percentage of the cross-tooth flux path iron losses associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port **HR**, and how much is attributed to the three winding thermal ports **HA**, **HB**, and **HC**.

**Dependencies**

To enable this parameter, set **Model** to Empirical.

## Version History

**Introduced in R2013b****Electrical connection ports update**

*Behavior changed in R2021b*



From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.
- [3] Mellor, P.H., R. Wrobel, and D. Holliday. "A computationally efficient iron loss model for brushless AC machines that caters for rated flux and field weakened operation." *IEEE Electric Machines and Drives Conference*. May 2009.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

BLDC | Hybrid Excitation PMSM

### Blocks

BLDC Commutation Logic | BLDC Current Controller with PWM Generation | BLDC Current Controller

### Topics

"Expand and Collapse Three-Phase Ports on a Block"

"List of Pre-Parameterized Components"

# Phase Permute

Permute phases of three-phase system

**Library:** Simscape / Electrical / Connectors & References



## Description

The Phase Permute block cyclically permutes (changes the order of) the phases of a three-phase system.

The block has two three-phase connections associated with its terminals. If you consider the side of the block labeled **~123** (**a1, b1, c1** in expanded view) as side 1 and the side of the block labeled **~231** (**a2, b2, c2**) as side 2, then the block connects phases as shown in the table.

Side 1 Phase	Connects to Side 2 Phase
<i>a1</i>	<i>c2</i>
<i>b1</i>	<i>a2</i>
<i>c1</i>	<i>b2</i>

## Ports

### Conserving

**~123 — Expandable three-phase port**  
electrical

Expandable composite (**a1, b1, c1**) three-phase port.

**~231 — Expandable three-phase port**  
electrical

Expandable composite (**a2, b2, c2**) three-phase port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Phase Splitter

## Phase Splitter

Expand or combine three electrical conserving ports

**Library:** Simscape / Electrical / Connectors & References



### Description

The Phase Splitter block expands a composite three-phase port into its constituent phases.

The expanded output ports are electrical conserving ports.

### Ports

#### Conserving

**~ — Expandable three-phase port**

electrical

Expandable composite (**a2,b2, c2**) three-phase port

**a, b, c — Constituent phases**

electrical

Constituent phases of the expanded three-phase port

### Version History

**Introduced in R2013b**

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

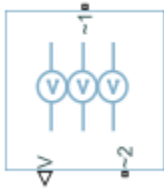
### See Also

Phase Permute

# Phase Voltage Sensor (Three-Phase)

Ideal three-phase phase voltage measurement

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Phase Voltage Sensor (Three-Phase) block represents an ideal three-phase voltage sensor. It measures the voltages across the three-phase ports **~1** and **~2** and outputs a single three-element, physical signal vector. Each element of the physical signal output vector is equal to the voltage in the respective phase.

---

**Note** You must connect the Phase Voltage Sensor (Three-Phase) block in shunt or in parallel with the three-phase element. For more information on how to use this block, see “Build and Simulate Composite and Expanded Three-Phase Models”.

---

## Ports

### Output

#### **V** – Three-elements vector

physical signal

Three-element physical signal vector output port associated with the phase voltages

### Conserving

#### **~1** – Three-phase port

electrical

Expandable three-phase port.

#### **~2** – Three-phase port

electrical

Expandable three-phase port.

## Parameters

#### **Modeling option** – Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## **Version History**

**Introduced in R2013b**

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Line Voltage Sensor (Three-Phase)

#### **Topics**

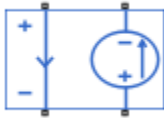
“Build and Simulate Composite and Expanded Three-Phase Models”

“Expand and Collapse Three-Phase Ports on a Block”

# PCCCS

Polynomial current-controlled current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The PCCCS (Polynomial Current-Controlled Current Source) block represents a current source whose output current value is a polynomial function of the current through the input ports. The following equations describe the current through the source as a function of time:

- If you specify an  $n$ -element vector of polynomial coefficients for the **Polynomial coefficients** parameter:

$$I_{out} = p(0) + p(1) * I_{in} + \dots + p(n - 1) * I_{in}^{n-1} + p(n) * I_{in}^n$$

- If you specify a scalar coefficient for the **Polynomial coefficients** parameter:

$$I_{out} = p * I_{in}$$

where:

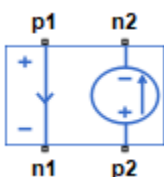
- $I_{in}$  is the current through the input ports.
- $p$  is the **Polynomial coefficients** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the output ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

Refer to the figure for port locations.



**Conserving**

**p1 – Positive voltage of the controlling input**

electrical

Electrical conserving port associated with the PCCCS positive voltage of the controlling input.

**n1 – Negative voltage of the controlling input**

electrical

Electrical conserving port associated with the PCCCS negative voltage of the controlling input.

**p2 – Positive output voltage**

electrical

Electrical conserving port associated with the PCCCS positive output voltage.

**n2 – Negative output voltage**

electrical

Electrical conserving port associated with the PCCCS negative output voltage.

**Parameters**

**Polynomial coefficients – Polynomial coefficients**

[0, 1] (default)

Polynomial coefficients that relate the input current to the output current, as described in the preceding section.

**Include parallel conductance, GMIN – Whether to include parallel conductance**

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

**Values of parallel conductance, GMIN**

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	



## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | PCCCS2 | PCCVS | PCCVS2 | PVCCS | PVCCS2 | PVCVS | PVCVS2

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## PCCCS2

Polynomial current-controlled current source with two controlling inputs

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



### Description

The PCCCS2 (Two-Input Polynomial Current-Controlled Current Source) block represents a current source whose output current value is a polynomial function of the currents through the pairs of controlling input ports. This equation describes the current through the source as a function of time:

$$I_{out} = p_1 + p_2 * I_{in1} + p_3 * I_{in2} + p_4 * I_{in1}^2 + p_5 I_{in1} * I_{in2} + p_6 * I_{in2}^2 + \dots$$

where:

- $I_{in1}$  is the current through the first pair of input ports.
- $I_{in2}$  is the current through the second pair of input ports.
- $p$  is the **Polynomial coefficients** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the output ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

### Ports

#### Conserving

##### 1+ — Positive voltage of the first controlling input

electrical

Electrical conserving port associated with the PCCCS2 positive voltage of the first controlling input.

##### 1- — Negative voltage of the first controlling input

electrical

Electrical conserving port associated with the PCCCS2 negative voltage of the first controlling input.

##### 2+ — Positive voltage of the second controlling input

electrical

Electrical conserving port associated with the PCCCS2 positive voltage of the second controlling input.

## 2- – Negative voltage of the second controlling input

electrical

Electrical conserving port associated with the PCCCS2 negative voltage of the second controlling input.

## + – Positive output voltage

electrical

Electrical conserving port associated with the PCCCS2 positive output voltage.

## - – Negative output voltage

electrical

Electrical conserving port associated with the PCCCS2 negative output voltage.

## Parameters

### Polynomial coefficients – Polynomial coefficients

[ 0 1 1 ] (default)

Polynomial coefficients that relate the input current to the output current, as described in the preceding section.

### Include parallel conductance, GMIN – Whether to include parallel conductance

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

### Values of parallel conductance, GMIN

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

## Version History

Introduced in R2009a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Environment Parameters | PCCCS | PCCVS | PCCVS2 | PVCCS | PVCCS2 | PVCVS | PVCVS2

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

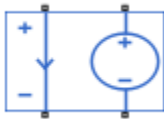
“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# PCCVS

Polynomial current-controlled voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The PCCVS (Polynomial Current-Controlled Voltage Source) block represents a voltage source whose output voltage value is a polynomial function of the current through the input ports. The following equations describe the voltage across the source as a function of time:

- If you specify an  $n$ -element vector of polynomial coefficients for the **Polynomial coefficients** parameter:

$$V_{out} = p(0) + p(1) * I_{in} + \dots + p(n - 1) * I_{in}^{n-1} + p(n) * I_{in}^n$$

- If you specify a scalar coefficient for the **Polynomial coefficients** parameter:

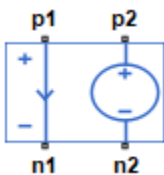
$$V_{out} = p * I_{in}$$

where:

- $I_{in}$  is the current through the input ports.
- $p$  is the **Polynomial coefficients** parameter value.

## Ports

Refer to the figure for port locations.



## Conserving

**p1 — Positive voltage of the controlling input**

electrical

Electrical conserving port associated with the PCCVS positive voltage of the controlling input.

**n1 — Negative voltage of the controlling input**

electrical

Electrical conserving port associated with the PCCVS negative voltage of the controlling input.

**p2 — Positive output voltage**

electrical

Electrical conserving port associated with the PCCVS positive output voltage.

**n2 — Negative output voltage**

electrical

Electrical conserving port associated with the PCCVS negative output voltage.

**Parameters****Polynomial coefficients — Polynomial coefficients**

[0, 1] (default)

The polynomial coefficients that relate the input current to the output voltage, as described in the preceding section.

**Version History**

Introduced in R2008a

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also****Simscape Blocks**

Environment Parameters | PCCCS | PCCCS2 | PCCVS2 | PVCCS | PVCCS2 | PVCVS | PVCVS2

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# PCCVS2

Polynomial current-controlled voltage source with two controlling inputs

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The PCCVS2 (Two-Input Polynomial Current-Controlled Voltage Source) block represents a voltage source whose output voltage value is a polynomial function of the currents through the pairs of controlling input ports. This equation describes the voltage across the source as a function of time:

$$V_{out} = p_1 + p_2 * I_{in1} + p_3 * I_{in2} + p_4 * I_{in1}^2 + p_5 I_{in1} * I_{in2} + p_6 * I_{in2}^2 + \dots$$

where:

- $I_{in1}$  is the current through the first pair of input ports.
- $I_{in2}$  is the current through the second pair of input ports.
- $p$  is the **Polynomial coefficients** parameter value.

## Ports

### Conserving

#### 1+ — Positive voltage of the first controlling input

electrical

Electrical conserving port associated with the PCCVS2 positive voltage of the first controlling input.

#### 1- — Negative voltage of the first controlling input

electrical

Electrical conserving port associated with the PCCVS2 negative voltage of the first controlling input.

#### 2+ — Positive voltage of the second controlling input

electrical

Electrical conserving port associated with the PCCVS2 positive voltage of the second controlling input.

#### 2- — Negative voltage of the second controlling input

electrical

Electrical conserving port associated with the PCCVS2 negative voltage of the second controlling input.

**+ – Positive output voltage**

electrical

Electrical conserving port associated with the PCCVS2 positive output voltage.

**- – Negative output voltage**

electrical

Electrical conserving port associated with the PCCVS2 negative output voltage.

**Parameters****Polynomial coefficients – Polynomial coefficients**

[ 0 1 1 ] (default)

The polynomial coefficients that relate the input current to the output voltage, as described in the preceding section.

**Version History****Introduced in R2009a****Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also****Simscape Blocks**

Environment Parameters | PCCCS | PCCCS2 | PCCVS | PVCCS | PVCCS2 | PVCVS | PVCVS2

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

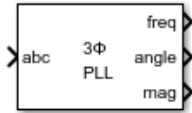
“Parameterize an Exponential Diode from SPICE Netlist”



# Phasor Measurement Unit (PMU, Three-Phase)

Frequency, phase, and amplitude measurement of a three-phase system

**Library:** Simscape / Electrical / Control / Measurements



## Description

The Phasor Measurement Unit (PMU, Three-Phase) block measures the frequency, phase, and amplitude of the positive sequence component of the input electrical signal in a three-phase system at a particular sample rate.

The block uses a system clock as the synchronous reference. This information is commonly collected from a GPS in the real world.

## Ports

### Input

#### **abc — Three-phase signal**

vector

Three-phase input signal, specified as a vector.

Data Types: `single` | `double`

### Output

#### **freq — Frequency of positive sequence component**

scalar

Measured frequency of the positive sequence component, in Hz, returned as a scalar.

Data Types: `single` | `double`

#### **angle — Phase angle of positive sequence component**

scalar

Measured phase angle of the positive sequence component, in degrees, returned as a scalar.

Data Types: `single` | `double`

#### **mag — Magnitude of positive sequence component**

scalar

Measured magnitude of the positive sequence component, returned as a scalar.

Data Types: `single` | `double`

## Parameters

### **Rated frequency (Hz) — Rated frequency**

60 (default) | 50

Rated frequency of the input signal, in Hz.

### **Sampling points per cycle — Sampling points per cycle**

64 (default) | 24 | 48

Number of sampling points per cycle.

### **Reporting rate factor — Reporting rate factor**

1 (default) | scalar

Ratio between the reporting rate and the sampling rate of the block.

## Version History

**Introduced in R2020a**

## See Also

Sinusoidal Measurement (PLL) | Sinusoidal Measurement (PLL, Three-Phase)

# Photodiode

Photodiode with incident flux input port

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Photodiode block represents a photodiode as a controlled current source and an exponential diode connected in parallel. The controlled current source produces a current  $I_p$  that is proportional to the radiant flux density:

$$I_p = DeviceSensitivity \cdot RadiantFluxDensity \quad (1-48)$$

where:

- *DeviceSensitivity* is the ratio of the current produced to the incident radiant flux density.
  - If you select **Specify measured current for given flux density** for the **Sensitivity parameterization** parameter, the block calculates this variable by converting the **Measured current** parameter value to units of amps and dividing it by the **Flux density** parameter values.
  - If you select **Specify current per unit flux density** for the **Sensitivity parameterization** parameter, this variable is defined by the **Device sensitivity** parameter value.
- *RadiantFluxDensity* is the incident radiant flux density.

To model dynamic response time, use the **Parameterization** parameter in the **Junction capacitance** tab to include the diode junction capacitance in the model.

The exponential diode model provides the following relationship between the diode current  $I$  and the diode voltage  $V$ :

$$I = IS \cdot \left( e^{\frac{qV}{NkT_{m1}}} - 1 \right)$$

where:

- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $N$  is the emission coefficient.
- $IS$  is the saturation current, which is equal to the **Dark current** parameter value.
- $T_{m1}$  is the temperature at which the diode parameters are specified, as defined by the **Measurement temperature** parameter value.

When  $(qV / NkT_{m1}) > 80$ , the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} - 79)e^{80}$ , which matches the gradient of the diode current at  $(qV / NkT_{m1}) = 80$  and extrapolates linearly. When  $(qV / NkT_{m1}) < -79$ ,

the block replaces  $e^{\frac{qV}{NkT_{m1}}}$  with  $(qV / NkT_{m1} + 80)e^{-79}$ , which also matches the gradient and extrapolates linearly. Typical electrical circuits do not reach these extreme values. The block provides this linear extrapolation to help convergence when solving for the constraints during simulation.

When you select **Use dark current** and **N** for the **Diode parameterization** parameter, you specify the diode in terms of the **Dark current** and **Emission coefficient N** parameters. When you select **Use dark current plus a forward bias I-V data point** for the **Diode parameterization** parameter, you specify the **Dark current** parameter and a voltage and current measurement point on the diode I-V curve. The block calculates  $N$  from these values as follows:

$$N = V_F / (V_t \log(I_F / I_S + 1))$$

where:

- $V_F$  is the **Forward voltage VF** parameter value.
- $V_t = kT_{m1} / q$ .
- $I_F$  is the **Current IF at forward voltage VF** parameter value.

The exponential diode model provides the option to include a junction capacitance:

- When you select **Fixed or zero junction capacitance** for the **Parameterization** parameter, the capacitance is fixed.
- When you select **Use parameters CJO, VJ, M & FC** for the **Parameterization** parameter, the block uses the coefficients  $CJO$ ,  $VJ$ ,  $M$ , and  $FC$  to calculate a junction capacitance that depends on the junction voltage.
- When you select **Use C-V curve data points** for the **Parameterization** parameter, the block uses three capacitance values on the C-V capacitance curve to estimate  $CJO$ ,  $VJ$ , and  $M$  and uses these values with the specified value of  $FC$  to calculate a junction capacitance that depends on the junction voltage. The block calculates  $CJO$ ,  $VJ$ , and  $M$  as follows:

- $CJO = C_1((V_{R2} - V_{R1}) / (V_{R2} - V_{R1}(C_2/C_1)^{-1/M}))^M$
- $VJ = -(-V_{R2}(C_1/C_2)^{-1/M} + V_{R1}) / (1 - (C_1/C_2)^{-1/M})$
- $M = \log(C_3/C_2) / \log(V_{R2}/V_{R3})$

where:

- $V_{R1}$ ,  $V_{R2}$ , and  $V_{R3}$  are the values in the **Reverse bias voltages [VR1 VR2 VR3]** vector.
- $C_1$ ,  $C_2$ , and  $C_3$  are the values in the **Corresponding capacitances [C1 C2 C3]** vector.

It is not possible to estimate  $FC$  reliably from tabulated data, so you must specify its value using the **Capacitance coefficient FC** parameter. In the absence of suitable data for this parameter, use a typical value of 0.5.

The reverse bias voltages (defined as positive values) should satisfy  $V_{R3} > V_{R2} > V_{R1}$ . This means that the capacitances should satisfy  $C_1 > C_2 > C_3$  as reverse bias widens the depletion region and hence reduces capacitance. Violating these inequalities results in an error. Voltages  $V_{R2}$  and  $V_{R3}$  should be well away from the Junction potential  $VJ$ . Voltage  $V_{R1}$  should be less than the Junction potential  $VJ$ , with a typical value for  $V_{R1}$  being 0.1 V.

The voltage-dependent junction is defined in terms of the capacitor charge storage  $Q_j$  as:

- For  $V < FC \cdot VJ$ :

$$Q_j = C_{J0} \cdot (VJ/(M - 1)) \cdot ((1 - V/VJ)^{1-M} - 1)$$

- For  $V \geq FC \cdot VJ$ :

$$Q_j = C_{J0} \cdot F_1 + (C_{J0}/F_2) \cdot (F_3 \cdot (V - FC \cdot VJ) + 0.5(M/VJ) \cdot (V^2 - (FC \cdot VJ)^2))$$

where:

- $F_1 = (VJ/(1 - M)) \cdot (1 - (1 - FC)^{1-M})$
- $F_2 = (1 - FC)^{1+M}$
- $F_3 = 1 - FC \cdot (1 + M)$

These equations are the same as used in [2], except that the temperature dependence of  $VJ$  and  $FC$  is not modeled. This model does not include the diffusion capacitance term that affects performance for high frequency switching applications.

The Photodiode block contains several options for modeling the dependence of the diode current-voltage relationship on the temperature during simulation. Temperature dependence of the junction capacitance is not modeled, this being a much smaller effect. For details, see the Diode reference page.

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- When you select **Use dark current plus a forward bias I-V curve data point** for the **Diode parameterization** parameter, choose a voltage near the diode turn-on voltage.

Typically this will be in the range from 0.05 to 1 Volt. Using a value outside of this region may lead to a poor estimate for  $N$ .

- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.

## Ports

### Output

#### **W – Incident flux**

physical signal

Physical signal port associated with the incident flux.

### Conserving

#### **+ – Positive terminal**

electrical

Electrical conserving port associated with the anode.

#### **- – Negative terminal**

electrical

Electrical conserving port associated with the cathode.

#### **H – Thermal port**

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

#### **Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Main

#### **Sensitivity parameterization – Sensitivity parameterization**

Specify measured current for given flux density (default) | Specify current per unit flux density

Select one of the following methods for sensitivity parameterization:

- Specify measured current for given flux density – Specify the measured current and the corresponding flux density. This is the default method.

- Specify current per unit flux density — Specify the device sensitivity directly.

#### Measured current — Measured current

25  $\mu\text{A}$  (default)

The current the block uses to calculate the device sensitivity.

#### Dependencies

This parameter is visible only when you select Specify measured current for given flux density for the **Sensitivity parameterization** parameter.

#### Flux density — Flux density

5  $\text{W}/\text{m}^2$  (default)

The flux density the block uses to calculate the device sensitivity.

#### Dependencies

This parameter is visible only when you select Specify measured current for given flux density for the **Sensitivity parameterization** parameter.

#### Device sensitivity — Device sensitivity

$5 \times 10^{-6} \text{ m}^2 \cdot \text{A}/\text{W}$  (default)

The current per unit flux density.

#### Dependencies

This parameter is visible only when you select Specify current per unit flux density for the **Sensitivity parameterization** parameter.

#### Diode parameterization — Diode parameterization

Use dark current plus a forward bias I-V data point (default) | Use dark current and N

Select one of the following methods for diode model parameterization:

- Use dark current plus a forward bias I-V data point — Specify the dark current and a point on the diode I-V curve. This is the default method.
- Use dark current and N — Specify dark current and emission coefficient.

#### Current I1 — Current at the forward-biased point

0.1 A (default)

The current at the forward-biased point on the diode I-V curve that the block uses to calculate  $I_S$  and  $N$ .

#### Dependencies

This parameter is visible only when you select Use dark current plus a forward bias I-V data point for the **Diode parameterization** parameter.

#### Voltage V1 — Voltage at the forward-biased point

1.3 V (default)

The corresponding voltage at the forward-biased point on the diode I-V curve that the block uses to calculate  $I_S$  and  $N$ .

**Dependencies**

This parameter is visible only when you select Use dark current plus a forward bias I-V data point for the **Diode parameterization** parameter.

**Dark current — Dark current**

5e-9 A (default)

The current through the diode when it is not exposed to light.

**Emission coefficient, N — Emission coefficient**

3 (default)

The diode emission coefficient or ideality factor.

**Dependencies**

This parameter is visible only when you select Use dark current and N for the **Diode parameterization** parameter.

**Ohmic resistance, RS — Series diode connection resistance**

0.1 Ohm (default)

The series diode connection resistance.

**Measurement temperature — Measurement temperature**

25 degC (default)

The temperature at which the I-V curve or dark current was measured.

**Junction Capacitance****Parameterization — Junction capacitance parameterization**

Fixed or zero junction capacitance (default) | Use C-V curve data points | Use parameters CJ0, VJ, M & FC

Select one of the following options for modeling the junction capacitance:

- Fixed or zero junction capacitance — Model the junction capacitance as a fixed value.
- Use C-V curve data points — Specify measured data at three points on the diode C-V curve.
- Use parameters CJ0, VJ, M & FC — Specify zero-bias junction capacitance, junction potential, grading coefficient, and forward-bias depletion capacitance coefficient.

**Junction capacitance — Junction capacitance**

60 pF (default)

Fixed value of the junction capacitance.

**Dependencies**

This parameter is visible only when you select Fixed or zero junction capacitance for the **Parameterization** parameter.



**Zero-bias junction capacitance,  $C_{J0}$  — Zero-bias junction capacitance**

60 pF (default)

Value of the capacitance placed in parallel with the exponential diode term.

**Dependencies**

This parameter is visible only when you select Use parameters  $C_{J0}$ ,  $V_J$ ,  $M$  &  $FC$  for the **Parameterization** parameter.

**Reverse bias voltages [VR1 VR2 VR3] — Reverse bias voltages**

[ 0.1 10 100 ] V (default)

A vector of the reverse bias voltage values at the three points on the diode C-V curve that the block uses to calculate  $C_{J0}$ ,  $V_J$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select Use C-V curve data points for the **Parameterization** parameter.

**Corresponding capacitances [C1 C2 C3] — Corresponding capacitances**

[ 45 3 6 ] pF (default)

A vector of the capacitance values at the three points on the diode C-V curve that the block uses to calculate  $C_{J0}$ ,  $V_J$ , and  $M$ .

**Dependencies**

This parameter is visible only when you select Use C-V curve data points for the **Parameterization** parameter.

**Junction potential,  $V_J$  — Junction potential**

1 V (default)

Junction potential.

**Dependencies**

This parameter is visible only when you select Use parameters  $C_{J0}$ ,  $V_J$ ,  $M$  &  $FC$  for the **Parameterization** parameter.

**Grading coefficient,  $M$  — Grading coefficient**

0.5 (default)

The grading coefficient.

**Dependencies**

This parameter is visible only when you select Use parameters  $C_{J0}$ ,  $V_J$ ,  $M$  &  $FC$  for the **Parameterization** parameter.

**Capacitance coefficient,  $FC$  — Capacitance coefficient**

0.5 (default)

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage.

**Dependencies**

This parameter is visible only when you select Use C-V curve data points or Use parameters CJ0, VJ, M & FC for the **Parameterization** parameter.

**Temperature Dependence****Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Use an I-V data point at second measurement temperature T2 | Specify saturation current at second measurement temperature T2 | Specify the energy gap EG

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Use an I-V data point at second measurement temperature T2 – If you select this option, you specify a second measurement temperature  $T_{m2}$ , and the current and voltage values at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify saturation current at second measurement temperature T2 – If you select this option, you specify a second measurement temperature  $T_{m2}$ , and saturation current value at this temperature. The model uses these values, along with the parameter values at the first measurement temperature  $T_{m1}$ , to calculate the energy gap value.
- Specify the energy gap EG – Specify the energy gap value directly.

**Current I1 at second measurement temperature – Current I1 at second measurement temperature**

0.07 A (default)

Specify the diode current  $I1$  value when the voltage is  $V1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 for the **Parameterization** parameter.

**Voltage V1 at second measurement temperature – Voltage V1 at second measurement temperature**

1.3 V (default)

Specify the diode voltage  $V1$  value when the current is  $I1$  at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 for the **Parameterization** parameter.

**Saturation current, IS, at second measurement temperature – Saturation current, IS, at second measurement temperature**

2.5e-7 A (default)

Specify the saturation current  $IS$  value at the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Second measurement temperature — Second measurement temperature**

125 degC (default)

Specify the value for the second measurement temperature.

**Dependencies**

This parameter is visible only when you select Use an I-V data point at second measurement temperature T2 or Specify saturation current at second measurement temperature T2 for the **Parameterization** parameter.

**Energy gap parameterization — Energy gap parameterization**

Use nominal value for silicon (EG=1.11eV) (default) | Use nominal value for 4H-SiC silicon carbide (EG=3.23eV) | Use nominal value for 6H-SiC silicon carbide (EG=3.00eV) | Use nominal value for germanium (EG=0.67eV) | Use nominal value for gallium arsenide (EG=1.43eV) | Use nominal value for selenium (EG=1.74eV) | Use nominal value for Schottky barrier diodes (EG=0.69eV) | Specify a custom value

Select a value for the energy gap from a list of predetermined options, or specify a custom value:

- Use nominal value for silicon (EG=1.11eV) — This is the default.
- Use nominal value for 4H-SiC silicon carbide (EG=3.23eV)
- Use nominal value for 6H-SiC silicon carbide (EG=3.00eV)
- Use nominal value for germanium (EG=0.67eV)
- Use nominal value for gallium arsenide (EG=1.43eV)
- Use nominal value for selenium (EG=1.74eV)
- Use nominal value for Schottky barrier diodes (EG=0.69eV)
- Specify a custom value — If you select this option, the **Energy gap, EG** parameter appears in the dialog box, to let you specify a custom value for *EG*.

**Dependencies**

This parameter is visible only when you select Specify the energy gap EG for the **Parameterization** parameter.

**Energy gap, EG — Energy gap**

1.11 eV (default)

Specify a custom value for the energy gap, *EG*.

**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Energy gap parameterization** parameter.

**Saturation current temperature exponent parameterization – Saturation current temperature exponent parameterization**

Use nominal value for pn-junction diode (XTI=3) (default) | Use nominal value for Schottky barrier diode (XTI=2) | Specify a custom value

Select one of the following options to specify the saturation current temperature exponent value:

- Use nominal value for pn-junction diode (XTI=3) – This is the default.
- Use nominal value for Schottky barrier diode (XTI=2)
- Specify a custom value – If you select this option, the **Saturation current temperature exponent, XTI** parameter appears in the dialog box, to let you specify a custom value for *XTI*.

**Saturation current temperature exponent, XTI – Saturation current temperature exponent**

3 (default)

Specify a custom value for the saturation current temperature exponent, *XTI*.

**Dependencies**

This parameter is visible only when you select Specify a custom value for the **Saturation current temperature exponent parameterization** parameter.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Specify the value for the temperature  $T_s$ , at which the device is to be simulated.

## Version History

Introduced in R2008a

## References

- [1] H. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition, Cambridge University Press, 1984.
- [2] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Diode | Light-Emitting Diode | Optocoupler

# Piecewise Linear Current Source

Time-dependent current source using lookup table

**Library:** Simscape / Electrical / Sources  
Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Piecewise Linear Current Source block represents a current source that you specify in lookup table form using a vector of time values and a vector of the corresponding current values. You must specify at least two time-current value pairs. The block generates a time-dependent current based on these time-current values using the selected interpolation and extrapolation methods. You have a choice of two interpolation methods and extrapolation methods. The output current is independent of the voltage across the terminals of the source.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance *GMIN*:

- By default, *GMIN* matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-121/\text{Ohm}$ .
- To change *GMIN*, add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the piecewise linear current source positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the piecewise linear current source negative terminal.

## Parameters

### Time specification – Vector of time values

[0, 1, 2, 3, 4] s (default)

Vector of time values as a tabulated 1-by-*n* array. The time values vector must be strictly monotonically increasing.

**Current at specified time – Vector of current values**

[0, 0, 0, 0, 0] A (default)

Vector of current values as a tabulated 1-by- $n$  array. The current values vector must be the same size as the time values vector.

**Interpolation method – Interpolation method**

Linear (default) | Smooth

Select the method that the block uses to determine the output current values at intermediate time points that are not specified in the preceding vectors:

- **Linear** — Prioritize performance by using a linear function.
- **Smooth** — Prioritize accuracy by producing a continuous curve with continuous first-order derivatives.

**Extrapolation method – Extrapolation method**

Nearest (default) | Linear

Select the method that the block uses to determine the output current values at time points that are outside the time range specified in the preceding vectors:

- **Nearest** — Select this option to use the nearest input value for extrapolation.
- **Linear** — Select this option to use a linear function.

**Include parallel conductance, GMIN – Whether to include parallel conductance**

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

**Values of parallel conductance, GMIN**

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

**Version History**

Introduced in R2008a

## References

- [1] D. Kahaner, Cleve Moler, and Stephen Nash *Numerical Methods and Software* Prentice Hall, 1988.
- [2] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Wetterling. *Numerical Recipes in C: The Art of Scientific Computing* . Cambridge University Press, 1992.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | Pulse Current Source | Piecewise Linear Voltage Source

### Functions

`subcircuit2ssc`

### Topics

“Parameterize an Exponential Diode from SPICE Netlist”  
“Converting a SPICE Netlist to Simscape Blocks”  
“Additional Parameterization Workflows”

# Piecewise Linear Voltage Source

Time-dependent voltage source using lookup table

**Library:** Simscape / Electrical / Sources  
Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Piecewise Linear Voltage Source block represents a voltage source that you specify in lookup table form using a vector of time values and a vector of the corresponding voltage values. You must specify at least two time-current value pairs. The block generates a time-dependent voltage based on these time-voltage values using the selected interpolation and extrapolation methods. You have a choice of two interpolation methods and extrapolation methods. The output voltage is independent of the current through the source.

## Ports

### Conserving

#### + — Positive voltage

electrical

Electrical conserving port associated with the piecewise linear voltage source positive voltage

#### - — Negative voltage

electrical

Electrical conserving port associated with the piecewise linear voltage source negative voltage

## Parameters

### Time specification — Vector of time values

[0, 1, 2, 3, 4] s (default)

Vector of time values as a tabulated 1-by- $n$  array. The time values vector must be strictly monotonically increasing.

### Current at specified time — Vector of current values

[0, 0, 0, 0, 0] V (default)

Vector of voltage values as a tabulated 1-by- $n$  array. The voltage values vector must be the same size as the time values vector.

### Interpolation method — Interpolation method

Linear (default) | Smooth



Select the method that the block uses to determine the output current values at intermediate time points that are not specified in the preceding vectors:

- **Linear** — Prioritize performance by using a linear function.
- **Smooth** — Prioritize accuracy by producing a continuous curve with continuous first-order derivatives.

### **Extrapolation method — Extrapolation method**

Nearest (default) | Linear

Select the method that the block uses to determine the output current values at time points that are outside the time range specified in the preceding vectors:

- **Nearest** — Select this option to use the nearest input value for extrapolation.
- **Linear** — Select this option to use a linear function.

## **Version History**

**Introduced in R2008a**

## **References**

[1] D. Kahaner, C. Moler, and S. Nash. *Numerical Methods and Software* . Prentice Hall, 1988.

[2] W.H. Press, B.P. Flannery, S.A. Teulkolsky, and W.T. Wetterling. *Numerical Recipes in C: The Art of Scientific Computing* . Cambridge University Press, 1992.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Environment Parameters | Pulse Voltage Source | Piecewise Linear Current Source

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

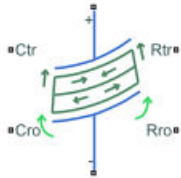
“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## Piezo Bender

Piezoelectric bimorph beam of rectangular cross-section

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



### Description

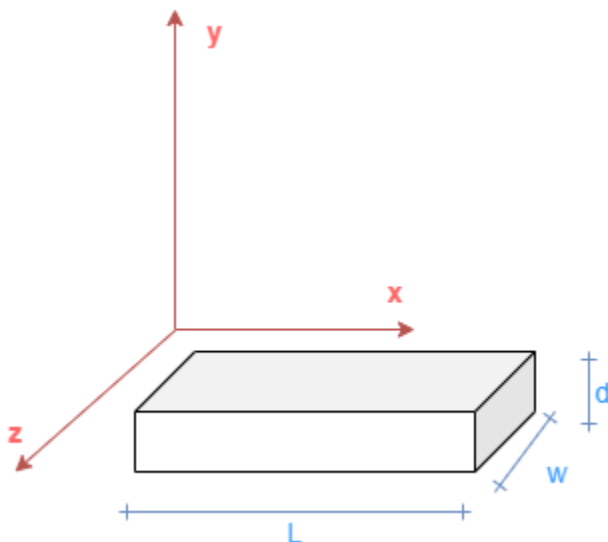
The Piezo Bender block models a piezoelectric bimorph beam of rectangular cross-section.

A piezo bender is a piezoelectric device that bends when you apply an electrical potential between its plates. Conversely, when a piezo bender bends, it generates an electrical potential.

A piezo bender comprises different rectangular layers of piezoelectric material with a polarization perpendicular to the stack. This polarization is alternated in each layer.

### Equations

This figure shows the cartesian reference frame



where:

- $L$  is the beam length.
- $w$  is the beam width.

- $d$  is the beam thickness.

These are the constitutive equations for a piezoelectric material in the stress-charge formulation,

$$T = [c]S - [e]^T E \quad (1-49)$$

$$D = [e]S - [\epsilon^S]E \quad (1-50)$$

where:

- $T$  is the stress field.
- $[c]$  is the compliance tensor.
- $S$  is the strain field.
- $[e]$  is the piezo stress coefficient tensor.
- $E$  is the electric field.
- $D$  is the electric displacement field.
- $[\epsilon^S]$  is the permittivity tensor at constant or zero strain.

To model the flexibility, the block uses the Euler-Bernoulli finite-element beam equations. The translation and rotation of each beam cross-section in function of the  $x$ -axis determine the beam kinematics.

This block considers only the forces applied in the  $y$  direction and the piezoelectric material is polarized to bend only in the  $x$ - $y$  plane. For this reason, to describe the kinematics, you only need to specify the vertical displacement in the  $y$  direction of the center of gravity of each cross-section,  $y(x)$ , and the rotation around the  $z$ -axis of each cross-section,  $\varphi_z(x)$ .

From the previous assumptions, the strain field in an Euler-Bernoulli beam subject to bending is equal to:

$$S_{xx}(x, y) = -y \frac{d\varphi_z}{dx}(x). \quad (1-51)$$

Because the electric field is constant between the positive and negative plates,  $E_y = \frac{v}{d}$ , the block substitutes equation 3 into equation 1:

$$T_{xx}(x, y) = -Ey \frac{d\varphi_z}{dx}(x) - e_{31} \frac{v}{d}.$$

In this equation,  $E = c_{11}$  is the Young modulus of the material and  $e_{31}$  is the (3,1) piezoelectric stress-charge coupling coefficient,  $\sigma_{xx} = -e_{31}E_y$ .

This equation defines the bending moment from the stress field:

$$M_z(x) = - \iint y T_{xx}(x, y) dS = \iint Ey^2 \frac{d\varphi_z}{dx}(x) + e_{31} \frac{v}{d} y dS.$$

Because the polarization of the material for  $y = \left[-\frac{d}{2}, 0\right]$  is the opposite than the polarization for  $y = \left[0, \frac{d}{2}\right]$ , the (3,1) piezoelectric stress-charge coupling coefficient changes sign and the bending moment is defined by

$$M_z(x) = \frac{E}{12} w d^3 \frac{d\varphi_z}{dx}(x) + w d e_{31} v = EI \frac{d\varphi_z}{dx}(x) + w d e_{31} v,$$

where  $I = \frac{1}{12} w d^3$  is the second moment of area of the rectangular cross-section.

In this equation, the first term is the classical equation of a beam subject to bending, and the second term is the electromechanical coupling due to the presence of a voltage across the piezoelectric material. This voltage produces a uniform electrical bending moment loaded along the beam.

The block then substitutes equation 3 into equation 2:

$$D_y(x, y) = -e_{31} y \frac{d\varphi_z}{dx}(x) + \frac{\epsilon v}{d}.$$

The electric charge inside a volume is equal to the Gauss integral of the electric displacement:

$$dq(x) = \oiint D_y dS = e_{31} d w dx \frac{d\varphi_z}{dx} = e_{31} d w d\varphi_z.$$

Then, this equation defines the charge accumulated between the two sections of the beam due to the piezoelectric effect:

$$q = e_{31} d w (\varphi_z(x_2) - \varphi_z(x_1)).$$

Finally, from the mechanical perspective, you can model a piezo bender as an Euler-Bernoulli beam loaded with a uniform torque that is proportional to the voltage:

$$M_z(x) = EI \frac{d\varphi_z}{dx}(x) + w d e_{31} v$$

From the electrical perspective, you can model a piezo bender as a capacitor with a charge source proportional to the bending angle:

$$q = e_{31} d w (\varphi_{zR} - \varphi_{zC}) + \frac{\epsilon w l}{d} v.$$

### Finite Element Formulation

To discretize and solve the Euler-Bernoulli equations with the piezoelectric coupling, the Piezo Bender block uses a Finite Element method.

The block discretizes the piezo bender beam into a number of slices in the length direction with the same width,  $w$ , and thickness,  $d$ . The length of each element is equal to the total length of the beam divided by the number of elements,  $l = \frac{L}{N_{elements}}$ .

This stiffness matrix of a finite element of an Euler-Bernoulli beam defines the relationship between the vertical displacement and rotational angle of each end of the beam element, and the corresponding forces and moments due to the beam elasticity:

$$\begin{bmatrix} F_C \\ T_C \\ F_R \\ T_R \end{bmatrix} = \begin{bmatrix} \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} \\ \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} \\ -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} \\ \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} \end{bmatrix} \begin{bmatrix} y_C \\ \varphi_C \\ y_R \\ \varphi_R \end{bmatrix}.$$

Then, to obtain the equations for a piezo bender beam element, add the coupling terms and the mass matrix for the inertia:

$$\frac{\rho l w d}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \begin{bmatrix} \frac{d^2 y_C}{dt^2} \\ \frac{d^2 \varphi_C}{dt^2} \\ \frac{d^2 y_R}{dt^2} \\ \frac{d^2 \varphi_R}{dt^2} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} & 0 \\ \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} & e_{31}wd \\ -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} & 0 \\ \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} & -e_{31}wd \\ 0 & -e_{31}wd & 0 & e_{31}wd & \frac{ewl}{d} \end{bmatrix} \begin{bmatrix} y_C \\ \varphi_C \\ y_R \\ \varphi_R \\ v \end{bmatrix} = \begin{bmatrix} F_C \\ T_C \\ F_R \\ T_R \\ q \end{bmatrix}.$$

Finally, this is the equation for a piezo bender beam element with damping:

$$\frac{\rho l w d}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \begin{bmatrix} \frac{d^2 y_C}{dt^2} \\ \frac{d^2 \varphi_C}{dt^2} \\ \frac{d^2 y_R}{dt^2} \\ \frac{d^2 \varphi_R}{dt^2} \\ 0 \end{bmatrix} + \begin{bmatrix} [B] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \frac{dy_C}{dt} \\ \frac{d\varphi_C}{dt} \\ \frac{dy_R}{dt} \\ \frac{d\varphi_R}{dt} \\ 0 \end{bmatrix} \\
 + \begin{bmatrix} \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} & 0 \\ \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} & e_{31}wd \\ -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} & 0 \\ \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} & -e_{31}wd \\ 0 & -e_{31}wd & 0 & e_{31}wd & \frac{\epsilon w l}{d} \end{bmatrix} \begin{bmatrix} y_C \\ \varphi_C \\ y_R \\ \varphi_R \\ v \end{bmatrix} = \begin{bmatrix} F_C \\ T_C \\ F_R \\ T_R \\ q \end{bmatrix}$$

where:

- $l$  is the element length.
- $w$  is the element width.
- $d$  is the element thickness.
- $I = \frac{1}{12}wt^3$  is the second moment of area.
- $E$  is the Young modulus.
- $m = \rho l w d$  is the element mass, where  $\rho$  is the mass density.
- $e_{31}$  is the (3,1) piezoelectric stress-charge coupling coefficient,  $\sigma_{xx} = -e_{31}E_y$ .
- $\epsilon$  is the electrical permittivity.
- $B = b_k K + b_m M$  is the damping matrix.
- $b_m$  is the Rayleigh damping coefficient proportional to mass.
- $b_k$  is the Rayleigh damping coefficient proportional to stiffness.

- $$K = \begin{bmatrix} \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} \\ \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} \\ -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} \\ \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} \end{bmatrix}$$
 is the stiffness finite element matrix.

- $$M = \frac{\rho l w d}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix}$$
 is the mass matrix.

- $y_C$  is the deflection along the y-axis of the left-end of the element.
- $y_R$  is the deflection along the y-axis of the right-end of the element.
- $\varphi_C$  is the rotation around the z-axis of the left-end of the element.
- $\varphi_R$  is the rotation around the z-axis of the right-end of the element.
- $F_C$  is the force along the y-axis of the left-end of the element.
- $F_R$  is the force along the y-axis of the right-end of the element.
- $T_C$  is the torque in the z-axis of the left-end of the element.
- $T_R$  is the torque in the z-axis of the right-end of the element.
- $v$  is the voltage across the top and bottom electrodes.
- $q$  is the accumulated charge between the electrodes and the piezoelectric material.

### Datasheet Parameterization

A datasheet of a piezo bender usually provides this data:

- Dimensions ( $l$ ,  $w$ ,  $d$ )
- Mass,  $m$
- Rated voltage,  $v_{rated}$
- Free deflection at rated voltage,  $y_{free}$
- Blocking force at rated voltage,  $F_{block}$
- Capacitance,  $C_{piezo}$
- First resonant frequency,  $f_1$

It is possible to calculate the fundamental material parameters of a piezo bender by using the datasheet parameters.

First, the block solves the voltage-force deflection relationships from the steady-state equations with no torque applied and clamped-free configuration:

$$\begin{bmatrix} \frac{12EI}{l^3} & -\frac{6EI}{l^2} & 0 \\ -\frac{6EI}{l^2} & \frac{4EI}{l} & -e_{31}wd \\ 0 & e_{31}wd & \frac{\varepsilon wl}{d} \end{bmatrix} \begin{bmatrix} y_R \\ \varphi_R \\ v \end{bmatrix} = \begin{bmatrix} F_R \\ 0 \\ q \end{bmatrix}.$$

These equations define the relationship between the tip deflection, voltage and tip force:

$$y_R = \frac{l^2(4F_R l + 6 d w e_{31} v)}{12EI} = \frac{l^2(4F_R l + 6 d w e_{31} v)}{E w d^3} = F_R \frac{4l^3}{E w d^3} + v \frac{6 e_{31} l^2}{E d^2}$$

$$y_{free} = \frac{6l^2 d w e_{31} v_{rated}}{12EI} = \frac{6 e_{31} l^2 v_{rated}}{E d^2}$$

$$F_{block} = -\frac{3 d w e_{31} v_{rated}}{2l}$$

The block computes the capacitance by assuming zero applied force:

$$C_{piezo} = \varepsilon \frac{l w}{d} + e_{31}^2 \frac{l d^2 w^2}{EI} = \varepsilon \frac{l w}{d} + e_{31}^2 \frac{12 l w}{E d} = \left( \varepsilon + 12 \frac{e_{31}^2}{E} \right) \frac{l w}{d}.$$

Finally, this equation shows the relationship between density and mass:

$$m = \rho l w d.$$

Once you have defined all the relationships between the fundamental and datasheet parameters, you can calculate the fundamental parameters with these equations:

$$e_{31} = -\frac{2 l F_{block}}{3 d w v_{rated}}$$

$$E = -\frac{4 F_{block} l^3}{y_{free} d^3 w}$$

$$\varepsilon = \frac{d}{l w} \left( C_{piezo} + \frac{4 F_{block} y_{free}}{v_{rated}^2} \right)$$

Then substitute these equations in the constitutive equations:

$$\frac{v}{v_{rated}} = \frac{F_R}{F_{block}} + \frac{y_R}{y_{free}}$$

$$q = C_{piezo} v + \frac{y_{free}}{v_{rated}} F_R$$

### Dynamics Datasheet Parameterization

You can calculate the first resonance frequency of a clamped-free beam of uniform cross section by using this equation:

$$2\pi f_0 = 1.855^2 \sqrt{\frac{EI}{m l^3}}.$$



The block then parameterizes the dynamics directly by specifying the desired natural frequency.

### Boundary Conditions

Beams have different boundary conditions at their left and right ends:

- Free — Both the displacement and rotation are equal to any value.
- Simply supported — The displacement is equal to 0.
- Clamped — Both the displacement and rotation are equal to 0.

This table shows the possible boundary configurations for a piezo bender beam.

Configuration	Model
Clamped-free	
Supported-supported	
Clamped-clamped	

### Ports

#### Conserving

##### + — Positive terminal

electrical

Electrical conserving port associated with the piezo bender positive terminal.

##### - — Negative terminal

electrical

Electrical conserving port associated with the piezo bender negative terminal.

**Ctr — Translational case**

mechanical

Mechanical translational conserving port associated with the case.

**Cro — Rotational case**

mechanical

Mechanical rotational conserving port associated with the case.

**Rtr — Translational rotor**

mechanical

Mechanical translational conserving port associated with the rotor.

**Rro — Rotational rotor**

mechanical

Mechanical rotational conserving port associated with the rotor.

**Parameters****Dimensions****Number of elements — Number of elements**

1 (default) | positive integer greater than or equal to 1

Number of elements in which the piezo bender beam is discretized using the Euler-Bernoulli finite elements equations.

**Total beam length — Total beam length**

3.175E-2 m (default) | positive scalar

Total length of the beam.

**Beam width — Beam width**

1.27E-2 m (default) | positive scalar

Width of the beam.

**Beam thickness — Beam thickness**

5.08E-4 m (default) | positive scalar

Thickness of the beam.

**Steady-State****Parameterization — Steady-state parameterization**

Specify from a datasheet for clamped-free configuration (default) | Specify from material properties

Whether to parameterize the steady state parameters from a datasheet or from material properties.

**Capacitance — Piezo bender capacitance**

15 nF (default) | positive scalar

Capacitance of the piezo bender.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet for clamped-free configuration.

**Rated drive voltage,  $V_{rated}$  — Rated drive voltage**

180 V (default) | positive scalar

Rated drive voltage.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet for clamped-free configuration.

**Free deflection at  $V_{rated}$  — Free deflection at  $V_{rated}$**

+0.233 mm (default) | nonzero value

Tip deflection when no force is applied. The sign of this parameter must be different from the sign of the **Blocking force at  $V_{rated}$**  parameter.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet for clamped-free configuration.

**Blocking force at  $V_{rated}$  — Blocking force at  $V_{rated}$**

-0.414 N (default) | scalar

Force that nullifies the tip deflection. The sign of this parameter must be different from the sign of the **Free deflection at  $V_{rated}$**  parameter.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from a datasheet for clamped-free configuration.

**Young modulus,  $E$  — Young modulus**

1.36628e11 N/m<sup>2</sup> (default) | positive scalar

Young modulus, or the modulus of elasticity in tension.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from material properties.

**Piezoelectric stress coefficient,  $e_{31}$  — Piezoelectric stress coefficient**

7.5459 C/m<sup>2</sup> (default) | nonzero scalar

(3,1) piezoelectric stress-charge coupling coefficient.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from material properties.

**Dielectric constant — Dielectric constant**

1.38965e-08 F/m (default) | positive scalar

Dielectric constant that represents the relative permittivity of the material. This parameter must be bigger than the vacuum permittivity.

**Dependencies**

To enable this parameter, set **Parameterization** to Specify from material properties.

**Dynamics****Inertial effects – Inertial effects modeling option**

On (default) | Off

Whether to model inertial effects.

**Parameterization – Inertia parameterization**

Specify from a datasheet for clamped-free configuration (default) | Specify from material properties

Whether to parameterize the inertia parameters from a datasheet or from material properties.

**Dependencies**

To enable this parameter, set **Inertial effects** to On.

**Beam mass – Beam mass**

1.6 g (default) | positive scalar

Mass of the beam.

**Dependencies**

To enable this parameter, set **Inertial effects** to On and **Parameterization** to Specify from material properties.

**First natural frequency – First natural frequency**

391 Hz (default) | positive scalar

First natural frequency.

**Dependencies**

To enable this parameter, set **Inertial effects** to On and **Parameterization** to Specify from a datasheet for clamped-free configuration.

**Rayleigh damping constant proportional to stiffness – Rayleigh damping constant proportional to stiffness**

1e-5 s (default) | nonnegative scalar

Damping coefficient proportional to stiffness, for Rayleigh damping.

**Rayleigh damping constant proportional to mass – Rayleigh damping constant proportional to mass**

0 1/s (default) | nonnegative scalar

Damping coefficient proportional to mass, for Rayleigh damping.

## Dependencies

To enable this parameter, set **Inertial effects** to On.

## Version History

Introduced in R2021a

## References

- [1] Tadmor, E. B., and G. Kosa. "Electromechanical Coupling Correction for Piezoelectric Layered Beams." *Journal of Microelectromechanical Systems*, vol. 12, no. 6, Dec. 2003, pp. 899-906. DOI.org (Crossref), doi:10.1109/JMEMS.2003.820286.
- [2] Benjeddou A, Trindade MA, Ohayon R. "A Unified Beam Finite Element Model for Extension and Shear Piezoelectric Actuation Mechanisms." *Journal of Intelligent Material Systems and Structures*. 1997;8(12):1012-1025. doi:10.1177/1045389X9700801202
- [3] Gavin, Henri P. "Beam Element Stiffness Matrices." CEE 421L. *Matrix Structural Analysis*. Duke University, 2014.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

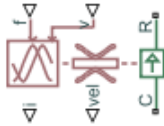
## See Also

Piezo Stack | Piezo Linear Actuator | Piezo Rotary Actuator

## Piezo Linear Actuator

Force-speed characteristics of linear piezoelectric traveling wave motor

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



### Description

The Piezo Linear Actuator block represents the force-speed characteristics of a linear piezoelectric traveling wave motor. The block represents the force-speed relationship of the motor at a level that is suitable for system-level modeling. To simulate the motor, the block uses the following models:

- “Mass and Friction Model for Unpowered Motor” on page 1-1376
- “Resonant Circuit Model for Powered Motor” on page 1-1376

#### Mass and Friction Model for Unpowered Motor

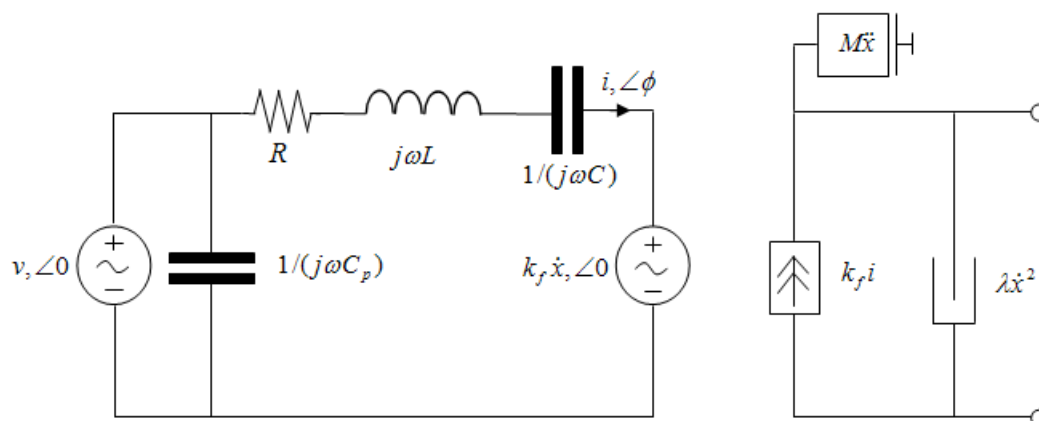
The motor is unpowered when the physical signal input  $v$  is zero. This corresponds to applying zero RMS volts to the motor. In this scenario, the block models the motor using the following elements:

- A mass whose value is the **Plunger mass** parameter value.
- A friction whose characteristics you specify using the parameter values in the **Motor-Off Friction** tab.

The block uses a SimscapeTranslational Friction block to model the friction component. For detailed information about the friction model, see the Translational Friction block reference page.

#### Resonant Circuit Model for Powered Motor

When the motor is active, Piezo Linear Actuator block represents the motor characteristics using the following equivalent circuit model.



In the preceding figure:

- The AC voltage source represents the block's physical signal input of frequency  $f$  and magnitude  $v$ .
- The resistor  $R$  provides the main electrical and mechanical damping term.
- The inductor  $L$  represents the rotor vibration inertia.
- The capacitor  $C$  represents the piezo crystal stiffness.
- The capacitor  $C_p$  represents the phase capacitance. This is the electrical capacitance associated with each of the two motor phases.
- The force constant  $k_f$  relates the RMS current  $i$  to the resulting mechanical force.
- The quadratic mechanical damping term,  $\lambda\dot{x}^2$ , shapes the force-speed curve predominantly at speeds close to maximum RPM.  $\dot{x}$  is the linear speed.
- The term  $M\dot{x}$  represents the plunger inertia.

At model initialization, the block calculates the model parameters  $R$ ,  $L$ ,  $C$ ,  $k_t$  and  $\lambda$  to ensure that the steady-state force-speed curve matches the values for the following user-specified parameters:

- **Rated force**
- **Rated speed**
- **No-load maximum speed**
- **Maximum (stall) force**

These parameter values are defined for the **Rated RMS voltage** and **Motor natural frequency** (or rated frequency) parameter values.

The quadratic mechanical damping term produces a quadratic force-speed curve. Piezoelectric motors force-speed curves can typically be approximated more accurately using a quadratic function than a linear one because the force-speed gradient becomes steeper as the motor approaches the maximum speed.

If the plunger mass  $M$  is not specified on the datasheet, you can select a value that provides a good match to the quoted response time. The response time is often defined as the time for the rotor to reach maximum speed when starting from rest, under no-load conditions.

The quality factor that you specify using the **Resonance quality factor** parameter relates to the equivalent circuit model parameters as follows:

$$Q = \frac{1}{R} \sqrt{\frac{L}{C}}$$

This term is not usually provided on a datasheet. You can calculate its value by matching the sensitivity of force to driving frequency.

To reverse the motor direction of operation, make the physical signal input  $v$  negative.

## Assumptions and Limitations

- When the motor is powered, the model is valid only between zero and maximum speed, for the following reasons:
  - Datasheets do not provide information for operation outside of normal range.

- Piezoelectric motors are not designed to operate in the powered braking and generating regions.

The block behaves as follows outside the valid operating region:

- Below zero speed, the model maintains a constant force with a zero speed value. The zero speed value is the **Maximum (stall) force** parameter value if the RMS input voltage equals the **Rated RMS voltage** parameter value, and the frequency input equals the **Motor natural frequency** parameter value.
- Above maximum speed, the model produces the negative force predicted by the equivalent circuit model, but limits the absolute value of the force to the zero-speed maximum force.
- The force-speed characteristics are most representative when operating the model close to the rated voltage and resonant frequency.

## Ports

### Input

#### **f** – Motor driving frequency

physical

Physical signal input port associated with the motor driving frequency, in Hz.

#### **v** – RMS supply voltage and direction of rotation

physical

Physical signal input magnitude port specifying the RMS supply voltage, and sign specifying the direction of rotation. If  $v$  is positive, then a positive force acts from port **C** to port **R**.

### Output

#### **i** – RMS phase current

physical

Physical signal output port associated with the RMS phase current.

#### **vel** – Rotor speed

physical

Physical signal output port associated with the linear speed of the rotor.

### Conserving

#### **C** – Case

mechanical

Mechanical translational conserving port associated with the motor case.

#### **R** – Rotor

mechanical

Mechanical translational conserving port associated with the motor rotor.



## Parameters

### Electrical Force

#### Motor natural frequency — Motor natural frequency

92 kHz (default)

Frequency at which the piezoelectric crystal naturally resonates. For most applications, set the input signal at port f to this frequency. To slow down the motor, for example in a closed-loop speed control, use a frequency slightly less than the motor natural frequency.

#### Rated RMS voltage — Rated RMS voltage

5.7 V (default)

Voltage at which the motor is designed to operate.

#### Rated force — Rated force

0.1 N (default)

Force the motor delivers at the rated RMS voltage.

#### Rated speed — Rated speed

50 mm/s (default)

Motor speed when the motor drives a load at the rated force.

#### No-load maximum speed — No-load maximum speed

150 mm/s (default)

Motor speed when driving no load and powered at the rated voltage and driving frequency.

#### Maximum (stall) force — Maximum force

0.15 N (default)

Maximum force the motor delivers when actively driving a load and powered at the rated voltage and frequency.

---

**Note** The **Holding force** parameter value, the load force the motor holds when stationary, may be greater than the **Maximum (stall) force** parameter value.

---

#### Resonance quality factor — Resonance quality factor

100 (default)

Quality factor  $Q$  that specifies how force varies as a function of driving frequency. Increasing the quality factor results in a much more rapid decrease in force as driving frequency is moved away from the natural frequency.

#### Capacitance per phase — Capacitance per phase

5 nF (default)

Electrical capacitance associated with each of the two motor phases.

## Mechanical

### Plunger mass — Plunger mass

0.3 g (default)

Mass of the moving part of the motor.

### Motor-Off Friction

#### Holding force — Holding force

0.3 N (default)

The sum of the Coulomb and the static frictions. It must be greater than or equal to the **Coulomb friction force** parameter value.

#### Coulomb friction force — Coulomb friction force

0.15 N (default)

The friction that opposes rotation with a constant force at any velocity.

#### Viscous friction coefficient — Viscous friction coefficient

1e-5 s\*N/mm (default) | nonnegative scalar

Proportionality coefficient between the friction force and the relative velocity.

#### Transition approximation coefficient — Transition approximation coefficient

0.1 s/mm (default)

The parameter sets the coefficient value that is used to approximate the transition between the static and the Coulomb frictions. For detailed information about the coefficient,  $c_v$ , see the SimscapeTranslational Friction block reference page.

#### Linear region velocity threshold — Linear region velocity threshold

0.1 mm/s (default)

The parameter sets the small vicinity near zero velocity, within which friction force is considered to be linearly proportional to the relative velocity. MathWorks recommends that you use values between 1e-6 and 1e-4 mm/s.

## Version History

Introduced in R2009a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

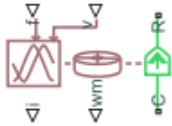
## See Also

Piezo Rotary Actuator

# Piezo Rotary Actuator

Torque-speed characteristics of rotary piezoelectric traveling wave motor

**Library:** Simscape / Electrical / Electromechanical / Mechatronic Actuators



## Description

The Piezo Rotary Actuator block represents the torque-speed characteristics of a piezoelectric traveling wave motor. The block represents the torque-speed relationship of the motor at a level that is suitable for system-level modeling. To simulate the motor, the block uses the following models:

- “Inertia and Friction Model for Unpowered Motor” on page 1-1381
- “Resonant Circuit Model for Powered Motor” on page 1-1381

### Inertia and Friction Model for Unpowered Motor

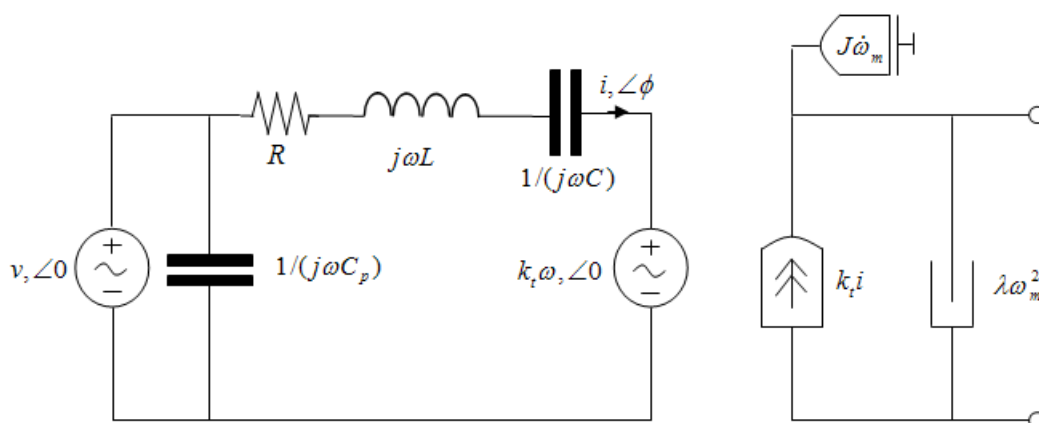
The motor is unpowered when the physical signal input  $v$  is zero. This corresponds to applying zero RMS volts to the motor. In this scenario, the block models the motor using the following elements:

- An inertia whose value is the **Rotor inertia** parameter value.
- A friction whose characteristics are determined by the parameter values in the **Motor-Off Friction** tab.

The block uses a SimscapeRotational Friction block to model the friction component. For detailed information about the friction model, see the Rotational Friction block reference page.

### Resonant Circuit Model for Powered Motor

When the motor is active, Piezo Rotary Actuator block represents the motor characteristics using the following equivalent circuit model.



In the preceding figure:

- The AC voltage source represents the block's physical signal input of frequency  $f$  and magnitude  $v$ .
- The resistor  $R$  provides the main electrical and mechanical damping term.
- The inductor  $L$  represents the rotor vibration inertia.
- The capacitor  $C$  represents the piezo crystal stiffness.
- The capacitor  $C_p$  represents the phase capacitance. This is the electrical capacitance associated with each of the two motor phases.
- The torque constant  $k_t$  relates the RMS current  $i$  to the resulting mechanical torque.
- The quadratic mechanical damping term,  $\lambda\omega_m^2$ , shapes the torque-speed curve predominantly at speeds close to maximum RPM.  $\omega_m$  is the mechanical rotational speed.
- The term  $J\dot{\omega}_m$  represents the rotor inertia.

At model initialization, the block calculates the model parameters  $R$ ,  $L$ ,  $C$ ,  $k_t$  and  $\lambda$  to ensure that the steady-state torque-speed curve matches the values of the following user-specified parameter values:

- **Rated torque**
- **Rated rotational speed**
- **No-load maximum rotational speed**
- **Maximum torque**

These parameter values are defined for the **Rated RMS voltage** and **Motor natural frequency** (or rated frequency) parameter values.

The quadratic mechanical damping term produces a quadratic torque-speed curve. Piezoelectric motors torque-speed curves can typically be approximated more accurately using a quadratic function than a linear one because the torque-speed gradient becomes steeper as the motor approaches the maximum speed.

If the rotor inertia  $J$  is not specified on the datasheet, you can select a value that provides a good match to the quoted response time. The response time is often defined as the time for the rotor to reach maximum speed when starting from rest, under no-load conditions.

The quality factor that you specify using the **Resonance quality factor** parameter relates to the equivalent circuit model parameters as follows:

$$Q = \frac{1}{R} \sqrt{\frac{L}{C}}$$

This term is not usually provided on a datasheet. You can calculate its value by matching the sensitivity of torque to driving frequency.

To reverse the motor direction of operation, make the physical signal input  $v$  negative.

## Assumptions and Limitations

- When the motor is powered, the model is valid only between zero and maximum speed, for the following reasons:
  - Datasheets do not provide information for operation outside of normal range.

- Piezoelectric motors are not designed to operate in the powered braking and generating regions.

The block behaves as follows outside the valid operating region:

- Below zero speed, the model maintains a constant torque that is the zero rpm torque value. The zero rpm torque value is the **Maximum torque** parameter value if the RMS input voltage equals the **Rated RMS voltage** parameter value, and the frequency input equals the **Motor natural frequency** parameter value.
- Above maximum speed, the model produces the negative torque predicted by the equivalent circuit model, but limits the absolute value of the torque to the zero-speed maximum torque.
- The torque-speed characteristics are most representative when operating the model close to the rated voltage and resonant frequency.

## Ports

### Input

#### **f – Motor driving frequency**

physical

Physical signal input port associated with the motor driving frequency, in Hz.

#### **v – RMS supply voltage and direction of rotation**

physical

Physical signal input magnitude port specifying the RMS supply voltage, and sign specifying the direction of rotation. If  $v$  is positive, then a positive torque acts from port **C** to port **R**.

### Output

#### **i – RMS phase current**

physical

Physical signal output port associated with the RMS phase current.

#### **$\omega_m$ – Rotor speed**

physical

Physical signal output port associated with the rotational speed of the rotor.

### Conserving

#### **C – Case**

mechanical

Mechanical translational conserving port associated with the motor case.

#### **R – Rotor**

mechanical

Mechanical translational conserving port associated with the motor rotor.

## Parameters

### Electrical Torque

#### Motor natural frequency — Motor natural frequency

40 kHz (default)

Frequency at which the piezoelectric crystal naturally resonates. For most applications, set the input signal at port **f** to this frequency. To slow down the motor, for example in a closed-loop speed control, use a frequency slightly less than the motor natural frequency. The default value is 40kHz.

#### Rated RMS voltage — Rated RMS voltage

130 V (default)

Voltage at which the motor is designed to operate.

#### Rated torque — Rated torque

0.5 N\*m (default)

Torque the motor delivers at the rated RMS voltage.

#### Rated rotational speed — Rated rotational speed

100 rpm (default)

Motor speed when the motor drives a load at the rated torque.

#### No-load maximum rotational speed — No-load maximum rotational speed

160 rpm (default)

Motor rotational speed when driving no load and powered at the rated voltage and driving frequency.

#### Maximum torque — Maximum torque

1 N\*m (default)

Maximum torque that the motor delivers when actively driving a load and powered at the rated voltage and frequency. The default value is 1N\*m.

---

**Note** The **Holding torque** parameter value, the load torque the motor holds when stationary, may be greater than the **Maximum torque** parameter value.

---

#### Resonance quality factor — Resonance quality factor

100 (default)

Quality factor  $Q$  that specifies how torque varies as a function of driving frequency. Increasing the quality factor results in a much more rapid decrease in torque as driving frequency is moved away from the natural frequency.

#### Capacitance per phase — Capacitance per phase

5 nF (default)

Electrical capacitance associated with each of the two motor phases.

## Mechanical

### Rotor inertia – Rotor inertia

200 g\*cm<sup>2</sup> (default)

Rotor resistance to change in motor motion.

### Motor-Off Friction

#### Holding torque – Holding torque

1.5 N\*m (default)

The sum of the Coulomb and the static frictions. It must be greater than or equal to the **Coulomb friction torque** parameter value.

#### Coulomb friction torque – Coulomb friction torque

1.0 N\*m (default)

The friction that opposes rotation with a constant torque at any velocity.

#### Viscous friction coefficient – Viscous friction coefficient

0.001 N\*m/(rad\*s) (default) | nonnegative scalar

Proportionality coefficient between the friction torque and the relative angular velocity.

#### Transition approximation coefficient – Transition approximation coefficient

10 s/rad (default)

The parameter sets the coefficient value that is used to approximate the transition between the static and the Coulomb frictions. For detailed information about the coefficient,  $c_v$ , see the SimscapeRotational Friction block reference page.

#### Linear region velocity threshold – Linear region velocity threshold

1e-4 rad/s (default)

The parameter sets the small vicinity near zero velocity, within which friction torque is considered to be linearly proportional to the relative velocity. MathWorks recommends that you use values in the range between 1e-5 and 1e-3rad/s.

## Version History

Introduced in R2009a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

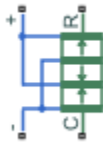
## See Also

Piezo Linear Actuator

## Piezo Stack

Electrical and force characteristics of piezoelectric stacked actuator

**Library:** Simscape / Electrical / Electromechanical / Mechatronics / Actuators



### Description

The Piezo Stack block represents the electrical and force characteristics of a piezoelectric stacked actuator using the following equations:

$$S = s^E T + d' E$$

$$D = d T + \varepsilon^T E$$

where

- $S$  is the strain tensor.
- $T$  is the stress tensor.
- $E$  is the electric field vector.
- $D$  is the electric displacement vector.
- $s^E$  is the elastic compliance matrix when subjected to a constant electric field.
- $d$  is the piezoelectric constant matrix.
- $\varepsilon^T$  is the permittivity measured at a constant stress.

---

**Note** The block models one-dimensional lumped parameter behavior, so  $S$ ,  $T$ ,  $E$  and  $D$  are all scalar values.

---

You can specify the block parameters that determine static force using either datasheet parameters or material properties, as determined by the value of the **Parameterization** parameter on the **Static Force** tab of the block dialog box.

The **Dynamic Forces** tab of the block dialog box lets you include optional effective mass and mechanical damping effects.

- If you specify a nonzero value for the **Effective mass** parameter or a finite value for the **Resonant frequency at constant field** parameter, the block attaches a lumped mass to the mechanical R port. When you specify a finite resonant frequency, the block calculates the effective mass to achieve the correct resonant frequency.
- If you specify a nonzero value for the **Damping** parameter or a finite value for the **Mechanical quality factor** parameter, the block adds a damping term across the **R** to **C** mechanical ports.



When you specify a mechanical quality factor,  $Q_m$ , the block calculates the damping from this parameter value as  $\sqrt{Mk}/Q_m$ , where  $k$  is the short-circuit device stiffness, or equivalently the stiffness at constant field.

A positive voltage across the electrical + to - ports creates a positive displacement acting from the mechanical **C** to **R** ports.

## Assumptions and Limitations

- The model does not include hysteresis effects.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the piezo stack positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the piezo stack negative terminal.

#### **C** — Case

mechanical

Mechanical translational conserving port associated with the case.

#### **R** — Rotor

mechanical

Mechanical translational conserving port associated with the rotor.

## Parameters

### Static Force

#### Parameterization — Static force parameterization

Specify from a datasheet (default) | Specify material properties

Select one of the following methods for static force parameterization:

- Specify from a datasheet — Provide datasheet parameters that the block converts to static force values.
- Specify material properties — Provide material properties that the block converts to static force values.

#### Stack area — Cross-sectional stack area

100 mm<sup>2</sup> (default)

Cross-sectional area of the stack.

**Stack length — Stack length**

36 mm (default)

Stack length when no load and no electrical potential are applied.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**No-load displacement at V0 volts — No-load unconstrained displacement**

0.038 (default)

Unconstrained displacement of the stack when a voltage of V0 volts is applied.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Blocking force at V0 volts — Blocking force**

3800 N (default)

Force the stack produces when a voltage of V0 volts is applied and the stack is physically prevented from expanding.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Test voltage V0 — Test voltage**

120 V (default)

Voltage used to determine the no-load displacement and blocking force.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Capacitance — Capacitance**

13 uF (default)

Capacitance.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Piezo layer thickness — Piezo layer thickness**

0.3 mm (default)

Thickness of each layer in the piezo stack.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Number of layers — Number of layers**

50 (default)

Number of layers in the piezo stack.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Piezoelectric charge constant — Piezoelectric charge constant**

500e-12 m/V (default)

Mechanical strain per unit electric field applied.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Dielectric constant — Dielectric constant**

2.1240e-08 F/m (default)

Permittivity or dielectric displacement per unit electric field measured at constant stress.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Elastic compliance — Elastic compliance**

19e-12 m<sup>2</sup>/N (default)

Strain produced in a piezoelectric material per unit of stress applied.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Dynamic Forces****Parameterization — Dynamic force parameterization**

Specify from a datasheet (default) | Specify material properties

Select one of the following methods for dynamic force parameterization:

- **Specify from a datasheet** — Provide datasheet parameters that the block converts to dynamic force values.
- **Specify material properties** — Provide material properties that the block converts to dynamic force values.

**Resonant frequency at constant field — Resonant frequency at constant field**

inf kHz (default)

Frequency at which the actuator naturally resonates if mechanically perturbed with the electrical ports shorted.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Mechanical quality factor — Mechanical quality factor**

inf (default)

Factor that affects the damping across the **R** and **C** mechanical ports.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify from a datasheet.

**Damping — Translational damping**

0 N/(m/s) (default)

Translational damping term.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

**Effective mass — Effective mass**

0 g (default)

Mass that approximates the distributed dynamics of the device and causes the stack to resonate at the correct frequency when attached to the mechanical R port. This mass is usually about one third of the actual stack mass.

**Dependencies**

This parameter is visible only when you set the **Parameterization** parameter to Specify material properties.

## Version History

Introduced in R2008b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Piezo Linear Actuator | Piezo Rotary Actuator

# PMLSM

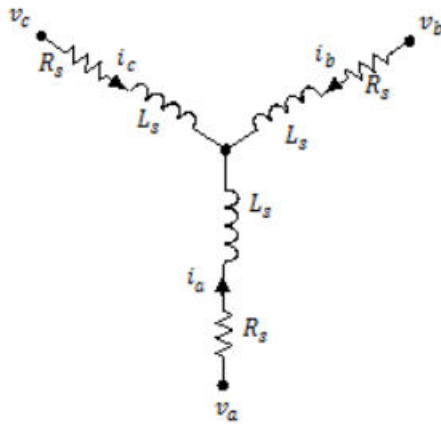
Permanent magnet linear synchronous motor with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet

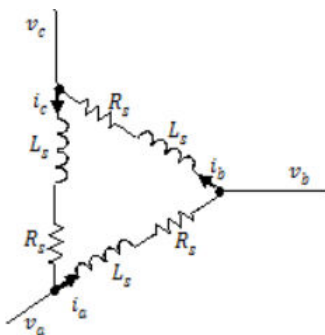


## Description

The PMLSM block models a permanent magnet linear synchronous motor with a three-phase wye-wound stator. The figure shows the equivalent electrical circuit for the stator windings.

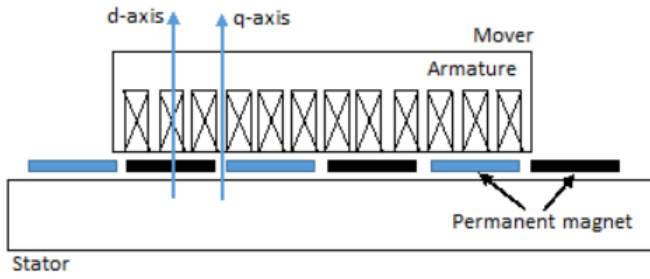


You can also model the permanent magnet linear synchronous motor in a delta-wound configuration by setting **Winding type** to Delta-wound.



## Motor Construction

This figure shows the motor construction.



### Equations

Voltages across the stator windings are defined by:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ , and  $\frac{d\psi_c}{dt}$  are the rates of change of magnetic flux in each stator winding.

The permanent magnet and the three windings contribute to the total flux linking each winding. The total flux is defined by:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix},$$

where:

- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the total fluxes linking each stator winding.
- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ ,  $L_{bc}$ ,  $L_{ca}$ , and  $L_{cb}$  are the mutual inductances of the stator windings.
- $\psi_{am}$ ,  $\psi_{bm}$ , and  $\psi_{cm}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of electrical angle, and are defined by:

$$\theta_e = N_p x + \text{rotor offset},$$

$$L_{aa} = L_s + L_m \cos(2\theta_e)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3))$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_e + \pi/6))$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/6 - 2\pi/3))$$

and

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_e + \pi/6 + 2\pi/3))$$

where:

- $\theta_e$  is the electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the d-axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the q-axis.
- $N_p = \frac{\pi}{\tau}$  and  $\tau$  is the polar pitch.
- $x$  is the distance.
- $L_s$  is the per-phase stator self-inductance. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the fluctuation in self-inductance and mutual inductance with changing angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The permanent magnet flux linking winding  $a$  is a maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore, the linked motor flux is defined by:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_e - 2\pi/3) \\ \psi_m \cos(\theta_e + 2\pi/3) \end{bmatrix}$$

where  $\psi_m$  is the permanent magnet flux linkage.

### Simplified Electrical Equations

Applying Park's transformation to the electrical equations produces an expression for force that is independent of the angle.

Park's transformation is defined by:

$$P = 2/3 \begin{bmatrix} \cos \theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin \theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

where  $\theta_e$  is the electrical angle defined as  $N_p x$ .

Using Park's transformation on the stator winding voltages and currents transforms them into the dq0 frame, which is independent of the angle:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

and



$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

Applying Park's transformation to the first two electrical equations produces the following equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N_p v i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N_p v (i_d L_d + \psi_m),$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

and

$$F = \frac{3}{2} N_p (i_q (i_d L_d + \psi_m) - i_d i_q L_q),$$

$$M \frac{dv}{dt} = F - F_L - B_m v,$$

where:

- $L_d = L_s + M_s + 3/2 L_m$ .  $L_d$  is the stator  $d$ -axis inductance.
- $L_q = L_s + M_s - 3/2 L_m$ .  $L_q$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 2M_s$ .  $L_0$  is the stator zero-sequence inductance.
- $R_s$  is the stator resistance per phase.
- $v$  is the linear speed.
- $N_p$  is the polar pitch factor.
- $M$  is the mass of the mover.
- $B_m$  is the damping.
- $F_L$  is the load force.

The PMLSM block uses the original, nonorthogonal implementation of the Park transform. If you try to apply the alternative implementation, you get different results for the dq0 voltage and currents.

The relationship between the force constant  $k_f$ , the back-emf constant  $k_e$ , and the permanent magnet flux linkage is defined as follows:

$$k_e = k_f = N_p \psi_m.$$

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and motor temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

~ — **Three-phase port**  
electrical

Expandable three-phase port.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**n – Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

**Dependencies**

To enable this port, set **Winding Type** to Wye-wound and **Zero sequence** to Include.

**R – Motor mover**

mechanical

Mechanical translational conserving port associated with the motor mover.

**C – Motor case**

mechanical

Mechanical translational conserving port associated with the motor case.

**HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Winding type — Windings configuration**

Wye-wound (default) | Delta-wound

Configuration for the windings:

- Wye-wound — The windings are wye-wound.
- Delta-wound — The windings are delta-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c** and the *c*-phase between ports **c** and **a**.

**Modeling fidelity — Modeling fidelity**

Constant Ld, Lq and PM (default) | Tabulated Ld, Lq and PM

Modeling fidelity:

- **Constant  $L_d$ ,  $L_q$  and  $PM$**  — The  $L_d$ ,  $L_q$ , and  $PM$  values are constant and defined by their respective parameters.
- **Tabulated  $L_d$ ,  $L_q$  and  $PM$**  — The  $L_d$ ,  $L_q$ , and  $PM$  values are computed online from DQ current lookup tables as follows:

$$L_d = f_1(i_d, i_q)$$

$$L_q = f_2(i_d, i_q)$$

$$\lambda_{PM} = f_3(i_d, i_q)$$

#### Dependencies

To enable this parameter, set **Stator parameterization** to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .

#### Polar pitch — Polar pitch

0.03 m (default) | scalar

Polar pitch.

#### Permanent magnet flux linkage parameterization — Permanent magnet flux linkage parameterization

Specify flux linkage (default) | Specify force constant | Specify back EMF constant

Permanent magnet flux linkage, specified as Specify flux linkage, Specify force constant, or Specify back EMF constant.

#### Permanent magnet flux linkage — Permanent magnet flux linkage

0.03 Wb (default) | scalar

Peak permanent magnet flux linkage with any of the stator windings.

#### Dependencies

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify flux linkage and **Modeling fidelity** to Constant  $L_d$ ,  $L_q$  and  $PM$ .

#### Force constant — Force constant

0.18 N/A (default) | scalar

Force constant for any of the stator windings.

#### Dependencies

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify force constant and **Modeling fidelity** to Constant  $L_d$ ,  $L_q$  and  $PM$ .

#### Back EMF constant — Back EMF constant

0.18 s\*V/m (default) | scalar

Back EMF constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify back EMF constant and **Modeling fidelity** to Constant Ld, Lq and PM.

**Stator parameterization — Stator parameterization**

Specify Ld, Lq, and L0 (default) | Specify Ls, Lm, and Ms

Stator parameterization, specified as Specify Ld, Lq, and L0 or Specify Ls, Lm, and Ms.

**Stator d-axis inductance, Ld — Stator d-axis inductance**

0.00019 H (default) | scalar

d-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Constant Ld, Lq and PM.

**Stator q-axis inductance, Lq — Stator q-axis inductance**

0.00025 H (default) | scalar

q-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Constant Ld, Lq and PM.

**Stator zero-sequence inductance, L0 — Stator zero-sequence inductance**

0.00016 H (default) | scalar

Zero-sequence inductance.

**Dependencies**

To enable this parameter either:

- Set **Winding Type** to Wye-wound, **Zero sequence** to Include, and **Stator parameterization** to Specify Ld, Lq, and L0.
- Set **Winding Type** to Delta-wound and **Stator parameterization** to Specify Ld, Lq, and L0.

**Direct-axis current vector, iD — Direct-axis current vector**

[-200, 0, 200] A (default) | vector

Direct-axis current vector, iD.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld, Lq and PM.

**Quadrature-axis current vector, iQ — Quadrature-axis current vector**

[-200, 0, 200] A (default) | vector

Quadrature-axis current vector,  $i_Q$ .

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld, Lq and PM.

**Ld matrix,  $L_d(i_d, i_q)$  – Ld matrix**

0.00019 \* ones(3, 3) H(default) | matrix

Ld matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld, Lq and PM.

**Lq matrix,  $L_q(i_d, i_q)$  – Lq matrix**

0.00025 \* ones(3, 3) H(default) | matrix

Lq matrix.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0 and **Modeling fidelity** to Tabulated Ld, Lq and PM.

**Permanent magnet flux linkage,  $PM(i_d, i_q)$  – Permanent magnet flux linkage**

0.03 \* ones(3, 3) Wb(default)

Permanent magnet flux linkage.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify Ld, Lq, and L0
- **Modeling fidelity** to Tabulated Ld, Lq and PM
- **Permanent magnet flux linkage parameterization** to Specify flux linkage

**Force constant matrix,  $k_f(i_d, i_q)$  – Force constant matrix**

0.18 \* ones(3, 3) N/A(default) | matrix

Force constant matrix.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify Ld, Lq, and L0
- **Modeling fidelity** to Tabulated Ld, Lq and PM
- **Permanent magnet flux linkage parameterization** to Specify force constant

**Back EMF constant matrix,  $k_e(i_d, i_q)$  – Back EMF constant matrix**

0.18 \* ones(3, 3) s\*V/m(default) | matrix

Back EMF constant matrix.

**Dependencies**

To enable this parameter, set:

- **Stator parameterization** to Specify Ld, Lq, and L0
- **Modeling fidelity** to Tabulated Ld, Lq and PM
- **Permanent magnet flux linkage parameterization** to Specify back EMF constant

**Stator self-inductance per phase, Ls — Stator self-inductance per phase**

0.0002 H (default) | scalar

Average self-inductance of each of the five stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator inductance fluctuation, Lm — Stator inductance fluctuation**

-0.00002 H (default) | scalar

Fluctuation in self-inductance and mutual inductance of the stator windings with angle.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator mutual inductance, Ms — Stator mutual inductance**

0.00002 H (default) | scalar

Average mutual inductance between the stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator resistance per phase, Rs — Stator resistance per phase**

0.013 Ohm (default) | scalar

Resistance of each of the stator windings.

**Zero sequence — Zero sequence option**

Include (default) | Exclude

Option to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. To prioritize model fidelity, use this default setting. Using this option:
  - Results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
  - Exposes a zero-sequence parameter.
- **Exclude** — Exclude zero-sequence terms. To prioritize simulation speed for desktop simulation or real-time deployment, select this option.

**Dependencies**

To enable this parameter, set **Winding Type** to Wye-wound.



**Angle definition — Reference point for angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the angle measurement. The default value is Angle between the a-phase magnetic axis and the d-axis. This definition is shown in the “Motor Construction” on page 1-1314 figure. When you select this value, the mover and a-phase fluxes are aligned when the mover angle is zero.

The other value you can choose for this parameter is Angle between the a-phase magnetic axis and the q-axis. When you select this value, the a-phase current generates maximum force when the mover angle is zero.

**Mechanical****Mass of the mover — Mass of the mover**

0.01 kg (default)

Mass of the motor mover.

**Linear damping — Linear damping**

0 N/(m/s) (default)

Linear damping.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature — Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Permanent magnet flux temperature coefficient — Permanent magnet flux temperature coefficient**

-0.001 1/K (default)

Fractional rate of change of permanent magnet flux density with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal mass for each stator winding — Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass — Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

## Version History

### Introduced in R2020a

### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. IEEE Press Power Systems Engineering Series, 1995. ISBN 0-7803-1145-0.
- [3] X. Zhang and J. Pan. *Nonlinear Robust Sliding Mode Control for PM Linear Synchronous Motors*. 2006 CES/IEEE 5th International Power Electronics and Motion Control Conference, Shanghai, 2006, pp. 1-5.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

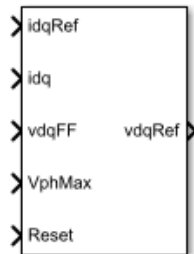
## See Also

PMSM | Hybrid Excitation PMSM | PMSM (Single-Phase)

# PMSM Current Controller

Discrete-time permanent magnet synchronous machine current controller

**Library:** Simscape / Electrical / Control / PMSM Control



## Description

The PMSM Current Controller block implements a discrete-time PI-based permanent magnet synchronous machine (PMSM) current controller in the rotor  $d$ - $q$  reference frame.

You typically use this block in a series of blocks making up a control structure.

- You can generate a current reference in the  $d$ - $q$  frame to be used as an input to this block with a PMSM Current Reference Generator.
- You can obtain a voltage reference in the  $abc$  domain by converting the output of this block using an Inverse Park Transform block.

You can see an example of a full control structure, from machine measurements to machine inputs, in the PMSM Field-Oriented Control block.

## Equations

The block is discretized using the backward Euler method due to its first-order simplicity and its stability.

Two PI current controllers implemented in the rotor reference frame produce the reference voltage vector:

$$v_d^{ref} = \left( K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) (i_d^{ref} - i_d) + v_{d\_FF},$$

and

$$v_q^{ref} = \left( K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) (i_q^{ref} - i_q) + v_{q\_FF},$$

where:

- $v_d^{ref}$  and  $v_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference voltages, respectively.
- $i_d^{ref}$  and  $i_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference currents, respectively.
- $i_d$  and  $i_q$  are the  $d$ -axis and  $q$ -axis currents, respectively.

- $K_{p\_id}$  and  $K_{p\_iq}$  are the proportional gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $K_{i\_id}$  and  $K_{i\_iq}$  are the integral gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $v_{d\_FF}$  and  $v_{q\_FF}$  are the feedforward voltages for the  $d$ -axis and  $q$ -axis, respectively, obtained from the machine mathematical equations and provided as inputs.
- $T_s$  is the sample time of the discrete controller.

### Zero Cancellation

Using PI control results in a zero in the closed-loop transfer function, which can result in undesired overshoot in the closed-loop response. This zero can be canceled by introducing a zero-cancellation block in the feedforward path. The zero cancellation transfer functions in discrete time are:

$$G_{ZC\_id}(z) = \frac{\frac{T_s K_{i\_id}}{K_{p\_id}}}{z + \left( \frac{T_s - \frac{K_{p\_id}}{K_{i\_id}}}{\frac{K_{p\_id}}{K_{i\_id}}} \right)}$$

and

$$G_{ZC\_iq}(z) = \frac{\frac{T_s K_{i\_iq}}{K_{p\_iq}}}{z + \left( \frac{T_s - \frac{K_{p\_iq}}{K_{i\_iq}}}{\frac{K_{p\_iq}}{K_{i\_iq}}} \right)}$$

### Voltage Saturation

Saturation must be imposed when the stator voltage vector exceeds the voltage phase limit  $V_{ph\_max}$ :

$$\sqrt{v_d^2 + v_q^2} \leq V_{ph\_max}$$

where  $v_d$  and  $v_q$  are the  $d$ -axis and  $q$ -axis voltages, respectively.

In the case of axis prioritization, the voltages  $v_1$  and  $v_2$  are introduced, where:

- $v_1 = v_d$  and  $v_2 = v_q$  for  $d$ -axis prioritization.
- $v_1 = v_q$  and  $v_2 = v_d$  for  $q$ -axis prioritization.

The constrained (saturated) voltages  $v_1^{sat}$  and  $v_2^{sat}$  are obtained as follows:

$$v_1^{sat} = \min(\max(v_1^{unsat}, -V_{ph\_max}), V_{ph\_max})$$

and

$$v_2^{sat} = \min(\max(v_2^{unsat}, -V_{2\_max}), V_{2\_max}),$$

where:

- $v_1^{unsat}$  and  $v_2^{unsat}$  are the unconstrained (unsaturated) voltages.

- $v_{2\_max}$  is the maximum value of  $v_2$  that does not exceed the voltage phase limit, given by

$$v_{2\_max} = \sqrt{(V_{ph\_max})^2 - (v_1^{sat})^2}.$$

In the case that the direct and quadrature axes have the same priority (d-q equivalence) the constrained voltages are obtained as follows:

$$v_d^{sat} = \min(\max(v_d^{unsat}, -V_{d\_max}), V_{d\_max})$$

and

$$v_q^{sat} = \min(\max(v_q^{unsat}, -V_{q\_max}), V_{q\_max}),$$

where

$$V_{d\_max} = \frac{V_{ph\_max}|v_d^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}|v_q^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

### Integral Anti-Windup

An anti-windup mechanism is employed to avoid saturation of integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}(v_d^{sat} - v_d^{unsat})$$

and

$$K_{i\_iq} + K_{aw\_iq}(v_q^{sat} - v_q^{unsat}),$$

where  $K_{aw\_id}$  and  $K_{aw\_iq}$  are the anti-windup gains for the  $d$ -axis and  $q$ -axis, respectively.

### Assumptions

- The plant model for direct and quadrature axis can be approximated with a first-order system.
- This control solution is used only for permanent magnet synchronous motors with sinusoidal flux distribution and field windings.

## Ports

### Input

#### idqRef – Reference currents

vector

Desired  $d$ - and  $q$ -axis currents for control of a PMSM, in A.

Data Types: single | double

**idq — Measured currents**

vector

Actual  $d$ - and  $q$ -axis currents of the controlled PMSM, in A.Data Types: `single` | `double`**vdqFF — Feedforward voltages**

vector

Feedforward pre-control voltages, in V.

Data Types: `single` | `double`**VphMax — Maximum phase voltage**

scalar

Maximum allowable voltage in each phase, in V.

Data Types: `single` | `double`**Reset — External reset**

scalar

External reset signal (rising edge) for integrators.

Data Types: `single` | `double`**Output****vdqRef — Reference voltages**

vector

Desired  $d$ - and  $q$ -axis voltages for control of a PMSM, in V.Data Types: `single` | `double`**Parameters****Control Parameters****D-axis current proportional gain — D-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for direct-axis current control.

**D-axis current integral gain — D-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for direct-axis current control.

**D-axis current anti-windup gain — D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for direct-axis current control.

**Q-axis current proportional gain — Q-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for quadrature-axis current control.

**Q-axis current integral gain — Q-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for quadrature-axis current control.

**Q-axis current anti-windup gain — Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for quadrature-axis current control.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, you can specify the sample time explicitly.

**Axis prioritization — Axis prioritization for voltage limiter**

q-axis (default) | d-axis | d-q equivalence

Prioritize or maintain the ratio between the  $d$ - and  $q$ -axes when the block limits voltage.

**Enable zero cancellation — Feedforward zero-cancellation**

off (default) | on

Enable or disable zero-cancellation on the feedforward path.

**Enable pre-control voltage — Pre-control voltage**

on (default) | off

Enable or disable pre-control voltage.

## Version History

**Introduced in R2017b**

## References

- [1] Bernardes, T., V. F. Montagner, H. A. Gründling, and H. Pinheiro. "Discrete-time sliding mode observer for sensorless vector control of permanent magnet synchronous machine." *IEEE Transactions on Industrial Electronics*. Vol. 61, Number 4, 2014, pp. 1679-1691.
- [2] Carpiuc, S., and C. Lazar. "Fast real-time constrained predictive current control in permanent magnet synchronous machine-based automotive traction drives." *IEEE Transactions on Transportation Electrification*. Vol.1, Number 4, 2015, pp. 326-335.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

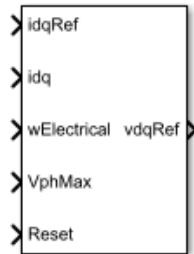
PMSM Current Controller with Pre-Control | PMSM Current Reference Generator



# PMSM Current Controller with Pre-Control

Discrete-time permanent magnet synchronous machine current controller with pre-control

**Library:** Simscape / Electrical / Control / PMSM Control



## Description

The PMSM Current Controller with Pre-Control block implements a discrete-time PI-based permanent magnet synchronous machine (PMSM) current controller in the rotor  $d$ - $q$  reference frame with internal feedforward pre-control.

You typically use this block in a series of blocks making up a control structure.

- You can generate a current reference in the  $d$ - $q$  frame to be used as an input to this block with a PMSM Current Reference Generator.
- You can obtain a voltage reference in the  $abc$  domain by converting the output of this block using an Inverse Park Transform block.

You can see an example of a full control structure, from machine measurements to machine inputs, in the PMSM Field-Oriented Control block.

## Equations

The block is discretized using the backward Euler method due to its first-order simplicity and its stability.

Two PI current controllers implemented in the rotor reference frame produce the reference voltage vector:

$$v_d^{ref} = \left( K_{p\_id} + K_{i\_id} \frac{T_{sZ}}{z-1} \right) (i_d^{ref} - i_d) + v_{d\_FF},$$

and

$$v_q^{ref} = \left( K_{p\_iq} + K_{i\_iq} \frac{T_{sZ}}{z-1} \right) (i_q^{ref} - i_q) + v_{q\_FF},$$

where:

- $v_d^{ref}$  and  $v_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference voltages, respectively.
- $i_d^{ref}$  and  $i_q^{ref}$  are the  $d$ -axis and  $q$ -axis reference currents, respectively.

- $i_d$  and  $i_q$  are the  $d$ -axis and  $q$ -axis currents, respectively.
- $K_{p\_id}$  and  $K_{p\_iq}$  are the proportional gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $K_{i\_id}$  and  $K_{i\_iq}$  are the integral gains for the  $d$ -axis and  $q$ -axis controllers, respectively.
- $T_s$  is the sample time of the discrete controller.
- $v_{d\_FF}$  and  $v_{q\_FF}$  are the feedforward voltages for the  $d$ -axis and  $q$ -axis, respectively.

The feedforward voltages are obtained from the machine mathematical equations:

$$v_{d\_FF} = -\omega_e L_q i_q^{ref},$$

and

$$v_{q\_FF} = \omega_e (L_d i_d^{ref} + \psi_m),$$

where:

- $\omega_e$  is the rotor electrical velocity.
- $L_d$  and  $L_q$  are the  $d$ -axis and  $q$ -axis inductances, respectively.
- $\psi_m$  is the permanent magnet flux linkage.

### Zero Cancellation

Using PI control results in a zero in the closed-loop transfer function, which can result in undesired overshoot in the closed-loop response. This zero can be canceled by introducing a zero-cancellation block in the feedforward path. The zero cancellation transfer functions in discrete time are:

$$G_{ZC\_id}(z) = \frac{\frac{T_s K_{i\_id}}{K_{p\_id}}}{z + \left( \frac{T_s - \frac{K_{p\_id}}{K_{i\_id}}}{\frac{K_{p\_id}}{K_{i\_id}}} \right)},$$

and

$$G_{ZC\_iq}(z) = \frac{\frac{T_s K_{i\_iq}}{K_{p\_iq}}}{z + \left( \frac{T_s - \frac{K_{p\_iq}}{K_{i\_iq}}}{\frac{K_{p\_iq}}{K_{i\_iq}}} \right)}.$$

### Voltage Saturation

Saturation must be imposed when the stator voltage vector exceeds the voltage phase limit  $V_{ph\_max}$ :

$$\sqrt{v_d^2 + v_q^2} \leq V_{ph\_max},$$

where  $v_d$  and  $v_q$  are the  $d$ -axis and  $q$ -axis voltages, respectively.

In the case of axis prioritization, the voltages  $v_1$  and  $v_2$  are introduced, where:

- $v_1 = v_d$  and  $v_2 = v_q$  for  $d$ -axis prioritization.
- $v_1 = v_q$  and  $v_2 = v_d$  for  $q$ -axis prioritization.

The constrained (saturated) voltages  $v_1^{sat}$  and  $v_2^{sat}$  are obtained as follows:

$$v_1^{sat} = \min(\max(v_1^{unsat}, -V_{ph\_max}), V_{ph\_max})$$

and

$$v_2^{sat} = \min(\max(v_2^{unsat}, -V_{2\_max}), V_{2\_max}),$$

where:

- $v_1^{unsat}$  and  $v_2^{unsat}$  are the unconstrained (unsaturated) voltages.
- $v_{2\_max}$  is the maximum value of  $v_2$  that does not exceed the voltage phase limit, given by

$$v_{2\_max} = \sqrt{(V_{ph\_max})^2 - (v_1^{sat})^2}.$$

In the case that the direct and quadrature axes have the same priority (d-q equivalence), the constrained voltages are obtained as follows:

$$v_d^{sat} = \min(\max(v_d^{unsat}, -V_{d\_max}), V_{d\_max})$$

and

$$v_q^{sat} = \min(\max(v_q^{unsat}, -V_{q\_max}), V_{q\_max}),$$

where:

$$V_{d\_max} = \frac{V_{ph\_max}|v_d^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}|v_q^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

### Integral Anti-Windup

An anti-windup mechanism is employed to avoid saturation of integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}(v_d^{sat} - v_d^{unsat})$$

and

$$K_{i\_iq} + K_{aw\_iq}(v_q^{sat} - v_q^{unsat}),$$

where  $K_{aw\_id}$ ,  $K_{aw\_iq}$ , and  $K_{aw\_if}$  are the anti-windup gains for the  $d$ -axis,  $q$ -axis, and field controllers, respectively.

## Assumptions

- The plant model for direct and quadrature axis can be approximated with a first-order system.
- This control solution is used only for permanent magnet synchronous motors with sinusoidal flux distribution and field windings.

## Ports

### Input

#### **idqRef — Reference currents**

vector

Desired  $d$ - and  $q$ -axis currents for control of a PMSM, in A.

Data Types: `single` | `double`

#### **idq — Measured currents**

vector

Actual  $d$ - and  $q$ -axis currents of the controlled PMSM, in A.

Data Types: `single` | `double`

#### **wElectrical — Measured electrical velocity**

vector

Rotor electrical velocity used for feedforward pre-control, in rad/s.

Data Types: `single` | `double`

#### **VphMax — Maximum phase voltage**

scalar

Maximum allowable voltage in each phase, in V.

Data Types: `single` | `double`

#### **Reset — External reset**

scalar

External reset signal (rising edge) for integrators.

Data Types: `single` | `double`

### Output

#### **vdqRef — Reference voltages**

vector

Desired  $d$ - and  $q$ -axis voltages for control of a PMSM, in V.

Data Types: `single` | `double`

## Parameters

### Control Parameters

#### **D-axis current proportional gain — D-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for direct-axis current control.

#### **D-axis current integral gain — D-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for direct-axis current control.

#### **D-axis current anti-windup gain — D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for direct-axis current control.

#### **Q-axis current proportional gain — Q-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for quadrature-axis current control.

#### **Q-axis current integral gain — Q-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for quadrature-axis current control.

#### **Q-axis current anti-windup gain — Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for quadrature-axis current control.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, you can specify the sample time explicitly.

#### **Axis prioritization — Axis prioritization for voltage limiter**

q-axis (default) | d-axis | d-q equivalence

Prioritize or maintain the ratio between  $d$ - and  $q$ -axes when the block limits voltage.

#### **Enable zero cancellation — Feedforward zero-cancellation**

off (default) | on

Enable or disable zero-cancellation on the feedforward path.

#### **Enable pre-control voltage — Pre-control voltage**

on (default) | off

Enable or disable pre-control voltage.

### Pre-Control Parameters

#### D-axis current vector, $i_d$ (A) — D-axis current breakpoint vector

$[-200, 0, 200]$ A (default) | monotonically increasing vector

Direct-axis current vector used in the lookup tables for parameters determination. For constant machine parameters, do not change the default.

#### Q-axis current vector, $i_q$ (A) — Q-axis current breakpoint vector

$[-200, 0, 200]$ A (default) | monotonically increasing vector

Quadrature-axis current vector used in the lookup tables used to determine parameters. For constant machine parameters, do not change the default.

#### Ld matrix, $L_d(i_d, i_q)$ (H) — D-axis inductance lookup data

$0.0002 * \text{ones}(3, 3)$ H (default) | positive matrix

$L_d$  matrix used as lookup-table data. For constant machine parameters change only the constant factor, for example,  $L_d * \text{ones}(3, 3)$ .

#### Lq matrix, $L_q(i_d, i_q)$ (H) — Q-axis inductance lookup data

$0.0002 * \text{ones}(3, 3)$ H (default) | positive matrix

$L_q$  matrix used as lookup-table data. For constant machine parameters change only the constant factor, e.g.,  $L_q * \text{ones}(3, 3)$ .

#### Permanent magnet flux linkage matrix, $PM(i_d, i_q)$ (Wb) — Flux linkage lookup data

$0.04 * \text{ones}(3, 3)$ Wb (default) | real matrix

Permanent magnet flux linkage matrix used in the lookup table. For constant machine parameters change only the constant factor, for example  $psim * \text{ones}(3, 3)$ .

## Version History

Introduced in R2017b

## References

- [1] Bernardes, T., V. F. Montagner, H. A. Gründling, and H. Pinheiro. "Discrete-time sliding mode observer for sensorless vector control of permanent magnet synchronous machine." *IEEE Transactions on Industrial Electronics*. Vol. 61, Number 4, 2014, pp. 1679-1691.
- [2] Carpiuc, S., and C. Lazar. "Fast real-time constrained predictive current control in permanent magnet synchronous machine-based automotive traction drives." *IEEE Transactions on Transportation Electrification*. Vol.1, Number 4, 2015, pp. 326-335.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

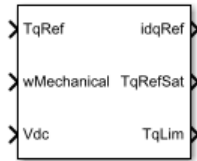
### **Blocks**

PMSM Current Controller | PMSM Current Reference Generator

## PMSM Current Reference Generator

Permanent magnet synchronous machine current reference generator

**Library:** Simscape / Electrical / Control / PMSM Control



### Description

The PMSM Current Reference Generator block implements a current reference generator for permanent magnet synchronous machine (PMSM) current control in the rotor  $d$ - $q$  reference frame.

You typically use this block in a series of blocks making up a control structure.

- You can generate a voltage reference in the  $d$ - $q$  frame by placing this block before a PMSM Current Control or PMSM Current Control with Pre-Control block.
- You can implement velocity control by placing this block after a Velocity Controller block.

You can see an example of a full control structure, from machine measurements to machine inputs, in the PMSM Field-Oriented Control block.

### Equations

The PMSM Current Reference Generator block can obtain the current reference using one of these methods:

- Zero  $d$ -axis control (ZDAC)
- User defined lookup tables
- Automatically generated lookup tables

For the ZDAC method, the block sets the  $d$ -axis current reference  $i_d^{ref}$  to zero and determines the  $q$ -axis current reference  $i_q^{ref}$  using the torque equation:

$$i_d^{ref} = 0,$$

and

$$i_q^{ref} = \frac{2T_{ref}}{3p\psi_m},$$

where:

- $T_{ref}$  is the reference torque input.
- $p$  is the number of pole pairs.
- $\psi_m$  is the permanent magnet flux linkage.



For operation below the base speed of the synchronous machine, ZDAC is a suitable method. Above base speed, a field weakening controller is required to adjust the  $d$ -axis reference.

To pregenerate optimal current references for several operating points offline, define two lookup tables using the user-defined lookup table approach:

$$i_d^{ref} = f(n_m, T_{ref}, v_{dc}),$$

and

$$i_q^{ref} = g(n_m, T_{ref}, v_{dc}),$$

where:

- $n_m$  is the rotor angular velocity.
- $v_{dc}$  is the DC-link voltage of the converter.

To let the block create the lookup tables, choose the automatically generated lookup table approach. The block generates the lookup table using two strategies:

- Maximum torque per ampere
- Field weakening

The selection between the two strategies is based on the modulation index, which can be computed as follows:

$$M = \frac{V_s}{kV_{ph\_max}},$$

where  $V_s$  is the stator voltage amplitude,  $k$  is the modulation factor, and  $V_{ph\_max}$  is the maximum allowable phase voltage. In the case that the modulation index is greater than 1, the block generates current references using the field weakening procedure. Otherwise, current references are computed using the maximum torque per ampere procedure.

### Maximum Torque Per Ampere

You can generate current references in the constant torque region (occurring below rated speed) by using the maximum torque per ampere (MTPA) strategy.

The direct and quadrature components of the stator current are written in terms of angle and magnitude as:

$$i_d = -I_s \sin\beta,$$

and

$$i_q = I_s \cos\beta,$$

where:

- $\beta$  is the angle of the stator current vector.
- $I_s$  is the stator current amplitude.

Using the angle-magnitude variant of the d-q currents, the PMSM torque equation is written as:

$$T_e = \frac{3p}{2}\psi_m I_s \cos\beta + \frac{3p}{4}(L_q - L_d)I_s^2 \sin 2\beta,$$

where  $L_d$  and  $L_q$  are the direct and quadrature inductances, respectively.

To obtain fast transient response and maximize torque with the smallest possible stator current amplitude, MTPA imposes  $(dT_e)/d\beta = 0$  to the torque equation, which yields

$$-\frac{3p}{2}\psi_m I_s \sin\beta + \frac{3p}{2}(L_q - L_d)I_s^2(\cos^2\beta - \sin^2\beta) = 0.$$

The MTPA  $d$ -axis current  $i_{d\_mtpa}$  is written in terms of the  $q$ -axis component  $i_{q\_mtpa}$  by substituting the  $d$ - $q$  currents back from their angle and magnitude variants:

$$i_{d\_mtpa} = \frac{\psi_m}{2(L_q - L_d)} - \sqrt{\frac{\psi_m^2}{4(L_q - L_d)^2} + i_{q\_mtpa}^2}.$$

Finally, by plugging the previous equation into the  $d$ - $q$  variant of the PMSM torque equation, the following polynomial is obtained:

$$9p^2(L_q - L_d)^2 i_{q\_mtpa}^4 + 6T_{ref} p \psi_m i_{q\_mtpa} - 4T_{ref}^2 = 0.$$

The  $q$ -axis component is obtained by solving this polynomial.

### Field Weakening

You can generate current references in the above rated speed region by using the field weakening (FW) strategy.

Above the rated speed, the stator voltage is limited by the power converter and the available DC-link voltage. The maximum stator voltage is:

$$V_s = \sqrt{v_d^2 + v_q^2} \leq V_{ph\_max},$$

where  $V_{ph\_max}$  is the maximum available stator phase voltage.

The steady-state voltage equations for PMSMs are

$$v_d = R_s i_d - \omega_e L_q i_q,$$

and

$$v_q = R_s i_q + \omega_e (L_d i_d + \psi_m).$$

For rotor speeds above rated, the stator resistance is negligible, and the field weakening  $d$ -axis current component  $i_{d\_fw}$  is obtained in terms of the  $q$ -axis component  $i_{q\_fw}$  from the  $v_q$  steady-state equation:

$$i_{d\_fw} = -\frac{\psi_m}{L_d} + \frac{1}{L_d} \sqrt{\frac{V_{ph\_max}^2}{\omega_e^2} - (L_q i_{q\_fw})^2},$$

Finally, by plugging the  $i_{d\_fw}$  equation into the PMSM torque equation, the following polynomial is obtained:

$$9p^2(L_d - L_q)^2 L_q^2 \omega_e^2 i_{q\_fw}^4 + (9p^2 \psi_m^2 L_q^2 \omega_e^2 - 9p^2(L_d - L_q)^2 V_{ph\_max}^2) i_{q\_fw}^2 - 12T_{ref} p \psi_m L_d L_q \omega_e^2 i_{q\_fw} + 4T_{ref}^2 L_d^2 \omega_e^2 = 0$$

The  $q$ -axis component is obtained by solving this polynomial.

### Assumptions

The machine parameters are constants.

### Limitations

The automatically generated current references introduce latency in the presimulation phase. For medium-power PMSM drives the latency is around 300 ms.

## Ports

### Input

#### TqRef — Reference torque

scalar

Desired mechanical torque produced by the PMSM, in N\*m.

Data Types: `single` | `double`

#### wMechanical — Rotor mechanical speed

scalar

Mechanical angular velocity of the rotor, obtained via direct measurement of the PMSM, in rad/s.

Data Types: `single` | `double`

#### Vdc — DC-link voltage

scalar

DC-link voltage of the converter, in V. For the ZDAC method, this value is used to limit the output reference torque and torque limit. For the lookup table method, this value is used as an input to the lookup tables.

Data Types: `single` | `double`

### Output

#### idqRef — Reference currents

vector

Reference  $d$ - and  $q$ -currents to be given as inputs to a PMSM current controller, in A.

Data Types: `single` | `double`

#### TqRefSat — Reference torque

scalar

Reference torque saturated by the calculated torque limit **TqLim**, in N\*m.

Data Types: `single` | `double`

**TqLim — Torque limit**

scalar

Torque limit imposed by both the electrical and mechanical constraints of the system, in N\*m.

Data Types: single | double

**Parameters****General Parameters****Nominal dc-link voltage (V) — Rated DC voltage**

300V (default) | positive number

Nominal DC-link voltage of the electrical source.

**Maximum power (W) — Rated power**

30000W (default) | positive number

Maximum allowable PMSM power.

**Maximum torque (N\*m) — Rated torque**

250N\*m (default) | positive number

Maximum allowable PMSM torque.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, you can specify the sample time explicitly.

**Reference Generation Strategy****Current references — Current reference strategy**

Zero d-axis control (default) | Lookup-table based | Automatically generated lookup-table

Select the strategy for determining current references.

**Mechanical speed vector, wMechanical (rpm) — Rotor speed lookup vector**

[0, 3000] rpm (default) | positive monotonically increasing vector

Speed vector used in the lookup tables for determining current references.

**Torque reference vector, TqRef (N\*m) — Torque reference lookup vector**

[-100, 0, 100]N\*m (default) | positive monotonically increasing vector

Torque vector used in the lookup tables for determining current references.

**DC-link voltage vector, Vdc (V) — DC-link voltage lookup vector**

[300, 350]V (default) | positive monotonically increasing vector

DC-link voltage vector used in the lookup tables for determining current references.

**D-axis current reference matrix,  $id(w_{\text{Mechanical}}, T_{\text{qRef}}, V_{\text{dc}})$  (A) — Reference d-axis current values**

zeros(2,3,2)A (default) | real matrix

Direct-axis current reference lookup data.

**Q-axis current reference matrix,  $iq(w_{\text{Mechanical}}, T_{\text{qRef}}, V_{\text{dc}})$  (A) — Reference q-axis current values**

zeros(2,3,2)A (default) | real matrix

Quadrature-axis current reference lookup data.

**PWM method — Pulse width modulation method**

SVM: space vector modulation (default) | SPWM: sinusoidal PWM

Specify the waveform technique.

**Dependencies**This parameter is only visible when **Current references** is set to Automatically generated lookup-table.**Modulation factor — Modulation factor**

1 (default) | positive scalar

Safety factor used to compute the maximum allowed phase voltage for current references generation.

**Dependencies**This parameter is only visible when **Current references** is set to Automatically generated lookup-table.**Number of pole pairs — Pole pairs**

8 (default) | positive integer

Number of permanent magnet pole pairs on the rotor.

**Permanent magnet flux linkage (Wb) — PM Flux Linkage**

0.04Wb (default) | positive scalar

Peak permanent magnet flux linkage.

**D-axis inductance (H) — Inductance of d-axis**

0.00024 (default) | positive scalar

Direct-axis inductance.

**Q-axis inductance (H) — Inductance of q-axis**

0.00029 (default) | positive scalar

Quadrature-axis inductance.

**Stator resistance (Ohm) — Resistance of stator**

0.01 (default) | positive scalar

Stator resistance per phase.

## **Version History**

**Introduced in R2017b**

### **References**

- [1] Haque, M. E., L. Zhong, and M. F. Rahman. "Improved trajectory control for an interior permanent magnet synchronous motor drive with extended operating limit." *Journal of Electrical & Electronics Engineering*. Vol. 22, Number 1, 2003, p. 49.
- [2] Carpiuc, S., C. Lazar, and D. I. Patrascu. "Optimal Torque Control of the Externally Excited Synchronous Machine." *Control Engineering and Applied Informatics*. Vol. 14, Number 2, 2012, pp. 80-88.

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

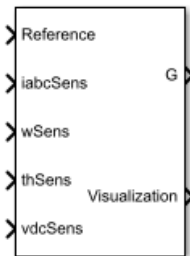
#### **Blocks**

PMSM Current Controller | PMSM Current Controller with Pre-Control

# PMSM Field-Oriented Control

Permanent magnet synchronous machine field-oriented control

**Library:** Simscape / Electrical / Control / PMSM Control



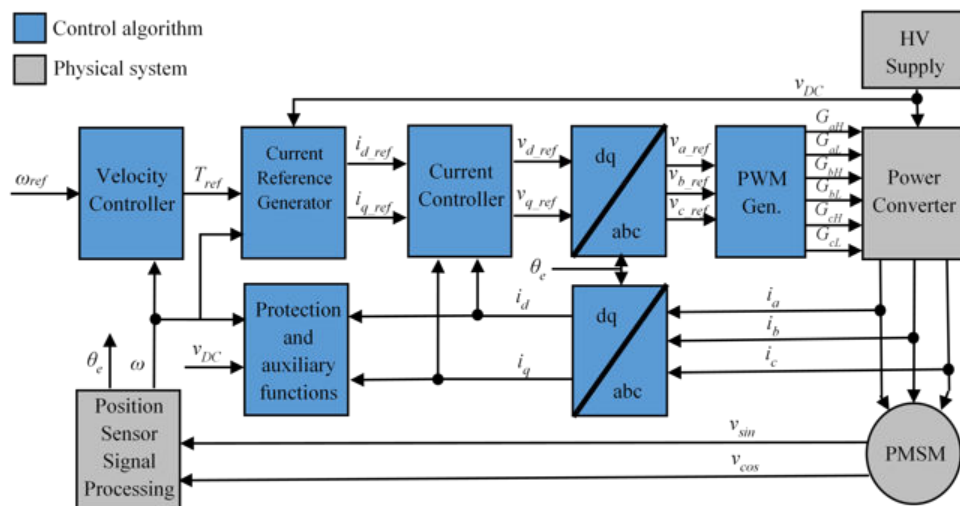
## Description

The PMSM Field-Oriented Control block implements a field-oriented control structure for a permanent magnet synchronous machine (PMSM). Field Oriented Control (FOC) is a performant AC motor control strategy that decouples torque and flux by transforming the stationary phase currents to a rotating frame. Use FOC when rotor speed and position are known and your application requires:

- High torque and low current at startup.
- High efficiency.

## Equations

The PMSM FOC structure decouples the torque and flux by using the rotor d-q reference frame. This diagram shows the overall architecture of the block.



In the diagram:

- $\omega$  and  $\omega_{ref}$  are the measured and reference angular velocities, respectively.

- $T_{ref}$  is the reference electromagnetic torque.
- $i$  and  $v$  are stator currents and voltages and subscripts  $d$  and  $q$  represent the  $d$ -axis and  $q$ -axis, and subscripts  $a$ ,  $b$ , and  $c$ , represent the three stator windings.
- $\theta_e$  is the rotor electrical angle.
- $G$  is a gate pulse, subscripts  $H$  and  $L$ , represent high and low, and subscripts  $a$ ,  $b$ , and  $c$  represent the three stator windings.

You can choose to implement either velocity or torque control with the `Control mode` parameter. The block implements velocity control exactly as shown in the diagram. The block implements torque control by removing the Velocity Controller block and accepting the reference torque directly.

### Assumptions

The machine parameters are known.

### Limitations

The control structure is implemented with a single sample rate.

## Ports

### Input

#### Reference — System reference

scalar

System reference specified as torque reference in N\*m or velocity reference in rad/s, depending on the control mode selected.

Data Types: `single` | `double`

#### iabcSens — Measured phase currents

vector

Measured stator phase currents, in A.

Data Types: `single` | `double`

#### wSens — Rotor speed

scalar

Measured mechanical angular velocity of rotor, in rad/s.

Data Types: `single` | `double`

#### thSens — Rotor angle

scalar

Measured mechanical angle of rotor, in rad.

Data Types: `single` | `double`

#### vdcSens — DC-link voltage

scalar

Measured DC-link voltage, in V.



Data Types: `single` | `double`

## Output

### G – Gate pulses

vector

Six pulse waveforms that determine switching behavior in the attached power converter.

Data Types: `single` | `double`

### Visualization – Visualization signals

bus

Bus containing signals for visualization, including:

- Reference
- `wElectrical`
- `iabc`
- `theta`
- `Vdc`
- `PwmEnable`
- `TqRef`
- `TqLim`
- `idqRef`
- `idq`
- `vdqRef`
- `modWave`

Data Types: `single` | `double`

## Parameters

### General

#### Winding Type – Control parameterization

`Wye-wound` (default) | `Delta-wound`

Specify either a wye-wound or a delta-wound parameterization for the reference values of the controller.

#### Control Mode – Control mode strategy

`Torque control` (default) | `Velocity control`

Specify either a torque control or velocity control strategy.

#### Nominal dc-link voltage (V) – Rated DC voltage

`300 V` (default) | positive number

Nominal DC-link voltage of the electrical source.

**Maximum power (W) — Maximum power**

35000 W (default) | positive number

Maximum machine power.

**Maximum torque (N\*m) — Maximum torque**

250 N\*m (default) | positive number

Maximum machine torque.

**Number of rotor pole pairs — Pole pairs**

8 (default) | positive integer

Number of permanent magnet pole pairs on the rotor.

**Inverter dc-link voltage threshold (V) — DC-link voltage threshold**

100 V (default) | positive number

Voltage threshold to activate the power inverter.

**Fundamental sample time (s) — Block sample time**

5e-6 (default) | positive number

Fundamental sample time for the block.

**Control sample time (s) — Control sample time**

1e-4 (default) | positive number

Sample time for the control system.

**Outer Loop****Control Type — Control type strategy**

PI control (default) | P control | P-PI control

Specify the type of the control strategy.

**Controller proportional gain — Proportional gain of PI controller**

1 (default) | positive number

Proportional gain of the PI controller.

**Controller integral gain — Integral gain of PI controller**

1 (default) | positive number

Integral gain of the PI controller.

**P controller proportional gain — Proportional gain of P controller**

1 (default) | positive number

Proportional gain of P controller.

**Integral anti-windup gain — Anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller.

**Current references — Current reference strategy**

Zero d-axis control (default) | Lookup-table based | Automatically generated lookup-table

Select the current reference strategy.

**Mechanical speed vector,  $w_{\text{Mechanical}}$  (rpm) — Rotor speed lookup vector**

[0, 3000] rpm (default) | positive monotonically increasing vector

Speed vector used in the lookup tables for determining current references.

**Torque reference vector,  $T_{\text{qRef}}$  (N\*m) — Torque reference lookup vector**

[-100, 0, 100] N\*m (default) | positive monotonically increasing vector

Torque vector used in the lookup tables for determining current references.

**DC-link voltage vector,  $V_{\text{dc}}$  (V) — DC-link voltage lookup vector**

[300, 350] V (default) | positive monotonically increasing vector

: DC-link voltage vector used in the lookup tables for determining current references.

**D-axis current reference matrix,  $i_{\text{d}}(w_{\text{Mechanical}}, T_{\text{qRef}}, V_{\text{dc}})$  (A) — Reference d-axis current values**

$\text{zeros}(2, 3, 2)$  A (default) | real matrix

Direct-axis current reference lookup data.

**Q-axis current reference matrix,  $i_{\text{q}}(w_{\text{Mechanical}}, T_{\text{qRef}}, V_{\text{dc}})$  (A) — Reference q-axis current values**

$\text{zeros}(2, 3, 2)$  A (default) | real matrix

Quadrature-axis current reference lookup data.

**Modulation factor — Modulation factor**

1 (default) | positive scalar

Safety factor used to compute the maximum allowed phase voltage for current references generation.

**Dependencies**

This parameter is only visible when **Current references** is set to Automatically generated lookup-table.

**Permanent magnet flux linkage (Wb) — PM Flux Linkage**

0.04 Wb (default) | positive scalar

Peak permanent magnet flux linkage.

**D-axis inductance (H) — Inductance of d-axis**

0.00024 (default) | positive scalar

Direct-axis inductance.

**Q-axis inductance (H) — Inductance of q-axis**

0.00029 (default) | positive scalar

Quadrature-axis inductance.

**Stator resistance (0hm) — Resistance of stator**

0.01 (default) | positive scalar

Stator resistance per phase.

**Inner Loop****D-axis current proportional gain — D-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for direct-axis current control.

**D-axis current integral gain — D-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for direct-axis current control.

**D-axis current anti-windup gain — D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for direct-axis current control.

**Q-axis current proportional gain — Q-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for quadrature-axis current control.

**Q-axis current integral gain — Q-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for quadrature-axis current control.

**Q-axis current anti-windup gain — Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for quadrature-axis current control.

**Axis prioritization — Axis prioritization for voltage limiter**

q-axis (default) | d-axis | d-q equivalence

Prioritize or maintain ratio between  $d$ - and  $q$ -axis when the block limits voltage.

**Enable zero cancellation — Feedforward zero cancellation**

off (default) | on

Enable or disable zero cancellation on the feedforward path.

**Enable pre-control voltage — Precontrol voltage**

on (default) | off

Enable or disable precontrol voltage.

**Machine parameters — Machine parameterization**

Constant parameters (default) | Lookup table based parameters

Specify how to parameterize the machine.

- **Constant parameters** — Specify machine parameters that are constant throughout the simulation.
- **Lookup table based parameters** — Specify machine parameters as lookup tables that depend on current.

**D-axis inductance for feed-forward pre-control (H) — Feedforward d-axis inductance**

0.00024 (default) | positive scalar

Direct-axis inductance for feedforward precontrol.

**Q-axis inductance for feedforward precontrol (H) — Feedforward q-axis inductance**

0.00029 (default) | positive scalar

Quadrature-axis inductance for feed-forward pre-control.

**Permanent magnet flux linkage for feedforward pre-control (H) — Feedforward flux linkage**

0.04 (default) | scalar

Permanent magnet flux linkage for feedforward pre-control.

**D-axis current vector,  $i_d$  (A) — D-axis current breakpoint vector**

[-200, 0, 200] A (default) | monotonically increasing vector

Direct-axis current vector used in the lookup tables for parameters determination. For constant machine parameters, do not change the default.

**Q-axis current vector,  $i_q$  (A) — Q-axis current breakpoint vector**

[-200, 0, 200] A (default) | monotonically increasing vector

Quadrature-axis current vector used in the lookup tables for parameters determination. For constant machine parameters, do not change the default.

**$L_d$  matrix,  $L_d(i_d, i_q)$  (H) — D-axis inductance lookup data**

0.0002 \* ones(3, 3) H (default) | positive matrix

$L_d$  matrix used as lookup table data. For constant machine parameters change only the constant factor, for example,  $L_d * \text{ones}(3, 3)$ .

**$L_q$  matrix,  $L_q(i_d, i_q)$  (H) — Q-axis inductance lookup data**

0.0002 \* ones(3, 3) H (default) | positive matrix

$L_q$  matrix used as lookup table data. For constant machine parameters change only the constant factor, for example,  $L_q * \text{ones}(3, 3)$ .

**Permanent magnet flux linkage matrix,  $PM(i_d, i_q)$  (Wb) — Flux linkage lookup data**

0.04 \* ones(3, 3) Wb (default) | real matrix

Permanent magnet flux linkage matrix used in the lookup table. For constant machine parameters change only the constant factor, for example,  $psim * \text{ones}(3, 3)$ .

## PWM

### PWM method — Pulse width modulation method

SVM: space vector modulation (default) | SPWM: sinusoidal PWM

Specify the waveform technique.

### Sampling mode — Wave-sampling method

Natural (default) | Asymmetric | Symmetric

Specify whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### Switching frequency (Hz) — Switching rate

1000 (default) | positive integer

Specify the rate at which you want the switches in the power converter to switch.

## Version History

Introduced in R2017b

## References

- [1] Bernardes, T., V. F. Montagner, H. A. Gründling, and H. Pinheiro. "Discrete-time sliding mode observer for sensorless vector control of permanent magnet synchronous machine." *IEEE Transactions on Industrial Electronics*. Vol. 61, Number 4, 2014, pp. 1679-1691.
- [2] Carpiuc, S., and C. Lazar. "Fast real-time constrained predictive current control in permanent magnet synchronous machine-based automotive traction drives." *IEEE Transactions on Transportation Electrification*. Vol.1, Number 4, 2015, pp. 326-335.
- [3] Haque, M. E., L. Zhong, and M. F. Rahman. "Improved trajectory control for an interior permanent magnet synchronous motor drive with extended operating limit." *Journal of Electrical & Electronics Engineering*. Vol. 22, Number 1, 2003, p. 49.
- [4] Yang, N., G. Luo, W. Liu, and K. Wang. "Interior permanent magnet synchronous motor control for electric vehicle using look-up table." *In 7th International Power Electronics and Motion Control Conference*. Vol. 2, 2012, pp. 1015-1019.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

PMSM Current Controller | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

**Topics**

“Three-Phase PMSM Drive”

## PMSM Field-Weakening Controller

Permanent magnet synchronous machine field-weakening controller

**Library:** Simscape / Electrical / Control / PMSM Control



### Description

The PMSM Field-Weakening Controller block implements a field-weakening controller for a permanent magnet synchronous machine (PMSM).

Use this block to enforce phase voltage constraints on a current-controlled PMSM. The block decreases the PMSM phase voltage by adjusting the angle of the reference current vector when the voltage vector magnitude exceeds its limit. The block does not adjust the amplitude of the current vector.

You can use this block as part of a PMSM control system:

- Use the zero  $d$ -axis control technique to generate an unconstrained current reference vector to drive the PMSM. You can implement this strategy with the PMSM Current Reference Generator block.
- Use this block to adjust the angle of the current reference vector in order to satisfy voltage phase constraints.
- Use a PMSM Current Controller to generate a voltage reference vector to drive the PMSM.

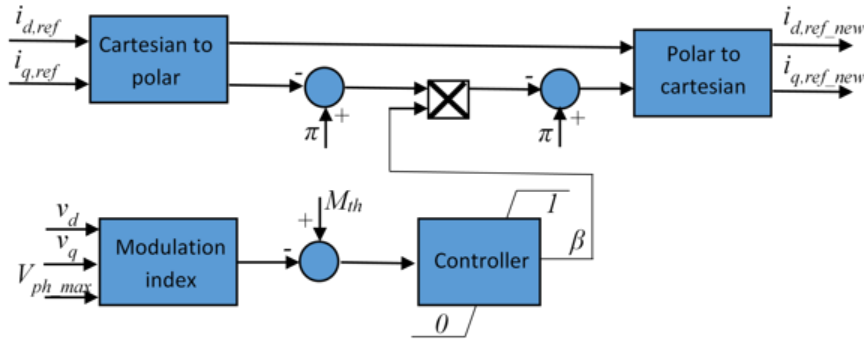
### Equations

An internal integral controller outputs a factor  $\beta \in [0,1]$ , which is determined by how closely the required stator voltage approaches the saturated voltage value at any instant in time:

- When the required stator voltage exceeds the limit,  $\beta$  tends to 0, decreasing the  $q$ -axis current.
- When the required stator voltage is within its limit,  $\beta$  tends to 1 and the angle remains unchanged.

This diagram shows the structure of the field-weakening controller.





In the diagram, you provide the modulation index threshold  $M_{th}$  as an input parameter to the block, and the block computes the modulation index  $M$  as the ratio between the actual phase voltage and the maximum available phase voltage  $V_{ph\_max}$ :

$$M = \frac{\sqrt{v_d^2 + v_q^2}}{V_{ph\_max}}$$

where  $v_d$  and  $v_q$  are the  $d$ -axis and  $q$ -axis components of the voltage vector.

## Ports

### Input

#### idqRef — Reference currents

vector

Desired  $d$ - and  $q$ -axis currents for control of permanent magnet synchronous motor, in A.

Data Types: single | double

#### vdq — Voltages

vector

Direct and quadrature axis voltages of permanent magnet synchronous motor, in V.

Data Types: single | double

#### VphMax — Maximum phase voltage

scalar

Maximum allowable voltage in each phase, in V.

Data Types: single | double

### Output

#### idqRefFW — Field-weakening reference currents

vector

Field-weakening reference direct and quadrature axis currents, in A.

Data Types: single | double

## Parameters

### **Modulation index threshold – Modulation index threshold**

1 (default) | positive number

Reference modulation index.

### **Field-weakening controller integral gain – Integral gain**

100 (default) | positive number

Integrator gain of the field-weakening controller.

### **Integral anti-windup gain – Anti-windup gain**

10 (default) | positive number

Anti-windup gain of the field-weakening controller.

### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, set the sample time explicitly.

## Version History

Introduced in R2017b

## References

- [1] Wai, J., and T. M. Jahns. "A new control technique for achieving wide constant power speed operation with an interior PM alternator machine." *In Industry Applications Conference*. Vol. 2, 2001, pp. 807-814.

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

### **Blocks**

PMSM Current Reference Generator | PMSM Current Controller

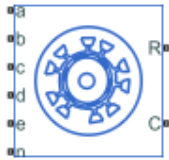
### **Topics**

"PMSM Field-Weakening Control"

## PMSM (Five-Phase)

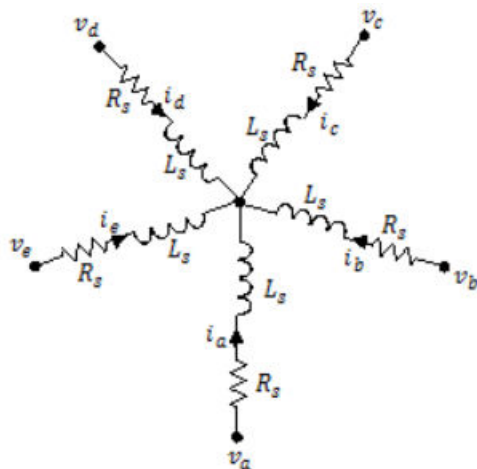
Five-phase permanent magnet synchronous motor with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet

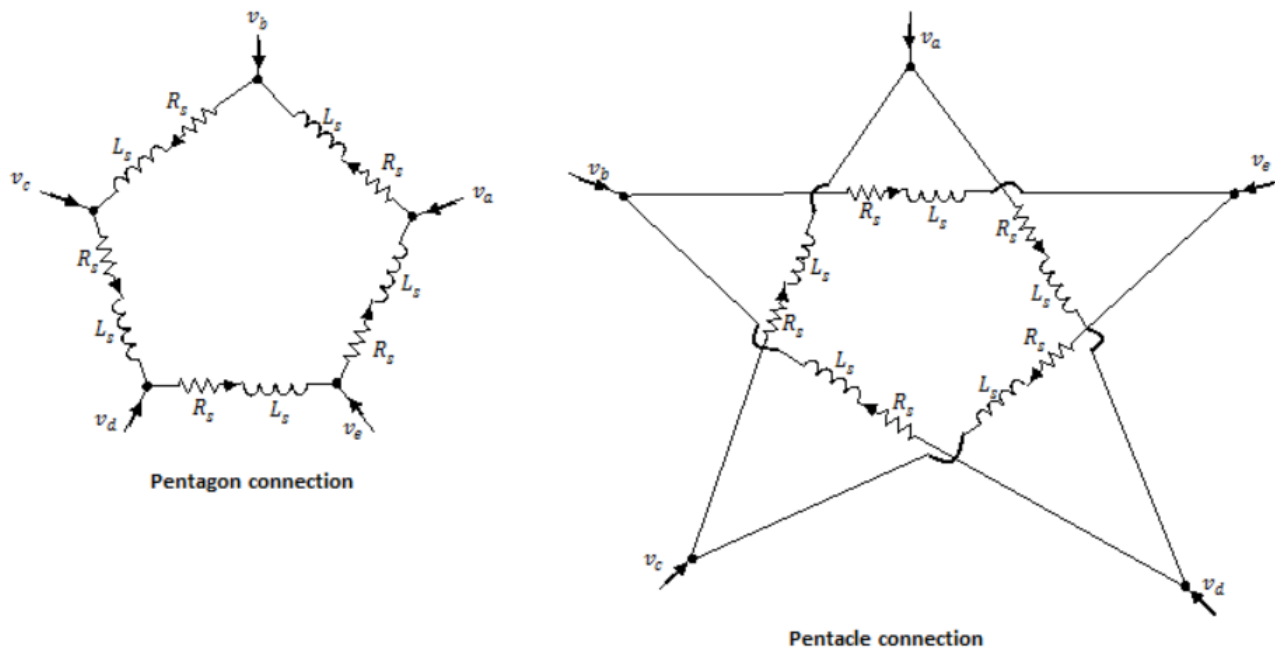


### Description

The PMSM (Five-Phase) block models a permanent magnet synchronous motor with a five-phase star-wound stator. The figure shows the equivalent electrical circuit for the star-connected stator windings.

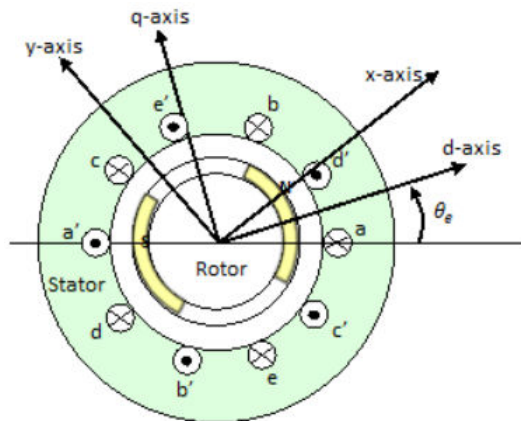


You can also model the permanent magnet synchronous motor either in a pentagon-wound or a pentacle-wound configuration by setting **Winding type** to Pentagon-wound or Pentacle-wound.



### Motor Construction

This figure shows the motor construction with a single pole-pair on the rotor.



Permanent magnets generate a rotor magnetic field that creates a sinusoidal rate of change of flux based on the rotor angle.

For the axes convention, when you set the **Rotor angle definition** parameter to **Angle** between the a-phase magnetic axis and the d-axis, the a-phase and permanent magnet fluxes align when the rotor mechanical angle,  $\theta_r$ , is zero. When you set the **Rotor angle definition** parameter to **Angle** between the a-phase magnetic axis and the q-axis, the rotor mechanical angle is the angle between the a-phase magnetic axis and the rotor q-axis.

### Equations

The voltages across the stator windings are

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \\ v_e \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 \\ 0 & 0 & 0 & R_s & 0 \\ 0 & 0 & 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \\ i_e \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \\ \frac{d\psi_d}{dt} \\ \frac{d\psi_e}{dt} \end{bmatrix},$$

where:

- $v_a, v_b, v_c, v_d,$  and  $v_e$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a, i_b, i_c, i_d,$  and  $i_e$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}, \frac{d\psi_b}{dt}, \frac{d\psi_c}{dt}, \frac{d\psi_d}{dt},$  and  $\frac{d\psi_e}{dt}$  are the rates of change for the magnetic flux in each stator winding.

The permanent magnet and the five windings contribute to the total flux linking each winding. The total flux is

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \\ \psi_d \\ \psi_e \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} & L_{ad} & L_{ae} \\ L_{ba} & L_{bb} & L_{bc} & L_{bd} & L_{be} \\ L_{ca} & L_{cb} & L_{cc} & L_{cd} & L_{ce} \\ L_{da} & L_{db} & L_{dc} & L_{dd} & L_{de} \\ L_{ea} & L_{eb} & L_{ec} & L_{ed} & L_{ee} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \\ i_e \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{dm} \\ \psi_{em} \end{bmatrix},$$

where:

- $\psi_a, \psi_b, \psi_c, \psi_d,$  and  $\psi_e$  are the total fluxes that link each stator winding.
- $L_{aa}, L_{bb}, L_{cc}, L_{dd},$  and  $L_{ee}$  are the self-inductances of the stator windings.
- $L_{ab}, L_{ac}, L_{ba},$  and so on, are the mutual inductances of the stator windings.
- $\psi_{am}, \psi_{bm}, \psi_{cm}, \psi_{dm},$  and  $\psi_{em}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of the rotor electrical angle and defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/5))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e - 4\pi/5))$$

$$L_{dd} = L_s + L_m \cos(2(\theta_e + 4\pi/5))$$

$$L_{ee} = L_s + L_m \cos(2(\theta_e + 2\pi/5))$$

$$L_{ab} = L_{ba} = L_{ce} = L_{ec} = -M_s + L_m \cos(2(\theta_e - 2\pi/5))$$

$$\begin{aligned}
 L_{bc} = L_{cb} &= -M_s + L_m \cos(2(\theta_e - 6\pi/5)) \\
 L_{cd} = L_{dc} = L_{be} = L_{eb} &= -M_s + L_m \cos(2\theta_e) \\
 L_{de} = L_{ed} &= -M_s + L_m \cos(2(\theta_e + 6\pi/5)) \\
 L_{ea} = L_{ae} = L_{bd} = L_{db} &= -M_s + L_m \cos(2(\theta_e + 2\pi/5)) \\
 L_{ac} = L_{ca} &= -M_s + L_m \cos(2(\theta_e - 4\pi/5)) \\
 L_{ad} = L_{da} &= -M_s + L_m \cos(2(\theta_e + 4\pi/5))
 \end{aligned}$$

where:

- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\pi/2$  if you define the rotor electrical angle with respect to the  $d$ -axis, or  $\theta$  if you define the rotor electrical angle with respect to the  $q$ -axis.
- $L_s$  is the stator per-phase self-inductance. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the amount the self-inductance and mutual inductance fluctuate with the changing of the rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The permanent magnet flux linking winding  $a$ - $a'$  is at maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore, the linked motor flux is defined by:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{dm} \\ \psi_{em} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_r - 2\pi/5) \\ \psi_m \cos(\theta_r - 4\pi/5) \\ \psi_m \cos(\theta_r + 4\pi/5) \\ \psi_m \cos(\theta_r + 2\pi/5) \end{bmatrix},$$

where  $\psi_m$  is the permanent magnet flux linkage.

### Simplified Electrical Equations

To remove the rotor angle dependence for the inductive terms, you perform a transformation,  $T$ , on the motor equations.

The  $T$  transformation is defined by:

$$T(\theta_e) = \frac{2}{5} \begin{bmatrix} \sin \theta_e & \sin(\theta_e - 2\pi/5) & \sin(\theta_e - 4\pi/5) & \sin(\theta_e + 4\pi/5) & \sin(\theta_e + 2\pi/5) \\ \cos \theta_e & \cos(\theta_e - 2\pi/5) & \cos(\theta_e - 4\pi/5) & \cos(\theta_e + 4\pi/5) & \cos(\theta_e + 2\pi/5) \\ \sin \theta_e & \sin(\theta_e + 4\pi/5) & \sin(\theta_e - 2\pi/5) & \sin(\theta_e + 2\pi/5) & \sin(\theta_e - 4\pi/5) \\ \cos \theta_e & \cos(\theta_e + 4\pi/5) & \cos(\theta_e - 2\pi/5) & \cos(\theta_e + 2\pi/5) & \cos(\theta_e - 4\pi/5) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix},$$

where  $\theta_e$  is the electrical angle defined as  $N\theta_r$ .  $N$  is the number of pole pairs.

The transformation matrix has this pseudo-orthogonal property:

$$T^{-1}(\theta_e) = \frac{5}{2}T^t(\theta_e).$$

Using the  $T$  transformation on the stator winding voltages and currents transforms them to the dq0 and xy frames, which are independent of the rotor angle:

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_x \\ v_y \\ v_0 \end{bmatrix} = T \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \\ v_e \end{bmatrix}$$

$$\begin{bmatrix} \dot{i}_{ds} \\ \dot{i}_{qs} \\ \dot{i}_x \\ \dot{i}_y \\ \dot{i}_0 \end{bmatrix} = T \begin{bmatrix} \dot{i}_a \\ \dot{i}_b \\ \dot{i}_c \\ \dot{i}_d \\ \dot{i}_e \end{bmatrix}$$



Applying this transformation to the first two electrical equations produces these equations

$$v_{ds} = R_s i_{ds} + L_d \frac{di_{ds}}{dt} - N\omega i_{qs} L_q$$

$$v_{qs} = R_s i_{qs} + L_q \frac{di_{qs}}{dt} + N\omega (i_{ds} L_d + \psi_m)$$

$$v_x = R_s i_x + L_d \frac{di_x}{dt}$$

$$v_y = R_s i_y + L_q \frac{di_y}{dt}$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

$$T = \frac{5}{2} N (i_{qs} (i_{ds} L_d + \psi_m) - i_{ds} i_{qs} L_q)$$

where:

- $L_d = L_s + M_s + 5/2 L_m$ .  $L_d$  is the stator  $d$ -axis inductance.
- $L_q = L_s + M_s - 5/2 L_m$ .  $L_q$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 4M_s$ .  $L_0$  is the stator zero-sequence inductance.
- $\omega$  is the rotor mechanical rotational speed.
- $N$  is the number of rotor permanent magnet pole pairs.

### Alternative Flux Linkage Parameterization

You can parameterize the motor by using the back EMF or torque constants, which are more commonly given on motor datasheets, by using the **Permanent magnet flux linkage** parameter.

The back EMF constant is the peak voltage induced by the permanent magnet in the per-unit rotational speed of each of the phases. The relationship between the peak permanent magnet flux linkage and the back EMF is:

$$k_e = N \psi_m \cdot$$

The back EMF,  $e_{ph}$ , for one phase is:

$$e_{ph} = k_e \omega \cdot$$

The torque constant is the peak torque induced by the per-unit current of each of the phases. It is numerically identical in value to the back EMF constant when both are expressed in SI units:

$$k_t = N \psi_m \cdot$$

When  $L_d = L_q$  and the currents in all five phases are balanced, the combined torque  $T$  is:

$$T = \frac{5}{2} k_t i_q = \frac{5}{2} k_t I_{pk},$$

where  $I_{pk}$  is the peak current in any of the three windings.

The factor 5/2 is calculated from the steady-state sum of the torques from all phases. Therefore, the torque constant  $k_t$  can also be:

$$k_t = \frac{2}{5} \left( \frac{T}{I_{pk}} \right),$$

where  $T$  is the measured total torque when testing with a balanced three-phase current with a peak line voltage of  $I_{pk}$ . The RMS line voltage is:

$$k_t = \sqrt{\frac{2}{5}} \left( \frac{T}{i_{line, rms}} \right).$$

## Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

#### **b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

#### **c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

#### **d — d-phase**

electrical

Electrical conserving port associated with *d*-phase.

#### **e — e-phase**

electrical

Electrical conserving port associated with *e*-phase.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

**Dependencies**

To enable this port, set **Winding Type** to Star-wound and **Zero sequence** to Include.

**R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the motor rotor.

**C – Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

**HA – a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC – c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HD – d-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *d*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HE – e-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *e*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Winding type — Windings configuration**

Star-wound (default) | Pentagon-wound | Pentacle-wound

Configuration for the windings:

- **Star-wound** — The windings are star-wound.
- **Pentagon-wound** — The windings are pentagon-wound. The *a*-phase is connected between ports **a** and **b**, the *b*-phase between ports **b** and **c**, the *c*-phase between ports **c** and **d**, the *d*-phase between ports **d** and **e**, and the *e*-phase between ports **e** and **a**.
- **Pentacle-wound** — The windings are pentacle-wound. The *a*-phase is connected between ports **a** and **d**, the *b*-phase between ports **b** and **e**, the *c*-phase between ports **c** and **a**, the *d*-phase between ports **d** and **b**, and the *e*-phase between ports **e** and **c**.

**Number of pole pairs — Number of pole pairs**

6 (default) | scalar

Number of permanent magnet pole pairs on the rotor.

**Permanent magnet flux linkage parameterization — Permanent magnet flux linkage parameterization**

Specify flux linkage (default) | Specify torque constant | Specify back EMF constant

Permanent magnet flux linkage, specified as Specify flux linkage, Specify torque constant, or Specify back EMF constant.

**Permanent magnet flux linkage — Permanent magnet flux linkage**

0.03 Wb (default) | scalar

Peak permanent magnet flux linkage for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify flux linkage.

**Torque constant — Torque constant**

0.18 N\*m/A (default) | scalar

Torque constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify torque constant.

**Back EMF constant — Back EMF constant**

0.18 V/(rad/s) (default) | scalar

Back EMF constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify back EMF constant.

**Stator parameterization — Stator parameterization**

Specify Ld, Lq, and L0 (default) | Specify Ls, Lm, and Ms

Stator parameterization, specified as Specify Ld, Lq, and L0 or Specify Ls, Lm, and Ms.

**Stator d-axis inductance, Ld — Stator d-axis inductance**

0.00019 H (default) | scalar

*d*-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator q-axis inductance, Lq — Stator q-axis inductance**

0.00025 H (default) | scalar

*q*-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator zero-sequence inductance, L0 — Stator zero-sequence inductance**

0.00016 H (default) | scalar

Zero-sequence inductance.

**Dependencies**

To enable this parameter either:

- Set **Winding Type** to Star-wound, **Zero sequence** to Include, and **Stator parameterization** to Specify Ld, Lq, and L0.
- Set **Winding Type** to Pentagon-wound.

- Set **Winding Type** to Pentacle-wound.

### Stator self-inductance per phase, $L_s$ — Stator self-inductance per phase

0.0002 H (default) | scalar

Average self-inductance of each of the five stator windings.

#### Dependencies

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

### Stator inductance fluctuation, $L_m$ — Stator inductance fluctuation

-0.00002 H (default) | scalar

Amount that the self-inductance and mutual inductance of the stator windings fluctuate with the rotor angle.

#### Dependencies

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

### Stator mutual inductance, $M_s$ — Stator mutual inductance

0.00002 H (default) | scalar

Average mutual inductance between the stator windings.

#### Dependencies

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

### Stator resistance per phase, $R_s$ — Stator resistance per phase

0.013 Ohm (default) | scalar

Resistance of each of the stator windings.

### Zero sequence — Zero sequence option

Include (default) | Exclude

Whether to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. Select this option to prioritize model fidelity. Using this option:
  - Results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
  - Exposes a zero-sequence parameter in the **Impedances** settings.
- **Exclude** — Exclude zero-sequence terms. Select this option to prioritize simulation speed for desktop simulation or real-time deployment, select this option.

#### Dependencies

To enable this parameter, set the **Winding Type** to Star-wound.

### Rotor angle definition — Reference point for the rotor angle measurement

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. “Motor Construction” on page 1-1314 figure shows the default value, Angle between the a-phase magnetic axis and the d-axis. When you select this value, the rotor and a-phase fluxes align when the rotor angle is zero.

When you select Angle between the a-phase magnetic axis and the q-axis, the a-phase current generates maximum torque when the rotor angle is zero.

## **Mechanical**

### **Rotor inertia – Rotor inertia**

0.01 kg\*m<sup>2</sup> (default) | nonnegative scalar

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

### **Rotor damping – Rotor damping**

0 N\*m/(rad/s) (default) | nonnegative scalar

Rotary damping.

## **Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

### **Measurement temperature – Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

### **Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

### **Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 1/K (default) | scalar

Fractional rate at which the permanent magnet flux density changes with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

### **Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

### **Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.



## Version History

Introduced in R2020a

## References

- [1] L. Parsa and H. A. Toliyat. *Sensorless Direct Torque Control of Five-Phase Interior Permanent-Magnet Motor Drives*. IEEE Transactions on Industry Applications, vol. 43, no. 4, pp.952-959, July-Aug., 2007.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. IEEE Press Power Systems Engineering Series, 1995. ISBN 0-7803-1145-0.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

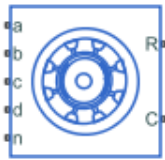
### See Also

PMSM | Hybrid Excitation PMSM | PMSM (Single-Phase)

## PMSM (Four-Phase)

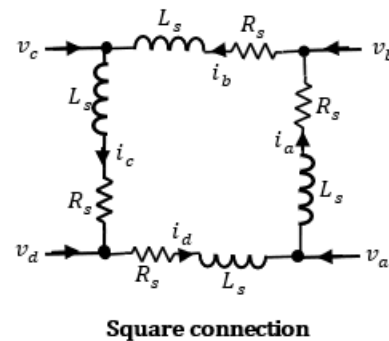
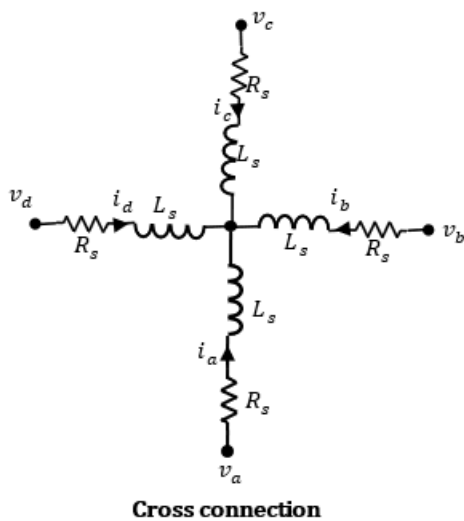
Four-phase permanent magnet synchronous motor with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



### Description

The PMSM (Four-Phase) block models a permanent magnet synchronous motor with a four-phase cross-wound or square-wound stator. The figure shows the equivalent electrical circuits for the cross-connected and square-connected stator windings.



### Motor Construction

Permanent magnets generate a rotor magnetic field that creates a sinusoidal rate of change of flux based on the rotor angle.

For the axes convention, when you set the **Rotor angle definition** parameter to Angle between the a-phase magnetic axis and the d-axis, the a-phase and permanent magnet fluxes align when the rotor mechanical angle,  $\theta_r$ , is zero. When you set the **Rotor angle definition** parameter to Angle between the a-phase magnetic axis and the q-axis, the rotor mechanical angle is the angle between the a-phase magnetic axis and the rotor q-axis.

### Equations

The voltages across the stator windings are

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 \\ 0 & 0 & R_s & 0 \\ 0 & 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \\ \frac{d\psi_d}{dt} \end{bmatrix},$$

where:

- $v_a, v_b, v_c,$  and  $v_d$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a, i_b, i_c,$  and  $i_d$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}, \frac{d\psi_b}{dt}, \frac{d\psi_c}{dt},$  and  $\frac{d\psi_d}{dt}$  are the rates of change for the magnetic flux in each stator winding.

The permanent magnet and the four windings contribute to the total flux linking each winding. The total flux is

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \\ \psi_d \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} & L_{ad} \\ L_{ba} & L_{bb} & L_{bc} & L_{bd} \\ L_{ca} & L_{cb} & L_{cc} & L_{cd} \\ L_{da} & L_{db} & L_{dc} & L_{dd} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{dm} \end{bmatrix},$$

where:

- $\psi_a, \psi_b, \psi_c,$  and  $\psi_d$  are the total fluxes that link each stator winding.
- $L_{aa}, L_{bb}, L_{cc},$  and  $L_{dd}$  are the self-inductances of the stator windings.
- $L_{ab}, L_{ac}, L_{ba},$  and so on, are the mutual inductances of the stator windings.
- $\psi_{am}, \psi_{bm}, \psi_{cm},$  and  $\psi_{dm}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of the rotor electrical angle and defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - \pi/2))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + \pi))$$

$$L_{dd} = L_s + L_m \cos(2(\theta_e + \pi/2))$$

$$L_{ab} = L_{ba} = -M_s + L_m \cos(2(\theta_e + \pi/4))$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/4 - \pi/2))$$

$$L_{cd} = L_{dc} = -M_s - L_m \cos(2(\theta_e + \pi/4 + \pi))$$

$$L_{da} = L_{da} = -M_s - L_m \cos(2(\theta_e + \pi/4 + \pi/2))$$

$$L_{ac} = L_{ca} = -M_s - L_m \cos(2(\theta_e + \pi))$$

$$L_{bd} = L_{db} = -M_s - L_m \cos(2(\theta_e + \pi - \pi/2))$$

where:

- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $-\pi/2$  if you define the rotor electrical angle with respect to the  $d$ -axis, or  $0$  if you define the rotor electrical angle with respect to the  $q$ -axis.
- $L_s$  is the stator per-phase self-inductance. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the amount the self-inductance and mutual inductance fluctuate with the changing of the rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The permanent magnet flux linking winding  $a$ - $a'$  is at maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore, the linked motor flux is defined by:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{dm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_r - \pi/2) \\ \psi_m \cos(\theta_r + \pi) \\ \psi_m \cos(\theta_r + \pi/2) \end{bmatrix},$$

where  $\psi_m$  is the permanent magnet flux linkage.

### Simplified Electrical Equations

To remove the rotor angle dependence for the inductive terms, you perform a transformation,  $T$ , on the motor equations.

The  $T$  transformation is defined by:

$$T(\theta_e) = \frac{1}{2} \begin{bmatrix} \cos \theta_e & \cos(\theta_e - \pi/2) & \cos(\theta_e + \pi) & \cos(\theta_e + \pi/2) \\ -\sin \theta_e & -\sin(\theta_e - \pi/2) & -\sin(\theta_e + \pi) & -\sin(\theta_e + \pi/2) \\ 1/\sqrt{2} & -1/\sqrt{2} & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix},$$

where  $\theta_e$  is the electrical angle defined as  $N\theta_r$ .  $N$  is the number of pole pairs.

The transformation matrix has this pseudo-orthogonal property:

$$T^{-1}(\theta_e) = 2T^t(\theta_e).$$

Using the  $T$  transformation on the stator winding voltages and currents transforms them to the dq0 and xy frames, which are independent of the rotor angle:

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_x \\ v_0 \end{bmatrix} = T \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \end{bmatrix}$$

$$\begin{bmatrix} \dot{i}_{ds} \\ \dot{i}_{qs} \\ \dot{i}_x \\ \dot{i}_0 \end{bmatrix} = T \begin{bmatrix} \dot{i}_a \\ \dot{i}_b \\ \dot{i}_c \\ \dot{i}_d \end{bmatrix}$$

Applying this transformation to the first two electrical equations produces these equations

$$v_{ds} = R_s i_{ds} + L_d \frac{di_{ds}}{dt} - N \omega i_{qs} L_q$$

$$v_{qs} = R_s i_{qs} + L_q \frac{di_{qs}}{dt} + N \omega (i_{ds} L_d + \psi_m)$$

$$v_x = R_s i_x + \frac{L_d + L_q}{2} \frac{di_x}{dt}$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

$$T = 2N(i_{qs}(i_{ds}L_d + \psi_m) - i_{ds}i_{qs}L_q)$$

where:

- $L_d = L_s + M_s + 2 L_m$ .  $L_d$  is the stator  $d$ -axis inductance.
- $L_q = L_s + M_s - 2 L_m$ .  $L_q$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 3M_s$ .  $L_0$  is the stator zero-sequence inductance.
- $\omega$  is the rotor mechanical rotational speed.
- $N$  is the number of rotor permanent magnet pole pairs.

#### Alternative Flux Linkage Parameterization

You can parameterize the motor by using the back EMF or torque constants, which are more commonly given on motor datasheets, by using the **Permanent magnet flux linkage** parameter.

The back EMF constant is the peak voltage induced by the permanent magnet in the per-unit rotational speed of each of the phases. The relationship between the peak permanent magnet flux linkage and the back EMF is:

$$k_e = N \psi_m \cdot$$

The back EMF,  $e_{ph}$ , for one phase is:

$$e_{ph} = k_e \omega \cdot$$

The torque constant is the peak torque induced by the per-unit current of each of the phases. It is numerically identical in value to the back EMF constant when both are expressed in SI units:

$$k_t = N \psi_m \cdot$$

When  $L_d = L_q$  and the currents in all four phases are balanced, the combined torque  $T$  is:

$$T = 2k_t i_q = 2k_t I_{pk},$$

where  $I_{pk}$  is the peak current in any of the four windings.

The factor 2 is calculated from the steady-state sum of the torques from all phases. Therefore, the torque constant  $k_t$  can also be:

$$k_t = \frac{1}{2} \left( \frac{T}{I_{pk}} \right),$$

where  $T$  is the measured total torque when testing with a balanced four-phase current with a peak line voltage of  $I_{pk}$ . The RMS line voltage is:

$$k_t = \sqrt{\frac{1}{2}} \left( \frac{T}{i_{line, rms}} \right).$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.



## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

#### **b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

#### **c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

#### **d — d-phase**

electrical

Electrical conserving port associated with *d*-phase.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

### Dependencies

To enable this port, set **Winding Type** to Cross-wound and **Zero sequence** to Include.

#### **R — Motor rotor**

mechanical

Mechanical rotational conserving port associated with the motor rotor.

#### **C — Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

#### **HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — *b*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — *c*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HD — *d*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *d*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Winding type — Windings configuration**

Cross-wound (default) | Square-wound

Configuration for the windings:

- **Cross-wound** — The windings are cross-wound.
- **Square-wound** — The windings are square-wound.

**Number of pole pairs — Number of pole pairs**

6 (default) | scalar

Number of permanent magnet pole pairs on the rotor.

**Permanent magnet flux linkage parameterization — Permanent magnet flux linkage**

Specify flux linkage (default) | Specify torque constant | Specify back EMF constant

Permanent magnet flux linkage, specified as Specify flux linkage, Specify torque constant, or Specify back EMF constant.

**Permanent magnet flux linkage — Permanent magnet flux linkage**

0.03 Wb (default) | scalar

Peak permanent magnet flux linkage for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify flux linkage.

**Torque constant — Torque constant**

0.18 N\*m/A (default) | scalar

Torque constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify torque constant.

**Back EMF constant — Back EMF constant**

0.18 V/(rad/s) (default) | scalar

Back EMF constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage** to Specify back EMF constant.

**Stator parameterization — Stator parameterization**

Specify Ld, Lq, and L0 (default) | Specify Ls, Lm, and Ms

Stator parameterization, specified as Specify Ld, Lq, and L0 or Specify Ls, Lm, and Ms.

**Stator d-axis inductance, Ld — Stator d-axis inductance**

0.00019 H (default) | scalar

d-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator q-axis inductance, Lq — Stator q-axis inductance**

0.00025 H (default) | scalar

$q$ -axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator zero-sequence inductance, L0 — Stator zero-sequence inductance**

0.00016 H (default) | scalar

Zero-sequence inductance.

**Dependencies**

To enable this parameter either:

- Set **Winding Type** to Cross-wound, **Zero sequence** to Include, and **Stator parameterization** to Specify Ld, Lq, and L0.
- Set **Winding Type** to Square-wound and **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator self-inductance per phase, Ls — Stator self-inductance per phase**

0.0002 H (default) | scalar

Average self-inductance of each of the four stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator inductance fluctuation, Lm — Stator inductance fluctuation**

-0.00002 H (default) | scalar

Amount that the self-inductance and mutual inductance of the stator windings fluctuate with the rotor angle.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator mutual inductance, Ms — Stator mutual inductance**

0.00002 H (default) | scalar

Average mutual inductance between the stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator resistance per phase, Rs — Stator resistance per phase**

0.013 Ohm (default) | scalar

Resistance of each of the stator windings.

**Rotor angle definition — Reference point for the rotor angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. When you select **Angle between the a-phase magnetic axis and the d-axis**, the rotor and *a*-phase fluxes align when the rotor angle is zero.

When you select **Angle between the a-phase magnetic axis and the q-axis**, the *a*-phase current generates maximum torque when the rotor angle is zero.

### Zero sequence — Zero sequence option

Include (default) | Exclude

Whether to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. Select this option to prioritize model fidelity. Using this option:
  - Results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
  - Exposes a zero-sequence parameter in the **Impedances** settings.
- **Exclude** — Exclude zero-sequence terms. Select this option to prioritize simulation speed for desktop simulation or real-time deployment.

### Dependencies

To enable this parameter, set the **Winding Type** to **Cross-wound**.

### Mechanical

#### Rotor inertia — Rotor inertia

0.01 kg\*m<sup>2</sup> (default) | nonnegative scalar

Inertia of the rotor attached to mechanical translational port **R**. The value can be zero.

#### Rotor damping — Rotor damping

0 N\*m/(rad/s) (default) | nonnegative scalar

Rotary damping.

### Thermal

To enable these parameters, set **Modeling option** to **Show thermal port**.

#### Measurement temperature — Rated temperature

298.15 k (default) | real scalar

Temperature for which the motor parameters are quoted.

#### Resistance temperature coefficient — Temperature coefficient

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all four windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

#### Permanent magnet flux temperature coefficient — Permanent magnet flux temperature coefficient

-0.001 1/K (default) | scalar

Fractional rate at which the permanent magnet flux density changes with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

Introduced in R2022a

## References

- [1] Della Torre, F., et al. *Symmetrical Components and Space-Vector Transformations for Four-Phase Networks*. IEEE Transactions on Power Delivery, vol. 23, no. 4, pp.2191-200, Oct., 2008.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. IEEE Press Power Systems Engineering Series, 1995. ISBN 0-7803-1145-0.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

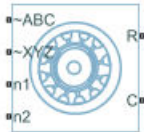
## See Also

PMSM | Hybrid Excitation PMSM | PMSM (Single-Phase) | PMSM (Five-Phase) | PMSM (Six-Phase)

## PMSM (Six-Phase)

Six-phase permanent magnet synchronous motor with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet

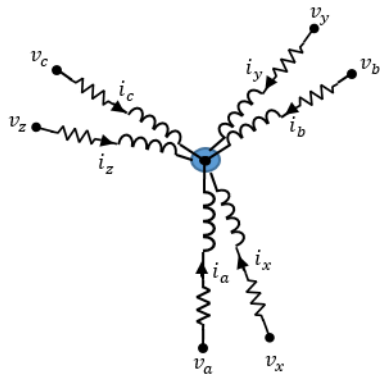


### Description

The PMSM (Six-Phase) block models a permanent magnet synchronous machine (PMSM) with a six-phase star-wound stator.

A six-phase PMSM has two groups of three-phase stator windings: the ABC group and the XYZ group. These two groups have a 30 degree phase shift.

The figure shows the equivalent electrical circuit for the stator windings.



### Equations

The voltages across the stator windings are

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & 0 & R_s & 0 & 0 \\ 0 & 0 & 0 & 0 & R_s & 0 \\ 0 & 0 & 0 & 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_x \\ i_y \\ i_z \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \\ \frac{d\psi_x}{dt} \\ \frac{d\psi_y}{dt} \\ \frac{d\psi_z}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages from port **~ABC** to neutral port **n1**.
- $v_x$ ,  $v_y$ , and  $v_z$  are the individual phase voltages from port **~XYZ** to neutral port **n2**.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing from port **~ABC** to port **n1**.
- $i_x$ ,  $i_y$ , and  $i_z$  are the currents flowing from port **~XYZ** to port **n2**.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ ,  $\frac{d\psi_c}{dt}$ ,  $\frac{d\psi_x}{dt}$ ,  $\frac{d\psi_y}{dt}$ , and  $\frac{d\psi_z}{dt}$  are the rates of change for the magnetic flux in each stator winding.

The permanent magnet and the six windings contribute to the total flux linking each winding. The total flux is

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \\ \psi_x \\ \psi_y \\ \psi_z \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} & L_{ax} & L_{ay} & L_{az} \\ L_{ba} & L_{bb} & L_{bc} & L_{bx} & L_{by} & L_{bz} \\ L_{ca} & L_{cb} & L_{cc} & L_{cx} & L_{cy} & L_{cz} \\ L_{xa} & L_{xb} & L_{xc} & L_{xx} & L_{xy} & L_{xz} \\ L_{ya} & L_{yb} & L_{yc} & L_{yx} & L_{yy} & L_{yz} \\ L_{za} & L_{zb} & L_{zc} & L_{zx} & L_{zy} & L_{zz} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_x \\ i_y \\ i_z \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{xm} \\ \psi_{ym} \\ \psi_{zm} \end{bmatrix},$$

where:

- $\psi_a$ ,  $\psi_b$ ,  $\psi_c$ ,  $\psi_x$ ,  $\psi_y$ , and  $\psi_z$  are the total fluxes that link each stator winding.
- $L_{aa}$ ,  $L_{bb}$ ,  $L_{cc}$ ,  $L_{xx}$ ,  $L_{yy}$ , and  $L_{zz}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ , and so on, are the mutual inductances of the stator windings.
- $\psi_{am}$ ,  $\psi_{bm}$ ,  $\psi_{cm}$ ,  $\psi_{xm}$ ,  $\psi_{ym}$ , and  $\psi_{zm}$  are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of the rotor electrical angle and defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$



$$\begin{aligned}
L_{aa} &= L_s + L_m \cos(2\theta_e) \\
L_{bb} &= L_s + L_m \cos\left(2\left(\theta_e - 2\frac{\pi}{3}\right)\right) \\
L_{cc} &= L_s + L_m \cos\left(2\left(\theta_e + 2\frac{\pi}{3}\right)\right) \\
L_{xx} &= L_s + L_m \cos\left(2\left(\theta_e - \frac{\pi}{6}\right)\right) \\
L_{yy} &= L_s + L_m \cos\left(2\left(\theta_e - 5\frac{\pi}{6}\right)\right) \\
L_{zz} &= L_s + L_m \cos\left(2\left(\theta_e + \frac{\pi}{2}\right)\right) \\
L_{ab} = L_{ba} &= -M_s - L_m \cos\left(2\left(\theta_e + \frac{\pi}{6}\right)\right) \\
L_{bc} = L_{cb} &= -M_s - L_m \cos\left(2\left(\theta_e + \frac{\pi}{6} - 2\frac{\pi}{3}\right)\right) \\
L_{ca} = L_{ac} &= -M_s - L_m \cos\left(2\left(\theta_e + \frac{\pi}{6} + 2\frac{\pi}{3}\right)\right) \\
L_{xy} = L_{yx} &= -M_s - L_m \cos(2\theta_e) \\
L_{yz} = L_{zy} &= -M_s - L_m \cos\left(2\left(\theta_e - 2\frac{\pi}{3}\right)\right) \\
L_{zx} = L_{xz} &= -M_s - L_m \cos\left(2\left(\theta_e + 2\frac{\pi}{3}\right)\right) \\
L_{ax} = L_{xa} &= \sqrt{3}M_s + L_m \cos\left(2\left(\theta_e - \frac{\pi}{12}\right)\right) \\
L_{ay} = L_{ya} &= -\sqrt{3}M_s + L_m \cos\left(2\left(\theta_e - 5\frac{\pi}{12}\right)\right) \\
L_{az} = L_{za} &= L_m \cos\left(2\left(\theta_e + \frac{\pi}{4}\right)\right) \\
L_{bx} = L_{xb} &= L_m \cos\left(2\left(\theta_e - 5\frac{\pi}{12}\right)\right) \\
L_{by} = L_{yb} &= \sqrt{3}M_s + L_m \cos\left(2\left(\theta_e + \frac{\pi}{4}\right)\right) \\
L_{bz} = L_{zb} &= -\sqrt{3}M_s + L_m \cos\left(2\left(\theta_e - \frac{\pi}{12}\right)\right) \\
L_{cx} = L_{xc} &= -\sqrt{3}M_s + L_m \cos\left(2\left(\theta_e + \frac{\pi}{4}\right)\right) \\
L_{cy} = L_{yc} &= L_m \cos\left(2\left(\theta_e - \frac{\pi}{12}\right)\right) \\
L_{cz} = L_{zc} &= \sqrt{3}M_s + L_m \cos\left(2\left(\theta_e - 5\frac{\pi}{12}\right)\right)
\end{aligned}$$

where:

- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the  $d$ -axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the  $q$ -axis.

- $L_s$  is the stator per-phase self-inductance. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the amount the self-inductance and mutual inductance fluctuate with the changing of the rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The permanent magnet flux linking winding  $a-a'$  is at maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore, the linked motor flux is defined by:

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \\ \psi_{xm} \\ \psi_{ym} \\ \psi_{zm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos \theta_e \\ \psi_m \cos(\theta_e - 2\pi/3) \\ \psi_m \cos(\theta_e + 2\pi/3) \\ \psi_m \cos(\theta_e - \pi/6) \\ \psi_m \cos(\theta_e - 5\pi/6) \\ \psi_m \cos(\theta_e + \pi/2) \end{bmatrix},$$

where  $\psi_m$  is the permanent magnet flux linkage.

### Simplified Electrical Equations

Applying a decoupled transformation to the block electrical equations produces an expression for torque that is independent of the rotor angle.

The decoupled transformation is defined by:

$$P(\theta_e) = \frac{1}{3} \begin{bmatrix} \cos \theta_e & \cos(\theta_e - 2\frac{\pi}{3}) & \cos(\theta_e + 2\frac{\pi}{3}) & \cos(\theta_e - \frac{\pi}{6}) & \cos(\theta_e - 5\frac{\pi}{6}) & \cos(\theta_e + \frac{\pi}{2}) \\ -\sin \theta_e & -\sin(\theta_e - 2\frac{\pi}{3}) & -\sin(\theta_e + 2\frac{\pi}{3}) & -\sin(\theta_e - \frac{\pi}{6}) & -\sin(\theta_e - 5\frac{\pi}{6}) & -\sin(\theta_e + \frac{\pi}{2}) \\ 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The transformation matrix,  $P$ , has this pseudo-orthogonal property:

$$P^{-1}(\theta_e) = 3P^T(\theta_e).$$

Using the decoupled transformation on the stator winding voltages and currents transforms them to the  $dq0$  frame, which is independent of the rotor angle.

To obtain the  $d$ -axis,  $q$ -axis, and zero-sequence stator voltages and flux linkages for the ABC and XYZ groups, apply the transformation to the voltage and flux linkage equations:

$$\begin{bmatrix} v_d \\ v_q \\ v_{z1} \\ v_{z2} \\ v_{01} \\ v_{02} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & 0 & R_s & 0 & 0 \\ 0 & 0 & 0 & 0 & R_s & 0 \\ 0 & 0 & 0 & 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_{z1} \\ i_{z2} \\ i_{01} \\ i_{02} \end{bmatrix} + \begin{bmatrix} -\psi_q \\ \psi_d \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} N\omega + \frac{d}{dt} \begin{bmatrix} \psi_d \\ \psi_q \\ \psi_{z1} \\ \psi_{z2} \\ \psi_{01} \\ \psi_{02} \end{bmatrix}$$

$$\begin{bmatrix} \psi_d \\ \psi_q \\ \psi_{z1} \\ \psi_{z2} \\ \psi_{01} \\ \psi_{02} \end{bmatrix} = \begin{bmatrix} L_d & 0 & 0 & 0 & 0 & 0 \\ 0 & L_q & 0 & 0 & 0 & 0 \\ 0 & 0 & L_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_{z1} \\ i_{z2} \\ i_{01} \\ i_{02} \end{bmatrix} + \begin{bmatrix} \psi_m \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where:

- $v_d, v_q, v_{z1}, v_{z2}, v_{01},$  and  $v_{02}$  are the  $d, q, z1,$  and  $z2$  components and zero-sequence stator voltages for the ABC and XYZ groups, defined by:

$$\begin{bmatrix} v_d \\ v_q \\ v_{z1} \\ v_{z2} \\ v_{01} \\ v_{02} \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_x \\ v_y \\ v_z \end{bmatrix}.$$

- $i_d, i_q, i_{z1}, i_{z2}, i_{01},$  and  $i_{02}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents for the ABC and XYZ groups, defined by:

$$\begin{bmatrix} i_d \\ i_q \\ i_{z1} \\ i_{z2} \\ i_{01} \\ i_{02} \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_x \\ i_y \\ i_z \end{bmatrix}.$$

- $L_d = L_s + 4M_s + 3L_m$  is the stator  $d$ -axis inductance.
- $L_q = L_s + 4M_s - 3L_m$  is the stator  $q$ -axis inductance.
- $L_0 = L_s - 2M_s$  is the stator zero-sequence inductance.
- $\omega$  is the rotor mechanical rotational speed.
- $N$  is the number of rotor permanent magnet pole pairs.

The torque equation is defined by:

$$T = 3N[i_q(i_d L_d + \psi_m) - i_d i_q L_q].$$

## Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **~ABC — Stator ABC windings**

electrical

Three-phase electrical port associated with the stator ABC windings.

#### **~XYZ — Stator XYZ windings**

electrical

Three-phase electrical port associated with the stator XYZ windings.

#### **n1 — ABC winding neutral point**

electrical

Electrical conserving port associated with the neutral point of the ABC winding configuration.

### Dependencies

To enable this port, set **Zero sequence** to Include.

#### **n2 — XYZ winding neutral point**

electrical

Electrical conserving port associated with the neutral point of the XYZ winding configuration.

### Dependencies

To enable this port, set **Zero sequence** to Include.

**R — Motor rotor**

mechanical rotational

Mechanical rotational conserving port associated with the motor rotor.

**C — Motor case**

mechanical rotational

Mechanical rotational conserving port associated with the motor case.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HX — x-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *x*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HY — y-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *y*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HZ — z-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding z.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Number of pole pairs — Number of pole pairs**

6 (default) | scalar

Number of permanent magnet pole pairs on the rotor.

**Permanent magnet flux linkage parameterization — Permanent magnet flux linkage parameterization**

Specify flux linkage (default) | Specify torque constant | Specify back EMF constant

Permanent magnet flux linkage, specified as Specify flux linkage, Specify torque constant, or Specify back EMF constant.

**Permanent magnet flux linkage — Permanent magnet flux linkage**

0.03 Wb (default) | scalar

Peak permanent magnet flux linkage for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify flux linkage.

**Torque constant — Torque constant**

0.18 N\*m/A (default) | scalar

Torque constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify torque constant.

**Back EMF constant — Back EMF constant**

0.18 V/(rad/s) (default) | scalar

Back EMF constant for any of the stator windings.

**Dependencies**

To enable this parameter, set **Permanent magnet flux linkage parameterization** to Specify back EMF constant.

**Stator parameterization — Stator parameterization**

Specify Ld, Lq, and L0 (default) | Specify Ls, Lm, and Ms

Stator parameterization, specified as Specify Ld, Lq, and L0 or Specify Ls, Lm, and Ms.

**Stator d-axis inductance, Ld — Stator d-axis inductance**

0.00019 H (default) | scalar

d-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator q-axis inductance, Lq — Stator q-axis inductance**

0.00025 H (default) | scalar

q-axis inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator zero-sequence inductance, L0 — Stator zero-sequence inductance**

0.00016 H (default) | scalar

Zero-sequence inductance.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ld, Lq, and L0.

**Stator self-inductance per phase, Ls — Stator self-inductance per phase**

0.0002 H (default) | scalar

Average self-inductance of each of the five stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify Ls, Lm, and Ms.

**Stator inductance fluctuation, Lm — Stator inductance fluctuation**

-0.00002 H (default) | scalar

Amount that the self-inductance and mutual inductance of the stator windings fluctuate with the rotor angle.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator mutual inductance,  $M_s$  — Stator mutual inductance**

0.00002 H (default) | scalar

Average mutual inductance between the stator windings.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator resistance per phase,  $R_s$  — Stator resistance per phase**

0.013 Ohm (default) | scalar

Resistance of each of the stator windings.

**Zero sequence — Zero sequence option**

Include (default) | Exclude

Whether to include or exclude zero-sequence terms.

- **Include** — Include zero-sequence terms. Select this option to prioritize model fidelity. Using this option results in an error for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Exclude** — Exclude zero-sequence terms. Select this option to prioritize simulation speed for desktop simulation or real-time deployment, select this option.

**Rotor angle definition — Reference point for the rotor angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. “Motor Construction” on page 1-1314 figure shows the default value, Angle between the a-phase magnetic axis and the d-axis. When you select this value, the rotor and  $a$ -phase fluxes aligns when the rotor angle is zero.

When you select Angle between the a-phase magnetic axis and the q-axis, the  $a$ -phase current generates maximum torque when the rotor angle is zero.

**Mechanical****Rotor inertia — Rotor inertia**

0.01 kg\*m<sup>2</sup> (default) | nonnegative scalar

Inertia of the rotor attached to the mechanical translational port **R**. The value can be zero.

**Rotor damping — Rotor damping**

0 N\*m/(rad/s) (default) | nonnegative scalar

Rotary damping.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.



**Measurement temperature – Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 1/K (default) | scalar

Fractional rate at which the permanent magnet flux density changes with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

Introduced in R2020b

## References

- [1] Krause, Paul, Oleg Wasynczuk, Scott Sudhoff, and Steven Pekarek, eds. *Analysis of electric machinery and drive systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013. <https://doi.org/10.1002/9781118524336>.
- [2] Su, Jian Yong, Jin Bo Yang, and Gui Jie Yang. *Research on Vector Control and PWM Technique of Six-Phase PMSM*. *Advanced Materials Research* 516-517 (May 2012): 1626–31. <https://doi.org/10.4028/www.scientific.net/AMR.516-517.1626>.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

PMSM | Synchronous Machine (Six-Phase)

# PMSM Torque Estimator

Estimate permanent magnet synchronous machine torque

**Library:** Simscape / Electrical / Control / PMSM Control



## Description

The PMSM Torque Estimator block implements a torque estimator for permanent magnet synchronous machines (PMSM).

Use this block to estimate the mechanical torque of a motor when it is not directly measurable. The block estimates torque using known machine parameters and the measured phase current vector in the  $dq0$  reference frame.

Use the Park Transform block to convert the measured phase current vector in the  $abc$  reference frame to the  $dq0$  reference frame.

## Equations

The block estimates the mechanical torque  $T_e$  of the PMSM using the torque equation in the  $d$ - $q$  rotor reference frame:

$$T_e = \frac{3p}{2}(\psi_m i_q + (L_d - L_q)i_d i_q),$$

where

- $p$  is the number of pole pairs of the PMSM.
- $\psi_m$  is the flux linkage of the permanent magnet.
- $L_d$  and  $L_q$  are the  $d$ - and  $q$ -axis inductances of the PMSM.
- $i_d$  and  $i_q$  are the  $d$ - and  $q$ -axis currents of the PMSM.

In practice, the machine parameters are not constants and depend on some physical phenomena. You can choose to define these parameters simply as constants or, more realistically, as functions of currents by using lookup tables.

## Assumptions

The machine parameters are known.

## Ports

### Input

**idq — Stator currents**

vector

Stator direct and quadrature currents of the PMSM, in A.

Data Types: `single` | `double`

## Output

### TqEst — Torque estimate

scalar

Estimated mechanical torque value of the PMSM, in N\*m.

Data Types: `single` | `double`

## Parameters

### Machine parameters — Parameter selection strategy

Constant parameters (default) | Lookup table based parameters

Specify the type of machine parameters, which can be in the form of constant values or tabulated data.

### Number of pole pairs — Pole pairs

8 (default) | positive integer

Number of permanent magnet pole pairs on the rotor.

### D-axis current vector, $i_d$ (A) — D-axis current breakpoint vector

$[-200, 0, 200]$ A (default) | monotonically increasing vector

Direct-axis current vector used in the lookup tables for parameters determination.

### Q-axis current vector, $i_q$ (A) — Q-axis current breakpoint vector

$[-200, 0, 200]$ A (default) | monotonically increasing vector

Quadrature-axis current vector used in the lookup tables for parameters determination.

### Ld matrix, $L_d(i_d, i_q)$ (H) — D-axis inductance lookup data

$0.0002 * \text{ones}(3, 3)$ H (default) | positive matrix

$L_d$  matrix used as lookup table data.

### Lq matrix, $L_q(i_d, i_q)$ (H) — Q-axis inductance lookup data

$0.0002 * \text{ones}(3, 3)$ H (default) | positive matrix

$L_q$  matrix used as lookup table data.

### Permanent magnet flux linkage matrix, $PM(i_d, i_q)$ (Wb) — Flux linkage lookup data

$0.04 * \text{ones}(3, 3)$ Wb (default) | real matrix

Permanent magnet flux linkage matrix used in the lookup table.

### D-axis inductance (H) — Inductance of d-axis

$0.0002$  (default) | positive scalar

Direct-axis inductance.

**Q-axis inductance (H) — Inductance of q-axis**

0.0002 (default) | positive scalar

Quadrature-axis inductance.

**Permanent magnet flux linkage (Wb) — PM Flux Linkage**

0.04Wb (default) | positive scalar

Peak permanent magnet flux linkage.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If you use this block inside a triggered subsystem, set the sample time to -1. If you use this block in a continuous variable-step model, set the sample time explicitly.

## Version History

Introduced in R2017b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

PMSM Current Controller | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

# PNP Bipolar Transistor

PNP bipolar transistor using enhanced Ebers-Moll equations

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The PNP Bipolar Transistor block uses a variant of the Ebers-Moll equations to represent an PNP bipolar transistor. The Ebers-Moll equations are based on two exponential diodes plus two current-controlled current sources. The PNP Bipolar Transistor block provides the following enhancements to that model:

- Early voltage effect
- Optional base, collector, and emitter resistances.
- Optional fixed base-emitter and base-collector capacitances.

The collector and base currents are [1]:

$$I_C = -IS \left[ \left( e^{-qV_{BE}/(kT_{m1})} - e^{-qV_{BC}/(kT_{m1})} \right) \left( 1 + \frac{V_{BC}}{V_A} \right) - \frac{1}{\beta_R} \left( e^{-qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

$$I_B = -IS \left[ \frac{1}{\beta_F} \left( e^{-qV_{BE}/(kT_{m1})} - 1 \right) + \frac{1}{\beta_R} \left( e^{-qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

Where:

- $I_B$  and  $I_C$  are base and collector currents, defined as positive into the device.
- $IS$  is the saturation current.
- $V_{BE}$  is the base-emitter voltage and  $V_{BC}$  is the base-collector voltage.
- $\beta_F$  is the ideal maximum current gain BF
- $\beta_R$  is the ideal maximum current gain BR
- $V_A$  is the forward Early voltage VAF
- $q$  is the elementary charge on an electron (1.602176e-19 Coulombs).
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).
- $T_{m1}$  is the transistor temperature, as defined by the **Measurement temperature** parameter value.

You can specify the transistor behavior using datasheet parameters that the block uses to calculate the parameters for these equations, or you can specify the equation parameters directly.

If  $-qV_{BC} / (kT_{m1}) > 40$  or  $-qV_{BE} / (kT_{m1}) > 40$ , the corresponding exponential terms in the equations are replaced with  $(-qV_{BC} / (kT_{m1}) - 39)e^{40}$  and  $(-qV_{BE} / (kT_{m1}) - 39)e^{40}$ , respectively. This helps prevent numerical issues associated with the steep gradient of the exponential function  $e^x$  at large values of  $x$ .

Similarly, if  $-qV_{BC} / (kT_{m1}) < -39$  or  $-qV_{BE} / (kT_{m1}) < -39$  then the corresponding exponential terms in the equations are replaced with  $(-qV_{BC} / (kT_{m1}) + 40)e^{-39}$  and  $(-qV_{BE} / (kT_{m1}) + 40)e^{-39}$ , respectively.

Optionally, you can specify fixed capacitances across the base-emitter and base-collector junctions. You also have the option to specify base, collector, and emitter connection resistances.

### Modeling Temperature Dependence

The default behavior is that dependence on temperature is not modeled, and the device is simulated at the temperature for which you provide block parameters. You can optionally include modeling the dependence of the transistor static behavior on temperature during simulation. Temperature dependence of the junction capacitances is not modeled, this being a much smaller effect.

When including temperature dependence, the transistor defining equations remain the same. The measurement temperature value,  $T_{m1}$ , is replaced with the simulation temperature,  $T_s$ . The saturation current,  $IS$ , and the forward and reverse gains ( $\beta_F$  and  $\beta_R$ ) become a function of temperature according to the following equations:

$$IS_{T_s} = IS_{T_{m1}} \cdot (T_s/T_{m1})^{XTI} \cdot \exp\left(-\frac{EG}{kT_s}(1 - T_s/T_{m1})\right)$$

$$\beta_{F_s} = \beta_{F_{m1}} \left(\frac{T_s}{T_{m1}}\right)^{XTB}$$

$$\beta_{R_s} = \beta_{R_{m1}} \left(\frac{T_s}{T_{m1}}\right)^{XTB}$$

where:

- $T_{m1}$  is the temperature at which the transistor parameters are specified, as defined by the **Measurement temperature** parameter value.
- $T_s$  is the simulation temperature.
- $IS_{T_{m1}}$  is the saturation current at the measurement temperature.
- $IS_{T_s}$  is the saturation current at the simulation temperature. This is the saturation current value used in the bipolar transistor equations when temperature dependence is modeled.
- $\beta_{F_{m1}}$  and  $\beta_{R_{m1}}$  are the forward and reverse gains at the measurement temperature.
- $\beta_{F_s}$  and  $\beta_{R_s}$  are the forward and reverse gains at the simulation temperature. These are the values used in the bipolar transistor equations when temperature dependence is modeled.
- $EG$  is the energy gap for the semiconductor type measured in Joules. The value for silicon is usually taken to be 1.11 eV, where 1 eV is 1.602e-19 Joules.
- $XTI$  is the saturation current temperature exponent.
- $XTB$  is the forward and reverse gain temperature coefficient.
- $k$  is the Boltzmann constant (1.3806503e-23 J/K).

Appropriate values for  $XTI$  and  $EG$  depend on the type of transistor and the semiconductor material used. In practice, the values of  $XTI$ ,  $EG$ , and  $XTB$  need tuning to model the exact behavior of a particular transistor. Some manufacturers quote these tuned values in a SPICE Netlist, and you can read off the appropriate values. Otherwise you can determine values for  $XTI$ ,  $EG$ , and  $XTB$  by using a datasheet-defined data at a higher temperature  $T_{m2}$ . The block provides a datasheet parameterization option for this.

You can also tune the values of  $XTI$ ,  $EG$ , and  $XTB$  yourself, to match lab data for your particular device. You can use Simulink Design Optimization software to help tune the values.

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- The block does not account for temperature-dependent effects on the junction capacitances.
- You may need to use nonzero ohmic resistance and junction capacitance values to prevent numerical simulation issues, but the simulation may run faster with these values set to zero.

### Ports

#### Conserving

##### **B — Base terminal**

electrical

Electrical conserving port associated with the transistor base terminal

##### **C — Collector terminal**

electrical

Electrical conserving port associated with the transistor collector terminal

##### **E — Emitter terminal**

electrical

Electrical conserving port associated with the transistor emitter terminal

**H — Thermal port**

thermal

Thermal conserving port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

**Main****Parameterization — Block parameterization**

Specify from a datasheet (default) | Specify using equation parameters directly

Select one of the following methods for block parameterization:

- **Specify from a datasheet** — Provide parameters that the block converts to equations that describe the transistor. The block calculates the forward Early voltage  $VAF$  as  $I_c/h_{oe}$ , where  $I_c$  is the **Collector current at which h-parameters are defined** parameter value, and  $h_{oe}$  is the **Output admittance  $h_{oe}$**  parameter value [2]. The block sets  $BF$  to the small-signal **Forward current transfer ratio  $h_{fe}$**  value. The block calculates the saturation current  $IS$  from the specified **Voltage  $V_{be}$**  value and the corresponding **Current  $I_b$  for voltage  $V_{be}$**  value when  $I_c$  is zero. This is the default method.
- **Specify using equation parameters directly** — Provide equation parameters  $IS$ ,  $BF$ , and  $VAF$ .

**Forward current transfer ratio  $h_{fe}$  — Forward current transfer ratio**

100 (default)

Small-signal current gain.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Output admittance  $h_{oe}$  — Output admittance**

50e-6 1/0hm (default)

Derivative of the collector current with respect to the collector-emitter voltage for a fixed base current.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.



**Collector current at which h-parameters are defined – Collector current at which h-parameters are defined**

-1 mA (default)

The h-parameters vary with operating point, and are defined for this value of the collector current.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Collector-emitter voltage at which h-parameters are defined – Collector-emitter voltage at which h-parameters are defined**

-5 V (default)

The h-parameters vary with operating point, and are defined for this value of the collector-emitter voltage.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Voltage Vbe – Voltage Vbe**

-0.55 V (default)

Base-emitter voltage when the base current is  $I_b$ . The [  $V_{be} I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Current Ib for voltage Vbe – Current Ib for voltage Vbe**

-0.5 mA (default)

Base current when the base-emitter voltage is  $V_{be}$ . The [  $V_{be} I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter.

**Forward current transfer ratio BF – Forward current transfer ratio**

100 (default)

Ideal maximum forward current gain.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Saturation current IS – Saturation current**

1e-14 A (default)

Transistor saturation current.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Forward Early voltage VAF — Forward Early voltage**

200 V (default)

In the standard Ebers-Moll equations, the gradient of the  $I_c$  versus  $V_{ce}$  curve is zero in the normal active region. The additional forward Early voltage term increases this gradient. The intercept on the  $V_{ce}$ -axis is equal to  $-VAF$  when the linear region is extrapolated.

**Dependencies**

This parameter is visible only when you select Specify using equation parameters directly for the **Parameterization** parameter.

**Reverse current transfer ratio BR — Reverse current transfer ratio**

1 (default)

Ideal maximum reverse current gain. This value is often not quoted in manufacturer datasheets because it is not significant when the transistor is biased to operate in the normal active region. When the value is not known and the transistor is not to be operated on the inverse region, use the default value of 1.

**Measurement temperature — Measurement temperature**

25 degC (default)

Temperature  $T_{m1}$  at which  $V_{be}$  and  $I_b$ , or  $I_S$ , are measured.

**Ohmic Resistance****Collector resistance RC — Collector resistance**

0.01 ohm (default)

Resistance at the collector.

**Emitter resistance RE — Emitter resistance**

1e-4 Ohm (default)

Resistance at the emitter.

**Zero bias base resistance RB — Zero bias base resistance**

1 Ohm (default)

Resistance at the base at zero bias.

**Capacitance****Base-collector junction capacitance — Base-collector junction capacitance**

5 pF (default)

Parasitic capacitance across the base-collector junction.

**Base-emitter junction capacitance – Base-emitter junction capacitance**

5 pF (default)

Parasitic capacitance across the base-emitter junction.

**Total forward transit time – Total forward transit time**

0 us (default)

Represents the mean time for the minority carriers to cross the base region from the emitter to the collector, and is often denoted by the parameter TF [1].

**Total reverse transit time – Total reverse transit time**

0 us (default)

Represents the mean time for the minority carriers to cross the base region from the collector to the emitter, and is often denoted by the parameter TR [1].

**Temperature Dependence****Parameterization – Temperature dependence parameterization**

None – Simulate at parameter measurement temperature (default) | Model temperature dependence

Select one of the following methods for temperature dependence parameterization:

- None – Simulate at parameter measurement temperature – Temperature dependence is not modeled, or the model is simulated at the measurement temperature  $T_{m1}$  (as specified by the **Measurement temperature** parameter on the **Main** tab). This is the default method.
- Model temperature dependence – Provide a value for simulation temperature, to model temperature-dependent effects. You also have to provide a set of additional parameters depending on the block parameterization method. If you parameterize the block from a datasheet, you have to provide values for a second [ *Vbe Ib* ] data pair and *h<sub>fe</sub>* at second measurement temperature. If you parameterize by directly specifying equation parameters, you have to provide the values for *XTI*, *EG*, and *XTB*.

**Forward current transfer ratio, h<sub>fe</sub>, at second measurement temperature – Forward current transfer ratio, h<sub>fe</sub>, at second measurement temperature**

125 (default)

Small-signal current gain at second measurement temperature. It must be quoted at the same collector-emitter voltage and collector current as for the **Forward current transfer ratio h<sub>fe</sub>** parameter on the **Main** tab.

**Dependencies**

This parameter is visible only when you select Specify from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Voltage Vbe at second measurement temperature – Voltage Vbe at second measurement temperature**

-0.45 V (default)

Base-emitter voltage when the base current is *Ib* and the temperature is set to the second measurement temperature. The [ *Vbe Ib* ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Current  $I_b$  for voltage  $V_{be}$  at second measurement temperature – Current  $I_b$  for voltage  $V_{be}$  at second measurement temperature**

-0.5 mA (default)

Base current when the base-emitter voltage is  $V_{be}$  and the temperature is set to the second measurement temperature. The [  $V_{be}$   $I_b$  ] data pair must be quoted for when the transistor is in the normal active region, that is, not in the saturated region.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Second temperature  $T_{m2}$  at which  $h_{fe}$ ,  $V_{be}$ , and  $I_b$  are measured.

**Dependencies**

This parameter is visible only when you select **Specify** from a datasheet for the **Parameterization** parameter on the **Main** tab.

**Current gain temperature coefficient,  $X_{TB}$  – Current gain temperature coefficient**

0 (default)

Current gain temperature coefficient value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Energy gap,  $E_G$  – Energy gap**

1.11 eV (default)

Energy gap value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Saturation current temperature exponent,  $X_{TI}$  – Saturation current temperature exponent**

3 (default)

Saturation current temperature coefficient value.

**Dependencies**

This parameter is visible only when you select **Specify** using equation parameters directly for the **Parameterization** parameter on the **Main** tab.

**Device simulation temperature – Device simulation temperature**

25 degC (default)

Temperature  $T_s$  at which the device is simulated.

## Version History

**Thermal network simlog paths***Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

## References

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.
- [2] H. Ahmed and P.J. Spreadbury. *Analogue and digital electronics for engineers*. 2nd Edition, Cambridge University Press, 1984.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Diode | NPN Bipolar Transistor

# Positive Supply Rail

Ideal positive supply rail

**Library:** Simscape / Electrical / Sources



## Description

The Positive Supply Rail block represents an ideal positive supply rail. Use this block instead of the SimscapeDC Voltage Source block to define the output voltage relative to the SimscapeElectrical Reference block that must appear in each model.

---

**Note** Do not attach more than one Positive Supply Rail block to *any* connected line.

---

## Ports

### Conserving

- — **Positive voltage**

electrical

Electrical conserving port associated to the positive supply rail positive voltage.

## Parameters

**Constant voltage — Constant output voltage**

1 V (default)

The voltage at the output port relative to the Electrical Reference block ground port.

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

DC Voltage Source | Negative Supply Rail

# Potentiometer

Rotary or linear-travel potentiometer controlled by physical signal

**Library:** Simscape / Electrical / Passive



## Description

The Potentiometer block represents a rotary or linear-travel potentiometer, with the wiper position controlled by the input physical signal.

If the potentiometer resistance changes linearly based on wiper position, then the resistance between the wiper position and port **L** is:

$$R_{WL} = \frac{R_0 - 2r}{x_{\max} - x_{\min}}(x - x_{\min}) + r$$

where

- $R_{WL}$  is the resistance between the wiper position and port **L**.
- $R_0$  is the total resistance between ports **L** and **R**.
- $x$  is the wiper position.
- $x_{\min}$  is the value of the wiper position when the wiper is at port **L**.
- $x_{\max}$  is the value of the wiper position when the wiper is at port **R**.
- $r$  is the residual resistance. This is the resistance between the wiper and port **L** when  $x$  is equal to  $x_{\min}$ . This is also the resistance between the wiper and port **R** when  $x$  is equal to  $x_{\max}$ .

If you specify **Logarithmic** for the potentiometer resistance **Taper** parameter, then the resistance between the wiper position and port **L** is:

$$R_{WL} = \begin{cases} A(e^{\lambda(x - x_{\min})} - 1) + r & \text{if resistance gradient is higher at R} \\ R_0 - A(e^{\lambda(x - x_{\min})} - 1) - r & \text{if resistance gradient is higher at L} \end{cases}$$

where  $A$  and  $\lambda$  are chosen such that  $R_{WL}$  at  $x_{\max}$  is  $R_0 - r$ , and  $R_{WL}$  at  $x = (x_{\max} + x_{\min}) / 2$  is equal to  $R_{av}$ , the resistance when the wiper is centered.

---

**Note** Potentiometers widely described as LOG or logarithmic taper are, in fact, exponential taper. That is, the gradient of the resistance between wiper and left-hand port increases as the resistance increases. The Potentiometer block implements this behavior.

---

For both linear and logarithmic tapers, the resistance between the wiper position and port **R** is:

$$R_{WR} = R_0 - R_{WL}$$

where

- $R_{WR}$  is the resistance between the wiper position and port **R**.
- $R_0$  is the total resistance between ports **L** and **R**.
- $R_{WL}$  is the resistance between the wiper position and port **L**.

## Ports

### Input

#### **x** — Wiper position control

physical signal

Physical signal input port controlling the wiper position.

### Conserving

#### **L** — Left pin

electrical

Electrical conserving port associated with the potentiometer left pin.

#### **R** — Right pin

electrical

Electrical conserving port associated with the potentiometer right pin.

### Dependencies

To enable this port, set **Number of visible ports** to Three: left, right, and wiper.

#### **w** — Wiper pin

electrical

Electrical conserving port associated with the potentiometer wiper pin.

## Parameters

#### **Number of visible ports** — Ports visibility option

Three: left, right, and wiper (default) | Two: left and wiper

Whether to enable the right end port of the potentiometer, **R**.

#### **Total resistance** — Total resistance

1000 Ohm (default)

Resistance between port **L** and port **R** when port **W** is open-circuit.

### Dependencies

To enable this parameter, set **Number of visible ports** to Three: left, right, and wiper.

#### **Residual resistance** — Residual resistance

1 Ohm (default) | positive scalar



Resistance measured between the wiper and the adjacent end port (**L** or **R**) when the wiper is at the end port. You can use this parameter to represent the inability of the wiper to reach the end of the track due to the mechanical stop. This value must be greater than or equal to zero. A typical value is  $5e-3$  times the total resistance.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Three: left, right, and wiper.

**PS input for wiper at L — PS input for wiper at L**

0 (default)

Value of the input physical signal at port **x** that corresponds to the wiper being located at port **L**.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Three: left, right, and wiper.

**PS input for wiper at R — PS input for wiper at R**

1 (default)

Value of the input physical signal at port **x** that corresponds to the wiper being located at port **R**.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Three: left, right, and wiper.

**Minimum resistance — Minimum resistance**

1 Ohm (default)

Minimum permissible resistance measured between the wiper and the left port, **L**, when the wiper is at the left port. The value of this parameter must be greater than 0 and lower than the value of the **Maximum resistance** parameter.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Two: left and wiper.

**Maximum resistance — Maximum resistance**

999 Ohm (default)

Maximum permissible resistance measured between the wiper and the left port, **L**. The value of this parameter must be greater than 0 and greater than the value of the **Minimum resistance** parameter.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Two: left and wiper.

**PS input for wiper at minimum resistance — PS input for wiper at minimum resistance**

0 (default)

Value of the input physical signal at port **x** that corresponds to the wiper being located at port **L**. The default value is 0.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Two: left and wiper.

**PS input for wiper at maximum resistance – PS input for wiper at maximum resistance**

1 (default)

Value of the input physical signal at port **x** that corresponds to the wiper being located the furthest from port **L**.

**Dependencies**

To enable this parameter, set **Number of visible ports** to Two: left and wiper.

**Taper – Resistance taper behavior**

Linear (default) | Logarithmic

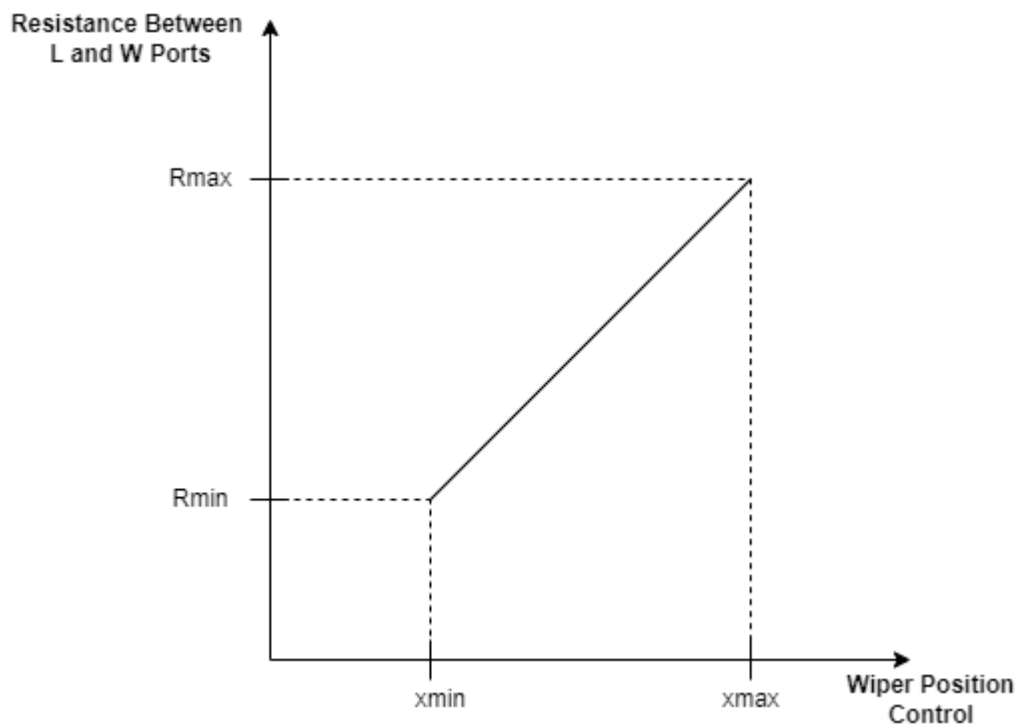
Resistance taper behavior of the potentiometer.

---

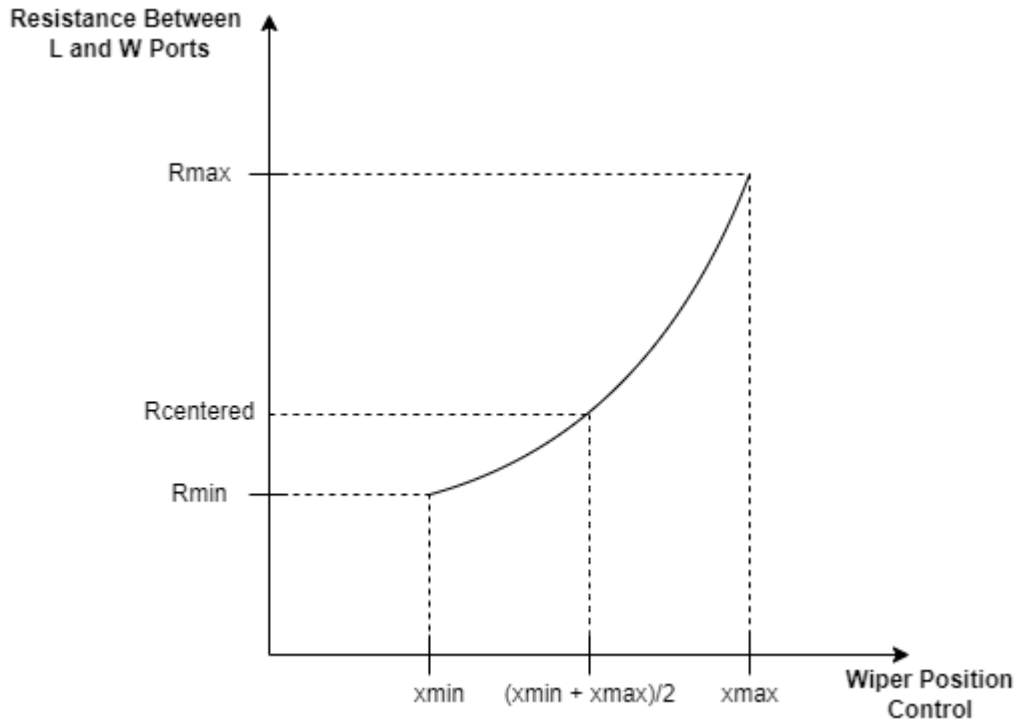
**Note** Linear and Logarithmic options do not model the same taper. You cannot parameterize the logarithmic potentiometer as a linear potentiometer, because the logarithmic taper is nonlinear.

---

If you set this parameter to **Linear**, this plot shows the relationship between the wiper control position and the resistance between **L** and **W** ports.



If you set this parameter to **Logarithmic**, this plot shows the relationship between the wiper control position and the resistance between **L** and **W** ports.



### Resistance when centered – Resistance when centered

200 Ohm (default)

If you set the **Resistance gradient** parameter to Higher at L, this parameter is the resistance between port L and port W when the wiper is centered. Otherwise, if you set **Resistance gradient** to Higher at R, this parameter is the resistance between port R and port W when the wiper is centered. Because the resistance taper is exponential in shape, the value of the **Resistance when centered** parameter must be less than half of the **Total resistance** parameter value.

#### Dependencies

To enable this parameter, set **Taper** to Logarithmic.

### Resistance gradient – Resistance gradient

Higher at R (default) | Higher at L

Whether the potentiometer resistance varies more rapidly at the left or the right end.

#### Dependencies

To enable this parameter, set **Taper** to Logarithmic.

## Version History

Introduced in R2010a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

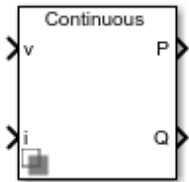
### **See Also**

Variable Resistor

# Power Measurement

Calculate single-phase real and reactive power

**Library:** Simscape / Electrical / Control / Measurements



## Description

The Power Measurement block measures the real and reactive power of an element in a single-phase network. The block outputs the power quantities for each frequency component you specify. For three-phase measurements, consider using the Three-Phase Power Measurement block.

Use this block to measure power for both sinusoidal and nonsinusoidal periodic signals.

Set the **Sample time** parameter to 0 for continuous-time operation, or explicitly for discrete-time operation.

Specify a vector of all frequency components to include in the power output using the **Harmonic numbers** parameter:

- To output the DC component, specify 0.
- To output the component corresponding to the fundamental frequency, specify 1.
- To output components corresponding to higher-order harmonics, specify  $n > 1$ .

## Equations

For each specified harmonic  $k$ , the block calculates the real power  $P_k$  and reactive power  $Q_k$  from the phasor equation:

$$P_k + jQ_k = G(V_k e^{j\theta V_k})(I_k e^{j\theta I_k}),$$

where:

- $G$  is equal to 0.25 for the DC component ( $k = 0$ ) and 0.5 for the AC components ( $k > 0$ ).
- $V_k e^{j\theta V_k}$  is the phasor representation of the  $k$ -component input voltage.
- $\overline{I_k e^{j\theta I_k}}$  is the complex conjugate of  $I_k e^{j\theta I_k}$ , the phasor representation of the  $k$ -component input current.

The block estimates the real-time  $k$ -component voltage and current phasors using these relationships:

$$V_k e^{j\theta V_k} = \frac{2}{T} \int_{-T}^t V(t) \sin(2\pi k F t) dt + j \frac{2}{T} \int_{-T}^t V(t) \cos(2\pi k F t) dt$$

$$I_k e^{j\theta I_k} = \frac{2}{T} \int_{-T}^t I(t) \sin(2\pi k F t) dt + j \frac{2}{T} \int_{-T}^t I(t) \cos(2\pi k F t) dt .$$

In these phasor equations:

- $V(t)$  and  $I(t)$  are the input voltage and current, respectively.
- $T$  is the period of the input signal, or equivalently the inverse of its base frequency  $F$ .

If the input signals have a finite number of harmonics  $n$ , the total real power  $P$  and total reactive power  $Q$  can be calculated from their components:

$$P = \sum_{k=0}^n P_k$$

$$Q = \sum_{k=1}^n Q_k .$$

The summation for  $Q$  does not include the DC component ( $k = 0$ ) because this component only contributes to real power.

## Ports

### Input

#### **v** – Input voltage

scalar

Voltage across element from which to measure power, in V.

Data Types: `single` | `double`

#### **i** – Input current

scalar

Current through element from which to measure power, in A.

Data Types: `single` | `double`

### Output

#### **P** – Real power

scalar or vector

Real power for selected frequency components, in W. If the **Harmonic numbers** parameter value is a scalar, this output is also a scalar.

Data Types: `single` | `double`

#### **Q** – Reactive power

scalar or vector

Reactive power for selected frequency components, in var. If the **Harmonic numbers** parameter value is a scalar, this output is also a scalar.

Data Types: `single` | `double`

## Parameters

### Base frequency (Hz) — Fundamental frequency

60 (default) | positive number

Fundamental frequency corresponding to component  $k=1$ .

### Harmonic numbers — Frequency components

[0 1 2 3] (default) | scalar or vector

Frequency components to include in the output. Specify either a scalar value corresponding to the desired component or a vector of all desired components.

- The value  $k = 0$  corresponds to the DC component.
- The value  $k = 1$  corresponds to the fundamental frequency.
- Values  $k > 1$  correspond to higher-level harmonics.

If you specify a vector, the order of the power outputs correspond to the order of this vector.

### Sample time — Block sample time

0 (default) | positive number

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For continuous operation, set this property to 0. For discrete operation, specify the sample time explicitly as a positive number. This block does not support inherited sample time.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

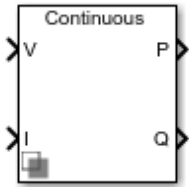
### Blocks

Three-Phase Power Measurement | Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL) | RMS Measurement

## Power Measurement (Three-Phase)

Calculate three-phase real and reactive power

**Library:** Simscape / Electrical / Control / Measurements



### Description

The Power Measurement (Three-Phase) block measures the real and reactive power of an element in a three-phase network. The block outputs the power quantities for each frequency component you specify in the selected symmetrical sequence.

Use this block to measure power for both sinusoidal and nonsinusoidal periodic signals. For single-phase power measurement, consider using the Power Measurement block.

Set the **Sample time** parameter to 0 for continuous-time operation, or explicitly for discrete-time operation.

Specify a vector of all frequency components to include in the power output using the **Harmonic numbers** parameter:

- To output the DC component, specify 0.
- To output the component corresponding to the fundamental frequency, specify 1.
- To output components corresponding to higher-order harmonics, specify  $n > 1$ .

### Equations

For each specified harmonic  $k$ , the block calculates the real power  $P_k$  and reactive power  $Q_k$  for the specified sequence from the phasor equation:

$$P_k + jQ_k = \frac{3}{2}(V_k e^{j\theta_{V_k}})(I_k e^{j\theta_{I_k}})^*$$

where:

- $V_k e^{j\theta_{V_k}}$  is the phasor representing the  $k$ -component voltage of the selected sequence.
- $(I_k e^{j\theta_{I_k}})^*$  is the complex conjugate of  $I_k e^{j\theta_{I_k}}$ , the phasor representing the  $k$ -component current of the selected sequence.

Select the symmetrical sequence used in the power calculation using the **Sequence** parameter:

- **Positive:**

$$V_k e^{j\theta_{V_k}} = V_{k+} e^{j\theta_{V_{k+}}}, \quad I_k e^{j\theta_{I_k}} = I_{k+} e^{j\theta_{I_{k+}}}$$



- Negative:

$$V_k e^{j\theta_{V_k}} = V_{k-} e^{j\theta_{V_{k-}}}, I_k e^{j\theta_{I_k}} = I_{k-} e^{j\theta_{I_{k-}}}$$

- Zero:

$$V_k e^{j\theta_{V_k}} = V_{k0} e^{j\theta_{V_{k0}}}, I_k e^{j\theta_{I_k}} = I_{k0} e^{j\theta_{I_{k0}}}$$

The block calculates the symmetrical set of  $+0$  voltage phasors from the set of  $abc$  voltage phasors using the symmetrical components transform  $S$ :

$$\begin{bmatrix} V_{k+} e^{j\theta_{V_{k+}}} \\ V_{k-} e^{j\theta_{V_{k-}}} \\ V_{k0} e^{j\theta_{V_{k0}}} \end{bmatrix} = S \begin{bmatrix} V_{ka} e^{j\theta_{V_{ka}}} \\ V_{kb} e^{j\theta_{V_{kb}}} \\ V_{kc} e^{j\theta_{V_{kc}}} \end{bmatrix}.$$

For more information about this transform, see Symmetrical Components Transform.

The block obtains this set of  $abc$  voltage phasors from the three-phase input voltage  $V(t)$  as:

$$\begin{bmatrix} V_{ka} e^{j\theta_{V_{ka}}} \\ V_{kb} e^{j\theta_{V_{kb}}} \\ V_{kc} e^{j\theta_{V_{kc}}} \end{bmatrix} = \frac{2}{T} \int_{-T}^t V(t) \sin(2\pi k F t) dt + j \frac{2}{T} \int_{-T}^t V(t) \cos(2\pi k F t) dt,$$

where  $T$  is the period of the input signal, or equivalently the inverse of its base frequency  $F$ .

The block calculates the symmetrical set of current phasors in the same way as it does the voltage.

If the input signals have a finite number of harmonics  $n$ , the total real power  $P$  and total reactive power  $Q$  for the specified sequence can be calculated from their components:

$$P = \sum_{k=0}^n P_k$$

$$Q = \sum_{k=1}^n Q_k.$$

The summation for  $Q$  does not include the DC component ( $k = 0$ ), because this component only contributes to real power.

## Ports

### Input

#### **V** — Input voltage

vector

Three-phase voltage across element from which to measure power, in V.

Data Types: `single` | `double`

### **I — Input current**

vector

Three-phase current through element from which to measure power, in A.

Data Types: `single` | `double`

### **Output**

#### **P — Real power**

scalar or vector | vector

Real power for selected frequency components, in W. If the **Harmonic numbers** parameter value is a scalar, this output is also a scalar.

Data Types: `single` | `double`

#### **Q — Reactive power**

scalar or vector | vector

Reactive power for selected frequency components, in var. If the **Harmonic numbers** parameter value is a scalar, this output is also a scalar.

Data Types: `single` | `double`

### **Parameters**

#### **Base frequency (Hz) — Fundamental frequency**

60 (default) | positive number

Fundamental frequency corresponding to component  $k=1$ .

#### **Harmonic numbers — Frequency components**

[0 1 2 3] (default) | scalar or vector

Frequency components to include in the output. Specify either a scalar value corresponding to the desired component or a vector of all desired components.

- The value  $k = 0$  corresponds to the DC component.
- The value  $k = 1$  corresponds to the fundamental frequency.
- Values  $k > 1$  correspond to higher-level harmonics.

If you specify a vector, the order of the power outputs correspond to the order of this vector.

#### **Sequence — Symmetrical sequence**

Positive (default) | Negative | Zero

Symmetrical sequence of the power output.

#### **Sample time — Block sample time**

0 (default) | positive number

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For continuous operation, set this property to 0. For discrete operation, specify the sample time explicitly as a positive number. This block does not support inherited sample time.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

**Introduced in R2017b**

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

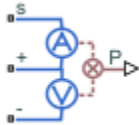
### Blocks

Power Measurement | Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL) | RMS Measurement

# Power Sensor

Ideal instantaneous or cycle-average power measurement

**Library:** Simscape / Electrical / Sensors & Transducers

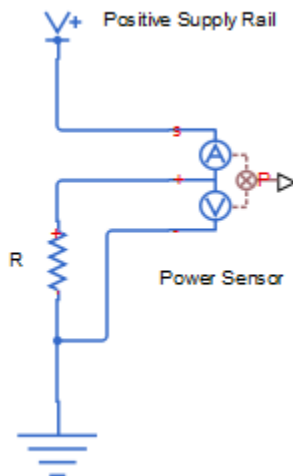


## Description

The Power Sensor block calculates the power taken by the load connected across the + and - terminals under the assumption that only the load is connected to the + terminal. Refer to the block icon for the arrangement of internal current and voltage sensors.

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

The following figure shows how you connect the block to measure power dissipated in a resistor.



For an alternative workflow using data logging to view component powers, see the “Buck Converter” example.

## Ports

### Output

**P – Measured power**

physical signal

Physical signal port that outputs the measured power

### **Conserving**

#### **S – Positive supply rail**

electrical

Electrical conserving port connected to the positive supply rail

#### **+ – Load positive terminal**

electrical

Electrical conserving port connected to the positive terminal of the load

#### **- – Load negative terminal**

electrical

Electrical conserving port connected to the negative terminal of the load

## **Parameters**

### **Measurement type – Measurement type**

Instantaneous power (default) | Instantaneous power

Select whether you want to measure Instantaneous power or Average power over a specified period. The default value is Instantaneous power.

### **Averaging period – Averaging period**

1e-4 s (default)

Fixed period of time for measuring the average power.

### **Dependencies**

This parameter is visible only when you select Average power over a specified period for the **Measurement type** parameter.

## **Version History**

**Introduced in R2012a**

## **Extended Capabilities**

### **C/C++ Code Generation**

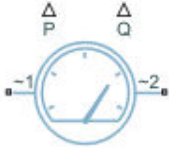
Generate C and C++ code using Simulink® Coder™.

## **See Also**

## Power Sensor (Three-Phase)

Three-phase ideal active and reactive power sensor

**Library:** Simscape / Electrical / Sensors & Transducers



### Description

The Power Sensor (Three-Phase) block implements an ideal sensor for active and reactive power measurement in balanced or unbalanced three-phase branches.

For balanced branches, the block returns the correct value of the active and reactive power at all time instants. This is the preferred option for a load-flow analysis.

For unbalanced branches, to estimate the magnitude and phase, the block measures the phasors of the three voltages and currents independently by using a one-moving-period Fourier transform. For this reason, the output magnitude and phase are zero during the first period.

### Unbalanced Branches Equations

The equivalent balanced phase voltage, current, and apparent power are:

$$V_e = \frac{1}{3} \sqrt{V_{ab_{RMS}}^2 + V_{bc_{RMS}}^2 + V_{ca_{RMS}}^2}$$

$$I_e = \sqrt{\frac{I_{a_{RMS}}^2 + I_{b_{RMS}}^2 + I_{c_{RMS}}^2 + \rho I_{n_{RMS}}^2}{3}}$$

$$S_e = 3V_e I_e$$

where

- $V_{ab_{RMS}}$ ,  $V_{bc_{RMS}}$ , and  $V_{ca_{RMS}}$  are the three phase-to-phase RMS voltages.
- $I_{a_{RMS}}$ ,  $I_{b_{RMS}}$ , and  $I_{c_{RMS}}$  are the RMS currents that flow through each of the three branches.
- $\rho$  is the ratio of neutral conductor resistance over phase conductor resistance, specified by the Neutral conductor resistance divided by phase conductor resistance on page 1-0 parameter.
- $I_{n_{RMS}}$  is the RMS neutral current.

The active power is then defined by:

$$P = V_{a_{RMS}} I_{a_{RMS}} \cos(\varphi_{V_a} - \varphi_{I_a}) + V_{b_{RMS}} I_{b_{RMS}} \cos(\varphi_{V_b} - \varphi_{I_b}) + V_{c_{RMS}} I_{c_{RMS}} \cos(\varphi_{V_c} - \varphi_{I_c}),$$

where:

- $V_{a_{RMS}}$ ,  $V_{b_{RMS}}$ , and  $V_{c_{RMS}}$  are the voltages of phase a, b, and c respectively.

- $\varphi_{V_a}$ ,  $\varphi_{V_b}$ , and  $\varphi_{V_c}$  are the phase shifts of the *a*-phase, *b*-phase, and *c*-phase voltages.
- $\varphi_{I_a}$ ,  $\varphi_{I_b}$ , and  $\varphi_{I_c}$  are the phase shifts of the *a*-phase, *b*-phase, and *c*-phase currents.

To compute the reactive power, the block first calculates its absolute value and then finds the sign by looking at the sign of the phase shift:

$$\begin{aligned} \text{abs}(Q) &= \sqrt{S_e^2 - P^2} \\ \text{sign}(Q) &= \text{sign}(\varphi) \\ Q &= \text{abs}(Q) * \text{sign}(Q) \end{aligned}$$

where  $\varphi = \varphi_{V_*} - \varphi_{I_*}$ . The \* symbol denotes the central phase in the  $[-\pi, \pi]$  range.

### Load-Flow Analysis

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize a machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”.

## Ports

### Conserving

#### ~1 — Three-phase port

electrical

Expandable three-phase electrical port.

#### ~2 — Three-phase port

electrical

Expandable three-phase electrical port.

### Output

#### P — Active power

physical

Physical signal port associated with the active power measurements.

#### Q — Reactive power

physical

Physical signal port associated with the reactive power measurements.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Load type – Load type**

Balanced (default) | Unbalanced

Type of load. Select **Balanced** if the three-phase branches are balanced, **Unbalanced** if the branches are not balanced.

**Rated frequency – Rated frequency**

60 Hz (default) | scalar

Rated frequency.

**Dependencies**

To enable this parameter, set **Load Type** to Unbalanced.

**Downstream configuration – Downstream configuration**

Without neutral (default) | With neutral

Whether the downstream configuration is with or without neutral.

**Dependencies**

To enable this parameter, set **Load Type** to Unbalanced.

**Neutral conductor resistance divided by phase conductor resistance – Ratio of neutral conductor resistance over phase conductor resistance**

1 (default) | scalar

Ratio of neutral conductor resistance over phase conductor resistance,  $\frac{R_{neutral}}{R_{phase}}$  ..

**Dependencies**

To enable this parameter, set **Downstream configuration** to With neutral.

## Version History

Introduced in R2020b

## References

[1] Willems, Jacques, "The IEEE standard 1459: What and why?", *2010 IEEE International Conference on Applied Measurements for Power Systems, Proceedings*: 41-46.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.



## **See Also**

[Current and Voltage Sensor \(Three-Phase\)](#) | [Line Voltage Sensor \(Three-Phase\)](#) | [Phase Voltage Sensor \(Three-Phase\)](#) | [Power Sensor](#)

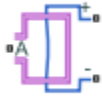
## **Topics**

[“Perform a Load-Flow Analysis Using Simscape Electrical”](#)  
[“Frequency and Time Simulation Mode”](#)

# Pressure Transducer

Behavioral model of generic pressure transducer that turns pressure measurement into voltage

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Pressure Transducer block models a generic pressure transducer that turns a pressure measurement into a voltage. The block lets you measure pressure in a variety of domains. Specify the domain type using the **Fluid port type** parameter.

The output voltage is linearly proportional to the pressure, and the block outputs zero volts if the pressure is less than zero. 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.

Optionally, if you set the **Dynamics** parameter to `Model transducer bandwidth`, then the dynamics of the sensor are approximated by a first-order lag. The lag is determined by the **Bandwidth** parameter. If you select this option, you must also specify an initial condition for the lag by using the **Measured pressure** variable target.

If running your simulation with a fixed-step solver, or generating code for hardware-in-the-loop testing, MathWorks recommends that you set the **Dynamics** parameter to `No dynamics – Suitable for HIL`, because this avoids the need for a small simulation time step if the sensor bandwidth is high.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

The **Measured pressure** variable target specifies the initial output for the sensor.

## Ports

### Conserving

#### A – Fluid port

Gas | Hydraulic | Thermal Liquid | Moist Air | Two-Phase Fluid

Conserving fluid port for pressure measurement. The port type is defined by the **Fluid port type** parameter value.

#### + – Positive terminal

electrical

Electrical conserving port associated with the positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the negative terminal.

## Parameters

### Fluid port type – Type of fluid port

Gas (default) | Hydraulic | Thermal Liquid | Moist Air | Two-Phase Fluid

Select the fluid port type for pressure measurement:

- Gas (default)
- Hydraulic
- Thermal Liquid
- Moist Air
- Two-Phase Fluid

### Pressure range – Maximum pressure

1e6 Pa (default)

Maximum pressure that the sensor can measure, in Pa.

### Operation mode – Reference point for pressure measurement

Absolute (default) | Gauge | Sealed-Gauge

Select one of the following options to define the reference point for the pressure measurement:

- **Absolute** — The pressure measurement is with respect to zero absolute pressure, that is, vacuum. This is the default option.
- **Gauge** — The pressure measurement is with respect to atmospheric pressure. Atmospheric pressure is defined by the Gas Properties block in the Simscape Foundation library.
- **Sealed-Gauge** — The pressure measurement is referenced to an internal sealed chamber. If you select this option, use the **Reference pressure** parameter to specify the reference point for pressure measurement.

### Reference pressure – Reference pressure

101325 Pa (default)

Reference pressure in the internal sealed chamber.

**Dependencies**

This parameter is visible only when you select `Sealed -Gauge` for the **Operation mode** parameter.

**Full-scale deflection — Full-scale deflection**

5 V (default)

Output voltage when the measured pressure is equal to, or greater than, the **Pressure range** parameter value.

**Output resistance — Transducer output resistance**

200 Ohm (default)

Output resistance of the transducer.

**Dynamics — Sensor dynamics option**

No dynamics – Suitable for HIL (default) | Model transducer bandwidth

Select one of the following options for modeling sensor dynamics:

- **No dynamics – Suitable for HIL** — Do not model sensor dynamics. Use this option when running your simulation fixed step or generating code for hardware-in-the-loop testing, because this avoids the need for a small simulation time step if the sensor bandwidth is high. This is the default option.
- **Model transducer bandwidth** — Model sensor dynamics with a first-order lag approximation, based on the **Bandwidth** parameter value. You can control the initial condition for the lag by specifying the **Measured pressure** variable target.

**Bandwidth — Bandwidth**

5 kHz (default)

Determines the value of the sensor lag.

**Dependencies**

This parameter is visible only when you select `Model transducer bandwidth` for the **Dynamics** parameter.

## Version History

**Introduced in R2012b****Fluid Port Type**

*Behavior changed in R2019a*

Prior to R2019a, the Pressure Transducer block had a pneumatic measurement port. Pneumatic blocks are no longer part of the Foundation library, but they are included in the Simscape product installation as an example custom library. The legacy Pressure Transducer block, with a pneumatic port, is now part of this custom library.

From R2019a forward, use the Gas library for modeling pneumatic systems, and use the latest version of the Pressure Transducer block with a gas port.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

# Primary Winding

(To be removed) Linear nonideal transformer winding

---

**Note** The Primary Winding block will be removed in a future release. Use the Winding, Eddy Current, and Fundamental Reluctance blocks instead. To learn how to create a custom transformer without this block, refer to “Push-Pull Buck Converter in Continuous Conduction Mode”.

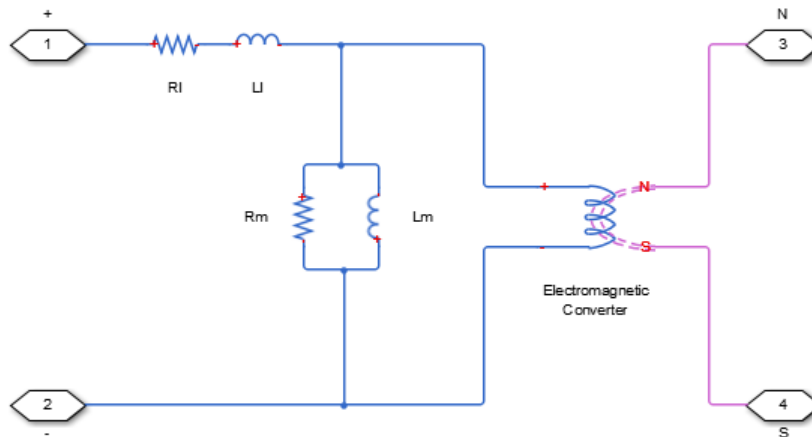
---

## Library

Simscape / Electrical / Power Systems / Passive Devices / Transformers / Fundamental Components

## Description

The Primary Winding block models linear nonideal winding of a transformer with linear winding leakage and linear core magnetization effects. Although magnetization effects occur in the magnetic core, it is common practice to place mathematically equivalent electrical components on the electrical winding and parameterize them using electrical parameters. The figure shows the equivalent circuit diagram for the primary winding.



- $R_l$  is the leakage resistance.
- $L_l$  is the leakage inductance.
- $R_m$  is the magnetization resistance.
- $L_m$  is the magnetization inductance.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

- +  
Positive electrical conserving port
- Negative electrical conserving port
- N  
North magnetic conserving port
- S  
South magnetic conserving port

## Parameters

### Main

#### Number of winding turns

Number of wire turns on the transformer winding. The default value is 10.

#### Leakage resistance

Power loss in the winding. The default value is  $1e-3$  Ohm.

#### Leakage inductance

Magnetic flux loss in the winding. The default value is  $1e-3$  H.

#### Core-loss resistance

Magnetic losses in the transformer core. The default value is  $1e6$  Ohm.

#### Magnetization inductance

Magnetic effects in the transformer core when operating in its linear region. The default value is  $1e6$  H.

## Version History

**Introduced in R2013b**

## See Also

Electromagnetic Converter | Secondary Winding

## Topics

“Three-Phase Custom Zigzag Transformer”

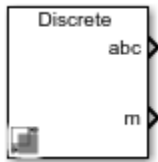
“Push-Pull Buck Converter in Continuous Conduction Mode”

“Push-Pull Buck Converter in Discontinuous Conduction Mode”

## Programmable Signal Generator (Three-Phase)

Discrete-time or continuous-time three-phase sinusoidal signal generator with programmable magnitude, phase, frequency, and overlapping harmonics

**Library:** Simscape / Electrical / Control / General Control



### Description

The Programmable Signal Generator (Three-Phase) generates a three-phase sinusoidal signal with programmable magnitude, phase, and frequency over time. You can specify one or two harmonics that can be overlapped on the signal fundamental frequency.

You can switch between continuous and discrete implementations of the block by using the **Sample time** parameter. To configure the block for continuous time, set **Sample time** to 0. To configure the block for discrete time, set **Sample time** to a positive, nonzero value.

### Ports

#### Output

##### **abc** — Three-phase sinusoidal signal

vector

Three-phase sinusoidal signal, returned as a vector.

Data Types: `single` | `double`

##### **m** — Signal measurements

vector

Magnitude, frequency, phase, and angle of the sine wave of all three phases of the sinusoidal signal, returned as a vector.

Data Types: `single` | `double`

### Parameters

#### **Sample time** — Block sample time

1e-5 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.



## Magnitude

### Variation configuration — Type of magnitude variation

Constant (default) | Ramp | Step | Modulation | Table

Type of variation to apply to the signal magnitude.

### Magnitude (value) — Magnitude of generated three-phase sine wave

100 (default) | scalar

Magnitude of the generated positive-sequence three-phase sine wave.

#### Dependencies

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

### Rate of change (value/s) — Magnitude rate of change

100 (default) | scalar

Rate of change of the magnitude.

#### Dependencies

To enable this parameter, set **Variation configuration** to Ramp.

### Step value (value) — Magnitude step value

100 (default) | scalar

Amplitude of the step variation for the signal magnitude.

#### Dependencies

To enable this parameter, set **Variation configuration** to Step.

### Modulation magnitude — Modulation magnitude

50 (default) | scalar

Magnitude of the modulation for the signal magnitude.

#### Dependencies

To enable this parameter, set **Variation configuration** to Modulation.

### Modulation frequency — Modulation frequency

10 (default) | scalar

Frequency of the modulation for the signal magnitude.

#### Dependencies

To enable this parameter, set **Variation configuration** to Modulation.

### Start time — Start time of magnitude variation

0.1 (default) | scalar

Time when the programmed variation on the magnitude starts.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Stop time — Stop time of magnitude variation**

0.6 (default) | scalar

Time when the programmed variation on the magnitude stops.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Table data — Magnitude table values**

[0 30 60 90] (default) | vector

Values of the signal magnitude that correspond to the time values specified in the **Time values** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Time values — Magnitude time values**

[0 0.3 0.6 0.9] (default) | vector

Time values when the signal magnitude changes to the values specified in the **Table data** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Frequency****Variation configuration — Type of frequency variation**

Constant (default) | Ramp | Step | Modulation | Table

Type of variation to apply to the signal frequency.

**Frequency (Hz) — Frequency of generated three-phase sine wave**

100 (default) | scalar

Frequency of the generated positive-sequence three-phase sine wave.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Rate of change (Hz/s) — Frequency rate of change**

100 (default) | scalar

Rate of change of the frequency.

**Dependencies**

To enable this parameter, set **Variation configuration** to Ramp.

**Step value (Hz) — Frequency step value**

100 (default) | scalar

Amplitude of the step variation for the signal frequency.

**Dependencies**

To enable this parameter, set **Variation configuration** to Step.

**Modulation magnitude — Frequency modulation magnitude**

50 (default) | scalar

Magnitude of the modulation for the signal frequency.

**Dependencies**

To enable this parameter, set **Variation configuration** to Modulation.

**Modulation frequency — Frequency modulation frequency**

10 (default) | scalar

Frequency of the modulation for the signal frequency.

**Dependencies**

To enable this parameter, set **Variation configuration** to Modulation.

**Start time — Start time of frequency variation**

0.1 (default) | scalar

Time when the programmed variation on the frequency starts.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Stop time — Stop time of frequency variation**

0.6 (default) | scalar

Time when the programmed variation on the frequency stops.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Table data — Frequency table values**

[0 30 60 90] (default) | vector

Values of the signal frequency that correspond to the time values specified in the **Time values** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Time values — Frequency time values**

[0 0.3 0.6 0.9] (default) | vector

Time values when the signal frequency changes to the values specified in the **Table data** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Phase****Variation configuration — Type of phase variation**

Constant (default) | Ramp | Step | Modulation | Table

Type of variation to apply to the phase magnitude.

**Phase shift (deg) — Phase shift of generated three-phase sine wave**

100 (default) | scalar

Phase shift of the generated positive-sequence three-phase sine wave.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Rate of change (deg/s) — Phase rate of change**

100 (default) | scalar

Rate of change of the phase.

**Dependencies**

To enable this parameter, set **Variation configuration** to Ramp.

**Step value (deg) — Phase step value**

100 (default) | scalar

Amplitude of the step variation for the signal phase.

**Dependencies**

To enable this parameter, set **Variation configuration** to Step.

**Modulation magnitude — Phase modulation magnitude**

50 (default) | scalar

Magnitude of the modulation for the signal phase.

**Dependencies**

To enable this parameter, set **Variation configuration** to Modulation.

**Modulation frequency — Phase modulation frequency**

10 (default) | scalar

Frequency of the modulation for the signal phase.

**Dependencies**

To enable this parameter, set **Variation configuration** to Modulation.

**Start time — Start time of phase variation**

0.1 (default) | scalar

Time when the programmed variation on the phase starts.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Stop time — Stop time of phase variation**

0.6 (default) | scalar

Time when the programmed variation on the phase stops.

**Dependencies**

To enable this parameter, set **Variation configuration** to Constant, Ramp, Step, or Modulation.

**Table data — Phase table values**

[0 30 60 90] (default) | vector

Values of the signal phase that correspond to the time values specified in the **Time values** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Time values — Phase time values**

[0 0.3 0.6 0.9] (default) | vector

Time values when the signal frequency changes to the values specified in the **Table data** parameter.

**Dependencies**

To enable this parameter, set **Variation configuration** to Table.

**Harmonics**

**Enable harmonic A — Harmonic A overlap**

off (default) | on

Whether harmonic A overlaps the fundamental frequency of the generated signal.

**Harmonic sequence — Type of harmonic A sequence**

Zero Sequence (default) | Positive Sequence | Negative Sequence

Type of sequence for harmonic A.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Harmonic order — Order of harmonic A**

3 (default) | scalar

Order of harmonic A that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Harmonic magnitude — Magnitude of harmonic A**

50 (default) | scalar

Magnitude of harmonic A that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Harmonic phase shift — Phase shift of harmonic A**

180 (default) | scalar

Phase shift of harmonic A that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Start time — Start time of harmonic A overlap**

0.1 (default) | scalar

Time when the harmonic A overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Stop time — Stop time of harmonic A overlap**

0.6 (default) | scalar

Time when the harmonic A stops overlapping the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic A** check box.

**Enable harmonic B — Harmonic B overlap**

off (default) | on

Whether harmonic B overlaps the fundamental frequency of the generated signal.

**Harmonic sequence — Type of sequence of harmonic B**

Zero Sequence (default) | Positive Sequence | Negative Sequence

Type of sequence for harmonic B.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

**Harmonic order — Order of harmonic B**

3 (default) | scalar

Order of harmonic B that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

**Harmonic magnitude — Magnitude of harmonic B**

50 (default) | scalar

Magnitude of harmonic B that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

**Harmonic phase shift — Phase shift of harmonic B**

180 (default) | scalar

Phase shift of harmonic B that overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

**Start time — Start time of harmonic B overlap**

0.1 (default) | scalar

Time when the harmonic B overlaps the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

**Stop time — Stop time of harmonic B overlap**

0.6 (default) | scalar

Time when the harmonic B stops overlapping the signal fundamental frequency.

**Dependencies**

To enable this parameter, select the **Enable harmonic B** check box.

## Version History

Introduced in R2020a

### See Also

Sine Wave Generator (Three-Phase) | Stair Generator

# Programmable Voltage Source

Single-phase AC voltage source with optional programmable magnitude, frequency, phase shift and DC offset

**Library:** Simscape / Electrical / Sources



## Description

The Programmable Voltage Source block models a single-phase AC voltage source with programmable magnitude, frequency, phase shift and DC offset. Choose the external mode to specify these quantities by physical input signals **M**, **F**, **Phi** and **DC**. Harmonics and noise can be included in the voltage source.

For relevant equations, see the Voltage Source block.

## Limitations

Simulating with harmonics enabled slows down simulation. If you include harmonics, choose a sample time such that harmonics are generated only at frequencies of interest, and not higher.

Simulating with noise enabled slows down simulation. If you include noise, choose a sample time such that noise is generated only at frequencies of interest, and not higher.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Input

#### DC — DC component

physical signal

Physical signal input associated with the DC component of the voltage.



**F — Frequency**

physical signal

Physical signal input associated with the frequency.

**M — Magnitude**

physical signal

Physical signal input associated with the magnitude.

**Phi — Phase shift**

physical signal

Physical signal input associated with the phase shift.

**Conserving****+ — Positive voltage**

electrical

Electrical conserving port associated with the positive voltage.

**- — Negative voltage**

electrical

Electrical conserving port associated with the negative voltage.

**Parameters****AC Magnitude****AC magnitude configuration — AC voltage magnitude configuration**

Constant (default) | RampStepModulationExternal

Configure the magnitude of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**AC voltage peak magnitude — AC voltage peak magnitude**

100 V (default)

AC voltage peak magnitude.

**Dependencies**This parameter is exposed when the **AC magnitude configuration** parameter is set to Constant, Ramp, Step, or Modulation.**Rate of change — AC voltage rate of change**

1 V/s (default)

AC voltage rate of change.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Ramp.

**Step amplitude — AC voltage step amplitude**

1 V (default)

AC voltage step amplitude.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Step.

**Modulation magnitude — AC voltage modulation magnitude**

1 V (default)

AC voltage modulation magnitude.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Modulation.

**Modulation frequency — AC voltage modulation frequency**

1 Hz (default)

AC voltage modulation frequency.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Modulation.

**Start time — AC voltage start time**

1s (default)

Simulation time for start of AC voltage.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Ramp, Step, or Modulation.

**Stop time — AC voltage stop time**

2s (default)

Simulation time for stop of AC voltage.

**Dependencies**

This parameter is exposed when the **AC magnitude configuration** parameter is set to Ramp, Step, or Modulation.

**Frequency****AC frequency configuration — AC frequency configuration**

Constant (default) | RampStepModulationExternal

Configure the frequency of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**AC voltage frequency — AC voltage frequency**

60 Hz (default)

AC voltage frequency.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Constant, Ramp, Step, or Modulation.

**Rate of change — AC frequency rate of change**

1 Hz/s (default)

AC frequency rate of change.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Ramp.

**Step amplitude — AC frequency step amplitude**

1 Hz (default)

AC frequency step amplitude.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Step.

**Modulation magnitude — AC frequency modulation magnitude**

1 Hz (default)

AC frequency modulation magnitude.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Modulation.

**Modulation frequency — AC frequency modulation frequency**

1 Hz (default)

AC frequency modulation frequency.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Modulation.

**Start time — AC frequency start time**

1s (default)

Simulation time for starting AC frequency.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Ramp, Step, or Modulation.

**Stop time — AC frequency start time**

2s (default)

Simulation time for stopping AC frequency.

**Dependencies**

This parameter is exposed when the **AC frequency configuration** parameter is set to Ramp, Step, or Modulation.

**Phase****AC phase shift configuration — AC phase shift configuration**

Constant (default) | RampStepModulationExternal

Configure the phase of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**AC voltage phase shift — AC voltage phase shift**

0 deg (default)

AC voltage phase shift.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Constant, Ramp, Step, or Modulation.

**Rate of change — AC phase shift rate of change**

1 deg/s (default)

AC phase shift rate of change.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Ramp.

**Step amplitude — AC phase shift step amplitude**

1 deg (default)

AC phase shift step amplitude.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Step.

**Modulation magnitude — AC phase shift modulation magnitude**

1 deg (default)

AC phase shift modulation magnitude.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Modulation.

**Modulation frequency – AC phase shift modulation frequency**

1 Hz (default)

AC phase shift modulation frequency.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Modulation.

**Start time – AC phase shift start time**

1s (default)

Simulation time for starting AC phase shift.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Ramp, Step, or Modulation.

**Stop time – AC phase shift start time**

2s (default)

Simulation time for stopping AC phase shift.

**Dependencies**

This parameter is exposed when the **AC phase shift configuration** parameter is set to Ramp, Step, or Modulation.

**DC Voltage**

**DC voltage configuration – DC voltage configuration**

Constant (default) | RampStepModulationExternal

Configure the DC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**DC voltage – DC voltage magnitude**

0 V (default)

DC voltage magnitude.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Constant, Ramp, Step, or Modulation.

**Rate of change — DC voltage rate of change**

1 V/s (default)

DC voltage rate of change.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Ramp.

**Step amplitude — DC voltage step amplitude**

1 V (default)

DC voltage step amplitude.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Step.

**Modulation magnitude — DC voltage modulation magnitude**

1 V (default)

DC voltage modulation magnitude.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Modulation.

**Modulation frequency — DC voltage modulation frequency**

1 Hz (default)

DC voltage modulation frequency.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Modulation.

**Start time — DC voltage start time**

1s (default)

Simulation time for start of DC voltage.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Ramp, Step, or Modulation.

**Stop time — DC voltage stop time**

2s (default)

Simulation time for stop of DC voltage.

**Dependencies**

This parameter is exposed when the **DC magnitude configuration** parameter is set to Ramp, Step, or Modulation.

## Harmonics

### Source harmonics — Source harmonics configuration

None (default) | Generate harmonics

Configure the source harmonics.

#### Dependencies

Selecting Generate harmonics exposes related parameters.

### Harmonic orders — Harmonic orders

[5, 7, 11, 13] (default)

Harmonic orders.

#### Dependencies

This parameter is exposed when the **Source harmonics** parameter is set to Generate harmonics.

### Harmonic to base magnitude ratios — Harmonic to base magnitude ratios

[.1, .1, .1, .1] (default)

Harmonic to base magnitude ratios. Specify the same number of elements as is specified for the **Harmonic orders** parameter.

#### Dependencies

This parameter is exposed when the **Source harmonics** parameter is set to Generate harmonics.

### Harmonic phase shifts — Harmonic phase shifts

[0, 0, 0, 0] deg (default)

Harmonic phase shifts. Specify the same number of elements as is specified for the **Harmonic orders** parameter.

#### Dependencies

This parameter is exposed when the **Source harmonics** parameter is set to Generate harmonics.

### Start time — Source harmonics start time

-infs (default)

Simulation time for start of source harmonics.

#### Dependencies

This parameter is exposed when the **Source harmonics** parameter is set to Generate harmonics.

### Stop time — Source harmonics stop time

infs (default)

Simulation time for stop of source harmonics.

#### Dependencies

This parameter is exposed when the **Source harmonics** parameter is set to Generate harmonics.

## Noise

### Noise mode — Option to include noise

Disabled (default) | Enabled

Noise configuration.

#### Dependencies

Selecting Enabled exposes related parameters.

### Power spectral density — Power spectral density

0 V<sup>2</sup>/Hz (default)

Single-sided spectrum noise power. Density function for the square of the voltage, commonly thought of as a power into a 1 ohm load. To avoid unit ambiguity, some datasheets quote noise voltage as a noise density with units of V/√Hz. In this case, enter the square of the noise density quoted in the datasheet as the parameter value.

Selecting Enabled for the **Noise mode** parameter exposes this parameter.

#### Dependencies

Selecting Enabled for the **Noise mode** parameter exposes this parameter.

### Repeatability — Random number seed control

Not repeatable (default) | Repeatable | Specify seed

The random number seed is the number that initializes the random number generator. The seed is 0 or a positive integer. To control the random number seed, set this parameter to:

- **Not repeatable** — The seed changes every time you simulate your model. The block resets the random seed using the MATLAB random number generator command:

```
seed = randi(2^32-1);
```

- **Repeatable** — The seed is the same random number at the start of every simulation. The block sets the value using the same MATLAB random number generator command used by the **Not repeatable** parameter.

When you add a Force Noise Source block to your model from the Sources library, the block generates and stores a random value for the repeated seed. When you make a copy of the Force Noise Source block from an existing block in a model, the copy generates a new random value for the repeated seed.

- **Specify seed** — The seed is a number that you specify using the **Seed** parameter. The **Seed** parameter is only available when you choose **Specify seed** for the **Repeatability** parameter.

#### Dependencies

Selecting Enabled for the **Noise mode** parameter exposes this parameter.

Selecting Repeatable or Specify seed exposes related parameters.

### Auto-generated seed used for repeatable option — Auto-generate seed

0 (default)

Seed is auto-generated.



**Dependencies**

Selecting Enabled for the **Noise mode** parameter and Repeatable for the **Repeatability** parameter exposes this parameter.

**Seed — Random number generation seed value**

0 (default)

The seed must be 0 or a positive integer. This parameter is only available when you select Specify seed for the **Repeatability** parameter.

**Dependencies**

Selecting Enabled for the **Noise mode** parameter and Specify seed for the **Repeatability** parameter exposes this parameter.

**Sample time — Time step period and offset**

1e-3 s (default) | [*step*, *offset*] s

The values of the time *step* period and the initial time *offset*. If you specify a scalar value for *step*, the block assumes an *offset* value of 0. To specify a nonzero value for the initial time offset, specify the parameter values using the vector [*step*, *offset*]. The *offset* value must be less than the *step* value and greater than or equal to zero.

**Dependencies**

Selecting Enabled for the **Noise mode** parameter exposes related parameters.

## Version History

Introduced in R2019a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Programmable Voltage Source (Three-Phase) | Voltage Source | Voltage Source (Three-Phase)

## Programmable Voltage Source (Three-Phase)

Three-phase voltage source with optional programmable AC magnitude, frequency, and phase shift

**Library:** Simscape / Electrical / Sources



### Description

The Programmable Voltage Source (Three-Phase) block models a three-phase voltage source with programmable AC magnitude, frequency, and phase shift. Choose the external mode to specify these quantities by physical input signals **M**, **F**, and **Phi**. You can add harmonics and internal impedance to the voltage source.

For relevant equations, see the Voltage Source (Three-Phase) block.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Ports

#### Input

##### **F — Frequency**

physical signal

Physical signal input associated with the frequency.

##### **M — Magnitude**

physical signal

Physical signal input associated with the magnitude.

##### **Phi — Phase shift**

physical signal

Physical signal input associated with the phase shift.

**Conserving****n — Neutral**  
electrical

Electrical conserving port associated with a neutral point.

**~ — Three-phase AC voltage**  
electrical

Expandable electrical conserving three-phase port associated with the AC voltage.

**Parameters****Modeling option — Whether to model composite or expanded three-phase ports**  
Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**AC Magnitude****AC magnitude configuration — AC voltage magnitude configuration**  
Constant (default) | Ramp | Step | Modulation | External

Configure the magnitude of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**Rated voltage (phase-to-phase RMS) — Rated AC voltage**  
100 V (default)

Rated AC voltage (phase-to-phase RMS).

**Dependencies**To enable this parameter, set **AC magnitude configuration** to Constant, Ramp, Step, or Modulation.**Rate of change — AC voltage rate of change**  
1 V/s (default)

AC voltage rate of change.

**Dependencies**To enable this parameter, set **AC magnitude configuration** to Ramp.

**Step amplitude – AC voltage step amplitude**

1 V (default)

AC voltage step amplitude.

**Dependencies**

To enable this parameter, set **AC magnitude configuration** to Step.

**Modulation magnitude – AC voltage modulation magnitude**

1 V (default)

AC voltage modulation magnitude.

**Dependencies**

To enable this parameter, set **AC magnitude configuration** to Modulation.

**Modulation frequency – AC voltage modulation frequency**

1 Hz (default)

AC voltage modulation frequency.

**Dependencies**

To enable this parameter, set **AC magnitude configuration** to Modulation.

**Start time – AC voltage start time**

1s (default)

Simulation time for start of AC voltage.

**Dependencies**

To enable this parameter, set **AC magnitude configuration** to Ramp, Step, or Modulation.

**Stop time – AC voltage stop time**

2s (default)

Simulation time for stop of AC voltage.

**Dependencies**

To enable this parameter, set **AC magnitude configuration** to Ramp, Step, or Modulation.

**Frequency**

**AC frequency configuration – AC frequency configuration**

Constant (default) | Ramp | Step | Modulation | External

Configure the frequency of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**Rated frequency — Rated frequency**

60 Hz (default)

Rated frequency.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Constant, Ramp, Step, or Modulation.

**Rate of change — AC frequency rate of change**

1 Hz/s (default)

AC frequency rate of change.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Ramp.

**Step amplitude — AC frequency step amplitude**

1 Hz (default)

AC frequency step amplitude.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Step.

**Modulation magnitude — AC frequency modulation magnitude**

1 Hz (default)

AC frequency modulation magnitude.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Modulation.

**Modulation frequency — AC frequency modulation frequency**

1 Hz (default)

AC frequency modulation frequency.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Modulation.

**Start time — AC frequency start time**

1s (default)

Simulation time for starting AC frequency.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Ramp, Step, or Modulation.

**Stop time — AC frequency start time**

2s (default)

Simulation time for stopping AC frequency.

**Dependencies**

To enable this parameter, set **AC frequency configuration** to Ramp, Step, or Modulation.

**Phase****AC phase shift configuration — AC phase shift configuration**

Constant (default) | Ramp | Step | Modulation | External

Configure the phase of the AC component of the voltage.

**Dependencies**

Selecting Constant, Ramp, Step, or Modulation exposes related parameters.

Selecting External exposes a physical signal input port.

**Phase shift — AC voltage phase shift**

0 deg (default)

AC voltage phase shift.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Constant, Ramp, Step, or Modulation.

**Rate of change — AC phase shift rate of change**

1 deg/s (default)

AC phase shift rate of change.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Ramp.

**Step amplitude — AC phase shift step amplitude**

1 deg (default)

AC phase shift step amplitude.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Step.

**Modulation magnitude — AC phase shift modulation magnitude**

1 deg (default)

AC phase shift modulation magnitude.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Modulation.

**Modulation frequency — AC phase shift modulation frequency**

1 Hz (default)

AC phase shift modulation frequency.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Modulation.

**Start time — AC phase shift start time**

1s (default)

Simulation time for starting AC phase shift.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Ramp, Step, or Modulation.

**Stop time — AC phase shift start time**

2s (default)

Simulation time for stopping AC phase shift.

**Dependencies**

To enable this parameter, set **AC phase shift configuration** to Ramp, Step, or Modulation.

**Impedance****Source impedance — Source impedance configuration**

None (default) | X/R Ratio | Series R | Series L | Series RL

Source impedance configuration.

**Dependencies**

Selecting X/R Ratio, Series R, Series L, or Series RL exposes related parameters.

**Short-circuit power level — Short-circuit power level**

1e6 V\*A (default)

Short-circuit power level.

**Dependencies**

To enable this parameter, set **Source impedance** to X/R Ratio.

**Source X/R ratio — X/R ratio**

15 (default)

Source X/R ratio.

**Dependencies**

To enable this parameter, set **Source impedance** to X/R Ratio.

**Source resistance — Resistance**

0.01 Ohm (default)

Source resistance.

**Dependencies**

To enable this parameter, set **Source impedance** to Series R or Series RL.

**Source inductance – Inductance**

3.97e-4 H (default)

Source inductance.

**Dependencies**

To enable this parameter, set **Source impedance** to Series L or Series RL.

**Parasitic parallel conductance – Parasitic parallel conductance**

1e-6 1/Ohm (default)

Parasitic parallel conductance.

**Dependencies**

To enable this parameter, set **Source impedance** to X/R Ratio, Series L, or Series RL.

**Harmonics****Source harmonics – Source harmonics configuration**

None (default) | Generate harmonics

Configuration for the source harmonics.

**Dependencies**

Selecting Generate harmonics exposes related parameters.

**Harmonic orders – Harmonic orders**

[5, 7, 11, 13] (default)

Harmonic orders.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.

**Harmonic to base magnitude ratios – Harmonic to base magnitude ratios**

[.1, .1, .1, .1] (default)

Harmonic to base magnitude ratios. Specify the same number of elements as is specified for the **Harmonic orders** parameter.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.

**Harmonic phase shifts – Harmonic phase shifts**

[0, 0, 0, 0] deg (default)

Harmonic phase shifts. Specify the same number of elements as is specified for the **Harmonic orders** parameter.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.



**Harmonic sequences — Harmonic sequences**

[1, 1, 1, 1] (default)

Harmonic sequences:

- 0 indicates zero-sequence
- 1 indicates positive-sequence
- 2 indicates negative-sequence

Specify the same number of elements as is specified for the **Harmonic orders** parameter.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.

**Start time — Source harmonics start time**

-infs (default)

Simulation time for start of source harmonics.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.

**Stop time — Source harmonics stop time**

infs (default)

Simulation time for stop of source harmonics.

**Dependencies**

To enable this parameter, set **Source harmonics** to Generate harmonics.

## Version History

Introduced in R2019a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Programmable Voltage Source | Voltage Source | Voltage Source (Three-Phase)

**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Proximity Sensor

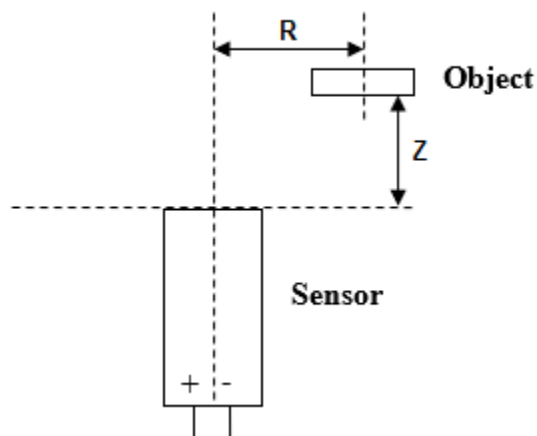
Ideal behavioral model of simple distance sensor

**Library:** Simscape / Electrical / Sensors & Transducers

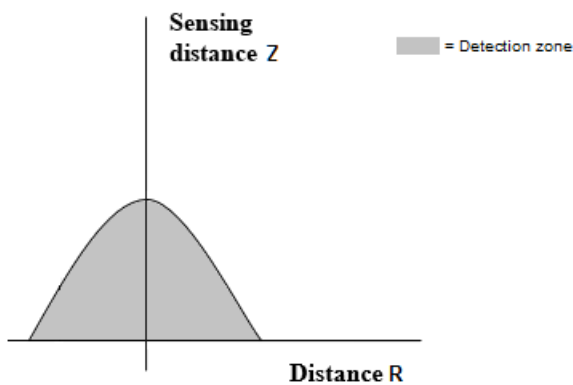


## Description

The Proximity Sensor block represents a simple proximity sensor. The sensing distance  $Z$  is defined as the distance normal to the sensor surface at which the sensor detects an object for a given radial offset  $R$ , as shown in the following figure.



This figure shows a typical sensing distance curve:



The sensor switches to detect the object if the  $R$  and  $Z$  inputs are below the sensing curve, in the detection zone.

The output is modeled by an electrical switch which can either be Normally Open (N.O.) or Normally Closed (N.C.) when no object is detected.

## Ports

### Input

#### **R – Radial distance**

physical signal

Physical signal port associated with the radial distance to the sensor.

#### **Z – Perpendicular distance**

physical signal

Physical signal port associated with the perpendicular distance to the sensor.

### Conserving

#### **+ – Positive voltage**

electrical

Electrical conserving port associated with the proximity sensor positive voltage.

#### **- – Negative voltage**

electrical

Electrical conserving port associated with the proximity sensor negative voltage.

## Parameters

#### **Vector of radial offset distances R – Vector of radial offset distances**

[-25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25] mm (default)

Vector of distances from the sensor to the object resolved into a plane tangential to the sensor head.

#### **Corresponding sensing distances Z – Corresponding sensing distances**

[0, 0, 5, 8, 9.5, 10, 9.5, 8, 5, 0, 0] mm (default)

Vector of distances from the sensor to the object resolved with respect to a normal vector at the sensor head.

#### **Output when not detected – Output when not detected**

Normally Open (N.O.) (default) | Normally Closed (N.C.)

Indicates whether the output is Normally Open (N.O.), meaning the output becomes closed only when the object is detected, or Normally Closed (N.C.), meaning the output becomes open only when the object is detected.

#### **Closed resistance R\_closed – Closed resistance**

0.01 Ohm (default)

Resistance between the + and - ports when the output contacts are closed.

**Open conductance G\_open — Open conductance**

1e-8 1/Ohm (default)

Conductance between the + and - ports when the output contacts are open.

**Version History**

**Introduced in R2008a**

**Extended Capabilities**

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

# PS Sensor

Generic linear transducer with electrical output

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The PS Sensor block represents a generic linear sensor. The block converts the physical signal input  $U$  into an electrical output  $Y$  across the + and - ports. The **Output type** parameter value determines which of the following electrical outputs the block produces:

- Output voltage
- Output current
- Output resistance

$Y$  is related to  $U$  as  $Y = \max(\min(A \cdot U + B, Y_{\max}), Y_{\min})$ , where  $Y_{\min}$  and  $Y_{\max}$  are minimum and maximum limits on the output, respectively.

## Ports

### Input

#### **U – Physical input signal**

physical signal

Physical input signal.

### Conserving

#### **+ – Positive voltage**

electrical

Electrical conserving port associated with the PS sensor positive voltage.

#### **- – Negative voltage**

electrical

Electrical conserving port associated with the PS sensor negative voltage.

## Parameters

#### **Output type – Sensor output type**

Variable voltage (default) | Variable current | Variable resistor

Indicates whether the sensor output is a Variable voltage of  $Y$  V, a Variable current of  $Y$  A, or Variable resistor with a value of  $Y$   $\Omega$ .

**Sensor gain, A – Output sensitivity**

1 (default)

Sensitivity of the output  $Y$  with respect to the input  $U$ ,  $dY/dU$ .

**Sensor offset, B – Sensor offset**

0 (default)

Output when the input  $U$  is zero. The output does not exceed the limits  $Y_{\max}$  and  $Y_{\min}$ .

**Maximum output, Ymax – Maximum output**

5 (default)

Upper limit on the sensor output. The following table shows the units of this parameter, which depend on the selected value of the **Output type** parameter.

Output type	Units
Variable voltage	V
Variable current	A
Variable resistor	$\Omega$

**Minimum output, Ymin – Minimum output**

0.01 (default)

Lower limit on the sensor output. The following table shows the units of this parameter, which depend on the selected value of the **Output type** parameter.

Output type	Units
Variable voltage	V
Variable current	A
Variable resistor	$\Omega$

If you select Variable resistance for the **Output type** parameter, the minimum resistance  $Y_{\min}$  must be greater than zero.

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Controlled Voltage Source | Controlled Current Source | Variable Resistor

## PTC Thermistor

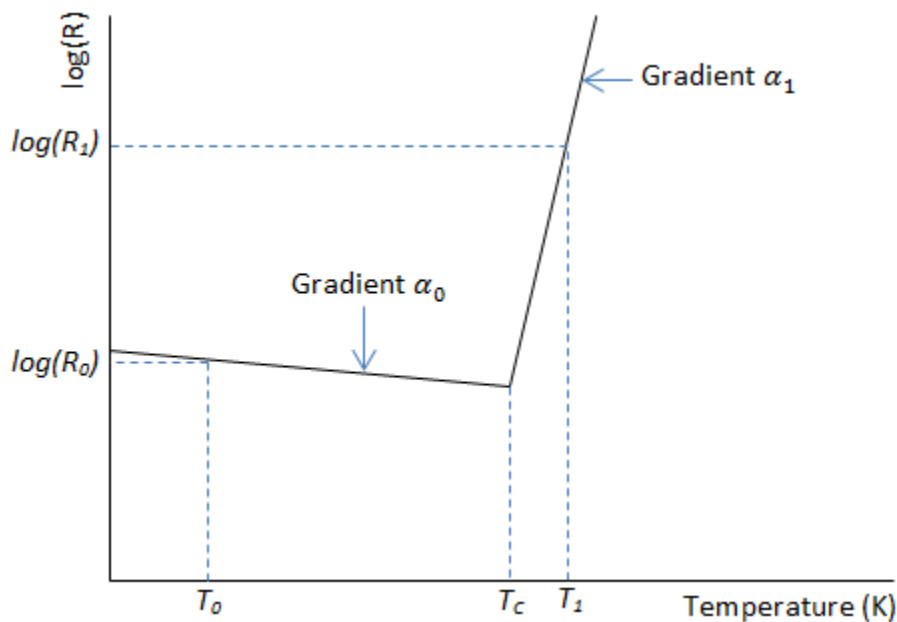
Switching type positive temperature coefficient (PTC) thermistor

**Library:** Simscape / Electrical / Sensors & Transducers



### Description

The PTC Thermistor block represents a switching type Positive Temperature Coefficient (PTC) thermistor. This type of thermistor has a decreasing resistance with temperature increasing up to the Curie temperature. Above the Curie temperature the resistance increases very rapidly with increasing temperature, as shown in the following plot. The region to the right of the Curie temperature is called the PTC regime. To represent a non-switching linear PTC thermistor, use the Thermal Resistor block.



For a switching type PTC thermistor, the resistance  $R$  at temperature  $T$  is given by

$$R = \begin{cases} R_0 e^{\alpha_0(T - T_0)} & \text{for } T < T_c \\ R_1 e^{\alpha_1(T - T_1)} & \text{for } T \geq T_c \end{cases}$$

$$T_c = \frac{\log(R_1) - \log(R_0) + \alpha_0 T_0 - \alpha_1 T_1}{\alpha_0 - \alpha_1}$$

where:

- $T_c$  is the Curie temperature.
- $R_0$  is the resistance at nominal temperature  $T_0$ .
- $R_1$  is the resistance at reference temperature  $T_1$ .
- $T_0$  is the nominal temperature at which the resistance is quoted, usually room temperature.  $T_0$  is less than the Curie temperature  $T_c$ .
- $T_1$  is the reference temperature, equal or greater than the Curie temperature  $T_c$ , which means that at this temperature the PTC regime is in force.
- $\alpha_0$  is the temperature coefficient at nominal temperature  $T_0$ .
- $\alpha_1$  is the temperature coefficient at reference temperature  $T_1$ .

The following equation describes the thermal behavior of the block:

$$Q = K_d t_c \frac{dT}{dt}$$

where:

- $Q$  is the net heat flow into port A.
- $K_d$  is the **Dissipation factor** parameter value.
- $t_c$  is the **Thermal time constant** parameter value.
- $dT/dt$  is the rate of change of the temperature.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### A — Thermal port

thermal

Thermal port.

#### + — Positive port

electrical

Electrical conserving port associated with the PTC thermistor positive port.

#### - — Negative port

electrical

Electrical conserving port associated with the PTC thermistor negative port.



## Parameters

### Electrical

#### Nominal resistance $R_0$ at $T_0$ — Nominal resistance at nominal temperature

1000 Ohm (default)

Nominal resistance of the thermistor at the nominal temperature. Many datasheets quote the nominal resistance at 25°C and list it as R25.

#### Temperature coefficient $\alpha_0$ at $T_0$ — Temperature coefficient at nominal temperature

-0.01 1/K (default) | negative scalar

Temperature coefficient at the nominal temperature. The value must be less than zero.

#### Nominal temperature $T_0$ — Nominal temperature

298.15 K (default)

Temperature at which the nominal resistance is measured.

#### Reference resistance $R_1$ at $T_1$ — Reference resistance at reference temperature

10000 Ohm (default)

Reference resistance of the thermistor at the reference temperature.

#### Temperature coefficient $\alpha_1$ at $T_1$ — Temperature coefficient at reference temperature

1 1/K (default) | positive scalar

Temperature coefficient at the reference temperature. The value must be greater than zero.

#### Reference temperature $T_1$ — Reference temperature

398.15 K (default)

Temperature at which the reference resistance is measured. This temperature must be in the PTC regime.

### Thermal

#### Thermal time constant — Thermal time constant

5.0 s (default)

Time it takes the sensor temperature to reach 63% of the final temperature change when a step change in ambient temperature occurs.

#### Dissipation factor — Dissipation factor

0.75e-3 W/K (default)

Thermal power required to raise the thermistor temperature by one K.

## Version History

Introduced in R2012b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Resistor | Thermal Resistor | Thermistor

# Pulse Current Source

Periodic square wave current source

**Library:** Simscape / Electrical / Sources  
Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Pulse Current Source block represents a current source whose output current value is a periodic square pulse as a function of time and is independent of the voltage across the terminals of the source. The following equations describe the current through the source as a function of time:

$$I_{out}(0) = I_1$$

$$I_{out}(TD) = I_1$$

$$I_{out}(TD + TR) = I_2$$

$$I_{out}(TD + TR + PW) = I_2$$

$$I_{out}(TD + TR + PW + TF) = I_1$$

$$I_{out}(TD + PER) = I_1$$

Where:

- $I_1$  is the output current at time zero.
- $I_2$  is the output current when the output is high.
- $TD$  is the time at which the pulse first starts.
- $TR$  is the time that it takes the output current to rise from  $I_1$  to  $I_2$ .
- $TF$  is the time it takes the output current to fall from  $I_2$  to  $I_1$ .
- $PW$  is the time width of the output pulse.
- $PER$  is the period of the output pulse.

The block determines the values at intermediate time points by linear interpolation.

The specified values for  $PW$  and  $PER$  have the following effect on the block output:

- If both  $PW$  and  $PER$  are infinite, the block produces a step response at time  $TD$ .
- If  $PER$  is infinite and  $PW$  is finite, the block produces a single pulse of width  $PW$  and infinite period.
- If  $PW$  is infinite and  $PER$  is finite, the block produces a step response with pulses of width  $TR$  to a value  $I_1$  every  $PER$  seconds.

- If  $PW > PER$ , the block produces a step response with pulses of width  $TR$  to a value  $I_1$  every  $PER$  seconds.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-121/0\Omega$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the pulse current source positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the pulse current source negative terminal.

## Parameters

### Initial value, I1 – Initial output current

0 A (default)

Value of the output current at time zero, in A.

### Pulse value, I2 – High output current

0 A (default)

Value of the output current when the output is high, in A.

### Pulse delay time, TD – Pulse delay time

0 s (default)

Time at which the pulse first starts, in s.

### Pulse rise time, TR – Pulse rise time

$1e-9$  s (default) | nonnegative scalar

Time it takes the output current to rise from the **Initial value, I1** value to the **Pulse value, I2** value. The value must be greater than or equal to 0.

### Pulse fall time, TF – Pulse fall time

$1e-9$  s (default) | nonnegative scalar

Time it takes the output current to fall from the **Pulse value, I2** value to the **Initial value, I1** value. The value must be greater than or equal to 0.

**Pulse width, PW — Output pulse time width**

Inf s (default) | positive scalar

Time width of the output pulse. The value must be greater than 0.

**Pulse period, PER — Output pulse period**

Inf s (default) | positive scalar

Period of the output pulse. For the default value, Inf s, the block produces a single pulse with an infinite period.

**Include parallel conductance, GMIN — Whether to include parallel conductance**

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

**Values of parallel conductance, GMIN**

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | Pulse Voltage Source | Piecewise Linear Current Source

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”  
“Parameterize an Exponential Diode from SPICE Netlist”

# Pulse Voltage Source

Periodic square wave voltage source

**Library:** Simscape / Electrical / Sources  
Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Pulse Voltage Source block represents a voltage source whose output voltage value is a periodic square pulse as a function of time and is independent of the current through the source. The following equations describe the output voltage as a function of time:

$$V_{out}(0) = V_1$$

$$V_{out}(TD) = V_1$$

$$V_{out}(TD + TR) = V_2$$

$$V_{out}(TD + TR + PW) = V_2$$

$$V_{out}(TD + TR + PW + TF) = V_1$$

$$V_{out}(TD + PER) = V_1$$

where:

- $V_1$  is the output voltage at time zero.
- $V_2$  is the output voltage when the output is high.
- $TD$  is the time at which the pulse first starts.
- $TR$  is the time that it takes the output voltage to rise from  $V_1$  to  $V_2$ .
- $TF$  is the time that it takes the output voltage to fall from  $V_2$  to  $V_1$ .
- $PW$  is the time width of the output pulse.
- $PER$  is the period of the output pulse.

The block determines the values at intermediate time points by linear interpolation.

The specified values for  $PW$  and  $PER$  have the following effect on the block output:

- If both  $PW$  and  $PER$  are infinite, the block produces a step response at time  $TD$ .
- If  $PER$  is infinite and  $PW$  is finite, the block produces a single pulse of width  $PW$  and infinite period.
- If  $PW$  is infinite and  $PER$  is finite, the block produces a step response with pulses of width  $TR$  to a value  $V_1$  every  $PER$  seconds.

- If  $PW > PER$ , the block produces a step response with pulses of width  $TR$  to a value  $V1$  every  $PER$  seconds.

## Ports

### Conserving

#### + — Positive voltage

electrical

Electrical conserving port associated with the pulse voltage source positive voltage

#### - — Negative voltage

electrical

Electrical conserving port associated with the pulse voltage source negative voltage

## Parameters

### Initial value, V1 — Initial output voltage

0 V (default)

Value of the output voltage at time zero, in V.

### Pulse value, V2 — High output voltage

0 V (default)

Value of the output voltage when the output is high, in V.

### Pulse delay time, TD — Pulse delay time

0 s (default)

Time at which the pulse first starts, in s.

### Pulse rise time, TR — Pulse rise time

1e-9 s (default) | nonnegative scalar

Time it takes the output voltage to rise from the **Initial value, V1** value to the **Pulse value, V2** value. The value must be greater than or equal to 0.

### Pulse fall time, TF — Pulse fall time

1e-9 s (default) | nonnegative scalar

Time it takes the output voltage to fall from the **Pulse value, V2** value to the **Initial value, V1** value. The value must be greater than or equal to 0.

### Pulse width, PW — Output pulse time width

Inf s (default) | positive scalar

Time width of the output pulse. The value must be greater than 0.

### Pulse period, PER — Output pulse period

Inf s (default) | positive scalar



Period of the output pulse. For the default value, `Inf`, the block produces a single pulse with an infinite period.

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

[Environment Parameters](#) | [Pulse Current Source](#) | [Piecewise Linear Voltage Source](#)

### Functions

`subcircuit2ssc`

### Topics

[“Additional Parameterization Workflows”](#)

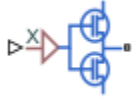
[“Converting a SPICE Netlist to Simscape Blocks”](#)

[“Parameterize an Exponential Diode from SPICE Netlist”](#)

# Push-Pull Output

Behavioral representation of CMOS complementary output stage

**Library:** Simscape / Electrical / Integrated Circuits

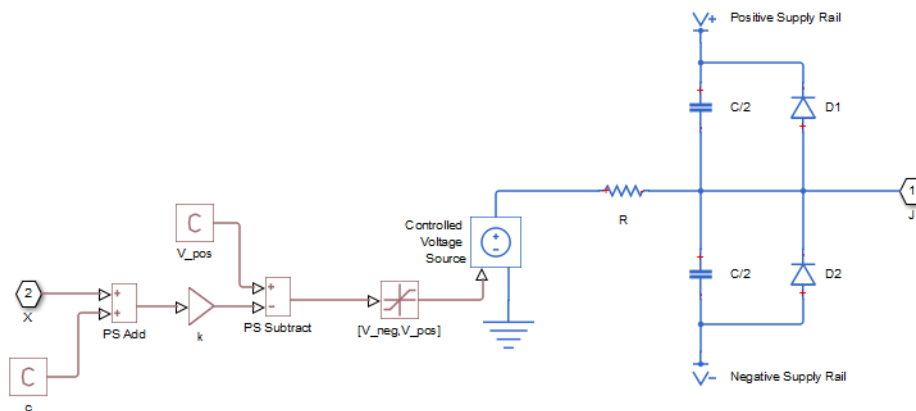


## Description

The Push-Pull Output block represents a CMOS complementary output stage behaviorally. 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.

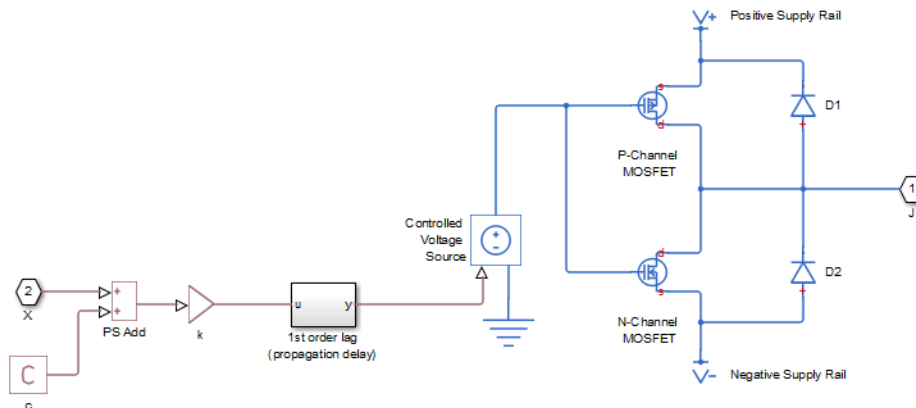
You can choose between are two output current-voltage relationships:

- **Linear** — The block represents the output as a voltage source plus series resistance and parallel capacitance, as shown in the following figure. The value you specify for the **Output resistance** parameter is assigned to the series resistance, and the capacitance values are determined by matching the RC time constant to the **Propagation delay** parameter value.



The input to the Controlled Voltage Source block is limited to be between the supply rails, and it is also inverted by subtraction from the supply voltage. The inversion makes it behave like a complementary output stage, with a high gate-source voltage resulting in a low output.

- **Quadratic** — The output stage is modeled by the two MOSFETs that constitute the complementary pair. The MOSFET parameters are derived from the output resistance values and short-circuit currents that you specify as mask parameters. The gate input demand is lagged to approximate the **Propagation delay** parameter value.



Both **Linear** and **Quadratic** output models add an offset and scale the physical input  $X$  so that the gate voltage is given by:

$$V_g = k \cdot (X + c) \quad (1-52)$$

where

- $k$  is the input signal scaling.
- $c$  is the input signal offset.

The offset and scaling can be used, for example, to match logical values for  $X$  (that is, range  $[0, 1]$ ) to  $[V_-, V_+]$  at the output pin. For example, if  $V_+ = 10\text{V}$  and  $V_- = 0$ , then to match the signal logical values to this voltage range, set  $c = -1$  and  $k = -10$ .

For both **Linear** and **Quadratic** output models, the protection diodes  $D1$  and  $D2$  act to limit the output voltage range. These diodes are Diode blocks from the Simscape Foundation library, that is, piecewise linear diodes defined by their forward voltage and on resistance. If the voltage across  $D1$  rises above the forward voltage, then the diode starts to conduct, and provided that the on resistance is low, it effectively prevents the output rising above  $V_+$  plus the diode forward voltage drop. An equivalent behavior results if the output voltage drops too low.

The output model is very similar to that used for the logic blocks. For a plot of a typical output  $V$ - $I$  characteristic when using the **Quadratic** output model, see [Selecting the Output Model for Logic Blocks](#).

---

**Note** This block is constructed out of blocks from the Simscape Physical Signals library (such as PS Add, PS Gain, and so on). Currently, the blocks in the Physical Signals library do not support unit propagation and checking. For more information, see “How to Work with Physical Units”.

---

## Assumptions and Limitations

- The block does not accurately model dynamic response.
- The **Quadratic** output model does not model any output capacitance effects. Add output capacitance externally to the block if required.

## Ports

### Input

#### **X – Input port**

physical signal

Physical port associated with the Push-Pull Output input.

### Conserving

#### **J – Output port**

electrical

Electrical conserving port associated with the block output. The port name is hidden on the block icon, but you can see it in simulation data logs.

## Parameters

### Input Scaling

#### **Input signal scaling, k – Input signal scaling**

1 V (default)

The input physical signal  $X$  is mapped to the gate voltage by  $V_g = k \cdot (X + c)$ , where  $k$  is the input signal scaling. Use this parameter in conjunction with the **Input signal offset, c** to map the range of  $X$  to the voltage range defined by the power supply.

#### **Input signal offset, c – Input signal offset**

0 (default)

The input physical signal  $X$  is mapped to the gate voltage by  $V_g = k \cdot (X + c)$ , where  $c$  is the input signal offset. Use this parameter in conjunction with the **Input signal scaling, k** to map the range of  $X$  to the voltage range defined by the power supply.

### Output Characteristics

#### **Output current-voltage relationship – Output model**

Linear (default) | Quadratic

Select the output model:

- **Linear** - the output voltage drops linearly with output current. This is the default option.
- **Quadratic** - the output voltage dependency on output current is defined by the quadratic I-V characteristics of the two output MOSFET devices.

#### **Output resistance – Output resistance**

25 Ohm (default)

Defines one over the slope of the output I-V characteristic.

### Dependencies

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Linear.

**Power rail voltages, [V- V+], used for measurements — Power rail voltages used for measurements**

[0, 5] V (default)

Defines the rail voltages for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic.

**Output resistance values at zero output current and at I\_OH when Vg=V- — Output resistance values at zero output current and at I\_OH when Vg=V-**

[25, 250] Ohm (default)

A row vector [ R\_OH1 R\_OH2 ] of two resistance values. The first value R\_OH1 is the gradient of the output voltage-current relationship when the complementary pair output is HIGH (Vg=V-) and there is no output current. The second value R\_OH2 is the gradient of the output voltage-current relationship when the output is HIGH and the output current is I\_OH.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic.

**Output current I\_OH when output is shorted to V- and Vg=V- — Output current I\_OH when output is shorted to V- and Vg=V-**

63 mA (default)

The resulting current when the output is HIGH (Vg=V-), but the load forces the output voltage to the negative supply rail.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic.

**Output resistance values at zero output current and at I\_OL when Vg=V+ — Output resistance values at zero output current and at I\_OL when Vg=V+**

[30, 800] Ohm (default)

A row vector [ R\_OL1 R\_OL2 ] of two resistance values. The first value R\_OL1 is the gradient of the output voltage-current relationship when the complementary pair output is LOW (Vg=V+) and there is no output current. The second value R\_OL2 is the gradient of the output voltage-current relationship when the output is LOW and the output current is I\_OL.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic.

**Output current I\_OL when output is shorted to V+ and Vg=V+ — Output current I\_OL when output is shorted to V+ and Vg=V+**

-45 mA (default)

The resulting current when the output is LOW (Vg=V+), but the load forces the output voltage to the positive supply voltage.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to reach 63.2% of its final value following a step change in the input, X. For Quadratic output, it is implemented by the lagged gate input demand.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Supply Voltage****Negative power rail voltage, V- — Negative power rail voltage**

0 V (default)

Negative power supply voltage applied to the N-channel MOSFET source pin.

**Positive power rail voltage, V+ — Positive power rail voltage**

5 V (default)

Positive power supply voltage applied to the P-channel MOSFET source pin.

**Initial Conditions****Initial output voltage — Initial output voltage**

0 V (default)

This parameter is used to set the voltage on the output capacitors so that the output voltage is initialized to the parameter's value.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Linear on the **Output Characteristics** tab.

**Initial input signal — Initial input signal**

0 V (default)

This parameter is used to initialize the propagation delay first-order lag such that there is no transient at time zero.

**Dependencies**

This parameter is visible only when the **Output current-voltage relationship** parameter is set to Quadratic on the **Output Characteristics** tab.

## **Version History**

**Introduced in R2011b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

## PVCCS

Polynomial voltage-controlled current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



### Description

The PVCCS (Polynomial Voltage-Controlled Current Source) block represents a current source whose output current value is a polynomial function of the voltage across the input ports. The following equations describe the current through the source as a function of time:

- If you specify an  $n$ -element vector of polynomial coefficients for the **Polynomial coefficients** parameter:

$$I_{out} = p(0) + p(1) * V_{in} + \dots + p(n - 1) * V_{in}^{n-1} + p(n) * V_{in}^n$$

- If you specify a scalar coefficient for the **Polynomial coefficients** parameter:

$$I_{out} = p * V_{in}$$

where:

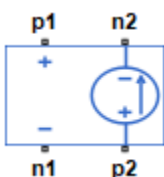
- $V_{in}$  is the voltage across the input ports.
- $p$  is the **Polynomial coefficients** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the output ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters to your model and set the **GMIN** parameter to the desired value.

### Ports

Refer to the figure for port locations.





## Conserving

### p1 – Positive voltage of the controlling input

electrical

Electrical conserving port associated with the PVCCS positive voltage of the controlling input.

### n1 – Negative voltage of the controlling input

electrical

Electrical conserving port associated with the PVCCS negative voltage of the controlling input.

### p2 – Positive output voltage

electrical

Electrical conserving port associated with the PVCCS positive output voltage.

### n2 – Negative output voltage

electrical

Electrical conserving port associated with the PVCCS negative output voltage.

## Parameters

### Polynomial coefficients – Polynomial coefficients

[0, 1] (default)

The polynomial coefficients that relate the input voltage to the output current, as described in the preceding section.

### Include parallel conductance, GMIN – Whether to include parallel conductance

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

#### Values of parallel conductance, GMIN

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

## **Version History**

Introduced in R2008a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Environment Parameters | PCCCS | PCCCS2 | PCCVS | PCCVS2 | PVCCS2 | PVCVS | PVCVS2

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## PVCCS2

Polynomial voltage-controlled current source with two controlling inputs

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



### Description

The PVCCS2 (Two-Input Polynomial Voltage-Controlled Current Source) block represents a current source whose output current value is a polynomial function of the voltages across the pairs of controlling input ports. This equation describes the current through the source as a function of time:

$$I_{out} = p_1 + p_2 * V_{in1} + p_3 * V_{in2} + p_4 * V_{in1}^2 + p_5 V_{in1} * V_{in2} + p_6 * V_{in2}^2 + \dots$$

where:

- $V_{in1}$  is the voltage across the first pair of input ports.
- $V_{in2}$  is the voltage across the second pair of input ports.
- $p$  is the **Polynomial coefficients** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the output ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

### Ports

#### Conserving

##### 1+ – Positive voltage of the first controlling input

electrical

Electrical conserving port associated with the PVCCS2 positive voltage of the first controlling input.

##### 1- – Negative voltage of the first controlling input

electrical

Electrical conserving port associated with the PVCCS2 negative voltage of the first controlling input.

##### 2+ – Positive voltage of the second controlling input

electrical

Electrical conserving port associated with the PVCCS2 positive voltage of the second controlling input.

**2- – Negative voltage of the second controlling input**

electrical

Electrical conserving port associated with the PVCCS2 negative voltage of the second controlling input.

**+ – Positive output voltage**

electrical

Electrical conserving port associated with the PVCCS2 positive output voltage.

**- – Negative output voltage**

electrical

Electrical conserving port associated with the PVCCS2 negative output voltage.

**Parameters**

**Polynomial coefficients – Polynomial coefficients**

[ 0 1 1 ] (default)

The polynomial coefficients that relate the input voltage to the output current, as described in the preceding section.

**Include parallel conductance, GMIN – Whether to include parallel conductance**

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

**Values of parallel conductance, GMIN**

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

**Version History**

Introduced in R2009a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | PCCCS | PCCCS2 | PCCVS | PCCVS2 | PVCCS | PVCVS | PVCVS2

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## PVCVS

Polynomial voltage-controlled voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



### Description

The PVCVS (Polynomial Voltage-Controlled Voltage Source) block represents a voltage source whose output voltage value is a polynomial function of the voltage across the input ports. The following equations describe the voltage across the source as a function of time:

- If you specify an  $n$ -element vector of polynomial coefficients for the **Polynomial coefficients** parameter:

$$V_{out} = p(0) + p(1) * V_{in} + \dots + p(n - 1) * V_{in}^{n-1} + p(n) * V_{in}^n$$

- If you specify a scalar coefficient for the **Polynomial coefficients** parameter:

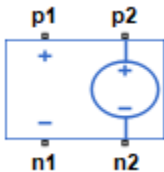
$$V_{out} = p * V_{in}$$

where:

- $V_{in}$  is the voltage across the input ports.
- $p$  is the **Polynomial coefficients** parameter value.

### Ports

Refer to the figure for port locations.



### Conserving

**p1 — Positive voltage of the controlling input**

electrical

Electrical conserving port associated with the PVCVS positive voltage of the controlling input.

**n1 — Negative voltage of the controlling input**

electrical

Electrical conserving port associated with the PVCVS negative voltage of the controlling input.

**p2 — Positive output voltage**

electrical

Electrical conserving port associated with the PVCVS positive output voltage.

**n2 — Negative output voltage**

electrical

Electrical conserving port associated with the PVCVS negative output voltage.

## Parameters

**Polynomial coefficients — Polynomial coefficients**

[0, 1] (default)

The polynomial coefficients that relate the input voltage to the output voltage, as described in the preceding section.

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | PCCCS | PCCCS2 | PCCVS | PCCVS2 | PVCCS | PVCCS2 | PVCVS2

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## PVCVS2

Polynomial voltage-controlled voltage source with two controlling inputs

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



### Description

The PVCVS2 (Two-Input Polynomial Voltage-Controlled Voltage Source) block represents a voltage source whose output voltage value is a polynomial function of the voltages across the pairs of controlling input ports. This equation describes the voltage across the source as a function of time:

$$V_{out} = p_1 + p_2 * V_{in1} + p_3 * V_{in2} + p_4 * V_{in1}^2 + p_5 V_{in1} * V_{in2} + p_6 * V_{in2}^2 + \dots$$

where:

- $V_{in1}$  is the voltage across the first pair of input ports.
- $V_{in2}$  is the voltage across the second pair of input ports.
- $p$  is the **Polynomial coefficients** parameter value.

### Ports

#### Conserving

##### 1+ — Positive input voltage of the first controlling input

electrical

Electrical conserving port associated with the PVCVS2 positive input voltage of the first controlling input.

##### 1- — Negative input voltage of the first controlling input

electrical

Electrical conserving port associated with the PVCVS2 negative input voltage of the first controlling input.

##### 2+ — Positive input voltage of the second controlling input

electrical

Electrical conserving port associated with the PVCVS2 positive input voltage of the second controlling input.

##### 2- — Negative input voltage of the second controlling input

electrical



Electrical conserving port associated with the PVCVS2 negative input voltage of the second controlling input.

#### **+ – Positive output voltage**

electrical

Electrical conserving port associated with the PVCVS2 positive output voltage.

#### **- – Negative output voltage**

electrical

Electrical conserving port associated with the PVCVS2 negative output voltage.

## **Parameters**

### **Polynomial coefficients – Polynomial coefficients**

[ 0 1 1 ] (default)

The polynomial coefficients that relate the input voltage to the output voltage, as described in the preceding section.

## **Version History**

Introduced in R2009a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Environment Parameters | PCCCS | PCCCS2 | PCCVS | PCCVS2 | PVCCS | PVCCS2 | PVCVS

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# PWM Generator

Generate pulse width modulated signal or waveform

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



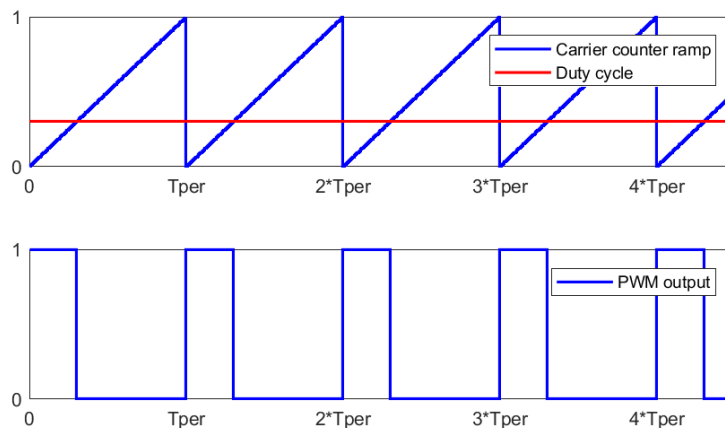
## Description

The PWM Generator block implements a PWM generator. The pulse width modulation technique controls power transfer from one electrical component to another by quickly switching between full power transfer and no power transfer.

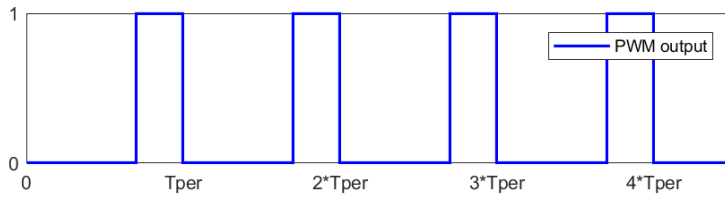
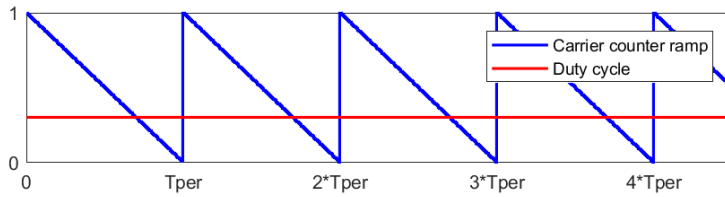
## Working Principle

The PWM generator block outputs either 1 when the duty cycle is greater than the carrier counter value, or 0 otherwise. You can set the period of each cycle by specifying the timer period  $T_{per}$ . You can change the initial output, or phase, of the PWM output by specifying one of three types of carrier counters:

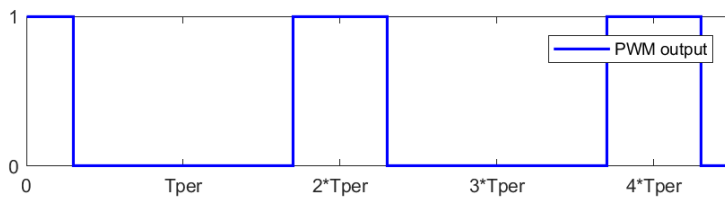
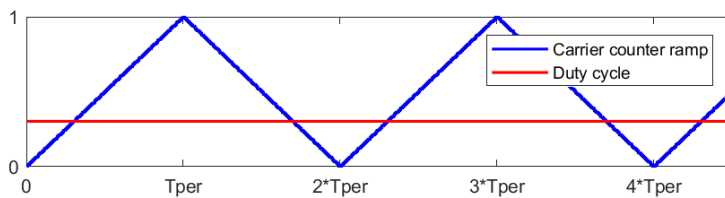
- Up counter — The PWM output signal initializes at the start of the on cycle. This graphic shows the carrier counter signal and the corresponding PWM output.



- Down counter — The PWM output signal initializes at the start of the off cycle. This graphic shows the carrier counter signal and the corresponding PWM output.



- Up-down counter — The PWM output signal initializes halfway through the on cycle. This graphic shows the carrier counter signal and the corresponding PWM output.



## Ports

### Input

#### DC — Duty cycle

scalar

Duty cycle in the range  $[0, 1]$ .

Data Types: `single` | `double`

### Output

#### PWM — PWM signal

scalar

Pulse width modulation signal.

Data Types: single | double

## Parameters

### Carrier counter — Carrier counter strategy

Up (default) | Down | Up-Down

Use the carrier counter strategy to change the initial behavior of the PWM output:

- Up counter — PWM output begins at the start of the on state.
- Down counter — PWM output begins at the start of the off state.
- Up-down counter — PWM output begins in the middle of the on state.

### Timer period (s) — PWM period

0.001 (default) | positive number

PWM timer period.

If you set **Carrier counter** to Up counter or Down counter, the **Timer period (s)** parameter is specified as  $\frac{1}{f_{sw}}$ , where  $f_{sw}$  is the switching frequency. If you set **Carrier counter** to Up-down

counter, the **Timer period (s)** parameter is specified as  $\frac{1}{2f_{sw}}$ .

### Phase delay (s) — Phase delay

0 (default) | nonnegative number

Phase delay, in seconds. Add a phase delay to change the initial state of the counter.

### Sample time — Block sample time

5e-5 (default) | positive number

Sample time for the block. For continuous-time simulation, set to zero. For discrete-time simulation, to ensure adequate resolution in the generated signal, specify a positive value that is less than or equal to  $T_{per}/10$ , where  $T_{per}$  is the **Timer period (s)**.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

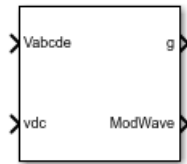
### Blocks

PWM Generator | PWM Generator (Three-phase, Two-level) | PWM Generator (Three-phase, Three-level) | Thyristor 6-Pulse Generator

## PWM Generator (Five-phase, Two-level)

Generate five-phase, two-level pulse width modulated waveform

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



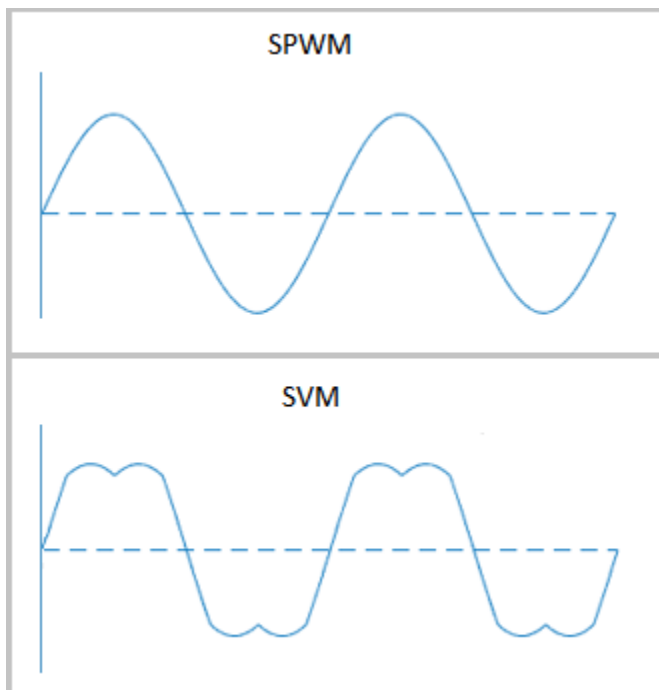
### Description

The PWM Generator (Five-phase, Two-level) block controls switching behavior for a five-phase, two-level power converter. The block:

- 1 Calculates on-gating and off-gating times based on the block inputs:
  - Five sinusoidal reference voltages, one per phase
  - A DC-link voltage
- 2 Uses the gating times to generate ten switch-controlling pulses.
- 3 Uses the gating times to generate modulation waveforms.

### Continuous PWM

The block only provides continuous pulse width modulation (PWM). The figure shows the general difference between continuous sinusoidal pulse width modulation (SPWM) and continuous space vector modulation (SVM) waveforms.



## Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

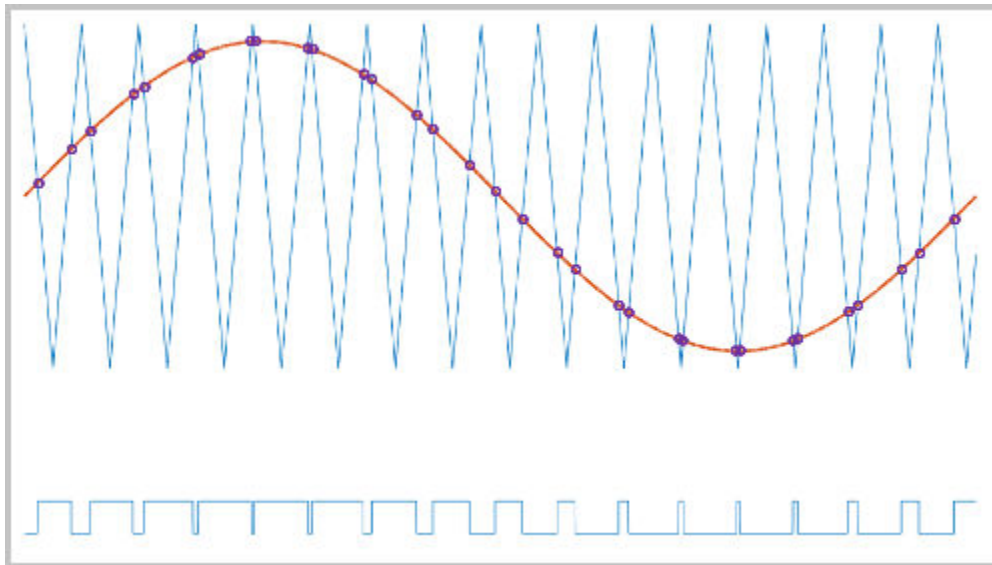
The PWM Generator (Five-phase, Two-level) block does not perform carrier-based PWM. Instead, the block uses input signals to calculate gating times and then uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that it outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a two-level, carrier-based PWM method:

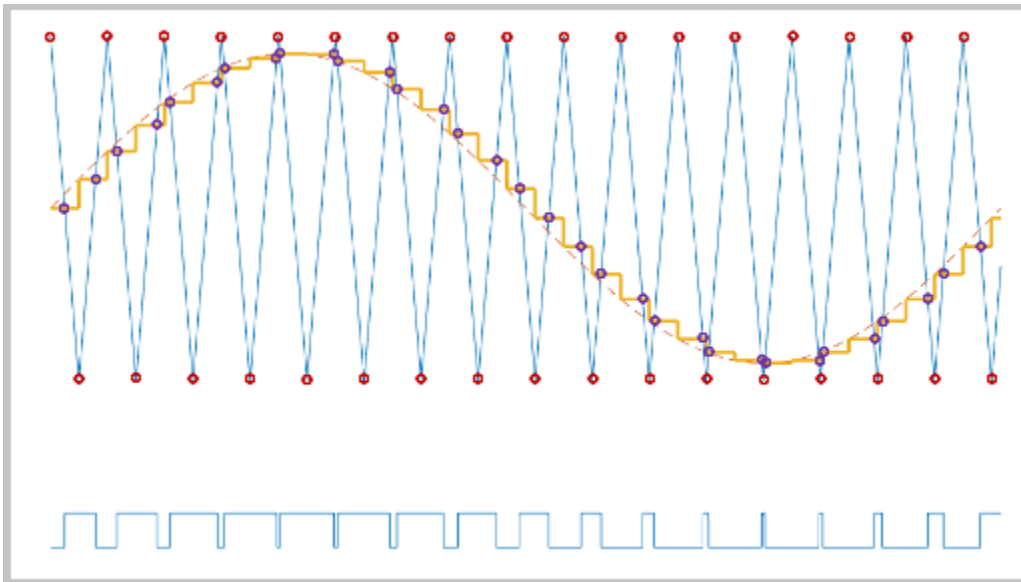
- 1 Samples a reference wave.
- 2 Compares the sample to a triangle carrier wave.
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave.

To determine switch-on and switch-off pulse behavior, a two-level carrier-based PWM generator uses these methods to sample the triangle wave:

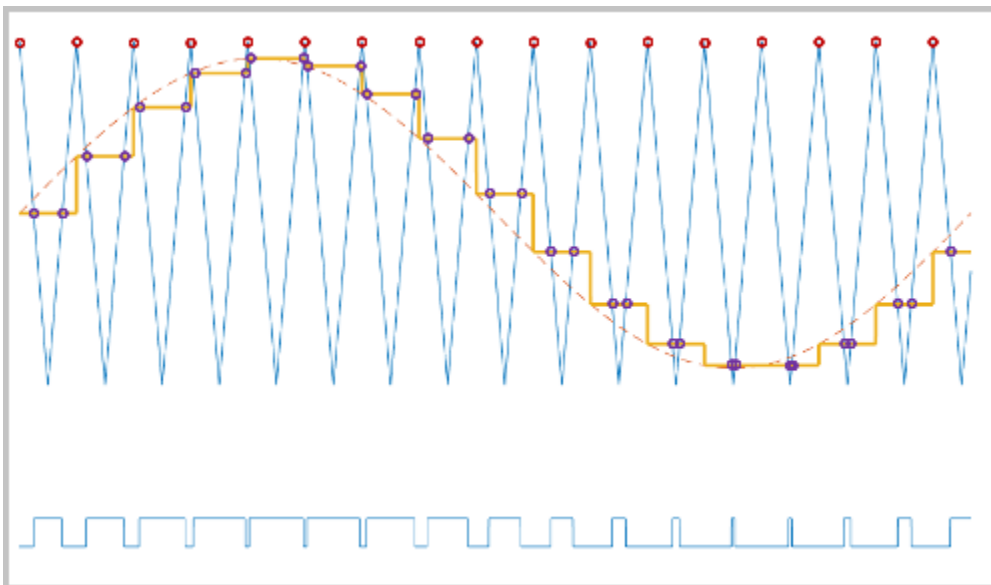
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs at only the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **vabcde — Five-phase sinusoidal reference signal**

vector

Sinusoidal voltages that you want the attached converter to output, specified as a vector of five elements with one element for each phase.

#### **vdc — DC-link voltage signal**

scalar

Positive real number for the DC-link voltage of the converter, specified as a scalar.

## Output

### **g – Gate control**

vector

Ten pulse waveforms that determine the switching behavior in the attached power converter, returned as a vector.

### **ModWave – Modulation wave**

vector

Modulation waves, returned as a vector of five elements, with one element for each phase.

## Parameters

### **Continuous PWM – Continuous pulse width modulation method**

SPWM: sinusoidal PWM (default) | SVM: space vector modulation

Continuous pulse width modulation method.

### **Sampling mode – Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### **Switching frequency (Hz) – Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch, in Hz.

### **Sample time (s) – Block sample time**

5e-5 (default) | positive scalar

Time interval between successive block executions, in seconds. To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the value of the **Switching frequency (Hz)** parameter.

## Version History

Introduced in R2021a

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.



## See Also

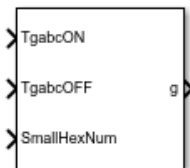
### Blocks

PWM Gate Signal Generator (Five-phase, Two-level) | PWM Timing and Waveform Generator (Five-phase, Two-level)

## PWM Gate Signal Generator (Three-phase, Three-level)

Generate twelve switch-controlling pulses for three-phase, three-level gating switching devices

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



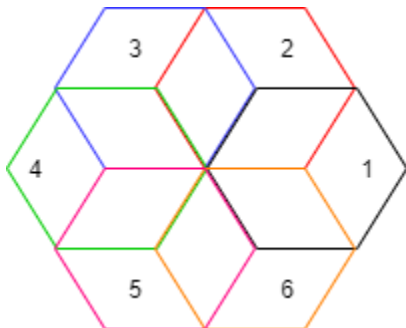
### Description

The PWM Gate Signal Generator (Three-phase, Three-level) block controls switching behavior for a three-phase, three-level power converter. The block uses the gating times provided as inputs to generate twelve switch-controlling pulses. You can obtain the gating time information using the PWM Timing and Waveform Generator (Three-phase, Three-level) block.

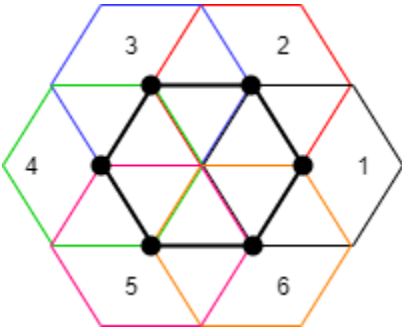
### Simplified Space-Vector PWM Method

To determine the switching sequence and the duration time of the voltage vector, the PWM Gate Signal Generator (Three-phase, Three-level) block uses a simplified space-vector pulse width modulation (SVPWM) method on the space-vector diagram of a three-level inverter [2].

This figure shows the six small hexagons that comprise the space-vector diagram of a three-level inverter:



Each of these six small hexagons represents the space-vector diagrams of a two-level inverter, with the diagrams centered on the six vertices of an inner small hexagon, as shown in this figure:



To simplify into the space-vector diagram of a two-level inverter, you must select one of the small hexagons by specifying a value at the **SmallHexNum** port. The block then transforms the three-level space-vector plane into a two-level space-vector plane by subtracting the original reference voltage vector by the amount of the center voltage vector of the hexagon you choose.

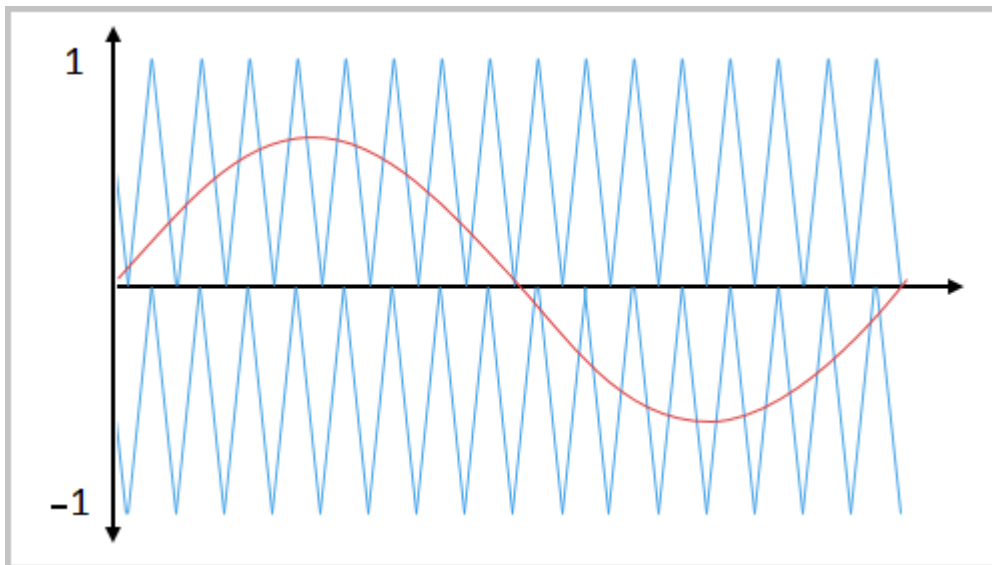
### Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Gate Signal Generator (Three-phase, Three-level) block does not perform carrier-based pulse width modulation. Instead, the block uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that the block outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a three-level, carrier-based PWM method:

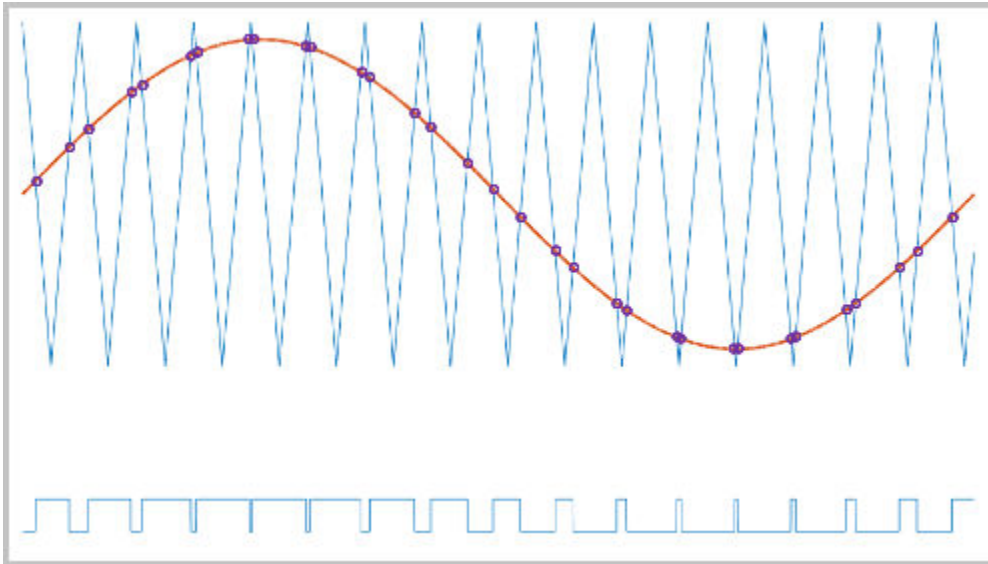
- 1 Samples a reference wave
- 2 Compares the sample to two parallel triangle carrier waves, separated by one level



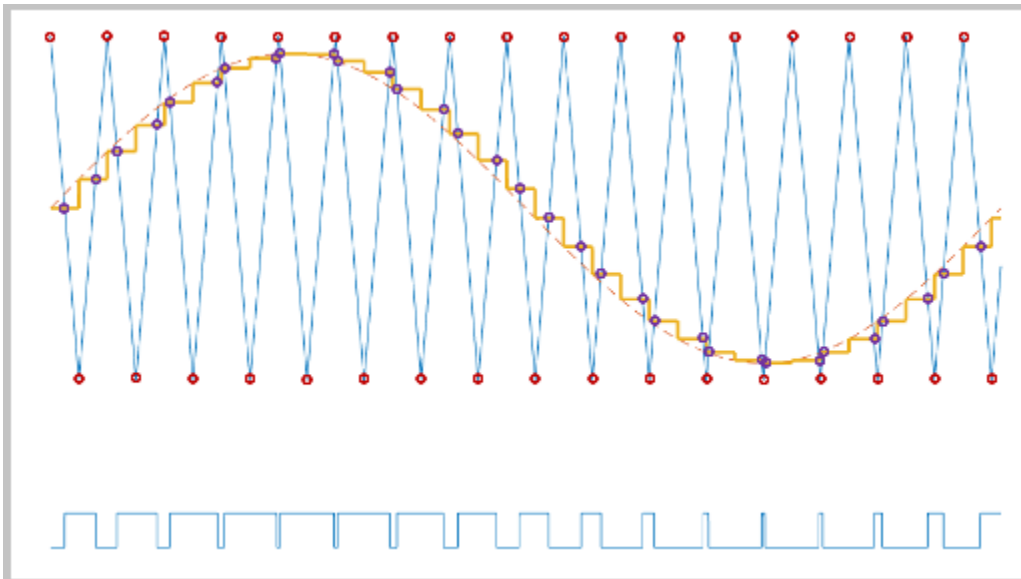
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

To determine switch-on and switch-off pulse behavior, a three-level carrier-based PWM generator uses these methods to sample each of the triangle waves:

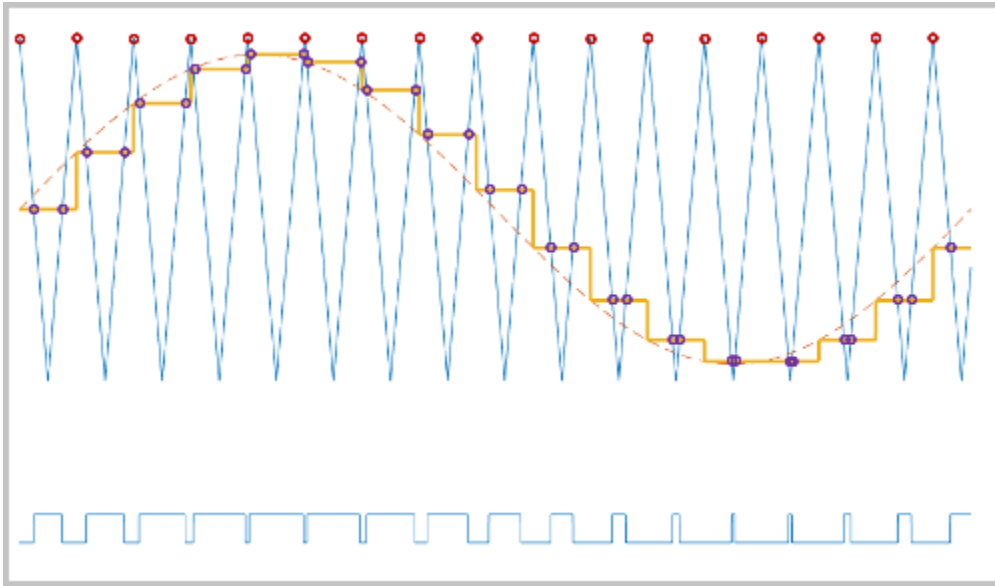
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs only at the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **TgabcON** — Switch-on gating time

vector

Switch-on gating time, specified as a vector of three elements, with one element for each phase.

#### **TgabcOFF** — Switch-off gating time

vector

Switch-off gating time, specified as a vector of three elements, with one element for each phase.

#### **SmallHexNum** — Small hexagon number

scalar

Number of one of the small hexagons that composes the space-vector diagrams of conventional two-level inverters, specified as a scalar between 1 and 6. For more information, see the “Simplified Space-Vector PWM Method” on page 1-1580 section.

### Output

#### **g** — Gate control

vector

Twelve pulse waveforms that determine switching behavior in the attached power converter, returned as a vector.

## Parameters

#### **Sampling mode** — Wave-sampling method

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

**Switching frequency (Hz) – Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch.

**Sample time (s) – Block sample time**

5e-5 (default) | positive scalar

Time interval between successive block executions. To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 * F_{sw})$ , where  $F_{sw}$  is the **Switching frequency (Hz)**.

## Version History

Introduced in R2020b

## References

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.
- [2] Seo, J. H., C. H. Choi, and D. S. Hyun. "A new simplified space-vector PWM method for three-level inverters." *IEEE Transactions on Power Electronics*, Vol. 16, No. 4, 2001, pp. 545-550.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

PWM Gate Signal Generator (Three-phase, Two-level) | PWM Timing and Waveform Generator (Three-phase, Three-level) | PWM Generator (Three-phase, Three-level) | Three-Level Converter (Three-Phase)

# PWM Gate Signal Generator (Three-phase, Two-level)

Generate six switch-controlling pulses for three-phase, two-level gating switching devices

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

The PWM Gate Signal Generator (Three-phase, Two-level) block controls switching behavior for a three-phase, two-level power converter. The block uses the gating times provided as inputs to generate six switch-controlling pulses. You can obtain the gating time information using the PWM Timing and Waveform Generator (Three-phase, Two-level) block.

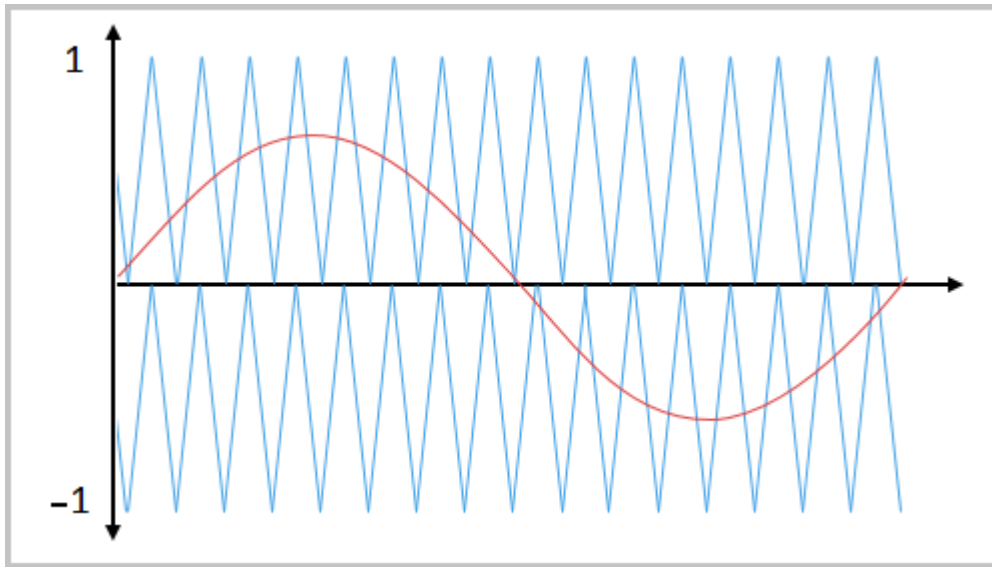
## Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Gate Signal Generator (Three-phase, Two-level) block does not perform carrier-based pulse width modulation (PWM). Instead, the block uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that the block outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a three-level, carrier-based PWM method:

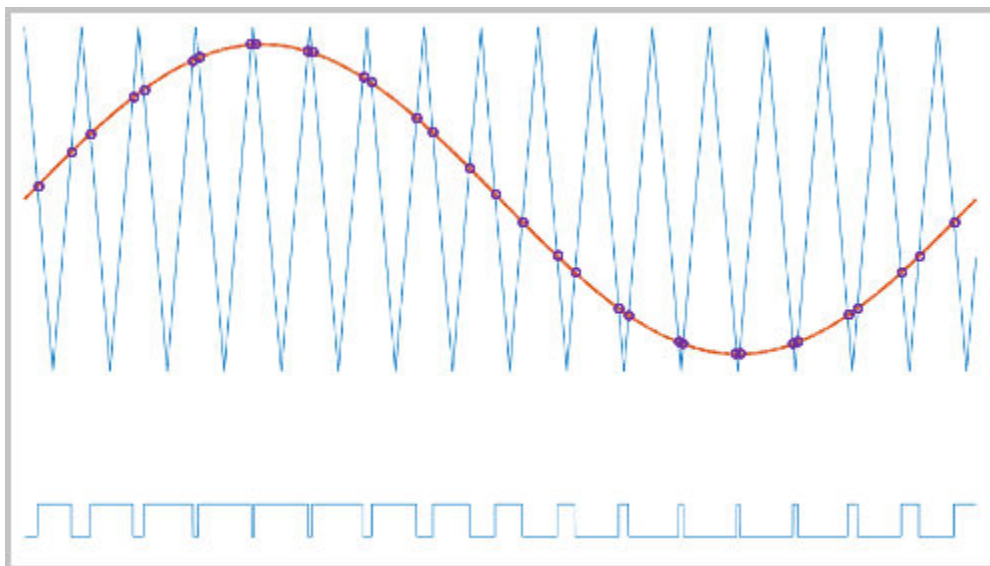
- 1 Samples a reference wave
- 2 Compares the sample to two parallel triangle carrier waves, separated by one level



- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

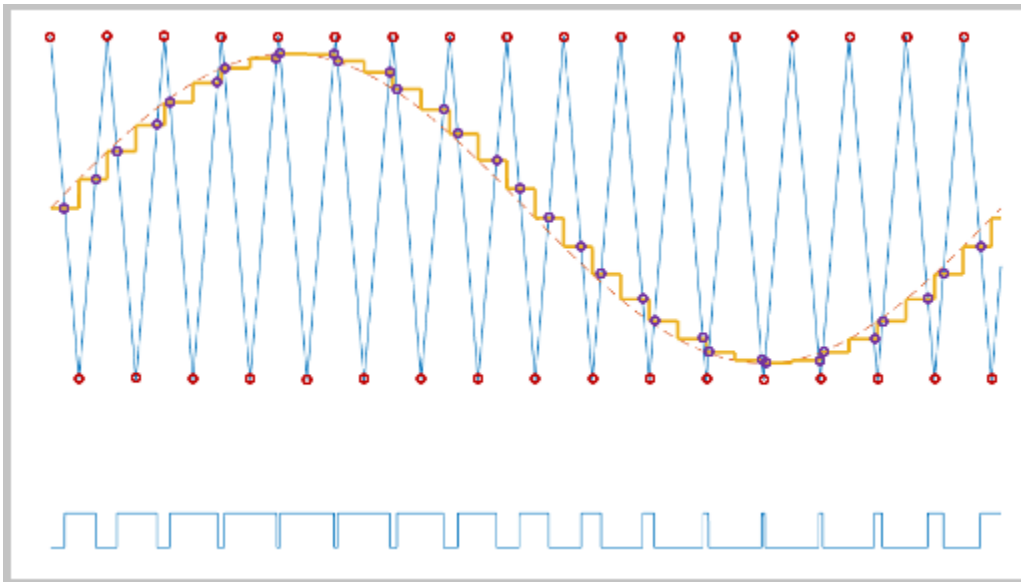
To determine switch-on and switch-off pulse behavior, a three-level carrier-based PWM generator uses these methods to sample each of the triangle waves:

- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.

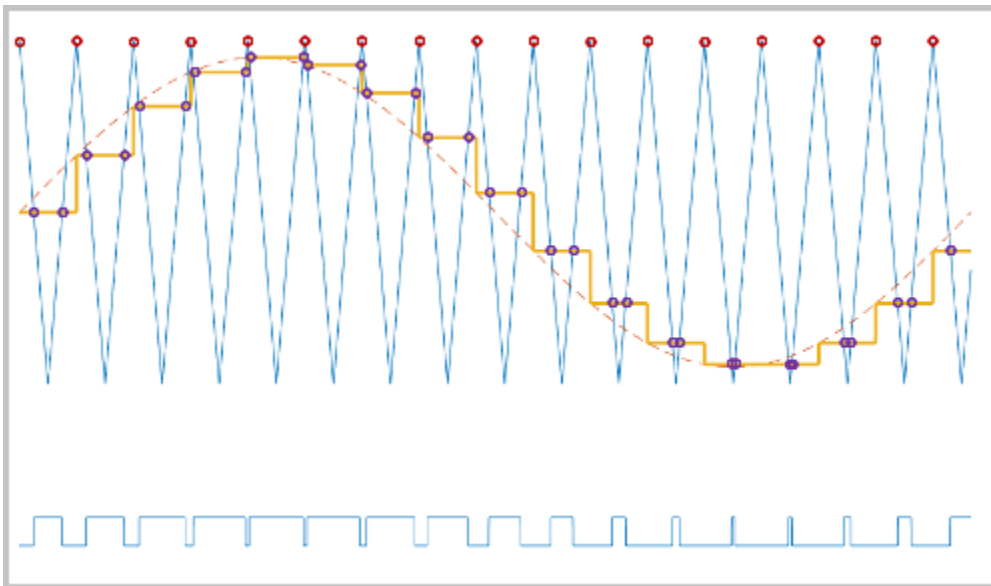


- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.





- Symmetric — Sampling occurs only at the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **Tgabc0N — Switch-on gating time**

vector

Switch-on gating time, specified as a vector of three elements, with one element for each phase.

#### **Tgabc0FF — Switch-off gating time**

vector

Switch-off gating time, specified as a vector of three elements, with one element for each phase.

## Output

### **g – Gate control**

vector

Six pulse waveforms that determine switching behavior in the attached power converter, returned as a vector.

## Parameters

### **Sampling mode – Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### **Switching frequency (Hz) – Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch.

### **Sample time (s) – Block sample time**

5e-5 (default) | positive scalar

Time interval between successive block executions. To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the **Switching frequency (Hz)**.

## Version History

Introduced in R2020b

## References

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.
- [2] Seo, J. H., C. H. Choi, and D. S. Hyun. "A new simplified space-vector PWM method for three-level inverters." *IEEE Transactions on Power Electronics*, Vol. 16, No. 4, 2001, pp. 545-550.

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

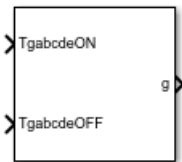
**See Also**

PWM Gate Signal Generator (Three-phase, Three-level) | PWM Generator (Three-phase, Two-level) |  
PWM Timing and Waveform Generator (Three-phase, Two-level) | Converter (Three-Phase)

# PWM Gate Signal Generator (Five-phase, Two-level)

Generate ten switch-controlling pulses for five-phase, two-level gating switching devices

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

The PWM Gate Signal Generator (Five-phase, Two-level) block controls the switching behavior for a five-phase, two-level power converter. The block uses the gating times provided as inputs to generate ten switch-controlling pulses. You can obtain the gating time information using the PWM Timing and Waveform Generator (Five-phase, Two-level) block.

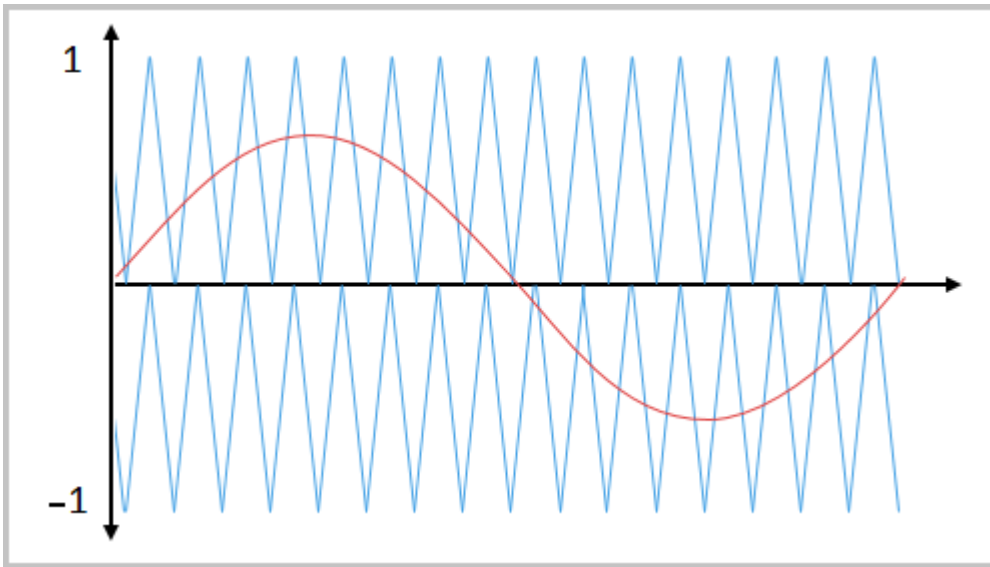
## Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Gate Signal Generator (Five-phase, Two-level) block does not perform carrier-based pulse width modulation (PWM). Instead, the block uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that the block outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a three-level, carrier-based PWM method:

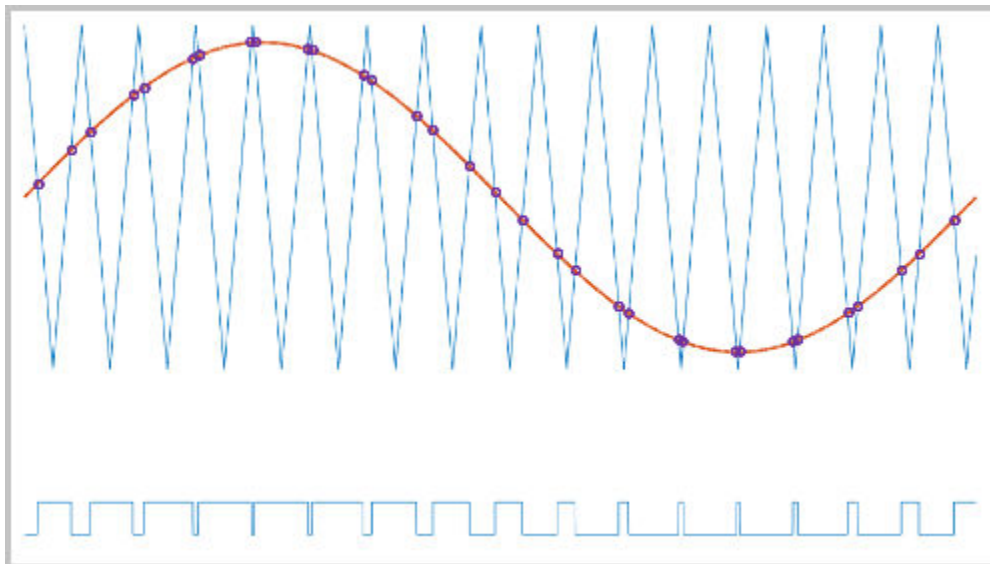
- 1 Samples a reference wave
- 2 Compares the sample to two parallel triangle carrier waves, separated by one level



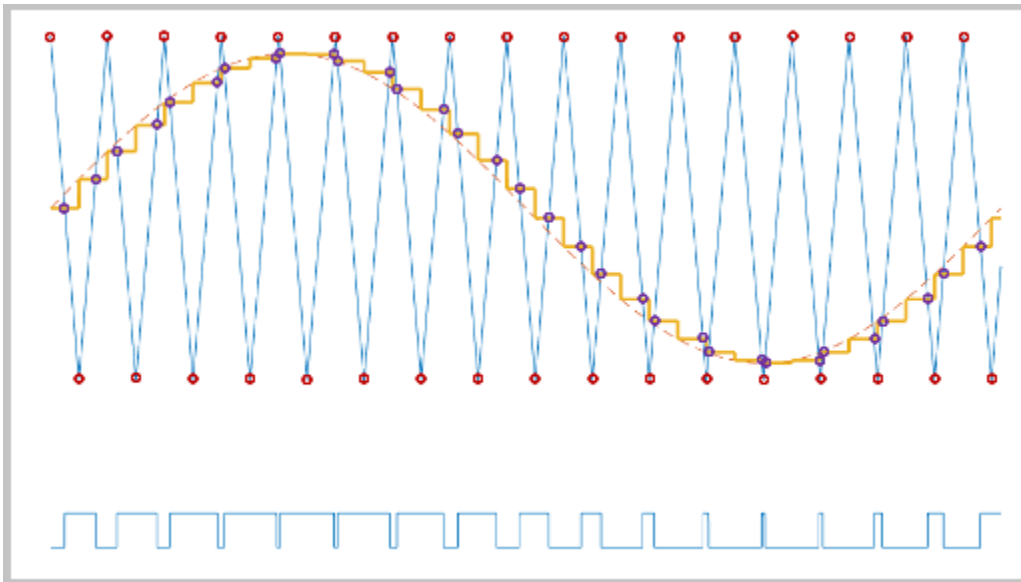
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

To determine switch-on and switch-off pulse behavior, a three-level carrier-based PWM generator uses these methods to sample each of the triangle waves:

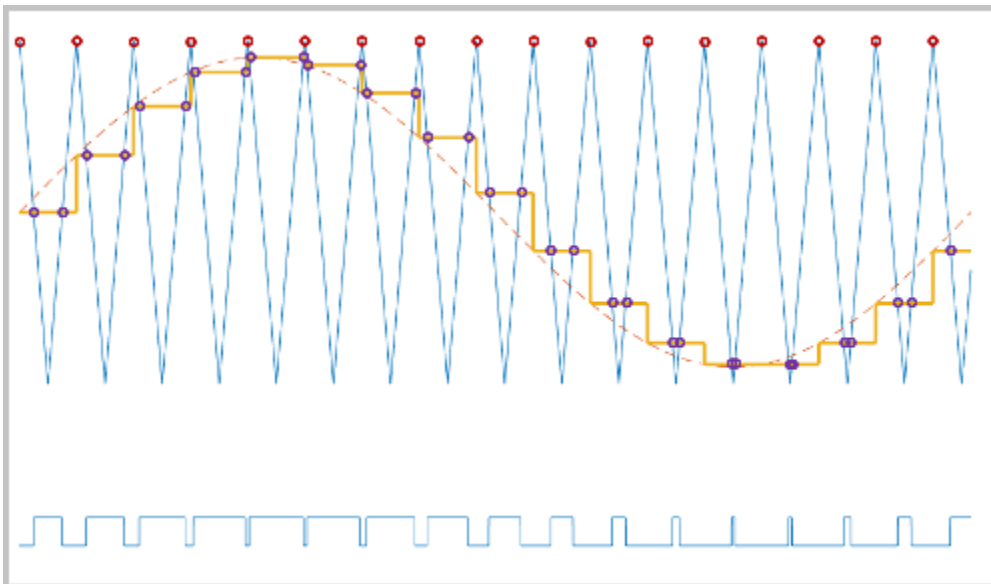
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs only at the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **Tgabcde0N — Switch-on gating time**

vector

Switch-on gating time, specified as a vector of five elements with one element for each phase.

#### **Tgabcde0FF — Switch-off gating time**

vector

Switch-off gating time, specified as a vector of five elements with one element for each phase.

## Output

### **g — Gate control**

vector

Ten pulse waveforms that determine the switching behavior in the attached power converter, returned as a vector.

## Parameters

### **Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### **Switching frequency (Hz) — Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch, in Hz.

### **Sample time (s) — Block sample time**

5e-5 (default) | positive scalar

Time interval between successive block executions, in seconds. To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the value of the **Switching frequency (Hz)** parameter.

## Version History

Introduced in R2021a

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

### **Blocks**

PWM Timing and Waveform Generator (Five-phase, Two-level) | PWM Generator (Five-phase, Two-level)

## PWM Generator (Multilevel)

Generate pulse width modulated signal or waveform for modular multilevel converters

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



### Description

The PWM Generator (Multilevel) block implements a pulse width modulation (PWM) generator for modular multilevel converters. The PWM technique controls power transfer from one electrical component to another by quickly switching between full power transfer and no power transfer.

The block calculates gate pulses for a modular, multilevel converter based on the number of power submodules and two converter topologies: half-bridge and full-bridge. You can specify the number of power submodules and the converter topology using the **Number of power submodules** and **Converter topology** parameters, respectively.

### Ports

#### Input

##### ref — Reference signal

vector

Reference signal to calculate the PWM for each power submodule of the multilevel converter, specified as a vector.

#### Output

##### PWM — PWM signal

vector

Pulse width modulation signal for each submodule, returned as a vector.

If you set the **Converter topology** parameter to **Half-bridge**, the output is a vector specified as:

$$G = \{G1\_SM1, G2\_SM1, G1\_SM2, G2\_SM2, \dots, G1\_SMN, G2\_SMN\}.$$

If you set the **Converter topology** parameter to **Full-bridge**, the output is a vector specified as:

$$G = \{G1\_SM1, G2\_SM1, G3\_SM1, G4\_SM1, G1\_SM2, G2\_SM2, G3\_SM2, G4\_SM2, \dots, G1\_SMN, G2\_SMN, G3\_SMN, G4\_SMN\},$$

where  $G_i\_SMj$  is the reference signal controlling the  $i$ -th gate of the  $j$ -th power submodule.  $i$  ranges from 1 to 2 in a half-bridge topology and from 1 to 4 in a full-bridge topology.  $j$  ranges from 1 to  $N$ , where  $N$  is the number of power submodules specified in the **Number of power submodules** parameter.

Data Types: single | double



## Parameters

### Converter topology — Modular multilevel converter topology

Half-bridge (default) | Full-bridge

Topology of the modular multilevel converter.

### Number of power submodules — Number of modular multilevel submodules

1 (default) | positive scalar

Number of power submodules of the modular multilevel converter.

### Timer period (s) — PWM period

0.001 (default) | positive scalar

PWM timer period. The **Timer period (s)** parameter is specified as  $\frac{1}{2f_{sw}}$ , where  $f_{sw}$  is the switching frequency.

### Sample time — Block sample time

5e-5 (default) | positive scalar

Sample time for the block. For continuous-time simulation, set to zero. For discrete-time simulation, to ensure adequate resolution in the generated signal, specify a positive value that is less than or equal to  $10^{-4} * T_{per}$ , where  $T_{per}$  is the **Timer period (s)**.

## Version History

Introduced in R2020b

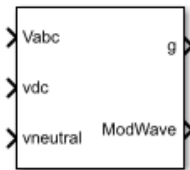
### See Also

PWM Generator | PWM Generator (Three-phase, Two-level) | PWM Generator (Three-phase, Three-level) | PWM Generator (Vienna Rectifier)

## PWM Generator (Three-phase, Three-level)

Generate three-phase, three-level pulse width modulated signal or waveform for gating switching devices

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



### Description

The PWM Generator (Three-phase, Three-level) block controls switching behavior for a three-phase, three-level power converter. The block:

- 1 Calculates on- and off-gating times based on the block inputs:
  - Three sinusoidal reference voltages
  - A DC-link voltage
  - A DC-link neutral point balance control signal
- 2 Uses the gating times to generate 12 switch-controlling pulses.
- 3 Uses the gating times to generate modulation waveforms.

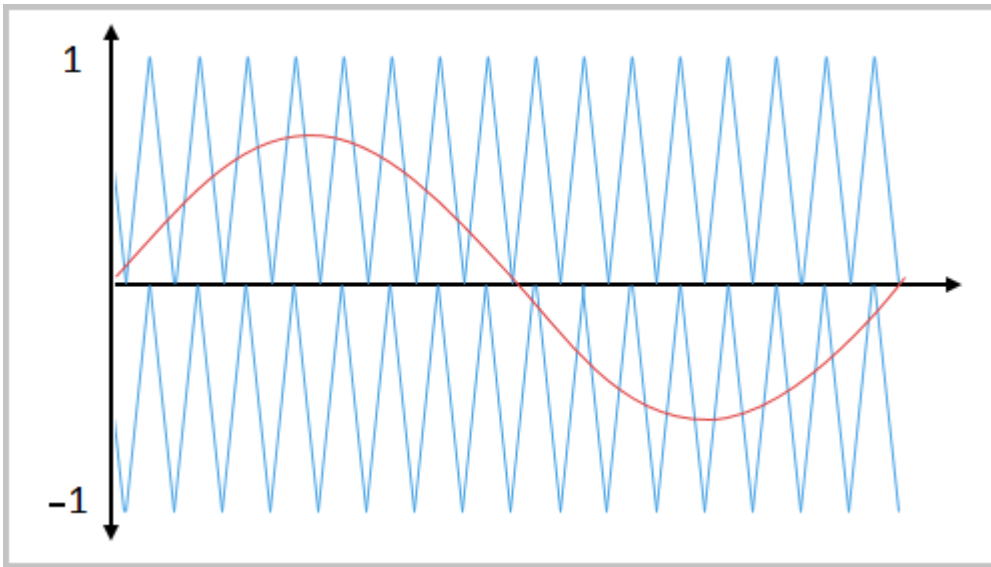
### Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Generator (Three-phase, Two-level) block does not perform carrier-based pulse width modulation (PWM). Instead, the block uses input signals to calculate gating times and then uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that it outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a three-level, carrier-based PWM method:

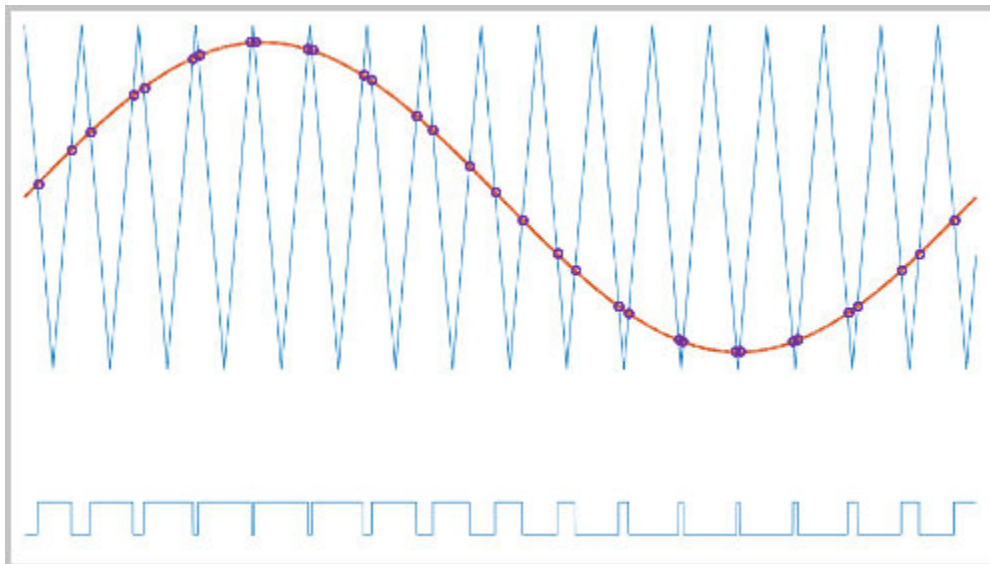
- 1 Samples a reference wave.
- 2 Compares the sample to two parallel triangle carrier waves, separated by one level.



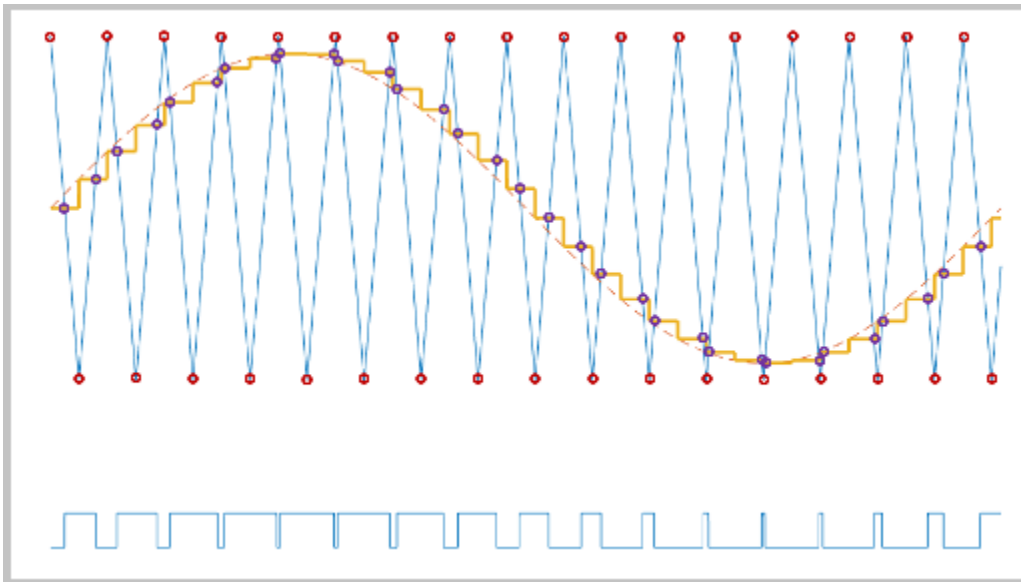
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave.

To determine switch-on and switch-off pulse behavior, a three-level carrier-based PWM generator uses these methods to sample each of the triangle waves:

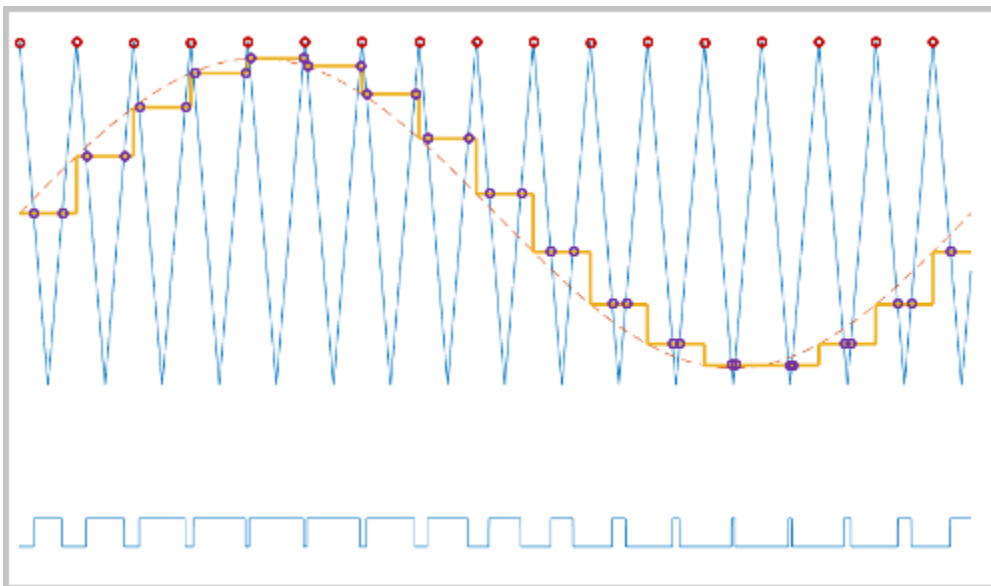
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs only at the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



### Overmodulation

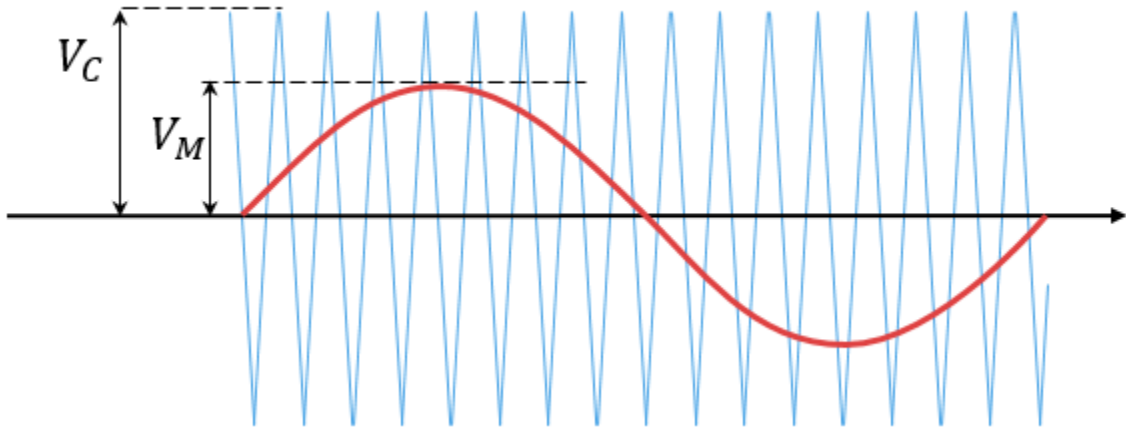
The modulation index, which measures the ability of the power converter to output a given voltage, is defined as

$$m = \frac{V_M}{V_C},$$

where

- $m$  is the modulation index.

- $V_m$  is the peak value of the modulation wave.
- $V_c$  is the peak value of the triangle carrier wave.



For three-phase SPWM,

$$V_{peak} = m \frac{v_{dc}}{2},$$

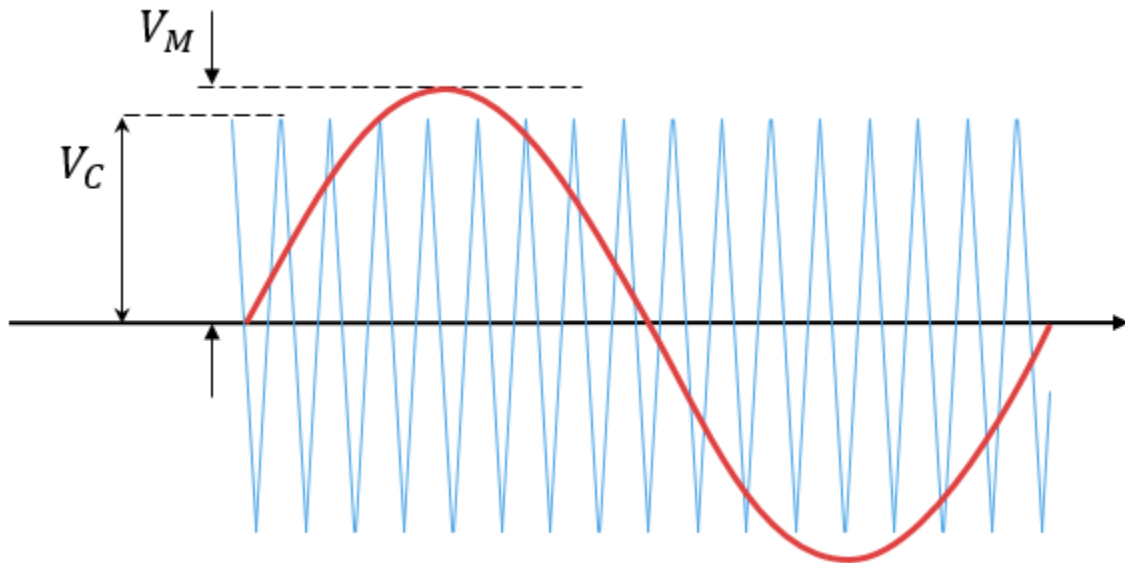
where

- $V_{peak}$  is the peak value of the fundamental component of the phase-to-neutral voltage.
- $v_{dc}$  is the DC-link voltage.

For three-phase space-vector PWM (SVM),

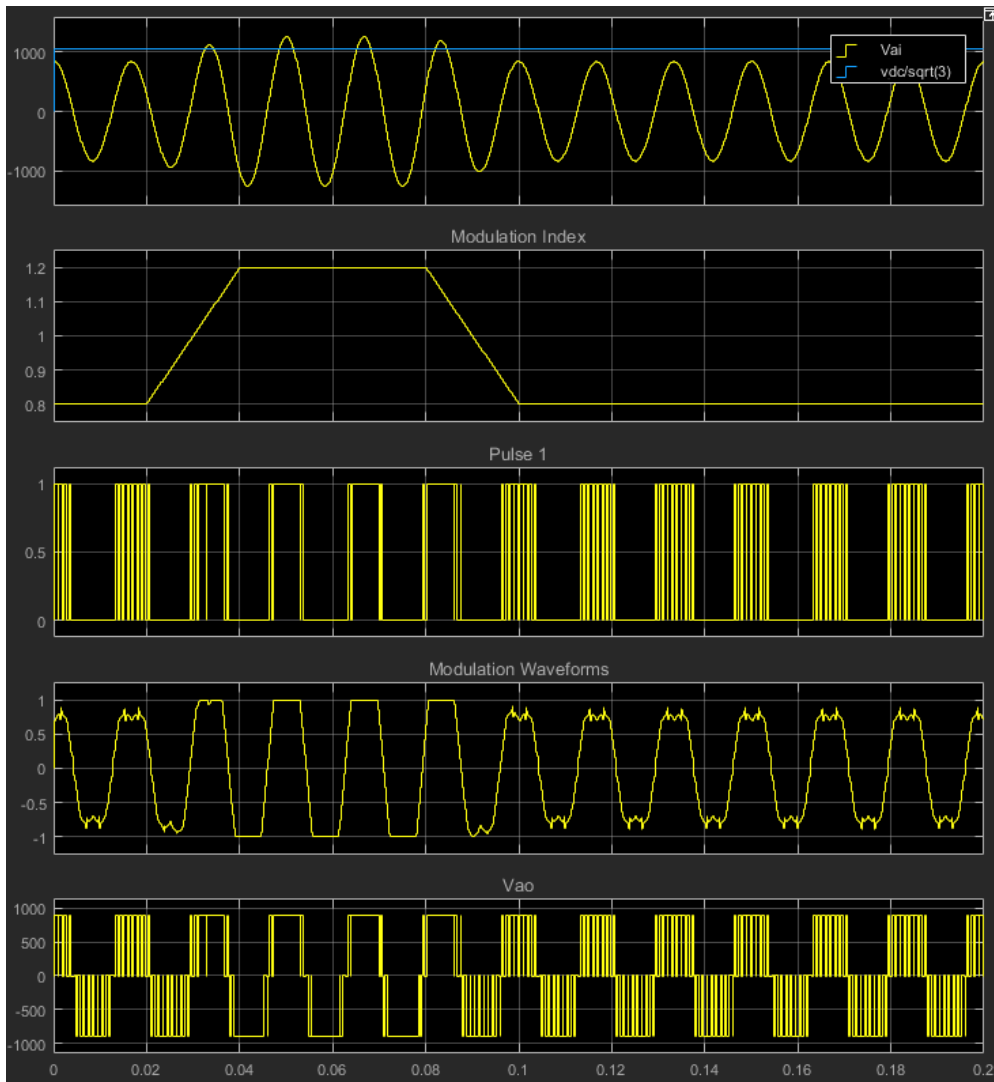
$$V_{peak} = m \frac{v_{dc}}{\sqrt{3}}.$$

For normal steady-state operation,  $0 < m \leq 1$ . If a transient, such as a load increase, causes the amplitude of  $V_m$  to exceed the amplitude of  $V_c$ , overmodulation ( $m > 1$ ) occurs



If overmodulation occurs, the output voltage of the power converter clamps to the positive or negative DC rail.

In the Three-Phase Three-Level PWM Generator example, the **Three-Level Controller** subsystem contains a 1800-V DC-link input, and a modulation index,  $m$ , of 0.8. For SVM, the maximal input voltage is  $1800/\sqrt{3}$  V, that is 1039.23 V. To demonstrate overmodulation, a transient is added at the beginning of the simulation. The transient forces the amplitudes of the reference voltages to exceed the amplitude of  $1/\sqrt{3}$  of the DC-link voltage. To highlight overmodulation, the scope includes simulation results for only one of the 12 output pulses and only the  $a$ -phase of the reference voltages, modulation waveforms, and output voltages.



The modulation index is greater than one between 0.03-0.09 seconds. During overmodulation:

- The pulse remains in the on or off position.
- The output voltage clamps to the positive or negative DC rail.

## Input/Output Ports

### Input

#### **vabc** — Three-phase sinusoidal reference signal

vector

Specify the three sinusoidal voltages, one per phase, that you want the attached converter to output.

#### **vdc** — DC-link voltage signal

scalar

Specify a positive real number for the DC-link voltage of the converter.

**vneutral — DC-link neutral point balance control**

scalar

This signal is the output from a feedback-control loop that balances the DC supply. The value of the signal must be a real number between  $-1$  and  $+1$ .

**Output****g — Gate control**

vector

12 pulse waveforms that determine switching behavior in the attached power converter.

**ModWave — Modulation wave**

vector

If you are generating code for a platform that has hardware with PWM capability, you can deploy the modulation wave to the hardware. Otherwise, this data is for reference only.

**Parameters****Continuous PWM — Continuous pulse width modulation method**

SPWM: sinusoidal PWM (default) | SVM: space vector modulation

Specify the waveform technique.

**Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

**Switching frequency (Hz) — Switching rate**

1e3 (default) | positive number

Specify the rate at which you want the switches in the power converter to switch.

**Sample time (s) — Block sample time**

5e-5 (default) | positive number

Specify the time interval between successive block executions (output calculations). To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the **Switching frequency (Hz)**.

**Version History****Introduced in R2016b****References**

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.



[2] Seo, J. H., C. H. Choi, and D. S. Hyun. "A new simplified space-vector PWM method for three-level inverters." *IEEE Transactions on Power Electronics*, Vol. 16, No. 4, 2001, pp. 545-550.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Three-Level Converter (Three-Phase)

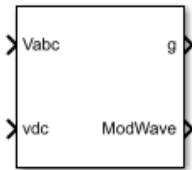
### **Blocks**

PWM Generator | PWM Generator (Three-phase, Two-level) | Thyristor 6-Pulse Generator

## PWM Generator (Three-phase, Two-level)

Generate three-phase, two-level pulse width modulated waveform

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



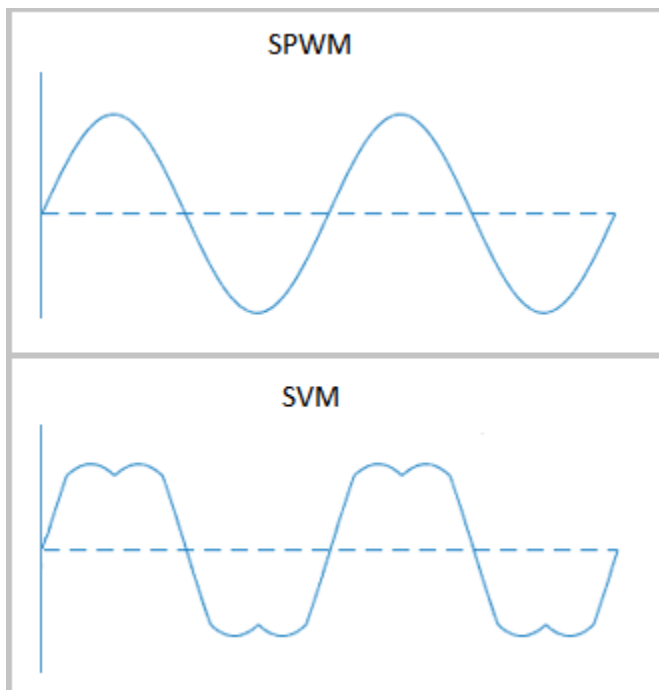
### Description

The PWM Generator (Three-phase, Two-level) block controls switching behavior for a three-phase, two-level power converter. The block:

- 1 Calculates on- and off-gating times based on the block inputs:
  - Three sinusoidal reference voltages, one per phase
  - A DC-link voltage
- 2 Uses the gating times to generate six switch-controlling pulses.
- 3 Uses the gating times to generate modulation waveforms.

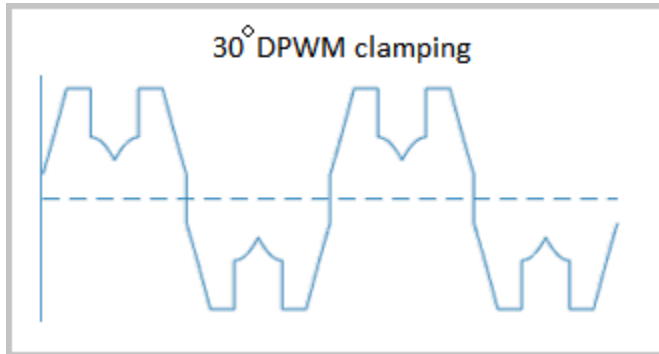
### Continuous and Discontinuous PWM

The block provides modes for both continuous and discontinuous pulse width modulation (PWM). The figure shows the general difference between continuous sinusoidal PWM (SPWM) and continuous space vector modulation (SVM) waveforms.

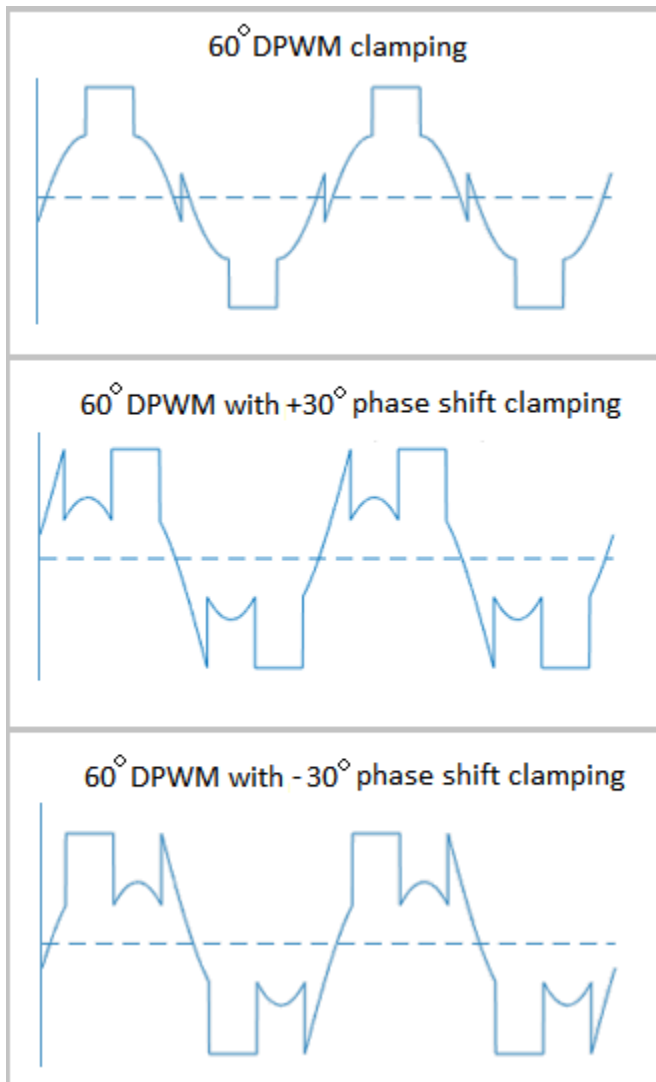


For discontinuous PWM (DPWM), the block clamps the modulation wave to the positive or negative DC rail for a total of 120 degrees during each fundamental period. During the clamping intervals, modulation discontinues.

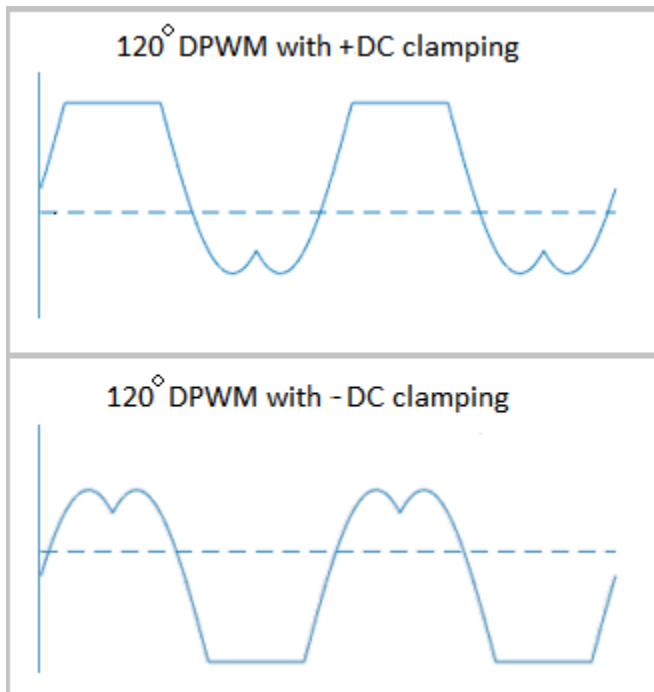
A waveform with 30-degree DPWM has four 30-degree intervals per fundamental period.



Selecting a positive or negative 30-degree phase shift affects the clamping intervals for 60-degree DPWM.



The figure shows the waveforms for positive and negative DC clamping for 120-degree DPWM.



### Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

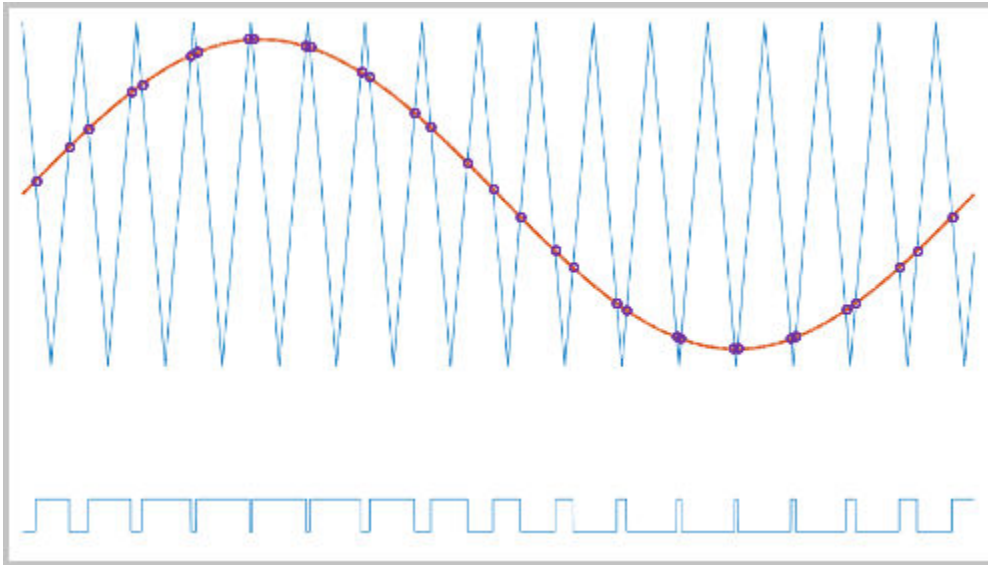
The PWM Generator (Three-phase, Two-level) block does not perform carrier-based PWM. Instead, the block uses input signals to calculate gating times and then uses the gating times to generate both the switch-controlling pulses and the modulation waveforms that it outputs.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a two-level, carrier-based PWM method:

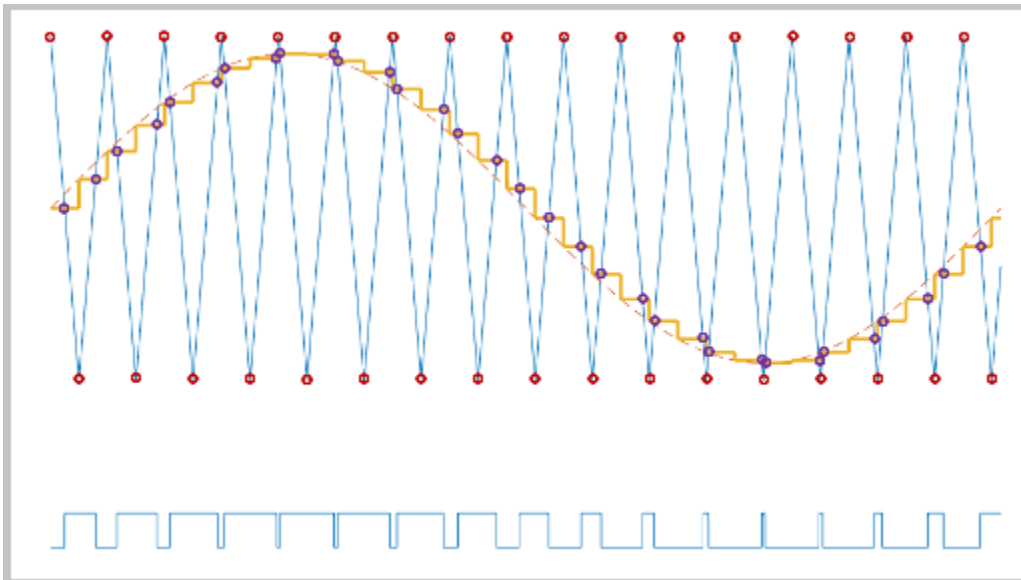
- 1 Samples a reference wave.
- 2 Compares the sample to a triangle carrier wave.
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave.

To determine switch-on and switch-off pulse behavior, a two-level carrier-based PWM generator uses these methods to sample the triangle wave:

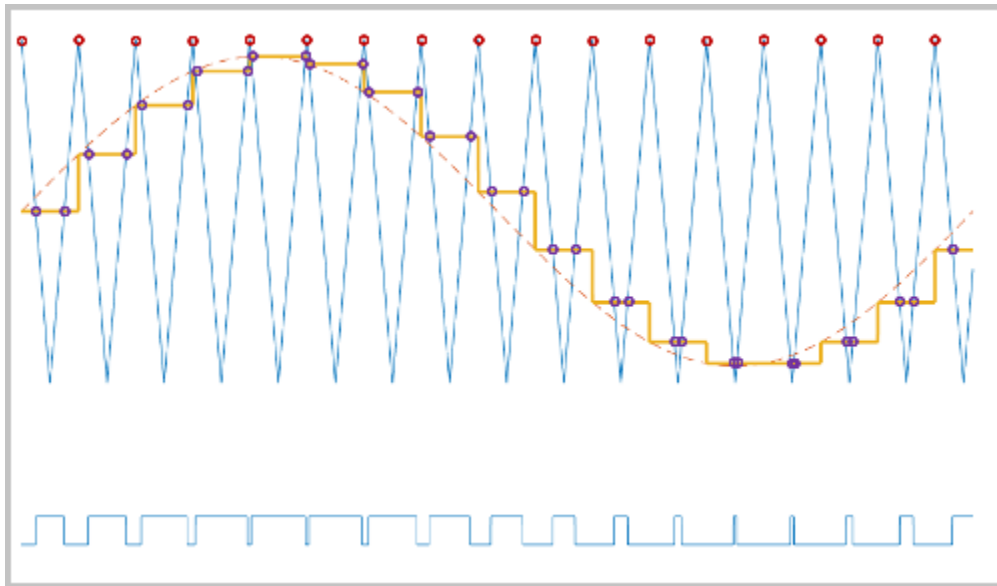
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs at only the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



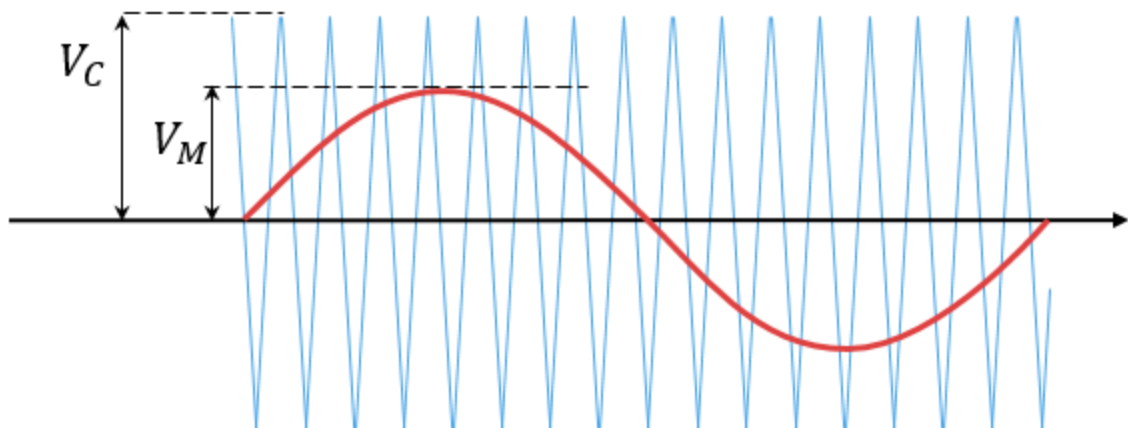
### Overmodulation

The modulation index, which measures the ability of the power converter to output a given voltage, is defined as

$$m = \frac{V_M}{V_C},$$

where

- $m$  is the modulation index.
- $V_m$  is the peak value of the modulation wave.
- $V_c$  is the peak value of the triangle carrier wave.



For three-phase SPWM,

$$V_{peak} = m \frac{v_{dc}}{2},$$

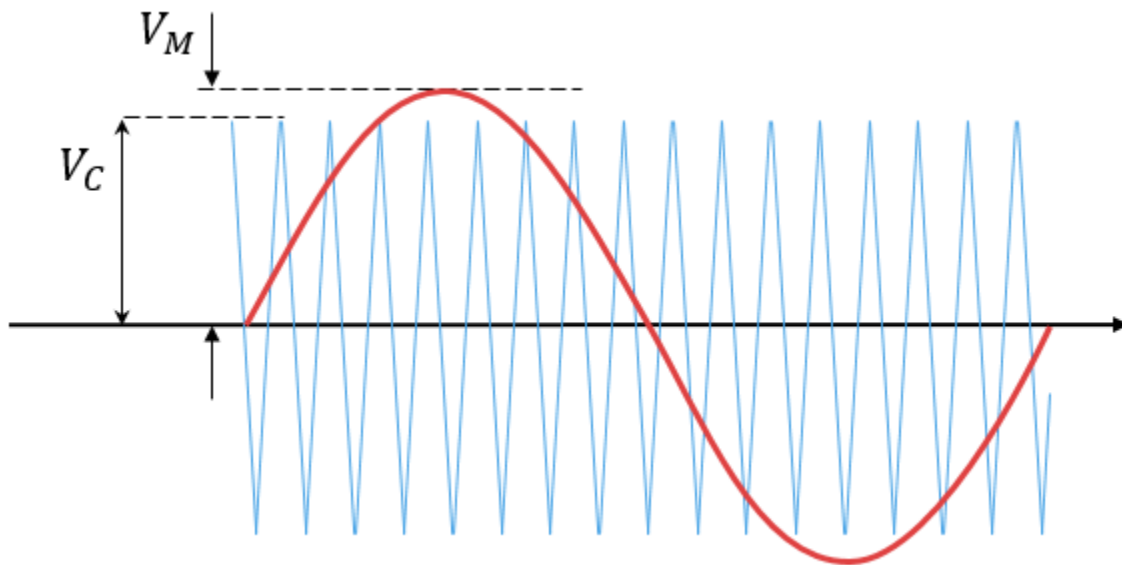
where

- $V_{peak}$  is the peak value of the fundamental component of the phase-to-neutral voltage.
- $v_{dc}$  is the DC-link voltage.

For three-phase space-vector PWM (SVM) and DPWM,

$$V_{peak} = m \frac{v_{dc}}{\sqrt{3}}.$$

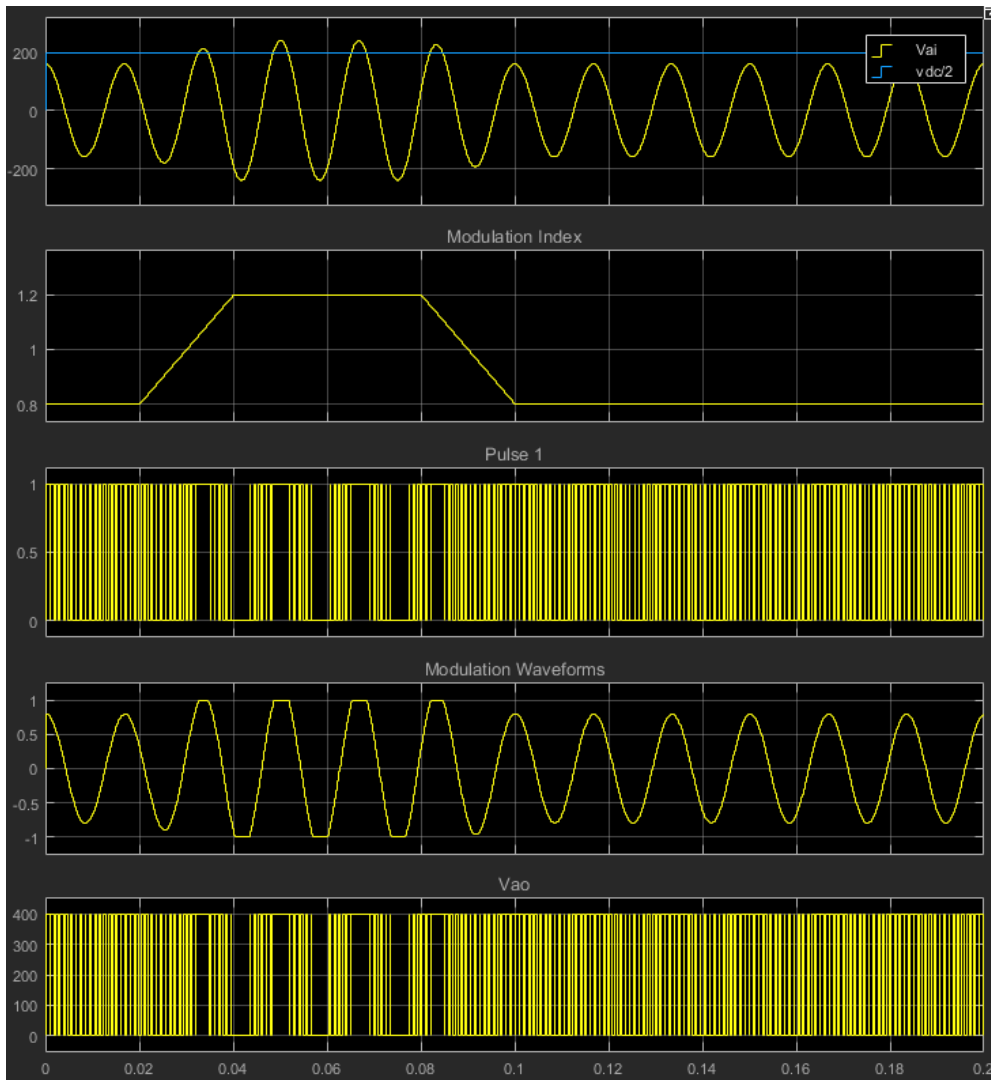
For normal steady-state operation,  $0 < m \leq 1$ . If a transient, such as a load increase, causes the amplitude of  $V_m$  to exceed the amplitude of  $V_c$ , overmodulation ( $m > 1$ ) occurs.



If overmodulation occurs, the output voltage of the power converter clamps to the positive or negative DC rail.

In the Three-Phase Two-Level PWM Generator example, the **Two-Level Controller** subsystem contains a 400-V DC-link input, and a modulation index,  $m$ , of 0.8. For SPWM, the maximal input voltage is  $400/2$ , that is, 200 V. To demonstrate overmodulation, a transient is added at the beginning of the simulation. The transient forces the amplitudes of the reference voltages to exceed the amplitude of  $1/2$  of the DC-link voltage. To highlight overmodulation, the scope includes simulation results for only one of the six output pulses and only the  $a$ -phase of the reference voltages, modulation waveforms, and output voltages.





The modulation index is greater than one between 0.03-0.09 seconds. During overmodulation:

- The pulse remains in the on or off position.
- The output voltage,  $V_{ao}$ , clamps to the positive or negative DC rail.

## Input/Output Ports

### Input

#### **vabc** — Three-phase sinusoidal reference signal

vector

Specify the three sinusoidal voltages, one per phase, that you want the attached converter to output.

#### **vdc** — DC-link voltage signal

scalar

Specify a positive real number for the DC-link voltage of the converter.

## Output

### **g – Gate control**

vector

Six pulse waveforms that determine switching behavior in the attached power converter.

### **ModWave – Modulation wave**

vector

If you are generating code for a platform that has hardware with PWM capability, you can deploy the modulation wave to the hardware. Otherwise, this data is only for your reference.

## Parameters

### **PWM mode – Pulse width modulation method**

Continuous PWM (CPWM) (default) | Discontinuous PWM (DPWM)

Discontinuous PWM clamps the waveform to the DC rail for a total of 120 degrees in each fundamental period. Continuous PWM does not.

### **Continuous PWM – Continuous pulse width modulation method**

SPWM: sinusoidal PWM (default) | SVM: space vector modulation

### **Dependencies**

The **Continuous PWM** parameter is only available when you set the **PWM mode** parameter to Continuous PWM (CPWM).

### **Sampling mode – Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### **Switching frequency (Hz) – Switching rate**

1e3 (default) | positive number

Specify the rate at which you want the switches in the power converter to switch.

### **Sample time (s) – Block sample time**

5e-5 (default) | positive number

Specify the time interval between successive block executions (output calculations). To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the **Switching frequency (Hz)**.

### **Discontinuous PWM (DPWM) – Clamping method**

60 DPWM: 60 degree discontinuous PWM (default)

Specify the method for distributing the 120 degrees per period during which the block clamps the modulation wave to the DC rail. Other options are:

- 60 DPWM (+30 degree shift): +30 degree shift from 60 DPWM

- 60 DPWM (-30 degree shift): -30 degree shift from 60 DPWM
- 30 DPWM: 30 degree discontinuous PWM
- 120 DPWM: positive dc component
- 120 DPWM: negative dc component

When the wave is clamped, modulation discontinues.

### Dependencies

The **Discontinuous PWM** parameter is only available when you set the **PWM mode** parameter to Discontinuous PWM (DPWM).

## Version History

Introduced in R2016b

### References

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.
- [2] Hava, A. M., R. J. Kerkman, and T. A. Lipo. "Simple Analytical and Graphical Methods for Carrier-Based PWM-VSI Drives." *IEEE Transactions on Power Electronics*, Vol. 14, No. 1, 1999, pp. 49-61.

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

#### Simscape Blocks

Converter (Three-Phase)

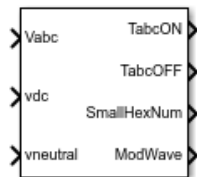
#### Blocks

PWM Generator | PWM Generator (Three-phase, Three-level) | Thyristor 6-Pulse Generator

## PWM Timing and Waveform Generator (Three-phase, Three-level)

Generate three-phase, three-level gating time information

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



### Description

The PWM Timing and Waveform Generator (Three-phase, Three-level) block controls switching behavior for a three-phase, three-level power converter. The block calculates on-gating and off-gating times and modulation waves based on these block inputs:

- Three sinusoidal reference voltages
- A DC-link voltage
- A DC-link neutral point balance control signal

This block can be used with average converters to speed up the simulation.

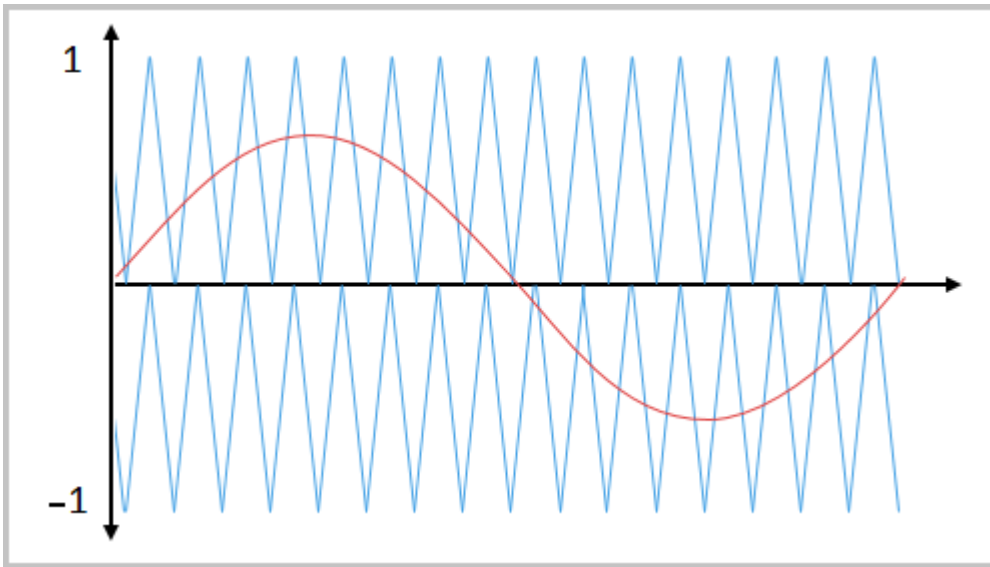
### Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Timing and Waveform Generator (Three-phase, Three-level) block does not perform carrier-based pulse width modulation (PWM). Instead, the block uses input signals to calculate the gating times.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a three-level, carrier-based PWM method:

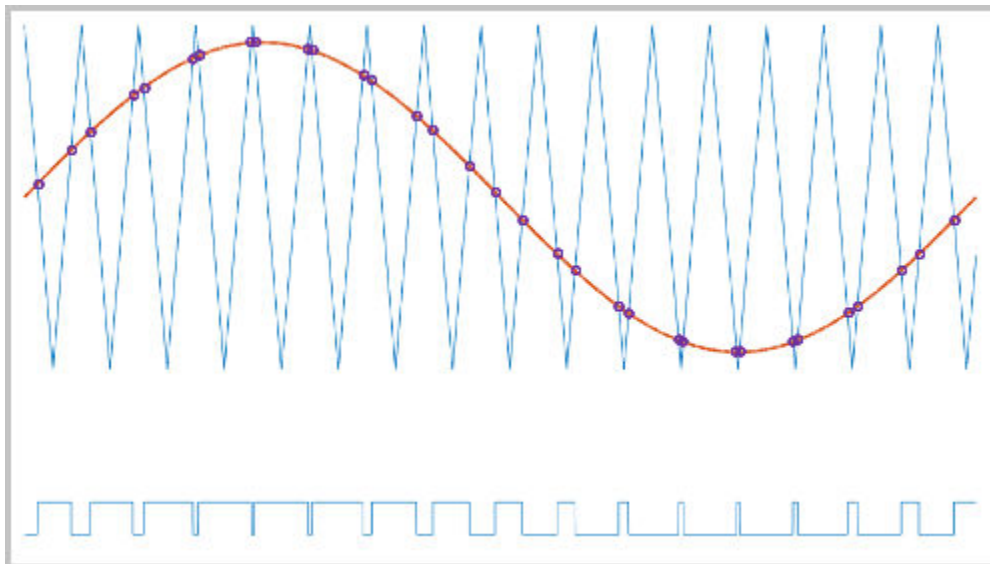
- 1 Samples a reference wave
- 2 Compares the sample to two parallel triangle carrier waves, separated by one level



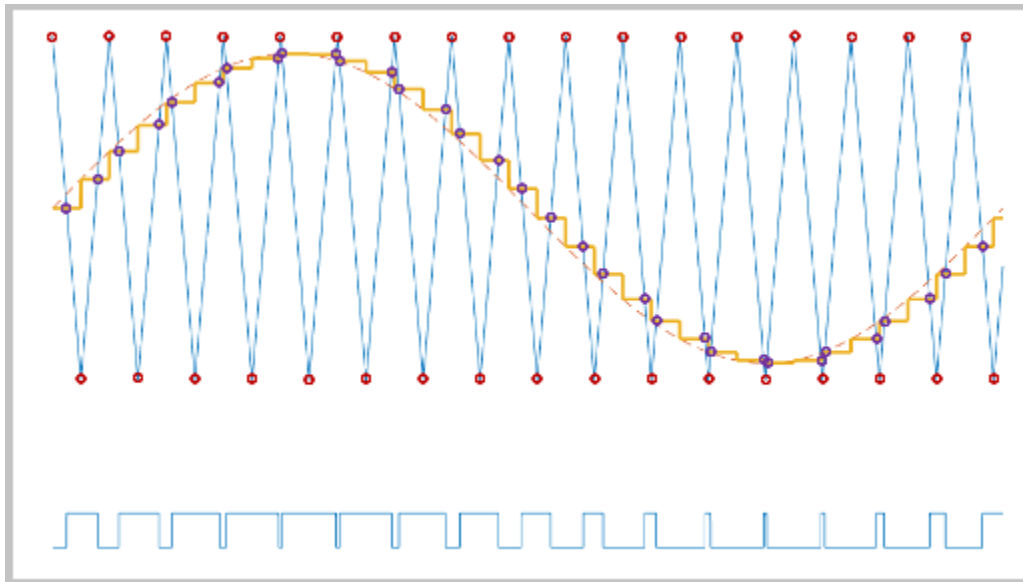
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

To determine switch-on and switch-off pulse behavior, a three-level carrier-based PWM generator uses these methods to sample each of the triangle waves:

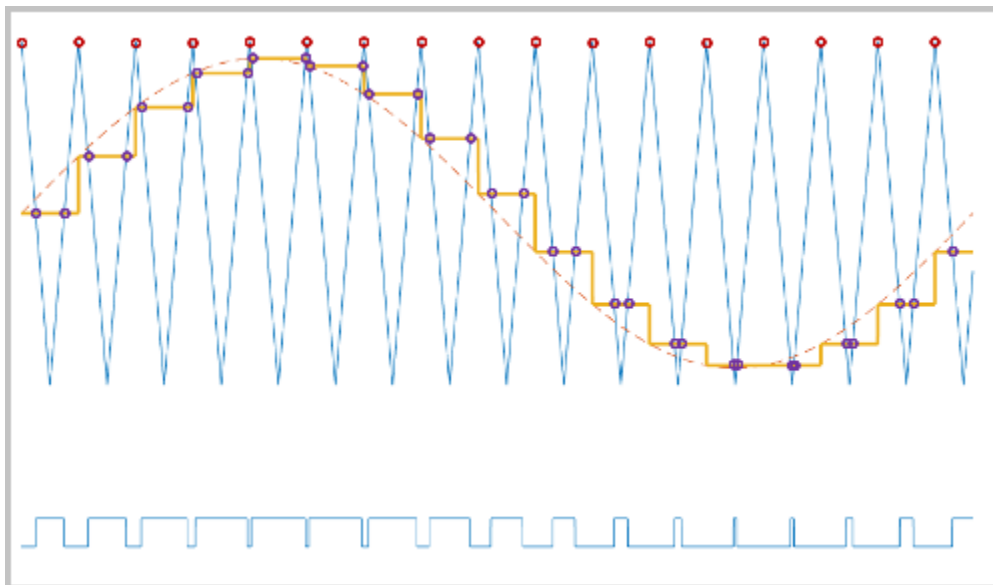
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs only at the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **vabc — Three-phase sinusoidal reference signal**

vector

Sinusoidal voltages that you want the attached converter to output, specified as a vector of three elements, with one element for each phase.

#### **vdc — DC-link voltage signal**

scalar

Positive real number for the DC-link voltage of the converter, specified as a scalar.

#### **vneutral — DC-link neutral point balance control**

scalar

Output from a feedback-control loop that balances the DC supply, specified as a real number between -1 and +1.

#### **Output**

#### **Tgabc0N — Switch-on gating time**

vector

Switch-on gating time, returned as a vector of three elements, with one element for each phase.

#### **Tgabc0FF — Switch-off gating time**

vector

Switch-off gating time, returned as a vector of three elements, with one element for each phase.

#### **SmallHexNum — Small hexagon number**

scalar

Number of the small hexagon that composes the space-vector diagrams of conventional two-level inverters, returned as a scalar between 1 and 6. For more information, see the “Simplified Space-Vector PWM Method” on page 1-1580 section in the PWM Gate Signal Generator (Three-phase, Three-level) block.

#### **ModWave — Modulation waves**

vector

Modulation waves, returned as a vector of three elements, with one element for each phase.

### **Parameters**

#### **Continuous PWM — Continuous pulse width modulation method**

SPWM: sinusoidal PWM (default) | SVM: space vector modulation

Specify the waveform technique.

#### **Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

#### **Switching frequency (Hz) — Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch.

## **Version History**

**Introduced in R2020b**

### **References**

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.
- [2] Seo, J. H., C. H. Choi, and D. S. Hyun. "A new simplified space-vector PWM method for three-level inverters." *IEEE Transactions on Power Electronics*, Vol. 16, No. 4, 2001, pp. 545-550.

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

#### **See Also**

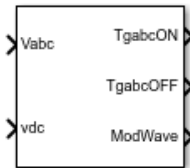
PWM Timing and Waveform Generator (Three-phase, Two-level) | PWM Generator (Three-phase, Three-level) | PWM Gate Signal Generator (Three-phase, Three-level) | Three-Level Converter (Three-Phase)



# PWM Timing and Waveform Generator (Three-phase, Two-level)

Generate three-phase, two-level gating time information

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

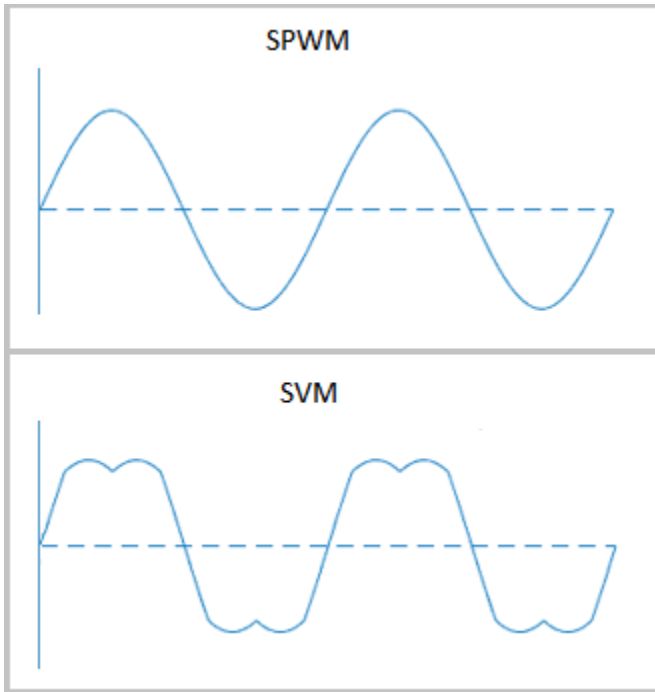
The PWM Timing and Waveform Generator (Three-phase, Two-level) block controls switching behavior for a three-phase, two-level power converter. The block calculates on-gating and off-gating times and modulation waves based on these block inputs:

- Three sinusoidal reference voltages
- A DC-link voltage

This block can be used with average converters to speed up the simulation.

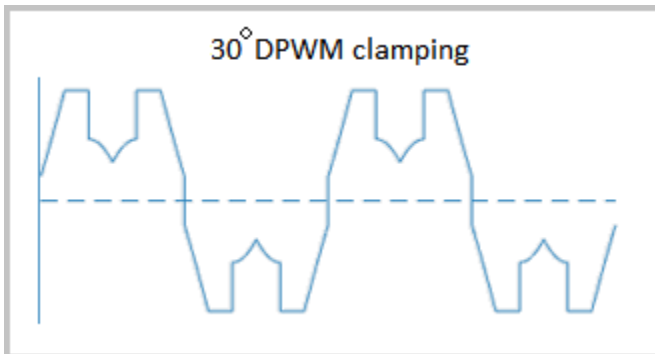
### Continuous and Discontinuous PWM

The block provides modes for both continuous and discontinuous pulse width modulation (PWM). The figure shows the general difference between continuous sinusoidal pulse width modulation (SPWM) and continuous space vector modulation (SVM) waveforms.

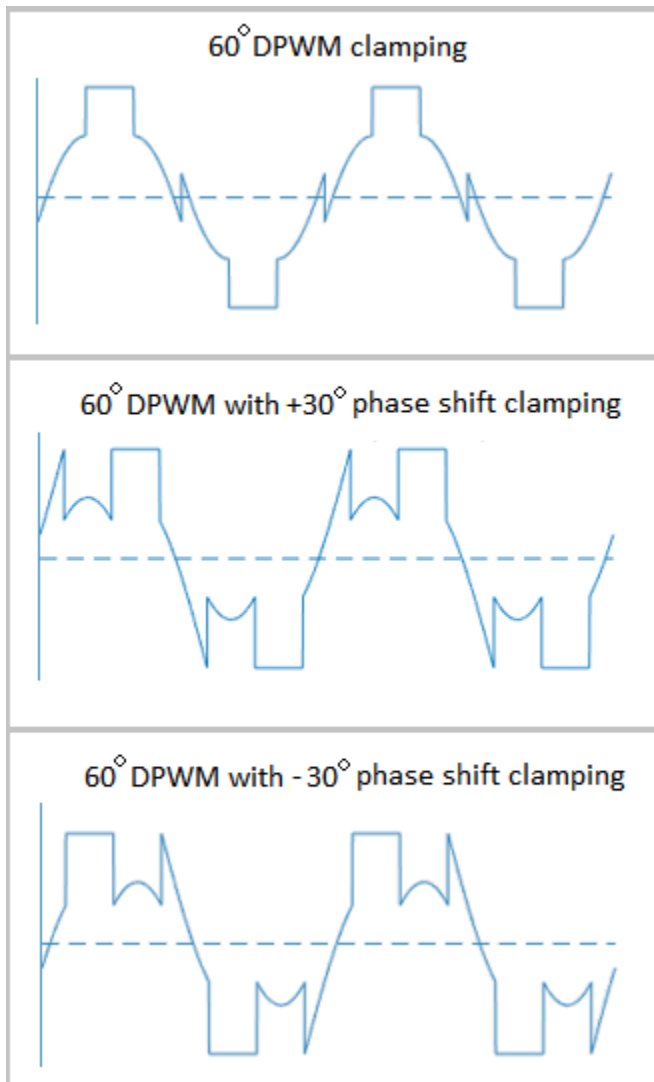


For discontinuous pulse width modulation (DPWM), the block clamps the modulation wave to the positive or negative DC rail for a total of 120 degrees during each fundamental period. During the clamping intervals, modulation discontinues.

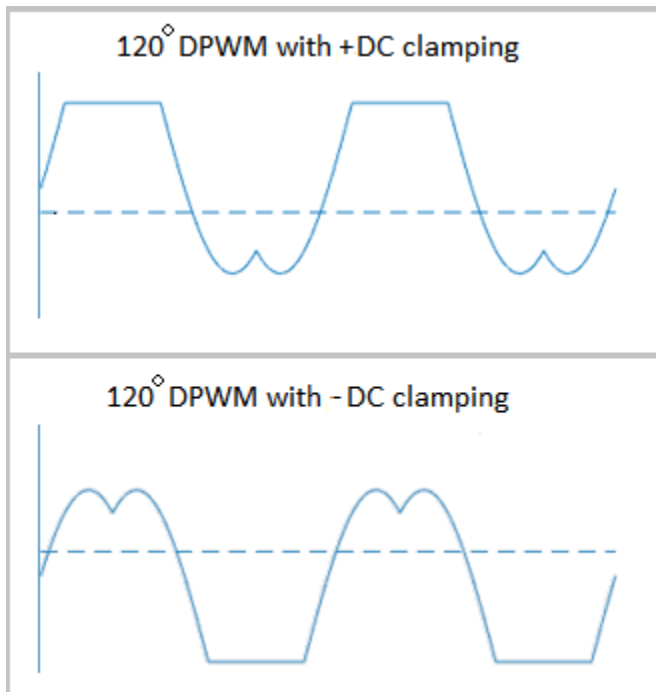
A waveform with 30-degree DPWM has four 30-degree intervals per fundamental period.



Selecting a positive or negative 30-degree phase shift affects the clamping intervals for 60-degree DPWM.



The figure shows the waveforms for positive and negative DC clamping for 120-degree DPWM.



### Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

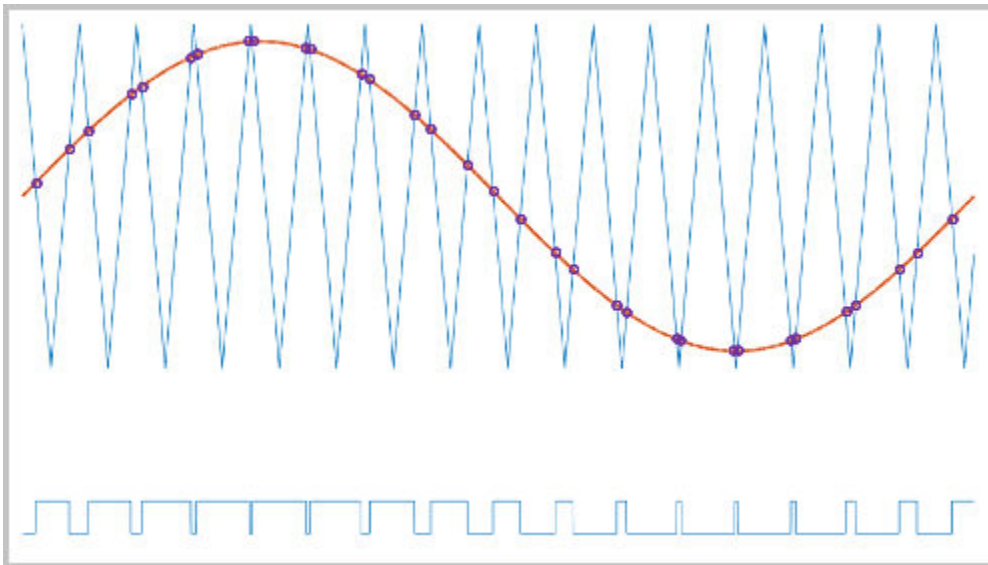
The PWM Timing and Waveform Generator (Three-phase, Two-level) block does not perform carrier-based PWM. Instead, the block uses input signals to calculate the gating times.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a two-level, carrier-based PWM method:

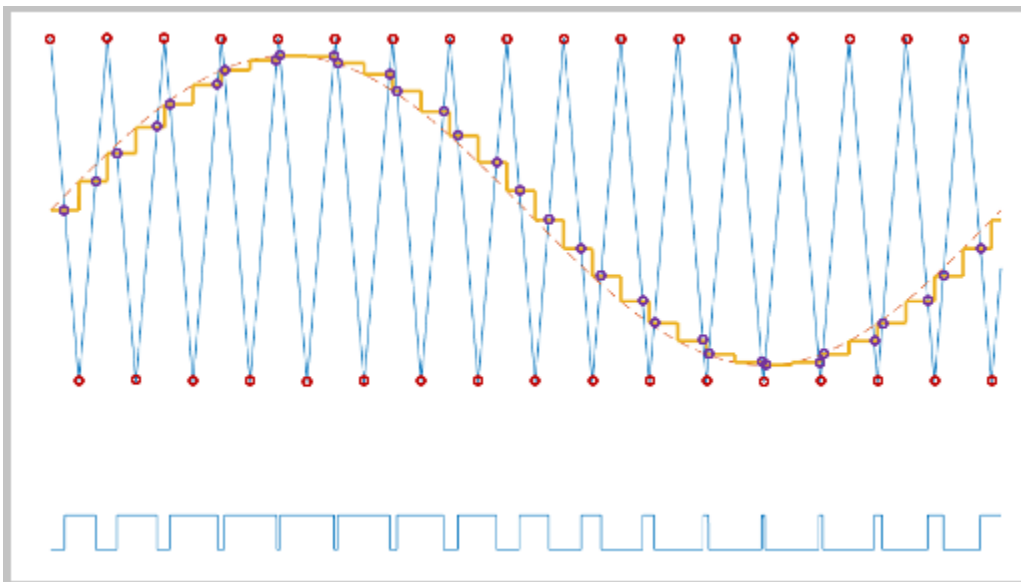
- 1 Samples a reference wave
- 2 Compares the sample to a triangle carrier wave
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

To determine switch-on and switch-off pulse behavior, a two-level carrier-based PWM generator uses these methods to sample the triangle wave:

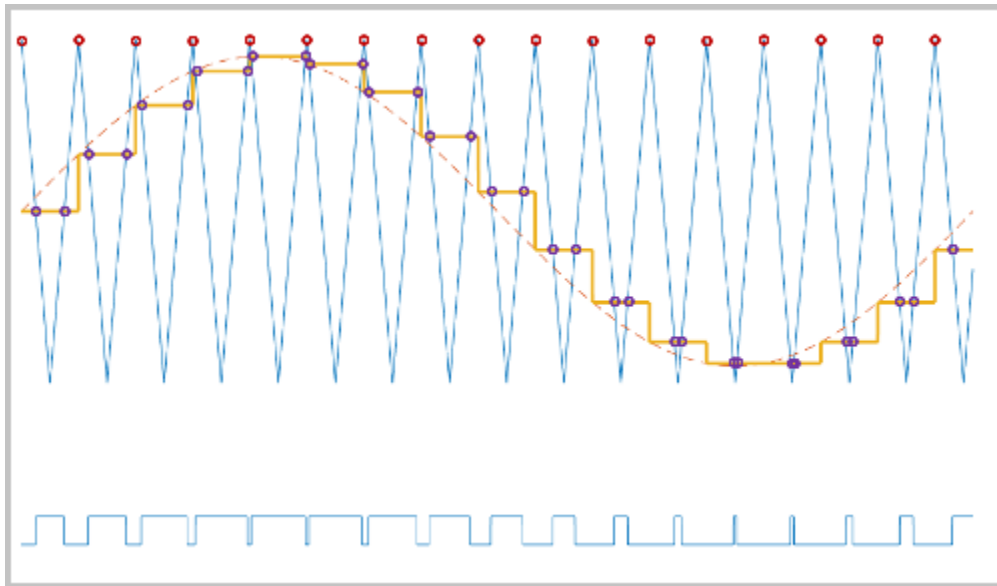
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs at only the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **Vabc — Three-phase sinusoidal reference signal**

vector

Sinusoidal voltages that you want the attached converter to output, specified as a vector of three elements, with one element for each phase.

#### **vdc — DC-link voltage signal**

scalar

Positive real number for the DC-link voltage of the converter, specified as a scalar.

### Output

#### **TgabcON — Switch-on gating time**

vector

Switch-on gating time, returned as a vector of three elements, with one element for each phase.

#### **TgabcOFF — Switch-off gating time**

vector

Switch-off gating time, returned as a vector of three elements, with one element for each phase.

#### **ModWave — Modulation waves**

vector

Modulation waves, returned as a vector of three elements, with one element for each phase.

## Parameters

### PWM mode — Pulse width modulation method

Continuous PWM (CPWM) (default) | Discontinuous PWM (DPWM)

Pulse width modulation method. Discontinuous PWM (DPWM) clamps the waveform to the DC rail for a total of 120 degrees in each fundamental period. Continuous PWM (CPWM) does not.

### Continuous PWM — Continuous pulse width modulation method

SVM: space vector modulation (default) | SPWM: sinusoidal PWM

Continuous pulse width modulation method.

### Dependencies

To enable this parameter, set **PWM mode** to Continuous PWM (CPWM).

### Discontinuous PWM (DPWM) — Clamping method

60 DPWM: 60 degree discontinuous PWM (default) | ...

Discontinuous pulse width modulation method. Specify the method for distributing the 120 degrees per period during which the block clamps the modulation wave to the DC rail. Options include:

- 60 DPWM: 60 degree discontinuous PWM
- 60 DPWM (+30 degree shift): +30 degree shift from 60 DPWM
- 60 DPWM (-30 degree shift): -30 degree shift from 60 DPWM
- 30 DPWM: 30 degree discontinuous PWM
- 120 DPWM: positive dc component
- 120 DPWM: negative dc component

When the wave is clamped, modulation discontinues.

### Dependencies

To enable this parameter, set **PWM mode** to Discontinuous PWM (DPWM).

### Sampling mode — Wave-sampling method

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

### Switching frequency (Hz) — Switching rate

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch.

## Version History

Introduced in R2020b

## References

- [1] Chung, D. W., J. S. Kim, and S. K. Sul. "Unified Voltage Modulation Technique for Real Time Three-Phase Power Conversion." *IEEE Transactions on Industry Applications*, Vol. 34, No. 2, 1998, pp. 374-380.
- [2] Seo, J. H., C. H. Choi, and D. S. Hyun. "A new simplified space-vector PWM method for three-level inverters." *IEEE Transactions on Power Electronics*, Vol. 16, No. 4, 2001, pp. 545-550.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

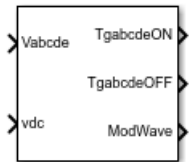
PWM Timing and Waveform Generator (Three-phase, Three-level) | PWM Generator (Three-phase, Two-level) | PWM Gate Signal Generator (Three-phase, Two-level) | Converter (Three-Phase)



# PWM Timing and Waveform Generator (Five-phase, Two-level)

Generate five-phase, two-level gating time information

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

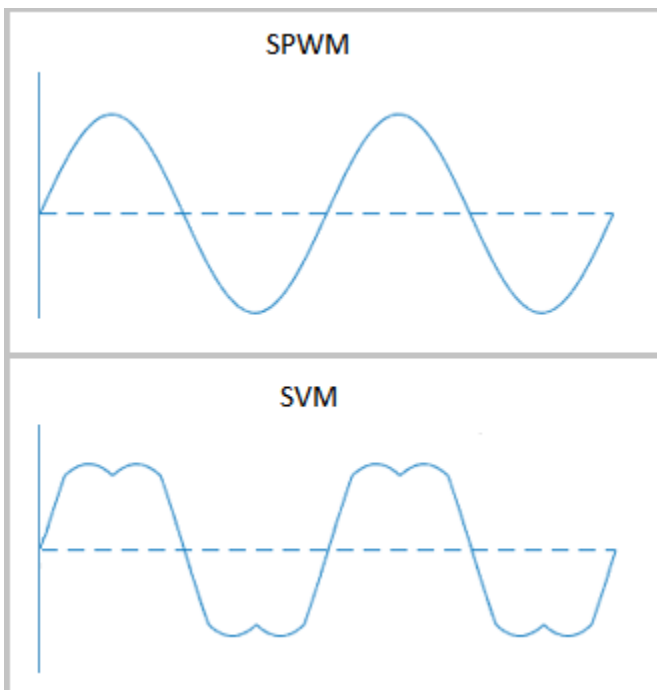
The PWM Timing and Waveform Generator (Five-phase, Two-level) block controls the switching behavior for a five-phase, two-level power converter. The block calculates on-gating and off-gating times and modulation waves based on these block inputs:

- Five sinusoidal reference voltages
- A DC-link voltage

This block can be used with average converters to speed up the simulation.

## Continuous PWM

The block only provides continuous pulse width modulation (PWM). The figure shows the general difference between continuous sinusoidal pulse width modulation (SPWM) and continuous space vector modulation (SVM) waveforms.



## Sampling Mode

This block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

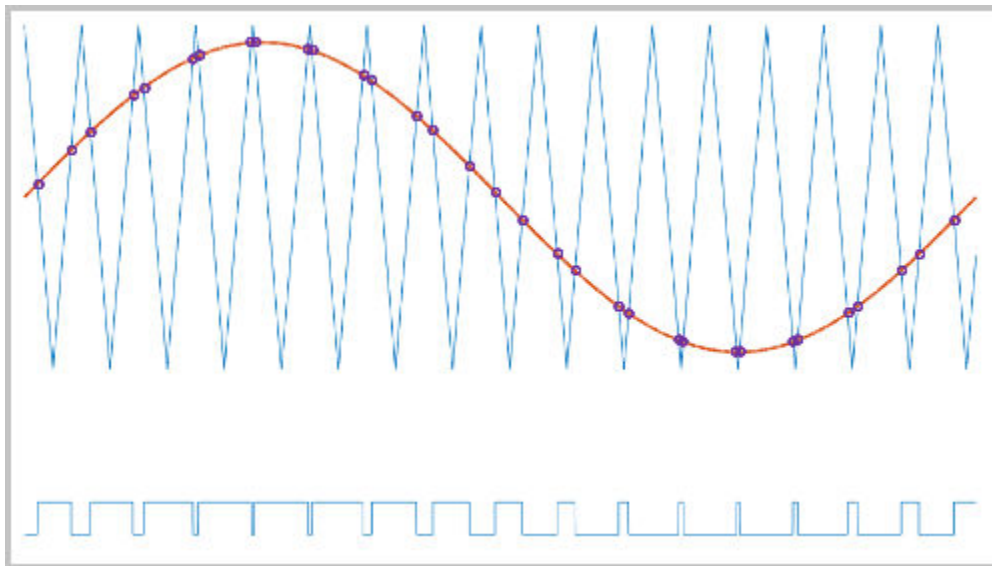
The PWM Timing and Waveform Generator (Five-phase, Two-level) block does not perform carrier-based PWM. Instead, the block uses input signals to calculate the gating times.

Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a two-level, carrier-based PWM method:

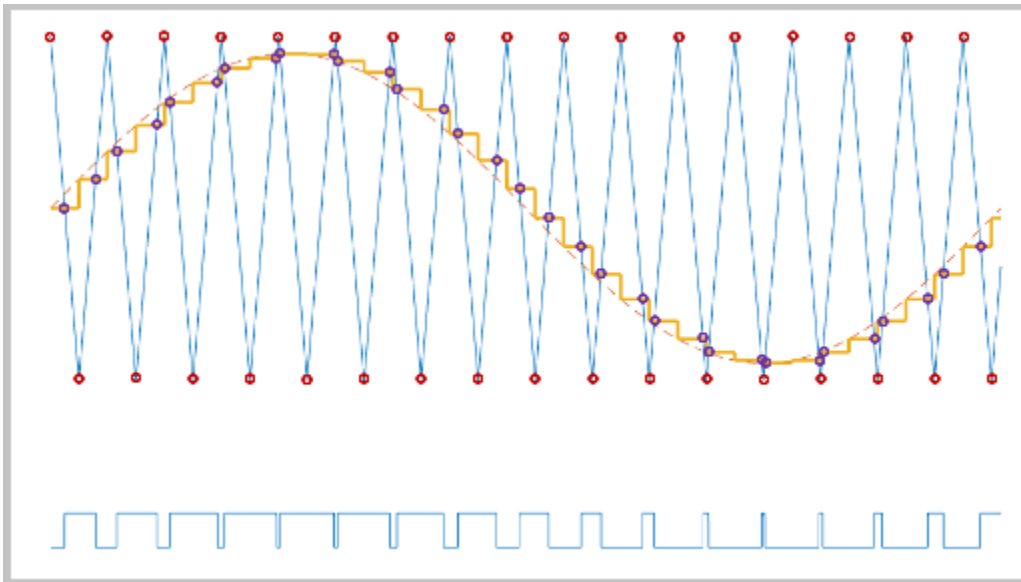
- 1 Samples a reference wave
- 2 Compares the sample to a triangle carrier wave
- 3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave

To determine switch-on and switch-off pulse behavior, a two-level carrier-based PWM generator uses these methods to sample the triangle wave:

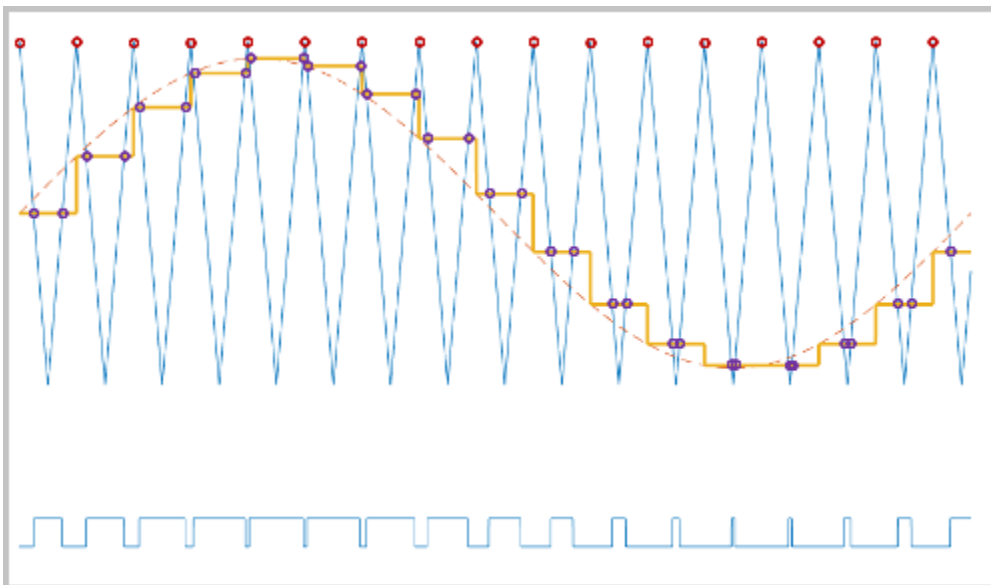
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- Asymmetric — Sampling occurs at the upper and lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling.



- Symmetric — Sampling occurs at only the upper boundary of the carrier wave. The comparison occurs at the intersection that follows the sampling.



## Ports

### Input

#### **vabcde — Five-phase sinusoidal reference signal**

vector

Sinusoidal voltages that you want the attached converter to output, specified as a vector of five elements with one element for each phase.

#### **vdc — DC-link voltage signal**

scalar

Positive real number for the DC-link voltage of the converter, specified as a scalar.

### **Output**

#### **TgabcdeON — Switch-on gating time**

vector

Switch-on gating time, returned as a vector of five elements with one element for each phase.

#### **TgabcdeOFF — Switch-off gating time**

vector

Switch-off gating time, returned as a vector of five elements with one element for each phase.

#### **ModWave — Modulation waves**

vector

Modulation waves, returned as a vector of five elements, with one element for each phase.

### **Parameters**

#### **Continuous PWM — Continuous pulse width modulation method**

SPWM: sinusoidal PWM (default) | SVM: space vector modulation

Continuous pulse width modulation method.

#### **Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Wave-sampling method. The sampling mode determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

#### **Switching frequency (Hz) — Switching rate**

1e3 (default) | positive scalar

Rate at which the switches in the power converter switch, in Hz.

## **Version History**

Introduced in R2021a

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

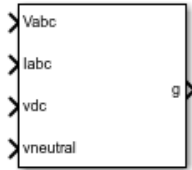
#### **Blocks**

PWM Gate Signal Generator (Five-phase, Two-level) | PWM Generator (Five-phase, Two-level)

# PWM Generator (Vienna Rectifier)

Vienna rectifier pulse-width modulation waveform generator

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

The PWM Generator (Vienna Rectifier) block:

- 1 Calculates on-gating and off-gating times based on the block inputs:
  - Three sinusoidal reference voltages, one per phase, that must be generated at the AC side of the Vienna rectifier, one per phase
  - Three filtered input sinusoidal current measurements of the Vienna rectifier, one per phase
  - A DC-link voltage
  - A DC-link neutral point balance control signal
- 2 Uses the gating times to generate three switch-controlling pulses.

The block uses a simplified space vector modulation method based on the equivalence between two-level and three-level converters to generate the control pulses [1].

The three-phase filtered input currents are used to identify the active voltage vector hexagons. The switching sequence selection and on-gating and off-gating times are determined by simplifying the space-vector diagram of a three-level inverter into that of a two-level inverter [2].

The DC link neutral point balance control signal that comes from the external DC link neutral control loop is used to adjust the time distribution of the redundant voltage vectors within one single switching period.

## Sampling Mode

The block allows you to choose natural, symmetric, or asymmetric sampling of the modulation wave.

The PWM Generator (Vienna Rectifier) block does not perform carrier-based pulse-width modulation (PWM). Instead, the block uses input signals to calculate gating times and uses the gating times to generate the switch-controlling pulses.

For more information about the sampling modes, see PWM Generator (Three-phase, Two-level).

## Ports

### Input

#### **Vabc — Three-phase sinusoidal reference signal**

vector

Three sinusoidal voltages, one per phase.

#### **Iabc — Three-phase current reference signal**

vector

Three currents, one per phase.

#### **vdc — DC link voltage signal**

positive scalar

DC link voltage of the converter. The value of the signal must be greater than 0.

#### **vneutral — DC link neutral point balance control**

scalar

Output of the feedback-control loop that balances the DC supply. The value of the signal must be between  $-1$  and  $+1$ .

### Output

#### **g — Gate control**

vector

Three-pulse waveforms that determine the switching behavior in the attached power converter.

## Parameters

#### **Sampling mode — Wave sampling method**

Natural (default) | Asymmetric | Symmetric

Sampling mode that determines whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

#### **Switching frequency (Hz) — Switching rate**

1e3 Hz (default) | positive number

Rate at which the switches in the power converter switch, in Hz.

#### **Sample time (s) — Block sample time**

5e-5 s (default) | positive number

Time interval between successive block executions, in s. To ensure adequate resolution in the generated signal, set this value to be less than or equal to  $1/(50 \cdot F_{sw})$ , where  $F_{sw}$  is the **Switching frequency (Hz)**.

## Version History

Introduced in R2019b

## References

- [1] R. Burgos, R. Lai, Y. Pei, F. Wang, D. Boroyevich, and J. Pou, "Space vector modulation for Vienna-type rectifiers based on the equivalence between two- and three-level converters: A carrier-based implementation", *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 1888-1898, Jul. 2008
- [2] J. H. Seo, C. H. Choi, and D. S. Hyun, "A new simplified space-vector PWM method for three-level inverters", *IEEE Trans. Power Electron.*, vol. 16, no. 4, pp. 545-550, Jul. 2001.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

### See Also

PWM Generator | PWM Generator (Three-phase, Two-level) | PWM Generator (Three-phase, Three-level)

# Quadrature Shaft Decoder

Decode incremental shaft encoder output

**Library:** Simscape / Electrical / Control / Observers



## Description

The Quadrature Shaft Decoder block measures the shaft rotation from an encoder block, such as the Incremental Shaft Encoder, and outputs the velocity and position.

The block counts the transitions of a pair of digital signals, *A* and *B*, that are positioned 90° out of phase. The block determines the direction and relative position by counting only one of the two signals changes per tick. The angular velocity is obtained by measuring the frequency of either signal.

## Ports

### Input

#### **A — Digital signal A**

scalar

Simulink digital signal *A*, specified as a scalar, which is 90° out-of-phase with digital signal *B*.

Data Types: `single` | `double`

#### **B — Digital signal B**

scalar

Simulink digital signal *B*, specified as a scalar, which is 90° out-of-phase with digital signal *A*.

Data Types: `single` | `double`

#### **Z — Index pulse**

scalar

Simulink signal, specified as a scalar, associated with the decoder index pulse that is used to reset the position calculation.

Data Types: `single` | `double`

### Output

#### **Velocity — Shaft rotational velocity**

scalar

Simulink signal, specified as a scalar, associated with the shaft rotational velocity.



Data Types: single | double

### **Position — Shaft angle of rotation**

scalar

Simulink signal, specified as a scalar, associated with the shaft angle of rotation.

Data Types: single | double

## **Parameters**

### **Pulses per revolution — Pulses per revolution**

4 (default) | integer > 0

Number of pulses per revolution of the encoder shaft.

### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time between consecutive block executions, in s. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, you can set this block to inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly by using a positive scalar.

## **Version History**

**Introduced in R2019b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

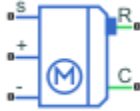
## **See Also**

Incremental Shaft Encoder

## RC Servo

Radio control servomotor with PWM-based angular position tracking and fault modeling

**Library:** Simscape / Electrical / Electromechanical / Brushed Motors



### Description

The RC Servo block 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.

The RC Servo block models the following effects:

- Torque-speed behavior based on DC motor equations
- Position tracking based on the input PWM signal pulse width
- Internal gear reduction ratio, including associated friction losses
- Mechanical end stops, to prevent the output shaft being driven out of range by the load
- Position measurement error
- Fault modeling

The motor equations are the same as those used by the DC Motor block, except that the inductance is not modeled. The RC Servo block determines the equation parameters using the stall torque and no-load speeds, and makes a correction to take account of the backdrive torque.

### Faults

The RC Servo block allows you to model several types of faults:

- Fail off — No electrical torque.
- Fail forward — Rotates in a positive direction to hit the upper end stop.
- Fail reverse — Rotates in a negative direction to hit the lower end stop.
- Failed winding — Torque is applied only if the motor rotor lines up with one of the two remaining functioning windings.

The block can trigger fault events:

- At a specific time
- When a current limit is exceeded for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault takes precedence. In other words, component fails no more than once per simulation.

You can choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

- This block has no optional thermal port.
- If you simulate the model with a fixed-step solver, for example, using a local solver, the step size must be small enough to get the required resolution of the input pulse width. MathWorks recommends that you use this block with variable step solvers for fast desktop simulation.

## Ports

### Conserving

#### **s – Shaft control**

electrical

Electrical conserving port associated with the PWM control signal. The output shaft angle demand is set by the pulse width of the voltage applied to this port.

#### **+ – Positive terminal**

electrical

Electrical conserving port associated with the motor positive terminal.

#### **- – Negative terminal**

electrical

Electrical conserving port associated with the motor negative terminal.

#### **R – Rotor**

mechanical rotational

Mechanical rotational conserving port associated with the rotor.

#### **C – Casing**

mechanical rotational

Mechanical rotational conserving port associated with the stator (casing).

## Parameters

### Electrical Torque

#### Stall torque — Maximum load torque

2.4 kg\*cm (default)

Maximum load torque that the RC servo can move without stalling (stopping).

#### Time to travel 60 degrees (no load) — Output shaft 60 degrees turn time

0.28 s (default)

The time for the output shaft to turn 60 degrees when the motor is driving no load.

#### Corresponding nominal voltage — DC supply voltage

4.8 V (default)

The DC supply voltage used when measuring stall torque and time to travel 60 degrees.

#### Rotational range — Output shaft angular range

[0 180] deg (default)

The output shaft angular range of the RC servo.

#### Corresponding pulse widths — Input pulse widths range

[550 2330] us (default)

The input pulse widths corresponding to the minimum and maximum output angles, as defined by the **Rotational range** parameter. Pulse widths outside of this range are clipped by the block to stay within this range.

### Control

#### Pulse threshold — Threshold voltage for pulse control

3 V (default)

The input pulse is detected as high when the voltage between the s and - ports is above this level.

#### Signal input resistance — Impedance

18 kOhm (default)

The electrical impedance measured between the s and - ports.

#### Angle resolution — Minimum error allowed between the demanded and measured output shaft angle

0.1 deg (default)

When the error between the demanded output shaft angle and measured output shaft angle drops below the angle resolution, the motor is powered off. This parameter models the hysteresis usually incorporated into an RC servo controller to prevent chatter around a set point.

#### Angle measurement error — Model error in angle measurement

0 deg (default)

This parameter allows you to model an angle measurement error, such as can happen due to a failing potentiometer angle sensor. For example, if you want to model the motor being powered against one of the hard stops, you can set a suitable angle measurement error to achieve this.

## Mechanical

**Backdrive torque (unpowered) — Load torque required to backdrive unpowered motor**  
0.5 kg\*cm (default)

Load torque required to backdrive the motor when it is unpowered. The block uses this value to determine the gear friction parameters.

**Gear reduction ratio — Approximate reduction ratio from motor shaft to RC servo output shaft**  
320 (default)

Reduction ratio from the DC motor shaft to the RC servo output shaft. This parameter affects only the impact of rotor inertia on equivalent inertia value at the output shaft. It has no impact on no-load speed. Therefore, the value does not have to be precise.

**Rotor inertia — Inertia of DC motor and rotor gearing**  
10 g\*cm<sup>2</sup> (default)

Inertia of the DC motor, plus inertia of the gearing reflected to the rotor (typically small if the gears are plastic).

**End-stop angles — Mechanical angle range**  
[-5 185] deg (default)

Mechanical end stops prevent rotation of the output shaft beyond the specified range. The range specified by the end stop angles must be larger than that specified by the **Rotational range** parameter.

**End-stop stiffness — Stiffness of mechanical end stops**  
1e6 N\*m/rad (default)

Stiffness of mechanical end stops.

**End-stop damping — Damping of mechanical end stops**  
0.01 N\*m/(rad/s) (default)

Damping of mechanical end stops.

## Faults

**Enable faults — Select Yes to enable faults modeling**  
No (default) | Yes

Select **Yes** to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

**Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs**  
None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- **None** — The block does not issue an assertion.
- **Warn** — The block issues a warning.
- **Error** — Simulation stops with an error.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Failure mode — Select the type of fault**

Fail off (default) | Fail forward | Fail reverse | Failed winding

Select the type of fault:

- **Fail off** — No electrical torque.
- **Fail forward** — Rotates in a positive direction to hit the upper end stop.
- **Fail reverse** — Rotates in a negative direction to hit the lower end stop.
- **Failed winding** — Torque is applied only if the motor rotor lines up with one of the two remaining functioning windings.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Enable temporal fault trigger — Select Yes to enable time-based fault triggering**

No (default) | Yes

Select Yes to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Simulation time for a fault event — Time before entering faulted state**

1 s (default)

Set the simulation time at which you want the block to enter the faulted state.

**Dependencies**

Enabled when the **Enable temporal fault trigger** parameter is set to Yes.

**Enable behavioral fault trigger — Select Yes to enable behavioral fault triggering**

No (default) | Yes

Select Yes to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Maximum permissible current — Current threshold to fault transition**

1 A (default)

Specify the maximum permissible current value. If the current exceeds this value for longer than the **Time to fail when exceeding maximum permissible current** parameter value, then the block enters the faulted state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Time to fail when exceeding current — Maximum length of time the current exceeds the threshold**

1 s (default)

Set the maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

## Version History

Introduced in R2017b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

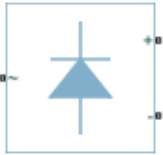
**See Also**

DC Motor

# Rectifier (Three-Phase)

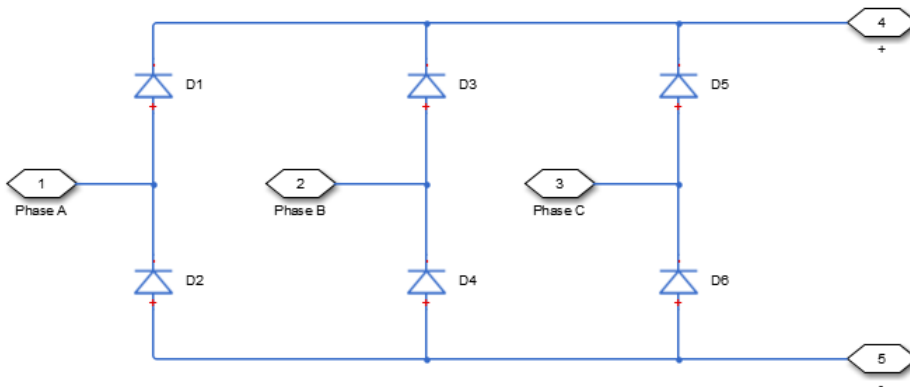
Uncontrolled three-phase AC to DC voltage

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Rectifier (Three-Phase) block models a three-arm diode bridge circuit that converts a three-phase AC voltage to a DC voltage. The figure shows the equivalent circuit for the three-arm diode bridge.



Using the Charge Dynamics tab of the block dialog box, you can choose the type of diode that the three-arm bridge circuit uses. The table shows you how to set the **Model dynamics** parameter based on your goals.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	No dynamics	Each arm of the bridge circuit uses a copy of the Diode block. The block dialog box does not display additional parameters.
Precisely specify reverse-mode charge dynamics.	Model charge dynamics	Each arm of the bridge circuit uses a copy of the commutation model of the Diode block. The block dialog box shows parameters relating to the commutation model of the block.



## Ports

### Conserving

#### ~ – Three-phase port

electrical

Expandable three-phase port.

#### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### a – a-phase

electrical

Electrical conserving port associated with *a*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### b – b-phase

electrical

Electrical conserving port associated with *b*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### c – c-phase

electrical

Electrical conserving port associated with *c*-phase.

#### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### + – Positive terminal

electrical

Electrical conserving port associated with the positive terminal

#### - – Negative terminal

electrical

Electrical conserving port associated with the negative terminal

## Parameters

### Main

#### Electrical connection – Electrical connection

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Forward voltage — Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - ports of each diode for the gradient of the diode i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the forward voltage for each diode.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of each reverse-biased diode.

**Charge Dynamics****Model dynamics — Diode charge dynamics**

No dynamics (default) | Model charge dynamics

Diode charge dynamics. The default value is No dynamics.

The charge dynamics options you can select are:

- No dynamics
- Model charge dynamics

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to Model charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to Model charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics`.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** 

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics`.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is `Specify reverse recovery time directly`.

If you select `Specify stretch factor` or `Specify reverse recovery charge`, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics`.

**Reverse recovery time,  $t_{rr}$  — Reverse recovery time**

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

**Dependencies**

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics` and the **Reverse recovery time parameterization** parameter is set to `Specify reverse recovery time directly`.

**Reverse recovery time stretch factor — Reverse recovery time stretch factor**

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

### Dependencies

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics` and the **Reverse recovery time parameterization** parameter is set to `Specify stretch factor`.

### Reverse recovery charge, $Q_{rr}$ — Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

### Dependencies

This parameter is visible only when the **Model dynamics** parameter is set to `Model charge dynamics` and the **Reverse recovery time parameterization** parameter is set to `Specify reverse recovery charge`.

For more information on these parameters, see `Diode`.

## Version History

### Introduced in R2013b

### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

`Converter (Three-Phase)` | `Average-Value Inverter (Three-Phase)` | `Average-Value Rectifier (Three-Phase)`

**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Resistor

Resistor including optional tolerance, operational limits, fault behavior, and noise

**Library:** Simscape / Electrical / Passive



## Description

The Resistor block represents a linear resistor, while letting you model the following effects:

- “Tolerances” on page 1-1648
- “Operating Limits” on page 1-1649
- “Faults” on page 1-1649
- “Thermal Noise” on page 1-1649
- “Model Thermal Effects” on page 1-1650

You can turn these modeling options on and off independently of each other. When all the additional options are turned off, the component behavior is identical to the Simscape Foundation library Resistor block.

In its simplest form, the Resistor block models a linear resistor, described with the following equation:

$$i = v/R$$

where:

- $i$  is current.
- $v$  is voltage.
- $R$  is resistance.

If you set the **Noise mode** parameter to Enabled, then the defining equations are augmented by a discrete variable  $i_N$  to represent thermal noise, as described in “Thermal Noise” on page 1-1649.

## Tolerances

You can apply tolerances to the nominal value you provide for the **Resistance** parameter. Datasheets typically provide a tolerance percentage for a given resistor type. The table shows how the block applies tolerances and calculates resistance based on the selected **Tolerance application** option.

Option	Resistance Value
None – use nominal value	$R$
Random tolerance	Uniform distribution: $R \cdot (1 - tol + 2 \cdot tol \cdot rand)$ Gaussian distribution: $R \cdot (1 + tol \cdot randn / nSigma)$

Option	Resistance Value
Apply maximum tolerance value	$R \cdot (1 + tol)$
Apply minimum tolerance value	$R \cdot (1 - tol)$

In the table,

- $R$  is the **Resistance** parameter value, nominal resistance.
- $tol$  is fractional tolerance, calculated from the percent-based **Tolerance (%)** parameter.
- $nSigma$  is the value you provide for the **Number of standard deviations for quoted tolerance** parameter.
- `rand` and `randn` are standard MATLAB functions for generating uniform and normal distribution random numbers.

---

**Note** If you choose the `Random tolerance` option and you are in "Fast Restart" mode, the random tolerance value is updated on every simulation if at least one between the fractional tolerance,  $tol$ , or the **Number of standard deviations for quoted tolerance**,  $nSigma$ , is set to Run-time and is defined with a variable (even if you do not modify that variable).

---

## Operating Limits

You can specify operating limits in terms of power and maximum working voltage. If you set the **Modeling option** parameter to `Show thermal port` (see "Model Thermal Effects" on page 1-1650), you can also specify operating limits in terms of temperature.

When an operating limit is exceeded, the block can either generate a warning or stop the simulation with an error. For more information, see the "Operating Limits" on page 1-1653 parameters section.

## Faults

The Resistor block allows you to model an electrical fault as an instantaneous change in resistance. The block can trigger fault events:

- At a specific time
- When a current limit is exceeded for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault takes precedence. In other words, the component fails no more than once per simulation.

When the resistor fails, its resistance is changed to the value you specify for the **Faulted zero-voltage resistance** parameter. You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

## Thermal Noise

The Resistor block can generate thermal noise current. If you set the **Noise mode** parameter to `Enabled`, then the defining equations are augmented by a discrete variable  $i_N$  to represent thermal noise:

$$i = v/R + i_N$$

If the sampling time is  $h$ , then the thermal noise is given by:

$$i_N = \sqrt{2kT/R} \frac{N(0, 1)}{\sqrt{h}}$$

where:

- $k$  is the Boltzmann constant,  $1.3806504 \times 10^{-23}$  J/K.
- $T$  is the temperature.
- $R$  is the resistance.
- $N$  is a Gaussian random number with zero mean and standard deviation of one.
- $2kT/R$  is the double-sided thermal noise power distribution (the single-sided equivalent is  $4kT/R$ ).

The block generates Gaussian noise by using the PS Random Number source in the Simscape Foundation library. You can control the random number seed by setting the **Repeatability** parameter:

- **Not repeatable** — Every time you simulate your model, the block resets the random seed using the MATLAB random number generator:

```
seed = randi(2^32-1);
```

- **Repeatable** — The block automatically generates a seed value and stores it inside the block, to always start the simulation with the same random number. This auto-generated seed value is set when you add a Resistor block from the block library to the model. When you make a new copy of the Resistor block from an existing one in a model, a new seed value is generated. The block sets the value using the MATLAB random number generator command shown above.
- **Specify seed** — If you select this option, the additional **Seed** parameter lets you directly specify the random number seed value.

### Model Thermal Effects

You can expose thermal ports to specify how the resistance value changes with temperature and to set the thermal mass. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains one thermal conserving port.

Use the **Variables** tab to set the initial temperature target.

With the thermal port exposed, the generated noise uses the temperature at the thermal port when determining the instantaneous noise value. Exposing the thermal port also extends the options on the **Operating Limits** tab as follows:

- The **Power rating** parameter becomes temperature dependent. You define a temperature up to which the full power rating is available, plus a higher temperature for which the power rating is reduced to zero. It is assumed that the power rating decreases linearly with temperature between these two values.
- An additional parameter, **Operating temperature range, [Tmin Tmax]**, lets you define the valid temperature range for block operation.



## Plotting and Display Option

If you expose the thermal port, you can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select **Basic characteristics** to plot the resistance, measured in SI units, in a MATLAB figure window using the current resistor parameters.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

This section appears only for the blocks with exposed thermal port. The **Temperature** variable lets you specify a high-priority target for the temperature at the start of simulation.

## Basic Assumptions and Limitations

Simulating with noise enabled slows down simulation. Choose the sample time ( $h$ ) so that noise is generated only at frequencies of interest, and not higher.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the resistor positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the resistor negative terminal.

#### H – Resistor thermal mass

thermal

Thermal conserving port that represents the resistor thermal mass.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and specify how the resistance value changes with temperature and to set the thermal mass.

## Main

### Resistance — Nominal resistance value

1 Ohm (default)

The nominal resistance value. Resistance value must be greater than zero.

### Tolerance (%) — Resistor tolerance, in percent

5 (default)

The resistor tolerance as defined on the manufacturer datasheet.

### Tolerance application — Select how to apply tolerance during simulation

None — use nominal value (default) | Random tolerance | Apply maximum tolerance value | Apply minimum tolerance value

Select how to apply tolerance during simulation:

- **None — use nominal value** — The block does not apply tolerance, uses the nominal resistance value. This is the default.
- **Random tolerance** — The block applies random offset to the resistance value, within the tolerance value limit. You can choose Uniform or Gaussian distribution for calculating the random number by using the **Tolerance distribution** parameter.
- **Apply maximum tolerance value** — The resistance is increased by the specified tolerance percent value.
- **Apply minimum tolerance value** — The resistance is decreased by the specified tolerance percent value.

### Tolerance distribution — Select the distribution type

Uniform (default) | Gaussian

Select the distribution type for random tolerance:

- **Uniform** — Uniform distribution
- **Gaussian** — Gaussian distribution

## Dependencies

Enabled when the **Tolerance application** parameter is set to **Random tolerance**.

### Number of standard deviations for quoted tolerance — Used for calculating the Gaussian random number

4 (default)

Number of standard deviations for calculating the Gaussian random number.

## Dependencies

Enabled when the **Tolerance distribution** parameter is set to **Gaussian**.

## Operating Limits

### Enable operating limits — Select Yes to enable reporting when the operational limits are exceeded

No (default) | Yes

Select **Yes** to enable reporting when the operational limits are exceeded. The associated parameters in the **Operating Limits** section become visible to let you select the reporting method and specify the operating limits in terms of power and maximum working voltage. Parameters that specify operating limits in terms of temperature are visible only for blocks with an exposed thermal port (see “Model Thermal Effects” on page 1-1650). The default value is **No**.

### Reporting when operating limits exceeded — Select the reporting method

Warn (default) | Error

Select what happens when an operating limit is exceeded:

- **Warn** — The block issues a warning.
- **Error** — Simulation stops with an error.

### Dependencies

Enabled when the **Enable operating limits** parameter is set to **Yes**.

### Maximum working voltage — Maximum voltage allowed for normal block operation

100 V (default)

Maximum voltage magnitude allowed for normal block operation.

### Dependencies

Enabled when the **Enable operating limits** parameter is set to **Yes**.

### Power rating — Maximum power allowed for normal block operation

1 W (default)

Maximum power allowed for normal block operation.

If you expose the thermal port of the block, this parameter becomes temperature dependent. The value you specify for the **Power rating** parameter applies up to the temperature specified by the **Temperature below which full power rating is available** parameter value. Then the power rating decreases linearly with temperature, until it becomes 0 at temperature specified by the **Temperature above which power rating is reduced to zero** parameter value.

### Dependencies

Enabled when the **Enable operating limits** parameter is set to **Yes**.

### Temperature below which full power rating is available — Maximum temperature where full power rating still applies

70 °C (default)

Maximum temperature where full power rating, specified by the **Power rating** parameter value, still applies.

**Dependencies**

To enable this parameter, set **Modeling option** to Show thermal port.

**Temperature above which power rating is reduced to zero — Temperature where power rating becomes 0**

155 °C (default)

Temperature where power rating becomes 0. Above this temperature, the simulation always issues an assertion regardless of dissipated power. This parameter value must be higher than **Temperature below which full power rating is available**.

**Dependencies**

To enable this parameter, set **Modeling option** to Show thermal port.

**Operating temperature range, [Tmin Tmax] — Minimum and maximum temperature values allowed for normal block operation**

[-50 150] °C (default)

A row vector of length 2 specifying minimum and maximum temperature values allowed for normal block operation. The first element is the lowest allowable operating temperature, and the second element is the largest allowable operating temperature.

**Dependencies**

To enable this parameter, set **Modeling option** to Show thermal port.

**Faults****Enable faults — Select Yes to enable faults modeling**

No (default) | Yes

Select Yes to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

**Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs**

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Faulted resistance — Resistance when block is in faulted state**

inf Ohm (default)

Resistance between the + and - ports when the block is in the faulted state.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Enable temporal fault trigger – Select Yes to enable time-based fault triggering**

No (default) | Yes

Select Yes to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Simulation time for fault event – Time before entering faulted state**

1 s (default)

Set the simulation time at which you want the block to enter the faulted state.

**Dependencies**

Enabled when the **Enable temporal fault trigger** parameter is set to Yes.

**Enable behavioral fault trigger – Select Yes to enable behavioral fault triggering**

No (default) | Yes

Select Yes to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

Enabled when the **Enable faults** parameter is set to Yes.

**Maximum permissible current – Current threshold to fault transition**

1 A (default)

Specify the maximum permissible current value. If the current exceeds this value for longer than the **Time to fail when exceeding maximum permissible current** parameter value, then the block enters the faulted state.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

**Time to fail when exceeding maximum permissible current – Maximum length of time the current exceeds the threshold**

1 s (default)

Set the maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

Enabled when the **Enable behavioral fault trigger** parameter is set to Yes.

## Noise

### Noise mode — Select whether to model thermal noise current

Disabled (default) | Enabled

Select whether to model thermal noise current:

- **Disabled** — No noise is produced by the resistor.
- **Enabled** — Resistor generates thermal noise current, and the associated parameters become visible in the **Noise** section.

### Sample time — Rate at which the noise source is sampled

1e-3 s (default)

Defines the rate at which the noise source is sampled. Choose it to reflect the frequencies of interest in your model. Making the sample time too small will unnecessarily slow down your simulation.

#### Dependencies

Enabled when the **Noise mode** parameter is set to Enabled.

### Repeatability — Select the noise control option

Not repeatable (default) | Repeatable | Specify seed

Select the noise control option:

- **Not repeatable** — The random sequence used for noise generation is not repeatable.
- **Repeatable** — The random sequence used for noise generation is repeatable, with a system-generated seed.
- **Specify seed** — The random sequence used for noise generation is repeatable, and you control the seed by using the **Seed** parameter.

#### Dependencies

Enabled when the **Noise mode** parameter is set to Enabled.

### Auto-generated seed used for repeatable option — Auto-generated random number seed

random real number

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Seed** parameter.

#### Dependencies

Enabled when the **Repeatability** parameter is set to Repeatable.

### Seed — Random number seed

0 (default)

Seed used by the noise random number generator.

**Dependencies**

Enabled when the **Repeatability** parameter is set to Specify seed.

**Device simulation temperature — Temperature of resistor at the start of the simulation**

25 °C (default)

The temperature of the resistor at the start of the simulation.

**Dependencies**

Enabled when the **Noise mode** parameter is set to Enabled.

For blocks with an exposed thermal port, this parameter is disabled. Instead, use the **Variables** tab to set the initial temperature target. For more information, see “Variables” on page 1-1651.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Resistance temperature coefficient — Specifies how the resistance value changes with temperature**

0.00393 1/K (default)

The coefficient  $\alpha$  in the equation that describes resistance as a function of temperature,  $R_T = R(1 + \alpha(T - T_0))$ . The default value is for copper.

**Measurement temperature — Temperature corresponding to nominal resistance**

25 °C (default)

The temperature  $T_0$ , for which the nominal resistance  $R$  is specified.

**Thermal mass — Thermal mass associated with port H**

100 J/K (default)

Thermal mass associated with the thermal port H. It represents the energy required to raise the temperature of the thermal port by one degree.

**Version History**

Introduced in R2009a

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

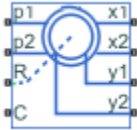
**See Also**

Fault | Resistor

# Resolver

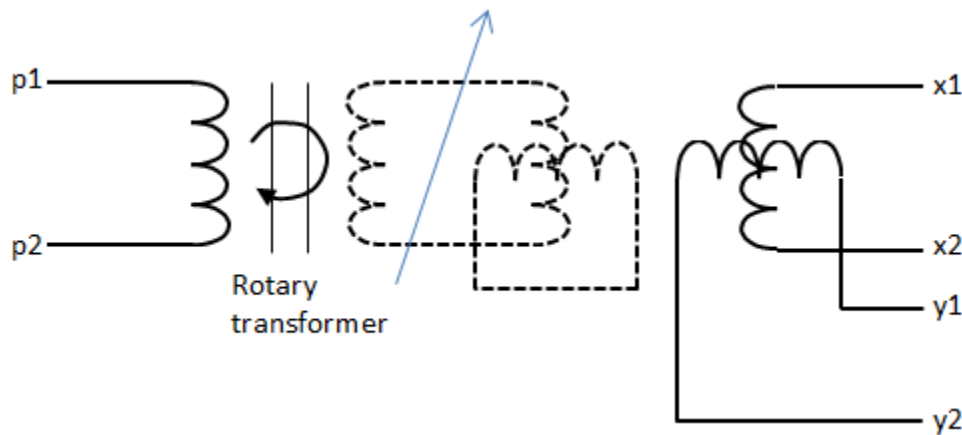
Rotary transformer that measures angle of rotation

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Resolver block models a generic resolver, which measures the electrical phase angle of a signal through electromagnetic coupling. The resolver 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.



Without loss of generality, it is assumed that the transformer between primary and rotor circuit is ideal with a ratio of 1:1. This results in the rotor current and voltage being equivalent to the primary current and voltage.

You have two options for defining the block equations:

- Omit the dynamics by neglecting the transformer inductive terms. This model is only valid if the sensor is driven by a sine wave because any DC component on the primary side will pass to the output side.
- Include the inductive terms, thereby capturing voltage amplitude loss and phase differences. This model is valid for any input waveform. Within this option, you can either specify the inductances and the peak coupling coefficient directly, or specify the transformation ratio and measured impedances, in which case the block uses these values to determine the inductive terms.



### Equations when Omitting Dynamics

The equations are based on the superposition of two ideal transformers, both with coupling coefficients that depend on rotor angle. The two ideal transformers have a common primary winding. See the Simscape Ideal Transformer block reference page for more information on modeling ideal transformers. The equations are:

$$K_x = R \cos(N \Theta)$$

$$K_y = R \sin(N \Theta)$$

$$v_x = K_x v_p$$

$$v_y = K_y v_p$$

$$i_p = -K_x i_x - K_y i_y$$

where:

- $v_p$  and  $i_p$  are the rotor (or equivalently primary) voltage and current, respectively.
- $v_x$  and  $i_x$  are the first secondary voltage and current, respectively.
- $v_y$  and  $i_y$  are the second secondary voltage and current, respectively.
- $K_x$  is the coupling coefficient for the first secondary winding.
- $K_y$  is the coupling coefficient for the second secondary winding.
- $R$  is the transformation ratio.
- $N$  is the number of pole pairs.
- $\Theta$  is the rotor angle.

### Equations when Including Dynamics

The equations are based on the superposition of two mutual inductors, both with coupling coefficients that depend on rotor angle. The two mutual inductors have a common primary winding. See the Simscape Mutual Inductor block reference page for more information on modeling mutual inductors. The equations are:

$$v_p = R_p i_p + L_p \frac{di_p}{dt} + \sqrt{L_p L_s} k \left( \cos(N\theta) \frac{di_x}{dt} + \sin(N\theta) \frac{di_y}{dt} \right)$$

$$v_x = R_s i_x + L_s \frac{di_x}{dt} + \sqrt{L_p L_s} k \cos(N\theta) \frac{di_p}{dt}$$

$$v_y = R_s i_y + L_s \frac{di_y}{dt} + \sqrt{L_p L_s} k \sin(N\theta) \frac{di_p}{dt}$$

where:

- $v_p$  and  $i_p$  are the rotor (or equivalently primary) voltage and current, respectively.
- $v_x$  and  $i_x$  are the first secondary voltage and current, respectively.
- $v_y$  and  $i_y$  are the second secondary voltage and current, respectively.
- $R_p$  is the rotor (or primary) resistance.

- $L_p$  is the rotor (or primary) inductance.
- $R_s$  is the stator (or secondary) resistance.
- $L_s$  is the stator (or secondary) inductance.
- $N$  is the number of pole pairs.
- $k$  is the coefficient of coupling.
- $\theta$  is the rotor angle.

It is assumed that coupling between the two secondary windings is zero.

Datasheets typically do not quote the coefficient of coupling and inductance parameters, but instead give the transformation ratio  $R$  and measured impedances. If you select **Specify transformation ratio and measured impedances** for the **Parameterization** parameter, then the values you provide are used to determine values for the equation coefficients, as defined above.

### **Variables**

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### **Assumptions and Limitations**

- The resolver draws no torque between the mechanical rotational ports **R** and **C**.
- The transformer between primary and rotor circuit is ideal with a ratio of 1:1.
- The coupling between the two secondary windings is zero.

### **Ports**

#### **Conserving**

##### **p1 – Primary winding positive terminal**

electrical

Electrical conserving port associated with the positive terminal of the primary winding.

##### **p2 – Primary winding negative terminal**

electrical

Electrical conserving port associated with the negative terminal of the primary winding.

##### **R – Resolver rotor**

mechanical rotational

Mechanical rotational conserving port connected to the rotor.

##### **C – Resolver case**

mechanical rotational

Mechanical rotational conserving port connected to the resolver case.

**x1 — Secondary winding x positive terminal**

electrical

Electrical conserving port associated with the positive terminal of secondary winding x.

**x2 — Secondary winding x negative terminal**

electrical

Electrical conserving port associated with the negative terminal of secondary winding x.

**y1 — Secondary winding y positive terminal**

electrical

Electrical conserving port associated with the positive terminal of secondary winding y.

**y2 — Secondary winding y negative terminal**

electrical

Electrical conserving port associated with the negative terminal of secondary winding y.

## Parameters

**Parameterization — Resolver parameterization**

Specify transformation ratio and omit dynamics (default) | Specify transformation ratio and measured impedances | Specify equation parameters directly

Select one of the following methods for block parameterization:

- **Specify transformation ratio and omit dynamics** — Provide values for transformation ratio, number of pole pairs, and initial rotor angle only. This model neglects the transformer inductive terms, and is only valid if the sensor is driven by a sine wave. The equations are based on the superposition of two ideal transformers, both with coupling coefficients that depend on rotor angle. For more information, see “Equations when Omitting Dynamics” on page 1-1659.
- **Specify transformation ratio and measured impedances** — Provide additional values to determine the transformer inductive terms, to model the voltage amplitude loss and phase differences. This model is valid for any input waveform. The equations are based on the superposition of two mutual inductors, both with coupling coefficients that depend on rotor angle. For more information, see “Equations when Including Dynamics” on page 1-1659.
- **Specify equation parameters directly** — Model the dynamics, but provide values for rotor and stator inductances and the peak coefficient of coupling, instead of transformation ratio and measured impedances. For more information, see “Equations when Including Dynamics” on page 1-1659. This model is valid for any input waveform.

**Transformation ratio — Peak output to input voltage ratio**

0.5 (default) | positive number

Ratio between the peak output voltage and the peak input voltage assuming negligible secondary voltage drop due to resistance and inductance.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to Specify transformation ratio and omit dynamics or Specify transformation ratio and measured

impedances. If you select **Specify transformation ratio and measured impedances** for the **Parameterization** parameter, then the transformation ratio takes the voltage drop due to primary winding resistance into account.

**Rotor resistance — Primary resistance**

70 Ohm (default) | positive number

Rotor ohmic resistance. This resistance is also referred to as the primary resistance.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to **Specify transformation ratio and measured impedances** or **Specify equation parameters directly**.

**Stator resistance — Secondary resistance**

180 Ohm (default) | positive number

Stator ohmic resistance. This resistance is also referred to as the secondary resistance. It is assumed that both secondaries have the same resistance.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to **Specify transformation ratio and measured impedances** or **Specify equation parameters directly**.

**Rotor reactance — Primary reactance**

100 Ohm (default) | positive number

Rotor reactance when the secondary windings are open-circuit. This reactance is also referred to as the primary reactance.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to **Specify transformation ratio and measured impedances**.

**Stator reactance — Secondary reactance**

300 Ohm (default) | positive number

Stator reactance when the primary winding is open-circuit. This reactance is also referred to as the secondary reactance.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to **Specify transformation ratio and measured impedances**.

**Frequency at which reactances and transformation ratio are specified — Sinusoidal source frequency**

10 kHz (default) | positive number

Frequency of the sinusoidal source used when measuring the reactances.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to **Specify transformation ratio and measured impedances**.

**Rotor inductance — Primary reactance**

0.0016 H (default) | positive number

Rotor or primary inductance,  $L_p$ .

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to Specify equation parameters directly.

**Stator inductance — Secondary reactance**

0.0048 H (default) | positive number

Stator or secondary inductance,  $L_s$ .

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to Specify equation parameters directly.

**Peak coefficient of coupling — Maximum coupling coefficient**

0.35 (default) | number between zero and one, exclusive

Peak coefficient of coupling between the primary and secondary windings.

**Dependencies**

To enable this parameter, set the **Parameterization** parameter to Specify equation parameters directly.

**Number of pole pairs — Rotor pole pairs**

1 (default) | positive number

Number of pole pairs on the rotor.

## Version History

Introduced in R2017b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

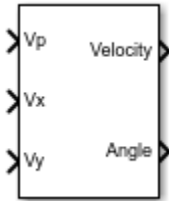
**See Also**

Incremental Shaft Encoder

# Resolver-to-Digital Converter

Resolver-to-digital converter

**Library:** Simscape / Electrical / Control / Observers



## Description

The Resolver-to-Digital Converter block models a transducer that converts the angular position or velocity of a rotating shaft to an electrical signal. Resolver-to-digital converters are commonly used in harsh, rugged environments, such as in fully electric vehicles.

The converted signal is proportional to the sine or cosine of the shaft angle.

A resolver sensor has one rotor winding with the exciter sine wave that is AC-coupled to two stator windings. The stator windings, a sine coil and a cosine coil, are mechanically positioned 90-degrees out-of-phase. As the rotor spins, the rotor position angle changes with respect to the stator windings. The resulting amplitude-modulated signals must then be gained, demodulated and post processed to extract angle and velocity information ([1] and [2]).

## Equations

The block uses a phase-locked loop (PLL) to extract the angle and the velocity of the rotating shaft. The error voltage used by the PI controller is obtained as:

$$V_e = V_y V_p \cos(N\theta) - V_x V_p \sin(N\theta),$$

where:

- $V_p$  is the excitation voltage.
- $V_x$  is the x voltage for the secondary winding of the resolver.
- $V_y$  is the y voltage for the secondary winding of the resolver.
- $N$  is the number of pole pairs for the resolver.
- $\theta$  is the angle.

Therefore, the velocity is obtained as:

$$\omega = K_p V_e + K_i \int V_e,$$

and the angle is computed from the velocity using:

$$\frac{d\theta}{dt} = \omega.$$

## Ports

### Input

#### **Vp — Voltage phase angle**

scalar

Simulink signal that corresponds to the voltage phase angle.

Data Types: single | double

#### **Vx — x-axis voltage**

scalar

Simulink signal that corresponds to the x-axis voltage.

Data Types: single | double

#### **Vy — y-axis voltage**

scalar

Simulink signal that corresponds to the y-axis voltage.

Data Types: single | double

### Output

#### **Velocity — Rotational velocity**

scalar

Simulink signal that corresponds to the rotational velocity.

Data Types: single | double

#### **Angle — Rotational angle**

scalar

Simulink signal that corresponds to the rotational angle.

Data Types: single | double

## Parameters

#### **Number of pole pairs — Pole pairs in the attached machine**

4 (default) | positive scalar integer

Number of pole pairs in the attached machine.

#### **Phase-locked loop proportional gain — PLL proportional gain**

15 (default) | positive scalar

Proportional gain for the phase-locked loop filter. This value determines how aggressively the PLL tracks and locks to the phase angle. Increase this value to more closely track step changes in the phase angle.

#### **Phase-locked loop integral gain — PLL integral gain**

2.5e5 (default) | positive scalar

Integral gain for the phase-locked loop filter. Increase this value to increase the rate at which steady-state error is eliminated in the phase angle. This value also determines how aggressively the PLL tracks and locks to the phase.

**Initial position (rad) — Initial phase angle**

0 rad (default) | scalar

Initial estimate of the phase angle. If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

---

**Note** If this block is in a masked subsystem, or another variant subsystem that allows you to switch between continuous operation or discrete operation, promote the sample time parameter to ensure correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

---

## Version History

Introduced in R2019b

## References

- [1] Santanu Sarma, V.K. Agrawal, Subramanya Udupa. *Software-Based Resolver-to-Digital Conversion Using a DSP*. IEEE Transactions on Industrial Electronics, 55, 371-379 . February 2008. ([https://www.researchgate.net/publication/3219673\\_Software-Based\\_Resolver-to-Digital\\_Conversion\\_Using\\_a\\_DSP](https://www.researchgate.net/publication/3219673_Software-Based_Resolver-to-Digital_Conversion_Using_a_DSP))
- [2] Ankur Verma, Anand Chellamuthu. *Design considerations for resolver-to-digital converters in electric vehicles*. Texas Instruments, Analog Applications Journal. 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

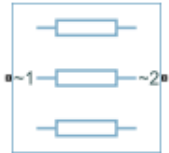
Sinusoidal Measurement (PLL) | Sinusoidal Measurement (PLL, Three-Phase)



## RLC (Three-Phase)

Three-phase impedance

**Library:** Simscape / Electrical / Passive / RLC Assemblies



### Description

The RLC (Three-Phase) block models a three-phase impedance with two three-phase connections. Each of the three identical impedance components can include any combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

Define the values for the R, L, and C components by specifying the appropriate block parameters. Do not set the parameter values to zero or infinity to remove terms; instead, select the correct option for the **Component structure** parameter.

For certain combinations of R, L, and C, for some circuit topologies, specify parasitic resistance or conductance values that help the simulation to converge numerically. These parasitic terms ensure that an inductor has a small parallel resistive path and that a capacitor has a small series resistance.

### Block parameterization

The following two tables list the block parameters for each **Component structure**, based on the selected **Parameterization** option:

- Specify by rated power
- Specify component values directly

Component Structure	Main Parameters	Parasitics Parameters	Initial Conditions Parameters
R	Resistance	None	None
L	Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
C	Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Series RL	Resistance Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
Series RC	Resistance Capacitance	None	Initial capacitor voltage [ Va Vb Vc ]

Component Structure	Main Parameters	Parasitics Parameters	Initial Conditions Parameters
Series LC	Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Series RLC	Resistance Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RL	Resistance Inductance	None	Initial inductor current [ Ia Ib Ic ]
Parallel RC	Resistance Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Parallel LC	Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RLC	Resistance Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]

### Variables

Use the **Variables** tab to specify the priority and initial target values for the block variables before simulation. For more information, see “Set Priority and Initial Target for Block Variables”.

The **Variables** tab is not visible if you set **Component structure** to R.

### Ports

#### Conserving

##### ~1 – Three-phase port

electrical

Expandable three-phase electrical conserving port.

##### ~2 – Three-phase port

electrical

Expandable three-phase electrical conserving port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### Component structure — Desired combination

R (default) | L | C | Series RL | Series RC | Series LC | Series RLC | Parallel RL | Parallel RC | Parallel LC | Parallel RLC

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

#### Resistance — Resistance

1 Ohm (default)

Resistance of each of the line impedances.

#### Dependencies

This parameter is visible only when you select a component structure that includes a resistor.

#### Inductance — Inductance

1e-3 H (default)

Inductance of each of the line impedances. This parameter is visible only when you select a component structure that includes an inductor.

#### Dependencies

This parameter is visible only when you select a component structure that includes an inductor.

#### Capacitance — Capacitance

1e-6 F (default)

Capacitance in each of the line impedances.

#### Dependencies

This parameter is visible only when you select a component structure that includes a capacitor.

### Parasitics

#### Parasitic series resistance — Parasitic series resistance

0 Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load.

**Parasitic parallel conductance — Parasitic parallel conductance**

0 1/Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load.

If you set **Component structure** to either **Series RL**, **Series LC**, or **Series RLC**, the parasitic parallel conductance is connected across the full branch of the block.

## Version History

Introduced in R2013b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Delta-Connected Load | Wye-Connected Load

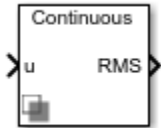
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# RMS Measurement

Calculate root-mean-square (RMS) properties of a signal

**Library:** Simscape / Electrical / Control / Measurements



## Description

The RMS Measurement block measures root-mean-square (RMS) properties of the input signal. You can use it to measure one of these properties:

- The total RMS of the input signal
- The RMS of the individual harmonics of the input signal that you specify.

Use the total RMS configuration with appropriate sensors to perform RMS voltage, current, or power analyses in your system.

You can use the harmonics configuration to perform total harmonic distortion analyses on systems with nonlinear loads such as:

- Converters
- Motor drives
- Inverters

## Equations

The total RMS value is calculated from the input signal  $x_{RMS}$  as:

$$x_{RMS}(t) = \sqrt{\frac{1}{T} \int_{-T}^t x(t)^2 dt},$$

where:

- $T$  is the period of the input signal, or equivalently the inverse of its base frequency  $F$ .
- $x$  is the input signal.

Because the calculation is performed over a period of time, the block requires  $T$  seconds to respond to a step change in the input signal. This condition also applies to startup.

The harmonic RMS component  $x_{k,RMS}$  for harmonic  $k$  is calculated as:

$$x_{k,RMS}(t) = G \left( \frac{2}{T} \right) \sqrt{\left( \int_{-T}^t x(t) \sin\left(\frac{2\pi kt}{T}\right) dt \right)^2 + \left( \int_{-T}^t x(t) \cos\left(\frac{2\pi kt}{T}\right) dt \right)^2},$$

where  $G$  is equal to 0.5 for the DC component ( $k = 0$ ) and  $1/\sqrt{2}$  for the AC components ( $k > 0$ ).

## Ports

### Input

#### **u — Input signal**

scalar

Periodic input signal.

Data Types: `single` | `double`

### Output

#### **RMS — Root-mean-square**

scalar or vector

Estimated RMS of the input signal. If you select **Specify harmonics**, the output is a vector with each element corresponding to a specified harmonic. Otherwise, the output is a scalar representing the total RMS.

Data Types: `single` | `double`

## Parameters

#### **Base frequency (Hz) — Fundamental frequency**

60 Hz (default) | scalar

Base frequency of the input signal corresponding to the first harmonic.

#### **Specify harmonics — RMS output mode**

`off` (default) | `on`

Specify whether to output the total RMS of the input signals, or the individual harmonics that you specify.

#### **Harmonic numbers — Harmonics specification**

[0 1 2] (default) | vector

Specify the harmonics for which to output an RMS.

### Dependencies

To enable this parameter, select the **Specify harmonics** parameter.

#### **Sample time — Block sample time**

0 (default) | positive number

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For continuous operation, set this property to 0. For discrete operation, specify the sample time explicitly as a positive number. This block does not support inherited sample time.

If this block is in a masked subsystem, or other variant subsystem that allows either continuous and discrete operation, promote the sample time parameter. Promoting the sample time parameter

ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2017b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

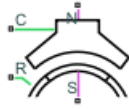
### **Blocks**

Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL)

## Rotating Air Gap

Air gap between stator tooth and rotating permanent magnet rotor

**Library:** Simscape / Electrical / Electromechanical



### Description

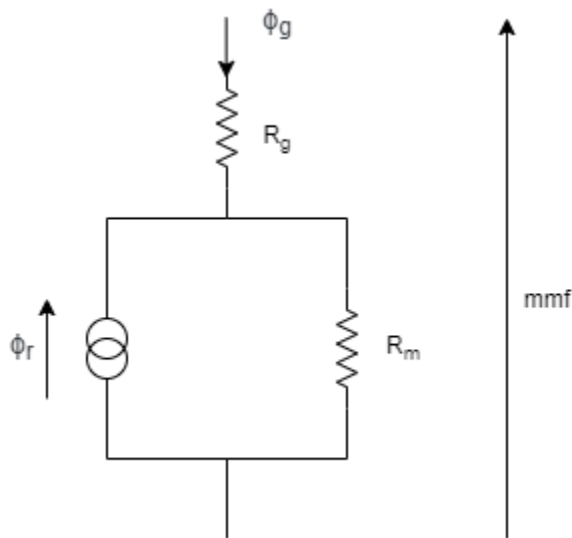
The Rotating Air Gap block models an air gap between a stator tooth and a rotating permanent magnet rotor. This block assumes that the rotor magnets are surface mounted and that the associated induced voltage is sinusoidal.

If the rotor angle is zero, specified by the **Rotor angle** variable in the **Variables** tab, then the rotor magnet perfectly aligns with the middle of the first stator tooth. The permanent magnet is then orientated to oppose the flux flow from port **N** to port **S**.

Use this block to create a magnetic representation of a permanent magnet synchronous motor (PMSM). For example, if you want to model a motor with nine stator poles, create nine copies of this block and set each of the **Stator tooth reference index** parameters to 1, 2, 3, 4, 5, 6, 7, 8, and 9, respectively.

### Equations

This figure shows the equivalent circuit for the air gap and the adjacent permanent magnet



where:

- $\phi_g$  is the magnetic flux that flows from the external magnetic circuit to port **N**.



- $R_g$  is the air gap reluctance.
- $mmf$  is the magnetomotive force across the rotating air gap component.
- $R_m$  is the permanent magnet reluctance.
- $\phi_r$  is the magnetic flux generated by the rotor permanent magnets in the angle range subtended by the stator tooth.

This equation defines the relationship between  $\phi_g$ ,  $mmf$ , and  $\phi_r$ :

$$\phi_g = \frac{mmf - R_m \phi_r}{R_m + R_g}.$$

If the back EMF is sinusoidal, the flux density of the permanent magnet rotor is defined by this equation

$$B_r = B_0 \cos(N\theta_s - N\theta_r)$$

where:

- $N$  is the **Number of rotor pole pairs**.
- $\theta_r$  is the rotor angle.
- $\theta_s$  is the stator angle.
- $B_0$  is the **Peak magnet flux density**, in Tesla.

Then, to obtain the permanent magnet flux linkage, integrate over the stator angle subtended by the stator tooth

$$\phi_r(\theta_r) = rl \int_{-\frac{\theta_{tooth}}{2}}^{\frac{\theta_{tooth}}{2}} [B_0 \cos(N\theta_s - N\theta_r)] d\theta_s$$

where:

- $r$  is the **Rotor radius**.
- $l$  is the **Tooth depth (in direction of shaft)**.

For an ideal PMSM, the  $\theta_{tooth}$  must be equal to  $2\pi/N_s$ , where  $N_s$  is the value of the **Number of stator teeth** parameter. Then the equation of the flux that flows through the equivalent circuit is obtained by solving the integral:

$$\phi_r(\theta_r) = 2B_0lr/Ns \sin\left(\frac{\pi N}{N_s}\right) \cos(N\theta_r).$$

To obtain the torque generated across the air gap, first calculate the total energy stored by the component:

$$E = \frac{1}{2} \phi_g^2 R_g + \frac{1}{2} (\phi_r(\theta_r))^2 R_m.$$

Then, to obtain the torque, differentiate with respect to the rotor angle:

$$\tau = \frac{\partial E}{\partial \theta_r} = -2B_0R_m l r \sin\left(\frac{\pi N}{N_s}\right) \sin(N\theta_r) (\phi_g + \phi_r(\theta_r)) / N.$$

Finally, calculate  $R_g$  and  $R_m$  in terms of geometry:

$$R_g = \frac{g}{\mu_0 A_g}$$
$$R_m = \frac{l_m}{\mu_r \mu_0 A_g}$$

where:

- $\mu_0$  is the permittivity of free space.
- $\mu_r$  is the relative permittivity of the permanent magnet.
- $g$  is the **Air gap**.
- $l_m$  is the magnet length.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **N – Magnetic stator connection**

magnetic

Magnetic conserving port associated with the stator.

#### **S – Magnetic rotor connection**

magnetic

Magnetic conserving port associated with the rotor.

#### **C – Motor case**

mechanical

Mechanical rotational conserving port associated with the motor case.

#### **R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the motor rotor.

## Parameters

#### **Number of rotor pole pairs – Rotor pole pairs**

5 (default) | positive scalar

Number of the pole pairs of the rotor. This parameter must be equal to or greater than 1.

**Number of stator teeth — Number of stator teeth**

9 (default) | positive scalar

Number of teeth of the stator. This parameter must be equal to or greater than 3.

**Stator tooth reference index — Stator tooth reference index**

1 (default) | positive scalar

Reference index of the stator tooth of the motor. This parameter must be between 1 and the value of the **Number of stator teeth** parameter.

For example, if you want to model a motor with nine stator poles, create nine copies of this block and set the **Stator tooth reference index** parameter for each of the Rotating Air Gap blocks to 1, 2, 3, 4, 5, 6, 7, 8, and 9, respectively.

**Peak magnet flux density — Peak flux density of magnet rotor**

0.4 T (default) | positive scalar

Peak flux density associated with the permanent magnet rotor. The flux density is sinusoidal with the rotor angle.

**Permanent magnet length (in direction of flux) — Magnet length in radial machine direction**

5 mm (default) | positive scalar

Length of the magnet in the radial machine direction or, equivalently, in the direction of the magnetic flux. This parameter must be less than the value of the **Rotor radius** parameter.

**Permanent magnet relative permeability — Permanent magnet relative permeability**

1.05 (default) | scalar

Relative permeability of the permanent magnets. Typically, you should set this value a little greater than 1 to reflect that the magnetic dipoles are already aligned in a permanent magnet.

**Air gap — Air gap in radial direction**

1 mm (default) | positive scalar

Length of the air gap in the radial direction.

**Rotor radius — Rotor radius**

65 mm (default) | positive scalar

Radius of the rotor.

**Tooth depth (in direction of shaft) — Stator tooth depth**

50 mm (default) | positive scalar

Length of the stator tooth in the direction of the rotating shaft.

## Version History

Introduced in R2021a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

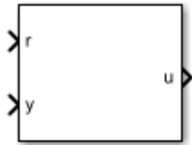
### **See Also**

PMSM

# RST Controller

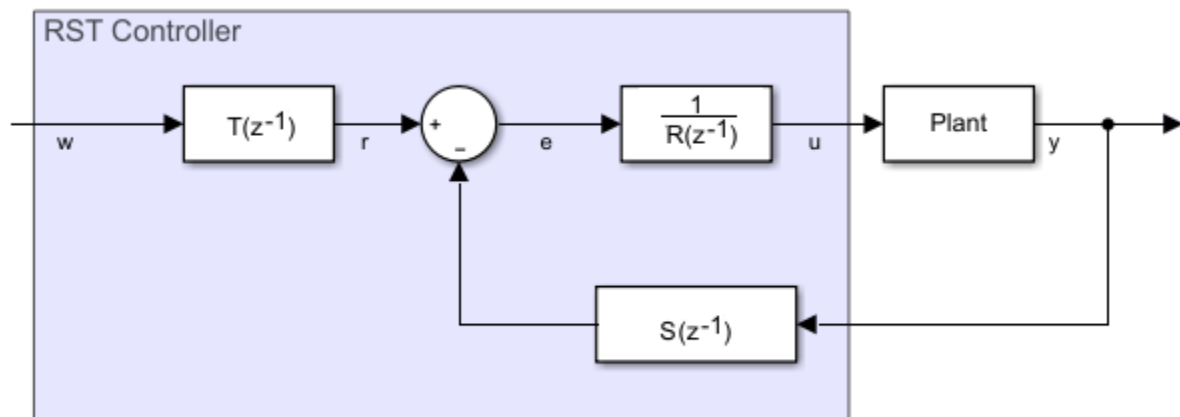
Predictive control using a polynomial representation

**Library:** Simscape / Electrical / Control / General Control



## Description

The RST Controller block implements a generalized predictive controller using a reference signal tracking polynomial representation. The diagram shows the equivalent circuit for the control algorithm.



## Equations

A controlled auto-regressive integrated moving average (CARIMA) model describes the plant:

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{e(k)C(z^{-1})}{D(z^{-1})}$$

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_{n_A}z^{-n_A}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + \dots + b_{n_B}z^{-n_B}$$

$$C(z^{-1}) = 1$$

$$D(z^{-1}) = 1 - z^{-1},$$

where:

- $d$  is the system dead-time.
- $y(k)$  is the plant output.
- $u(k)$  is the controller output.
- $e(k)$  is white noise with a zero-mean value.
- $A(z^{-1})$  and  $B(z^{-1})$  are the system polynomials.
- $n_A$  and  $n_B$  are the polynomials degrees.
- $C(z^{-1})$  and  $D(z^{-1})$  are the disturbance polynomials for obtaining the steady-state error.

The prediction model is given as

$$\widehat{y}(k+j|k) = G_{j-d}(z^{-1})D(z^{-1})z^{-d-1}u(k+j) + \frac{H_{j-d}(z^{-1})D(z^{-1})}{C(z^{-1})}u(k-1) + \frac{F_{j-d}(z^{-1})}{C(z^{-1})}y(k)$$

and

$$j = \overline{hi, hp}$$

where:

- $hi$  is the minimum prediction.
- $hp$  is the prediction horizon.

The future control sequence, computed at time  $k$ , is

$$u(k+j-1|k),$$

where

$$j = \overline{1, hc}$$

and  $hc$  is the control horizon.

The predicted values of the output is

$$\widehat{y}(k+j|k).$$

To determine the system polynomials,  $F_{j-d}(z^{-1})$ ,  $G_{j-d}(z^{-1})$ , and  $H_{j-d}(z^{-1})$ , the block uses two Diophantine equations. The first Diophantine equation is

$$\frac{C(z^{-1})}{A(z^{-1})D(z^{-1})} = E_{j-d}(z^{-1}) + z^{-j+d} \frac{F_{j-d}(z^{-1})}{A(z^{-1})D(z^{-1})},$$

where:

$$E_{j-d}(z^{-1}) = 1 + e_1z^{-1} + \dots + e_{n_E}z^{-n_E}$$

$$F_{j-d}(z^{-1}) = f_0 + f_1z^{-1} + \dots + f_{n_F}z^{-n_F}$$

$$n_E = j - d - 1$$

$$n_F = \max(n_A + n_D - 1, n_C - j + d)$$

The second Diophantine equation is

$$E_{j-d}(z^{-1})B(z^{-1}) = C(z^{-1})G_{j-d}(z^{-1}) + z^{-j+d}H_{j-d}(z^{-1}),$$

where:

$$G_{j-d}(z^{-1}) = g_0 + g_1z^{-1} + \dots + g_{n_G}z^{-n_G}$$

$$H_{j-d}(z^{-1}) = h_0 + h_1z^{-1} + \dots + h_{n_H}z^{-n_H}$$

$$n_G = j - d - 1$$

$$n_H = \max(n_C, n_B + d) - 1$$

The resulting prediction model is

$$\hat{y}(k+j|k) = G_{j-d}(z^{-1})D(z^{-1})z^{-d-1}u(k+j) + \hat{y}_0(k+j|k),$$

where

$$\hat{y}_0(k+j|k) = \frac{H_{j-d}(z^{-1})D(z^{-1})}{C(z^{-1})}u(k-1) + \frac{F_{j-d}(z^{-1})}{C(z^{-1})}y(k)$$

represents the free response of the system.

Using the matrix notation, the prediction model can be written as

$$= Gu_d + o,$$

where:

$$= [\hat{y}(k+hi|k), \hat{y}(k+hi+1|k), \dots, \hat{y}(k+hp|k)]^T$$

$$G = \begin{bmatrix} g_{hi-d-1} & \dots & g_0 & 0 & \dots & 0 \\ g_{hi-d} & \dots & g_1 & g_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ g_{hc-1} & \dots & \dots & \dots & \dots & g_0 \\ g_{hp-d-1} & \dots & \dots & \dots & \dots & g_{hp-hc-1} \end{bmatrix}$$

$$u_d = [D(z^{-1})u(k), \dots, D(z^{-1})u(k+hc-1)]^T$$

$$o = [\hat{y}_0(k+hi|k), \hat{y}_0(k+hi+1|k), \dots, \hat{y}_0(k+hp|k)]^T$$

To minimize tracking error and controller output, the block uses a cost function. To trade off between the minimization of the tracking error and the minimization of the controller output, the block uses a weighting factor,  $\lambda$ , such that

$$J = (Gu_d + o - w)^T(Gu_d + o - w) + \lambda u_d^T u_d$$

for

$$D(z^{-1})u(k+i) = 0$$

and

$$i \in [hc, hp - d - 1],$$

where  $w$  is the reference trajectory vector. Minimizing the cost function, yields the equation for the optimal control sequence:

$$u_d^* = (G^T G + \lambda I_{hc}) G^T [w - 0].$$

As  $\nu_j$  and  $j = \overline{hi, hp}$  are elements in the first row of the matrix  $(G^T G + \lambda I_{hc})^{-1} G^T$ , applying the receding horizon principle yields the control algorithm equation as

$$D(z^{-1})u(k) = \sum_{j=hi}^{hp} \nu_j [w(k+j|k) - \hat{y}_0(k+j|k)].$$

Substitution using  $\hat{y}_0(k+j|k) = \frac{H_{j-d}(z^{-1})D(z^{-1})}{C(z^{-1})}u(k-1) + \frac{F_{j-d}(z^{-1})}{C(z^{-1})}y(k)$  yields this form of the control algorithm equation:

$$\begin{aligned} C(z^{-1})D(z^{-1})u(k) &= - \sum_{j=hi}^{hp} \nu_j H_{j-d}(z^{-1})D(z^{-1})u(k-1) - \sum_{j=hi}^{hp} \nu_j F_{j-d}(z^{-1})y(k) \\ &+ \sum_{j=hi}^{hp} \nu_j C(z^{-1})w(k+j). \end{aligned}$$

The polynomial form of the control algorithm follows as

$$R(z^{-1})u(k) + S(z^{-1})y(k) = T(z^{-1})w(k+hp),$$

where:

$$R(z^{-1}) = \left( C(z^{-1}) + \sum_{j=hi}^{hp} \nu_j z^{-1} H_{j-d}(z^{-1}) \right) D(z^{-1}),$$

$$S(z^{-1}) = \sum_{j=hi}^{hp} \nu_j F_{j-d}(z^{-1}),$$

and

$$T(z^{-1}) = C(z^{-1}) \sum_{j=hi}^{hp} \nu_j z^{-hp+j}.$$

### Limitations

To obtain the  $R$ ,  $S$ , and  $T$  polynomials, use the discrete-time instead of the continuous-time transfer function.



## Ports

### Input

#### **r – Plant reference**

scalar

Plant system reference signal.

Data Types: `single` | `double`

#### **y – Plant output**

scalar

Plant system output signal.

Data Types: `single` | `double`

### Output

#### **u – Controller output**

scalar

Control system output signal.

Data Types: `single` | `double`

## Parameters

### **Controller parameterization – Parameterization method**

`Controller polynomials (default)` | `Generate polynomials`

Method for parameterizing the controller. If you know the discrete-time  $R$ ,  $S$ , and  $T$  polynomial values, select `Controller polynomials`. Otherwise, select `Generate polynomials`.

#### **Dependencies**

Selecting a parameterization method enables other parameters.

### **R polynomial – R polynomial values**

1 (default) | positive, scalar or vector

Vector of the  $R$  polynomials for the RST control.

#### **Dependencies**

Selecting `Controller polynomials` for the **Controller parameterization** parameter enables this parameter.

### **S polynomial – S polynomial values**

1 (default) | positive, scalar or vector

Vector of the  $S$  polynomials for the RST control.

#### **Dependencies**

Selecting `Controller polynomials` for the **Controller parameterization** parameter enables this parameter.

**T polynomial — T polynomial values**

1 (default) | positive, scalar or vector

Vector of the  $T$  polynomials for the RST control.

**Dependencies**

Selecting `Controller` polynomials for the **Controller parameterization** parameter enables this parameter.

**Model discrete transfer function numerator — Transfer function numerator**

1 (default) | scalar or vector

Numerator of the system discretized transfer function. To determine the discrete transfer function, if you have a license for Control System Toolbox™, use the `c2d` function.

**Dependencies**

Selecting `Generate` polynomials for the **Controller parameterization** parameter enables this parameter.

**Model discrete transfer function denominator — Transfer function denominator**

[1 0.5] (default) | vector

Denominator of the system discretized transfer function. To determine the discrete transfer function, if you have a license for Control System Toolbox, use the `c2d` function.

**Dependencies**

Selecting `Generate` polynomials for the **Controller parameterization** parameter enables this parameter.

**Control horizon (samples) — Number of control-horizon samples**

5 (default) | positive integer

Number of samples in the control horizon.

**Dependencies**

Selecting `Generate` polynomials for the **Controller parameterization** parameter enables this parameter.

**Control weighting factor — Weighting factor**

0.5 (default) | positive number

Weighting factor for the RST controller.

**Dependencies**

Selecting `Generate` polynomials for the **Controller parameterization** parameter enables this parameter.

**System dead time (samples) — Number of dead-time samples**

2 (default) | 0 or a positive integer

Number of samples of the dead time.

## Dependencies

Selecting `Generate polynomials` for the **Controller parameterization** parameter enables this parameter.

## Sample time (-1 for inherited) – Sampling interval

-1 (default) | default value or a positive number

Time interval between samples. If the block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

## Version History

Introduced in R2017b

## References

[1] Camacho, E. F. and C. Bordons. *Model Predictive Control*. Second Edition, London: Springer, 2007.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Smith Predictor Controller | State-Feedback Controller

## S-R Latch

Behavioral model of an S-R Latch

**Library:** Simscape / Electrical / Integrated Circuits / Logic



### Description

The S-R Latch block is an abstracted behavioral model of a set-reset latch. It does not model the internal individual MOSFET devices (see “Assumptions and Limitations” on page 1-1687 for details). Therefore, the block runs quickly during simulation but retains the correct I/O behavior.

If the gate voltage is greater than the threshold voltage  $V_{TH}$ , then the input taken is 1 (HIGH). Otherwise, the input is zero (LOW). The gate threshold voltage  $V_{TH}$  is halfway between the **Low level input voltage** ( $V_{IL}$ ) and **High level input voltage** ( $V_{IH}$ ) parameters.

The block output logic level is either HIGH or LOW, according to the logic levels of the gate inputs and the S-R latch truth table.

S	R	Q <sub>n</sub>
0	0	Q <sub>n-1</sub>
0	1	0
1	0	1
1	1	1

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block initial conditions depend on the output model selected:

- For **Linear** model, the high initial condition corresponds to the **High level output voltage** parameter value,  $V_{OH}$ , and the low initial condition corresponds to the **Low level output voltage** parameter value,  $V_{OL}$ .
- For **Quadratic** model, the high initial condition is a value close to the **Supply voltage** parameter value,  $V_{CC}$ , and the low initial condition is close to 0.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.
- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The behavior of this block is abstracted. In particular, response to input noise and inputs that are around the logic threshold voltage can be inaccurate. Also, dynamic response is approximate.
- The linear drop in output voltage as a function of output current is an approximation to the MOSFET or bipolar output behavior.
- Modeling of the output as a controlled voltage source is representative of a totem-pole or push-pull output stage. To model a device with an open-collector:
  - 1 Connect the output pin to the base of an NPN Bipolar Transistor or PNP Bipolar Transistor block.
  - 2 Set the **Output resistance** parameter to a suitable value.

## Ports

### Conserving

#### S — Set pin

electrical

Electrical conserving port associated with the S-R latch set pin.

#### R — Reset pin

electrical

Electrical conserving port associated with the S-R latch reset pin.

#### Q — Output pin

electrical

Electrical conserving port associated with the S-R latch output pin.

## Parameters

### Inputs

#### Low level input voltage — Low level input voltage

2 V (default)

Voltage value less than which the block interprets the input voltage as LOW.

**High level input voltage — High level input voltage**

3 V (default)

Voltage value greater than which the block interprets the input voltage as HIGH.

**Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero can result in faster simulation times.

**Outputs****Output current-voltage relationship — Output current-voltage relationship**

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

**Low level output voltage — Low level output voltage**

0 V (default)

Voltage value at the output when the output logic level is LOW.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**High level output voltage — High level output voltage**

5 V (default)

Voltage value at the output when the output logic level is HIGH.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Output resistance — Output resistance**

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

**Dependencies**

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

**Supply voltage — Supply voltage**

5 V (default)

Supply voltage value applied to the gate in your circuit.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at  $I_{OH}$  — Logic HIGH output resistance at zero current and at  $I_{OH}$**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current  $I_{OH}$  when shorted to ground — Logic HIGH output current  $I_{OH}$  when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at  $I_{OL}$  — Logic LOW output resistance at zero current and at  $I_{OL}$**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub> — Logic LOW output current I<sub>OL</sub> when shorted to V<sub>cc</sub>**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage V<sub>cc</sub>.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2009b



## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS NOT | CMOS OR | CMOS XOR | Schmitt Trigger

# Schmitt Trigger

Behavioral model of Schmitt trigger

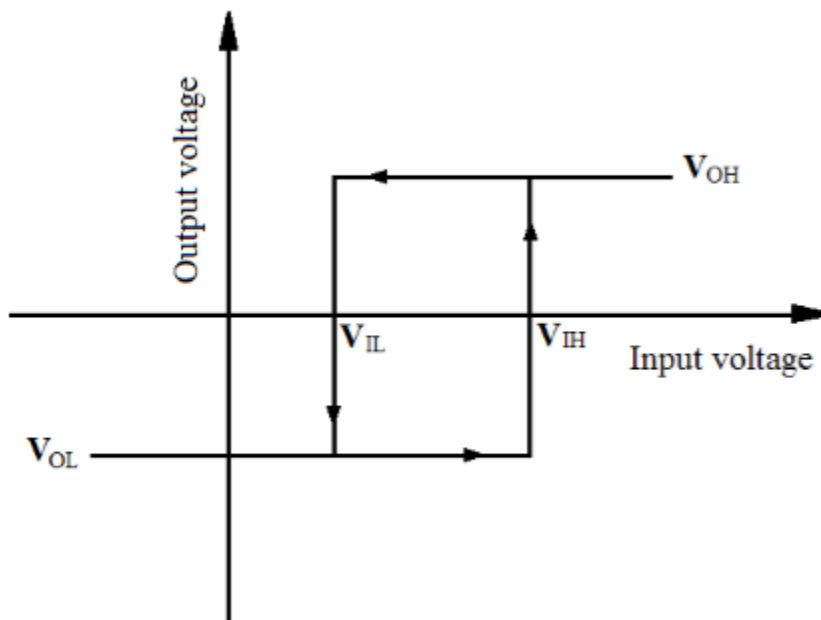
**Library:** Simscape / Electrical / Integrated Circuits / Logic



## Description

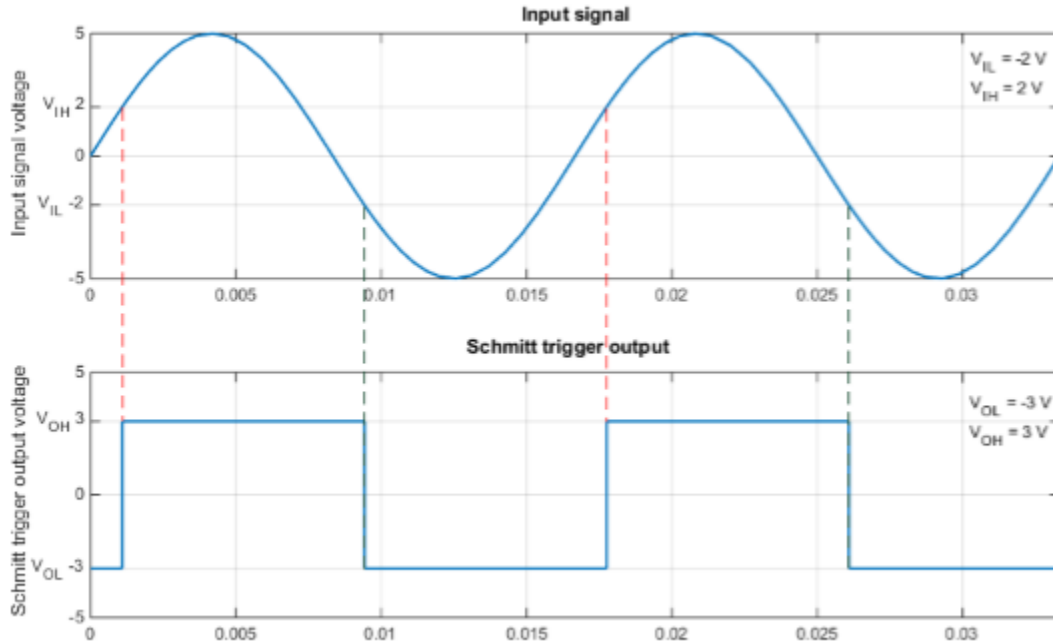
The Schmitt Trigger block implements a behavioral model of Schmitt trigger.

The block output logic level is **HIGH** when the input rises above the **High level input voltage** ( $V_{IH}$ ) value and does not go **LOW** until the input falls below the lower-valued **Low level input voltage** ( $V_{IL}$ ) value. This logic implements a hysteresis characteristic between input and output.



In the graphic,  $V_{OH}$  and  $V_{OL}$  correspond to the **High level output voltage** and **Low level output voltage** values, respectively.

The next figure shows a sample output of the block with parameters  $V_{IH} = 2V$ ,  $V_{IL} = -2V$ ,  $V_{OH} = 3V$ , and  $V_{OL} = -3V$ .



The block determines the logic levels of the gate inputs as follows:

- If the gate voltage is greater than the threshold voltage, the block interprets the input as logic 1.
- Otherwise, the block interprets the input as logic 0.

The threshold voltage is the voltage value at midpoint between the **High level input voltage** parameter value and the **Low level input voltage** parameter value.

---

**Note** To improve simulation speed, the block does not model all the internal individual MOSFET devices that make up the gate. See “Assumptions and Limitations” on page 1-1694 for details.

---

The block models the gate as follows:

- The gate inputs have infinite resistance and finite or zero capacitance.
- The gate output offers a selection of two models: **Linear** and **Quadratic**. For more information, see “Selecting the Output Model for Logic Blocks”. Use the **Output current-voltage relationship** parameter to specify the output model.
- You can specify propagation delay for both output models. For **Linear** output, the block sets the value of the gate output capacitor such that the resistor-capacitor time constant equals the **Propagation delay** parameter value. For **Quadratic** output, the gate input demand is lagged to approximate the **Propagation delay** parameter value.

The block output voltage depends on the output model selected:

- For **Linear** model, output high is the **High level output voltage** parameter value, and output low is the **Low level output voltage** parameter value.

- For **Quadratic** model, the output voltage for High and Low states is a function of the output current, as explained in “Quadratic Model Output and Parameters”. For zero load current, output high is  $V_{cc}$  (the **Supply voltage** parameter value), and output low is zero volts.

## Assumptions and Limitations

The block does not model the internal individual MOSFET devices that make up the gate (except for the final MOSFET pair if you select the **Quadratic** option for the **Output current-voltage relationship** parameter). This limitation has the following implications:

- The block does not accurately model the gate's response to input noise and inputs that are around the logic threshold voltage.
- The block does not accurately model dynamic response.

For circuits that involve a feedback path around a set of logic gates, you might need to set a nonzero propagation delay on one or more gates.

This block is implemented using event equations. This means that you must provide an initial output state that is consistent with the block input at time zero. For example, if you set initial output state HIGH, but the initial input voltage is below the **Low level input voltage**, then the initial output stays HIGH, the state only correcting itself when the input voltage rises above the **High level input voltage** value.

## Ports

### Conserving

#### **A — Input port**

electrical

Electrical conserving port associated with the Schmitt Trigger input.

#### **J — Output port**

electrical

Electrical conserving port associated with the Schmitt Trigger output.

## Parameters

### Inputs

#### **Low level input voltage — Low level input voltage**

2 V (default)

Voltage value below which the block interprets the input voltage as logic LOW.

#### **High level input voltage — High level input voltage**

3 V (default)

Voltage value above which the block interprets the input voltage as logic HIGH.

#### **Average input capacitance — Average input capacitance**

5 pF (default)

Fixed capacitance that approximates the input capacitance for a MOSFET gate. The MOSFET capacitance depends on the applied voltage. When you drive this block with another gate, the **Average input capacitance** produces a rise time similar to that of the MOSFET. You can usually find this capacitance value on a manufacturer datasheet. Setting this value to zero may result in faster simulation times.

## Outputs

### Output current-voltage relationship — Output current-voltage relationship

Linear (default) | Quadratic

Select the output model:

- Linear - This is the default value.
- Quadratic

### Low level output voltage — Low level output voltage

0 V (default)

Voltage value at the output when the output logic level is LOW.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### High level output voltage — High level output voltage

5 V (default)

Voltage value at the output when the output logic level is HIGH.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### Output resistance — Output resistance

25 Ohm (default)

Value of the series output resistor that is used to model the drop in output voltage resulting from the output current. You can derive this value from a datasheet by dividing the high-level output voltage by the maximum low-level output current.

#### Dependencies

This parameter is visible only when you select Linear for the **Output current-voltage relationship** parameter.

### Supply voltage — Supply voltage

5 V (default)

Supply voltage value applied to the gate in your circuit.

#### Dependencies

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Measurement voltage — Measurement voltage**

5 V (default)

The gate supply voltage for which mask data output resistances and currents are defined.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output resistance at zero current and at I<sub>OH</sub> — Logic HIGH output resistance at zero current and at I<sub>OH</sub>**

[25, 250] Ohm

A row vector [  $R_{OH1}$   $R_{OH2}$  ] of two resistance values. The first value  $R_{OH1}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and there is no output current. The second value  $R_{OH2}$  is the gradient of the output voltage-current relationship when the gate is logic HIGH and the output current is  $I_{OH}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic HIGH output current I<sub>OH</sub> when shorted to ground — Logic HIGH output current I<sub>OH</sub> when shorted to ground**

63 mA (default)

The resulting current when the gate is in the logic HIGH state, but the load forces the output voltage to zero.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output resistance at zero current and at I<sub>OL</sub> — Logic LOW output resistance at zero current and at I<sub>OL</sub>**

[30, 800] Ohm (default)

A row vector [  $R_{OL1}$   $R_{OL2}$  ] of two resistance values. The first value  $R_{OL1}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and there is no output current. The second value  $R_{OL2}$  is the gradient of the output voltage-current relationship when the gate is logic LOW and the output current is  $I_{OL}$ .

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Logic LOW output current I<sub>OL</sub> when shorted to Vcc — Logic LOW output current I<sub>OL</sub> when shorted to Vcc**

-45 mA (default)

The resulting current when the gate is in the logic LOW state, but the load forces the output voltage to the supply voltage Vcc.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Propagation delay — Propagation delay**

25 ns (default)

Time it takes for the output to swing from LOW to HIGH or HIGH to LOW after the input logic levels change.

**Protection diode on resistance — Protection diode on resistance**

5 Ohm (default)

The gradient of the voltage-current relationship for the protection diodes when forward biased.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Protection diode forward voltage — Protection diode forward voltage**

0.6 V (default)

The voltage above which the protection diode is turned on.

**Dependencies**

This parameter is visible only when you select Quadratic for the **Output current-voltage relationship** parameter.

**Initial Conditions****Output initial state — Output initial state**

Low (default) | High

Specify whether the initial output state of the block is High or Low. This parameter is used for both linear and quadratic output states, provided that the **Propagation delay** parameter is greater than zero and the Solver Configuration block does not have the **Start simulation from steady state** option selected.

## Version History

Introduced in R2015a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

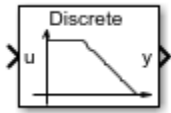
## See Also

CMOS AND | CMOS Buffer | CMOS NAND | CMOS NOR | CMOS NOT | CMOS OR | CMOS XOR | S-R Latch

## Second-Order Filter

Discrete-time or continuous-time low-pass, high-pass, band-pass, or band-stop second-order filter

**Library:** Simscape / Electrical / Control / General Control



### Description

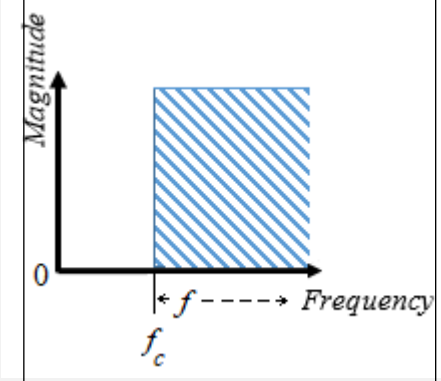
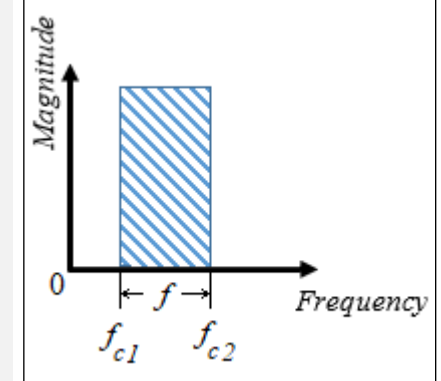
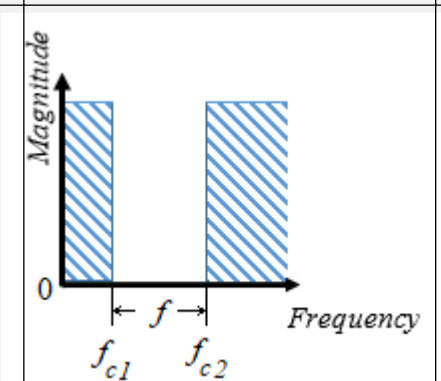
The Second-Order Filter block implements different types of second-order filters. Filters are useful for attenuating noise in measurement signals.

The block provides these filter types:

- Low pass — Allows signals,  $f$ , only in the range of frequencies below the cutoff frequency,  $f_c$ , to pass.
- High pass — Allows signals,  $f$ , only in the range of frequencies above the cutoff frequency,  $f_c$ , to pass.
- Band pass — Allows signals,  $f$ , only in the range of frequencies between two cutoff frequencies,  $f_{c1}$  and  $f_{c2}$ , to pass.
- Band stop — Prevents signals,  $f$ , only in the range of frequencies between two cutoff frequencies,  $f_{c1}$  and  $f_{c2}$ , from passing.

Filter Type	Frequency Range, $f$	
Low-Pass		$f < f_c$



Filter Type	Frequency Range, $f$	
<b>High-Pass</b>		$f > f_c$
<b>Band-Pass</b>		$f_{c1} < f < f_{c2}$
<b>Band-Stop</b>		$f_{c1} < f < f_{c2}$

### Equations

The second order derivative state equation for the filter is:

$$\frac{d^2x}{dt^2} = u - 2\zeta\omega_n\frac{dx}{dt} - \omega_n^2x$$

Where:

- $x$  is the filter internal state.
- $u$  is the filter input.

- $\omega_n$  is the filter natural frequency.
- $\zeta$  is the filter damping factor.

For each filter type, the table maps the block output,  $y(x)$ , as a function of the internal state of the filter, to the  $s$ -domain transfer function,  $G(s)$ .

Filter Type	Output, $y(x)$	Transfer Function, $G(s)$
<b>Low-Pass</b>	$\omega_n^2 x$	$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>High-Pass</b>	$\frac{d^2 x}{dt^2}$	$\frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>Band-Pass</b>	$2\zeta\omega_n \frac{dx}{dt}$	$\frac{2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>Band-Stop</b>	$\frac{d^2 x}{dt^2} + x$	$\frac{s^2 + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

For Initialization:

$$\dot{x}(0) = \left. \frac{dx}{dt} \right|_{t=0}$$

$$u(0) = u_1(0) + u_2(0)$$

$$u_1(0) = A_0 e^{j\varphi_0}$$

$$u_2(0) = b_0 e^{j\frac{\pi}{2}}$$

Where:

- $x(0)$  is the initial state of the filter.
- $u(0)$  is the initial input to the filter.
- $u_1(0)$  is the AC component of the steady-state initial input.
- $A_0$  is the initial amplitude.
- $\varphi_0$  is the initial phase.
- $u_2(0)$  is the DC component of the steady-state initial input.
- $b_0$  is the initial bias.

In the  $s$ -domain  $s = j\omega_0$ . Therefore, for the initial frequency,  $\omega_0$ :

$$\dot{x}(0) = \text{Im} \left( \frac{j\omega_0 u_1(0)}{-\omega_0^2 + j\omega_0 2\zeta\omega_n + \omega_n^2} \right)$$

$$x(0) = \text{Im} \left( \frac{\dot{x}(0)\omega_n^2}{j\omega_0} + u_2(0) \right)$$

## Ports

### Input

#### **u – Filter input**

scalar

Filter input.

Data Types: single | double

### Output

#### **y – Filtered output**

scalar

Filtered output.

Data Types: single | double

## Parameters

### Main

#### **Filter type – Filter type**

Low-pass (default) | High-pass | Band-pass | Band-stop

Type of second-order filter.

#### **Natural frequency (Hz) – Natural frequency**

60 (default) | positive scalar

Natural frequency, in Hz.

### Initial Conditions

#### **Damping factor – Damping factor**

0.707 (default) | nonnegative scalar

Damping factor of the filter.

#### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Initial amplitude — Initial amplitude** $\theta$  (default) | nonnegative scalar

Amplitude at the start of simulation.

**Initial phase (rad) — Initial phase** $\theta$  (default) | nonnegative scalar

Phase, in rad, at the start of simulation.

**Initial frequency (Hz) — Initial frequency** $\theta$  (default) | scalar

Frequency, in Hz, at the start of simulation.

**Initial bias — Initial bias** $\theta$  (default) | nonnegative scalar

Bias at the start of simulation.

## Version History

Introduced in R2018b

## References

[1] Agarwal, A. and Lang, J. H. *Foundations of Analog and Digital Electronic Circuits*. New York: Elsevier, 2005.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

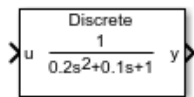
**See Also**

Low-Pass Filter (Discrete or Continuous) | SM PSS1A | Second-Order Low-Pass Filter (Discrete or Continuous) | Variable-Frequency Second-Order Filter | Washout (Discrete or Continuous)

# Second-Order Low-Pass Filter (Discrete or Continuous)

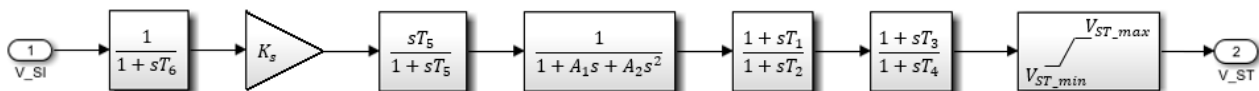
Discrete-time or continuous-time second-order low-pass filter

**Library:** Simscape / Electrical / Control / General Control



## Description

The Second-Order Low-Pass Filter (Discrete or Continuous) block implements a second-order low pass filter in conformance with IEEE Std 421.5-2016<sup>[1]</sup>. In the standard, the filter is a single input, single output signal conditioner that is used in the Power System Stabilizer PSS1A.



In the PSS1A, the filter accounts for some of the low-frequency effects of high-frequency torsional filters. It can also help shape the gain and phase characteristics of the stabilizer.

To switch between continuous and discrete implementations of the Second-Order Low-Pass Filter (Discrete or Continuous) block, adjust the **Sample time** parameter value.

## Initial Conditions

To specify the initial conditions of this block, set **Initialization** to:

- **Inherited from block input** — The block sets the state and output initial conditions to the initial input.
- **Specify as parameter** — The block sets the state initial condition to the value of **Initial state**.

## Ports

### Input

#### **u** — Filter input

scalar or vector

Filter input.

Data Types: single | double

### Output

#### **y** — Filtered output

scalar or vector

Filtered output. The output has the same number of elements as the input.

Data Types: `single` | `double`

## Parameters

### Coefficient of $s$ , $A1$ — $A1$ coefficient

0.1 (default) | positive scalar

$A1$ , the signal conditioning frequency filter coefficient of  $s$ .

### Coefficient of $s^2$ , $A2$ — $A2$ coefficient

0.2 (default) | positive scalar

$A2$ , the signal conditioning frequency filter coefficient of  $s^2$ .

### Initialization — Initial state specification

Inherited from block input (default) | Specify as parameter

Specify the initial state condition for this block. For more information, see “Initial Conditions” on page 1-1703.

### Initial state — Initial state

0 (default) | real number

Block initial state.

### Dependencies

To enable this parameter, set **Initialization** to `Specify as parameter`.

### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

**Introduced in R2018b**

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

[Low-Pass Filter \(Discrete or Continuous\)](#) | [Second-Order Filter](#) | [SM PSS1A](#) | [Variable-Frequency Second-Order Filter](#) | [Washout \(Discrete or Continuous\)](#)

# Secondary Winding

(To be removed) Linear nonideal transformer winding

---

**Note** The Secondary Winding block will be removed in a future release. Use the Winding block instead. To learn how to create a custom transformer without this block, refer to “Push-Pull Buck Converter in Continuous Conduction Mode”.

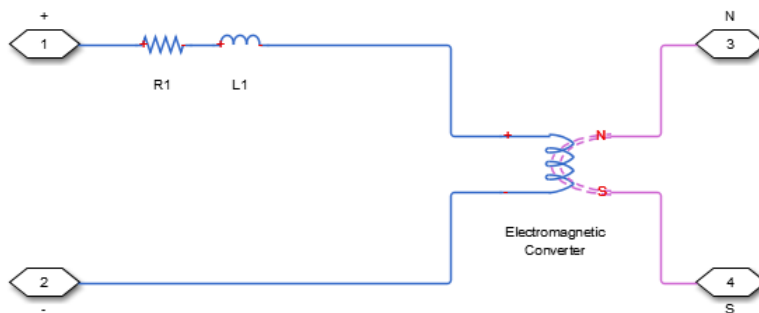
---

## Library

Simscape / Electrical / Power Systems / Passive Devices / Transformers / Fundamental Components

## Description

The Secondary Winding block models linear nonideal winding of a transformer with linear winding leakage effects. The figure shows the equivalent circuit diagram for the secondary winding.



- $R_1$  is the leakage resistance.
- $L_1$  is the leakage inductance.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

+

Positive electrical conserving port



- Negative electrical conserving port
- N North magnetic conserving port
- S South magnetic conserving port

## Parameters

### Main

#### Number of winding turns

Number of wire turns on the transformer winding. The default value is 10.

#### Leakage resistance

Power loss in the winding. The default value is  $1e-3$  Ohm.

#### Leakage inductance

Magnetic flux loss in the winding. The default value is  $1e-3$  H.

## Version History

Introduced in R2013b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Electromagnetic Converter | Winding

### Topics

“Three-Phase Custom Zigzag Transformer”

“Push-Pull Buck Converter in Continuous Conduction Mode”

“Push-Pull Buck Converter in Discontinuous Conduction Mode”

# Sequence Analyzer

Calculate the magnitude and phase of the positive, negative, or zero sequence of a three-phase signal

**Library:** Simscape / Electrical / Control / Measurements



## Description

The Sequence Analyzer block takes a three-phase signal (usually voltage) as the input and outputs the magnitude and phase of the positive, negative, or zero sequence.

The input signals are first converted to phasor representation based on the base frequency and harmonic specified in the block mask. The block then transforms the phasor into a sequence of components..

You can represent the zero, positive, and negative sequence components with  $V_0$ ,  $V_1$ ,  $V_2$  in phasor representation;  $V_a$ ,  $V_b$ ,  $V_c$  are the voltage of three phases. The sequence components are computed as:

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

where  $a = e^{j\frac{2\pi}{3}}$ .

## Limitations

- Because the computation of the three-phase signal requires integration over a full cycle, the output of the sequence analyzer is not accurate during the first period.
- If the base frequency or the harmonic change during the simulation, the output signal is not accurate.

## Ports

### Input

**u** — Periodic input signal

vector

Periodic input signal, specified as a vector of three elements.

### Output

**mag** — Magnitude

scalar | vector

Magnitude of the positive, negative, or zero sequence. Depending on the harmonic numbers, this output is a scalar or a vector.

### **angle — Phase shift**

scalar | vector

Phase shift, in degrees, of the positive, negative, or zero sequence. Depending on the harmonic numbers, this output is a scalar or a vector.

## **Parameters**

### **Base frequency (Hz) — Base frequency**

60 (default)

Base frequency of the input signal in Hz.

### **Harmonic numbers — Harmonic numbers**

[1 2] (default) | scalar or vector

Scalar or vector of the harmonic numbers. 1 stands for base frequency and 2 is the second harmonic.

### **Sequence — Sequence selection**

Positive (default) | Negative | Zero

Select whether the magnitude and phase of the sequence is positive, negative or zero.

### **Sample Time (s) — Block sample time**

0 (default)

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2019b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

# Serial-In Parallel-Out Shift Register

Discrete-time serial-in, parallel-out shift register

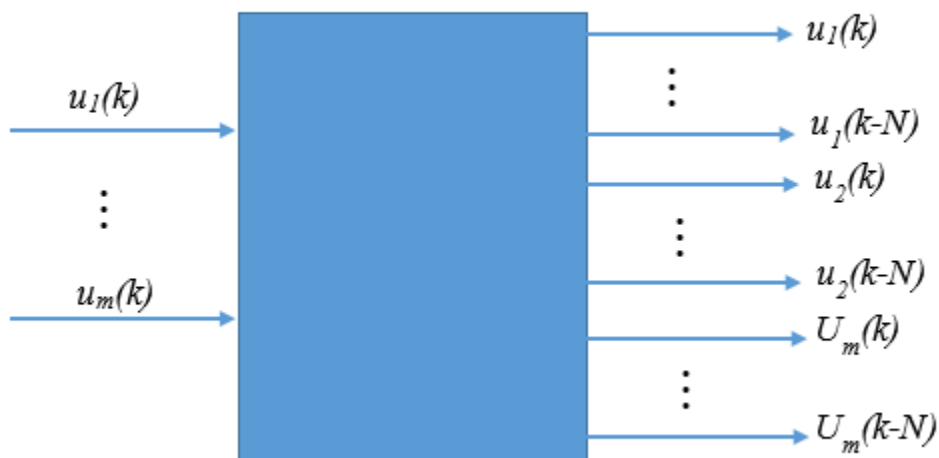
**Library:** Simscape / Electrical / Control / General Control



## Description

The Serial-In Parallel-Out Shift Register block implements a serial-in parallel-out shift register in discrete time. You can use a shift register to convert between serial and parallel interfaces or to implement a circuit delay or hardware stack.

This block outputs a vector of last  $N$  samples of the input signal. If the input signal is a vector, the block outputs the last  $N$  samples of each input signal.



## Ports

### Input

#### **u** – Serial input

scalar | vector

Serial input signal.

Data Types: single | double

## Output

### **y** — Parallel output

vector

Parallel output signal.

Data Types: `single` | `double`

## Parameters

### **Number of samples** — Number of input samples

4 (default) | positive integer

Number of register stages or samples.

### **Initial condition** — Initial value of the input sample

0 (default) | real number

Initial value of the  $N-1$  samples preceding simulation start time. The value must be a scalar or a vector of the same size as the input signal.

### **Sample time (-1 for inherited)** — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time, specify -1. For discrete-time, specify a positive integer. For continuous-time, specify 0.

## Version History

Introduced in R2018b

## Extended Capabilities

### **C/C++ Code Generation**

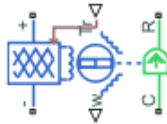
Generate C and C++ code using Simulink® Coder™.

## See Also

## Motor & Drive (System Level)

Generic motor and drive with closed-loop torque control

**Library:** Simscape / Electrical / Electromechanical / System-Level Modeling



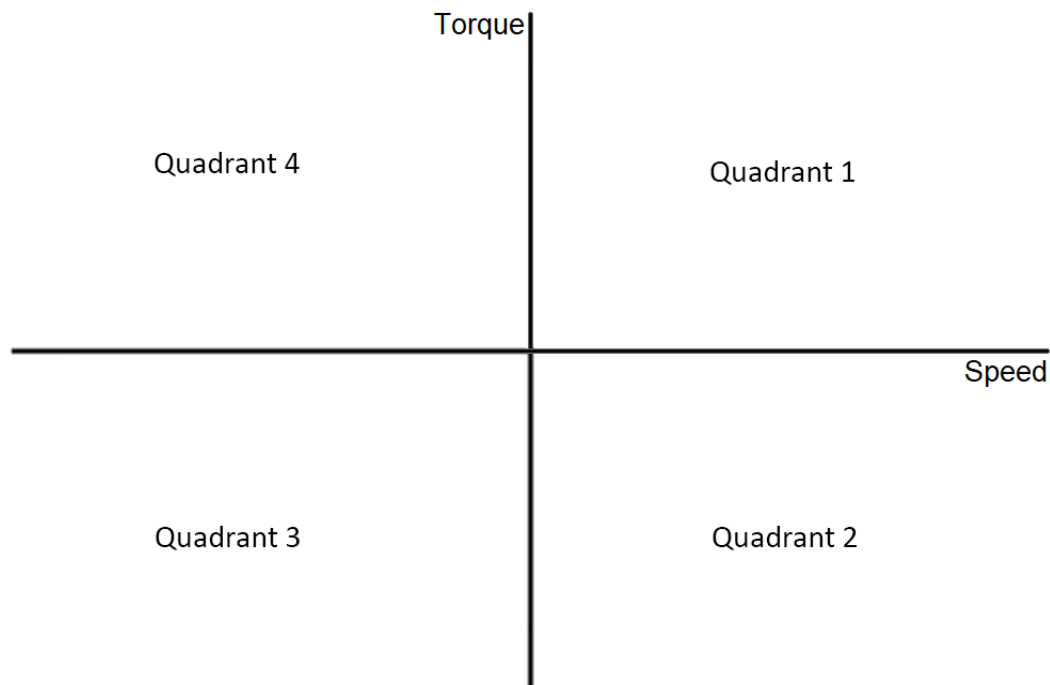
### Description

The Motor & Drive (System Level) block represents a generic motor and drive with closed-loop torque control. Use this block to model a brushless motor (such as a PMSM) and drive, or, more generally, traction and actuation systems implemented using a wide range of motor types.

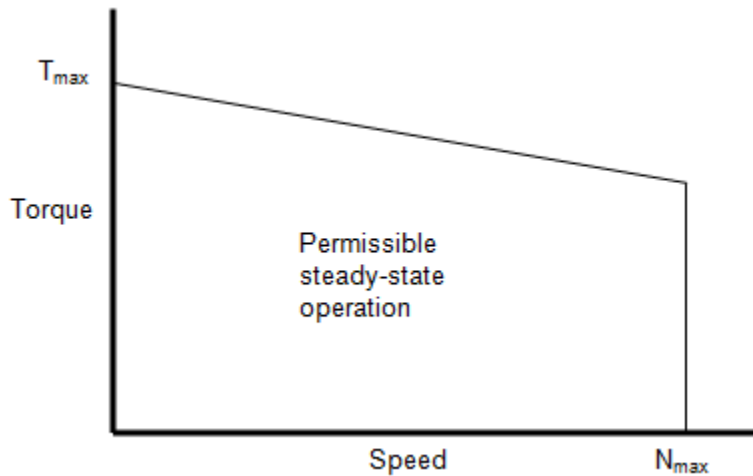
To enable fast simulation at a system level, this block abstracts the motor, drive electronics, and control.

The block permits only the range of torques and speeds that the torque-speed envelope defines. In the default block configuration, you specify this data in the block dialog box as a set of speed data points and corresponding maximum torque values.

This figure shows the Cartesian quadrants definition:

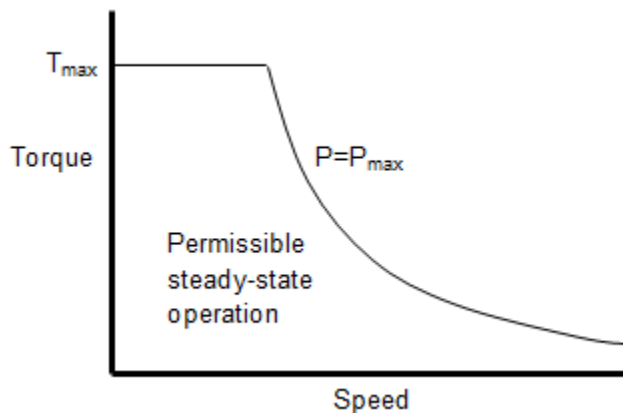


The following figure shows a typical torque-speed envelope for a torque-controlled motor and drive.



Specify the torque-speed envelope for the positive torque region only, that is, quadrants 1 and 4. If you specify only for positive speeds (quadrant 1 or, equivalently, the motoring region), then the quadrant 4 torque envelope is defined by the block as the mirror image of quadrant 1. The motor torque-speed envelope has the same profile when the motor is operating in a reverse direction (quadrants 2 and 3).

Instead of providing tabulated torque-speed data, you can specify a maximum torque and a maximum power. This results in the torque-speed envelope profile shown below. The other three operating quadrants are constrained by this same profile.



The block produces a positive torque acting from the mechanical C to R ports.

### Intermittent Over-Torque Operation

To over-torque the motor for short periods of time, set the **Allow intermittent over-torque** parameter to Yes. In this case, you must specify the values for both the **Continuous operation maximum torque envelope** and the **Intermittent operation maximum torque envelope** parameters. Internally the block determines which torque envelope to apply based on the torque demand history. You can over-torque the motor drive if the torque demand has been less than the continuous operation torque envelope for more than the value specified in the **Recovery time**

parameter. Over-torquing is disabled if over-torquing has been applied for longer than the value specified in the **Over-torque time limit** parameter.

For more application specific management of over-torquing, disable the over-torquing in the Motor & Drive block and externally implement the torque limiting in Simulink between the torque demand and the torque reference input port, **Tr**, of the Motor & Drive block.

### **Model Electrical Losses**

The block allows both simplified and tabulated definition of electrical losses. The default, simplified, behavior is to model the losses as the sum of the following four terms:

- A series resistance between the DC power supply and the motor drive.
- Fixed losses independent of torque and speed,  $P_0$ . Use this to account for fixed converter losses.
- A torque-dependent electrical loss  $k\tau^2$ , where  $\tau$  is the torque and  $k$  is a constant. This represents ohmic losses in the copper windings.
- A speed-dependent electrical loss  $k_w\omega^2$ , where  $\omega$  is the speed and  $k_w$  is a constant. This represents iron losses due to eddy currents.

This simplification of loss dependency on torque and speed might be sufficient for early design work, regardless of the underlying motor type and drive topology.

If you require more accuracy, or later in the design process, 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.

Finally, you can specify electrical losses by using tabulated efficiency data, instead of a single efficiency measurement or tabulated loss data. When using this option, also 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.

The best practice is to provide tabulated loss data as a function of speed and torque, rather than tabulated efficiency data, 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.



When you provide tabulated loss or efficiency data, you can also specify it as a function of speed, load torque, and DC supply voltage. This option is useful when the supply voltage is not regulated and can vary during the simulation. One example is an electric vehicle drivetrain that does not have a DC-DC regulator upstream of the motor drive. Use the Motor & Drive (System Level) block to model the motor drive and provide tabulated loss or efficiency values as a function of motor speed, load torque, and DC supply voltage.

### Plot Efficiency Map

The block allows you to visualize the map of the torque-speed envelope and the steady-state electrical efficiency of the motor and drive. To plot the efficiency map, either:

- Double-click the block in your block diagram and, in the **Description** tab, click **Plot efficiency map**.
- Right-click the block in your block diagram and, in the **Electrical** context menu, select **Plot efficiency map**.

This efficiency map does not include the losses generated by the **External supply series resistance** and **Rotor damping** parameters.

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port.
- **Show thermal port** — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Model Faults

You can use the physical signal input port **F** to simulate servomotor failure, as well as connecting and 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.

If a signal is connected to port **F**, then the block operates according to the parameter settings on the **Faults** tab. For example, if **Fault condition** is **Faulted if  $F \geq \text{Fault threshold}$** , then when the signal at port **F** rises above the **Fault threshold** value, the servomotor stops operating, zero current is taken from the supply side, and zero current is supplied to the load side.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Assumptions and Limitations

- The motor driver tracks a torque demand with a time constant  $T_c$ .
- Motor speed fluctuations due to mechanical load do not affect the motor torque tracking.
- The electrical connections must always be connected to a supply, such as a DC voltage source or a battery, that provides a positive voltage and is capable of delivering or absorbing the required power. To simulate turning the supply source on and off, see “Model Faults” on page 1-1715.

## Ports

### Input

#### **Tr** — Reference torque demand

physical

Physical input port associated with the reference torque demand.

#### **F** — Servomotor failure port

physical

Physical input port associated with the servomotor failure simulation.

### Dependencies

To enable this port, set **Enable open circuit fault** to On.

### Output

#### **W** — Mechanical speed output

physical

Physical output port associated with the mechanical speed.

### Conserving

#### **+** — Positive electrical DC supply

electrical

Electrical conserving port associated with the positive electrical DC supply.

#### **-** — Negative electrical DC supply

electrical

Electrical conserving port associated with the negative electrical DC supply.

#### **C** — Motor case

mechanical

Mechanical rotational conserving port associated with the motor case.

#### **R** — Motor rotor

mechanical

Mechanical rotational conserving port associated with the motor rotor.

**H – Thermal port**

thermal

Thermal port.

**Dependencies**To enable this port, set **Modeling option** to Show thermal port.**Parameters****Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

**Electrical Torque****Parameterize by – Parameterize by**

Tabulate torque envelope with speed (default) | Maximum torque and power | Tabulate torque envelope with speed and supply voltage

Select one of the following methods for block parameterization:

- **Tabulate torque envelope with speed** – Provide the vectors of rotational speeds and corresponding maximum torque values. This is the default option.
- **Maximum torque and power** – Define the torque-speed envelope by providing values for maximum permissible torque and motor power.
- **Tabulate torque envelope with speed and supply voltage** – Provide the vectors of rotational speeds, the vectors of DC supply voltages, and the corresponding maximum torque values.

**Allow intermittent over-torque – Intermittent over-torque option**

No (default) | Yes

Whether to allow intermittent over-torque. For more information on over-torquing, see “Intermittent Over-Torque Operation” on page 1-1713.

**Continuous operation maximum torque envelope,  $T_c(w)$  – Vector of maximum torque values for continuous operation**

[.09, .08, .07, 0] N\*m (default)

Maximum torque values for permissible steady-state operation. These values correspond to the speeds in the **Corresponding rotational speeds,  $w$**  parameter and define the torque-speed envelope for the motor.**Dependencies**To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed.**Continuous operation maximum torque envelope,  $T_c(w, V_{dc})$  – Vector of maximum torque values for continuous operation tabulated with speeds and voltages**

[.09, .09, .09; .08, .08, .08; .07, .07, .07; 0, 0, 0] N\*m (default) | 2D vector of positive values

Maximum torque values for permissible steady-state operation. These values correspond to the speeds in the **Corresponding rotational speeds,  $w$**  parameter and to the voltages in the **Corresponding DC supply voltage,  $V_{dc}$**  parameter. This parameter defines the envelope with speed and supply voltage for the motor.

**Dependencies**

To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed and supply voltage.

**Intermittent operation maximum torque envelope,  $T_i(w)$  – Vector of maximum torque values for intermittent operation**

[.36, .32, .28, 0] N\*m (default)

Maximum torque values during intermittent over-torquing operation.

**Dependencies**

To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed and **Allow intermittent over-torque** to Yes.

**Intermittent operation maximum torque envelope,  $T_i(w, V_{dc})$  – Vector of maximum torque values for intermittent operation tabulated with speeds and voltages**

[.36, .36, .36; .32, .32, .32; .28, .28, .28; 0, 0, 0] N\*m (default) | 2D vector of positive values

Maximum torque values during intermittent over-torquing operation tabulated with speeds and voltages. The values of this parameter must be greater than the values of the **Continuous operation maximum torque envelope,  $T_c(w, V_{dc})$**  parameter.

**Dependencies**

To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed and supply voltage and **Allow intermittent over-torque** to Yes.

**Corresponding rotational speeds,  $w$  – Vector of rotational speeds**

[0, 3750, 7500, 8000] rpm (default)

Rotational speeds for permissible steady-state operation. To avoid poor performance due to an infinite slope in the torque-speed curve, specify a vector of rotational speeds that does not contain duplicate consecutive values.

**Dependencies**

To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed.

**Corresponding DC supply voltage,  $V_{dc}$  – Vector of DC supply voltages**

[6, 12, 24] V (default) | vector of positive scalars

DC supply voltages for permissible steady-state operation. The values of this parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Parameterize by** to Tabulate torque envelope with speed and supply voltage.

**Continuous operation maximum torque – Maximum torque for continuous operation**

0.1 N\*m (default) | positive scalar

Maximum permissible motor torque during continuous operation.

**Dependencies**

To enable this parameter, set **Parameterize by** to Maximum torque and power.

**Continuous operation maximum power – Maximum power for continuous operation**

30 W (default) | positive scalar

Maximum permissible motor power during continuous operation.

**Dependencies**

To enable this parameter, set **Parameterize by** to Maximum torque and power.

**Intermittent operation maximum torque – Maximum torque for intermittent operation**

0.4 N\*m (default) | positive scalar

Maximum permissible motor torque during intermittent over-torquing operation.

**Dependencies**

To enable this parameter, set **Parameterize by** to Maximum torque and power and **Allow intermittent over-torque** to Yes.

**Intermittent operation maximum power – Maximum power for intermittent operation**

120 W (default) | positive scalar

Maximum permissible motor power during intermittent over-torquing operation.

**Dependencies**

To enable this parameter, set **Parameterize by** to Maximum torque and power and **Allow intermittent over-torque** to Yes.

**Over-torque time limit – Over-torque time limit**

30 s (default) | positive scalar

Maximum amount of time for which you can continuously apply over-torquing.

**Dependencies**

To enable this parameter, set **Allow intermittent over-torque** to Yes.

**Recovery time – Recovery time**

60 s (default) | positive scalar

Amount of time for which the torque demand has to be less than the continuous operation torque envelope to be able to apply over-torquing again.

**Dependencies**

To enable this parameter, set **Allow intermittent over-torque** to Yes.

**Torque control time constant, T<sub>c</sub> – Torque control time constant**

0.02 s (default) | nonnegative scalar

Time constant with which the motor driver tracks a torque demand.

## Electrical Losses

### Parameterize losses by — Electrical loss parameterization

Single efficiency measurement (default) | Tabulated loss data as a function of speed and torque | Tabulated efficiency data as a function of speed and torque | Tabulated loss data as a function of speed, torque, and DC supply voltage | Tabulated efficiency data as a function of speed, torque, and DC supply voltage

Select one of the following methods for electrical loss parameterization:

- **Single efficiency measurement** — Model the losses as the sum of the four terms, listed in the block description, at a single measurement point.
- **Tabulated loss data as a function of speed and torque** — Determine the losses by two-dimensional table lookup based on the provided tabulated data for motor speeds, load torques, and corresponding losses.
- **Tabulated efficiency data as a function of speed and torque** — Determine the losses by two-dimensional table lookup based on the provided tabulated data for motor speeds, load torques, and corresponding efficiencies.
- **Tabulated loss data as a function of speed, torque, and DC supply voltage** — Determine the losses by three-dimensional table lookup based on the provided tabulated data for motor speeds, load torques, DC supply voltages, and corresponding losses.
- **Tabulated efficiency data as a function of speed, torque, and DC supply voltage** — Determine the losses by three-dimensional table lookup based on the provided tabulated data for motor speeds, load torques, DC supply voltages, and corresponding efficiencies.

See “Model Electrical Losses” on page 1-1714 for details.

### Motor and driver overall efficiency (percent) — Motor and driver overall efficiency 100 (default)

The block defines overall efficiency as

$$\eta = 100 \frac{\tau_0 \omega_0}{\tau_0 \omega_0 + P_0 + k \tau_0^2 + k_w \omega_0^2}$$

where:

- $\tau_0$  represents the **Torque at which efficiency is measured**.
- $\omega_0$  represents the **Speed at which efficiency is measured**.
- $P_0$  represents the **Fixed losses independent of torque or speed**.
- $k \tau_0^2$  represents the torque-dependent electrical losses.
- $k_w \omega_0^2$  represents the speed-dependent iron losses.

At initialization, the block solves the efficiency equation for  $k$ . The block neglects losses associated with the rotor damping.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Single efficiency measurement.

**Speed at which efficiency is measured — Speed at which efficiency is measured**  
3750 rpm (default)

Speed that the block uses to calculate torque-dependent electrical losses.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Single efficiency measurement.

**Torque at which efficiency is measured — Torque at which efficiency is measured**  
0.08 Nm (default)

Torque that the block uses to calculate torque-dependent electrical losses.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Single efficiency measurement.

**Iron losses at measurement speed — Iron losses at measurement speed**  
0 W (default)

Iron losses at the speed and torque at which efficiency is defined.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Single efficiency measurement.

**Fixed losses independent of torque and speed — Fixed losses independent of torque and speed**  
0 W (default)

Fixed electrical loss associated with the driver when the motor current and torque are zero.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Single efficiency measurement.

**Vector of speeds (w) for tabulated losses — Vector of speeds (w) for tabulated losses**  
[-8000, -4000, 0, 4000, 8000] rpm (default)

Vector of speed values, to be used for table lookup when calculating losses.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated loss data as a function of speed and torque, Tabulated loss data as a function of speed, torque, and DC supply voltage, Tabulated efficiency data as a

function of speed and torque, or Tabulated efficiency data as a function of speed, torque, and DC supply voltage.

**Vector of torques (T) for tabulated losses – Vector of torques (T) for tabulated losses**

[0, .03, .06, .09] Nm (default)

Vector of speed values, to be used for table lookup when calculating losses.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated loss data as a function of speed and torque, Tabulated loss data as a function of speed, torque, and DC supply voltage, Tabulated efficiency data as a function of speed and torque, or Tabulated efficiency data as a function of speed, torque, and DC supply voltage.

**Vector of DC supply voltages (v) for tabulated losses – Vector of DC supply voltages (v) for tabulated losses**

[100, 250, 400] V (default)

The vector of DC supply voltages, to be used for table lookup when calculating losses.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated loss data as a function of speed, torque, and DC supply voltage or Tabulated efficiency data as a function of speed, torque, and DC supply voltage.

**Corresponding losses, P(w,T) – Corresponding losses**

[1.49, 1.67, 2.21, 3.1; .42, .69, 1.14, 2.03; .06, .24, .78, 1.68; .42, .69, 1.14, 2.03; 1.49, 1.67, 2.21, 3.1] W (default)

Tabulated values for electrical losses as a function of speed and torque, to be used for 2D table lookup. Each value in the matrix specifies the losses for a specific combination of speed and torque. The matrix size must match the dimensions defined by the speed and torque vectors.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated loss data as a function of speed and torque.

**Corresponding efficiency (percent), E(w,T) – Corresponding efficiency (percent), E(w,T)**

[95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95] (default)

Tabulated efficiency values, in percent, as a function of speed and torque, to be used for 2D table lookup. Each value in the matrix specifies the efficiency for a specific combination of speed and torque. The matrix size must match the dimensions defined by the speed and torque vectors. 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.



**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated efficiency data as a function of speed and torque.

**Corresponding losses,  $P(w,T,v)$  — Corresponding losses,  $P(w,T,v)$** 

ones(5, 4, 3) W (default)

Tabulated values for electrical losses as a function of speed, torque, and DC supply voltage, to be used for 3D table lookup. Each value in the matrix specifies the losses for a specific combination of speed, torque, and DC supply voltage. The matrix size must match the dimensions defined by the three vectors.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated loss data as a function of speed, torque, and DC supply voltage.

**Corresponding efficiency (percent),  $E(w,T,v)$  — Corresponding efficiency (percent),  $E(w,T,v)$** 

95 \* ones(5, 4, 3) (default)

Tabulated efficiency values, in percent, as a function of speed, torque, and DC supply voltage, to be used for 3D table lookup. Each value in the matrix specifies the efficiency for a specific combination of speed, torque, and DC supply voltage. The matrix size must match the dimensions defined by the three vectors. 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.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter to Tabulated efficiency data as a function of speed, torque, and DC supply voltage.

**External supply series resistance — External supply series resistance**

0 Ohm (default) | nonnegative scalar

Equivalent resistance used in series with the DC supply to model electrical losses that are proportional to the driver supply current. The block assumes that the DC supply current is approximately constant under constant load conditions.

**Faults****Enable open circuit fault — Open circuit fault option**

Off (default) | On

Whether to enable open circuit fault.

**Fault condition — Fault condition**

Faulted if  $F \geq$  Fault threshold (default) | Faulted if  $F \leq$  Fault threshold

Selects whether the fault is triggered by a signal that is high or low:

- **Faulted if  $F \geq$  Fault threshold** — Motor & Drive (System Level) block is disabled if the signal at port **F** rises above the threshold value.

- **Faulted** if  $F \leq \text{Fault threshold}$  — Motor & Drive (System Level) block is disabled if the signal at port **F** falls below the threshold value.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Fault threshold — Fault threshold**

0.5 (default)

The threshold value used to detect a fault.

**Dependencies**

To enable this parameter, set **Enable open circuit fault** to On.

**Mechanical****Rotor inertia — Rotor inertia**

$5e-6 \text{ kg}\cdot\text{m}^2$  (default) | nonnegative scalar

Rotor resistance to change in motor motion. The value can be zero.

**Rotor damping — Rotor damping**

$1e-5 \text{ N}\cdot\text{m}/(\text{rad}/\text{s})$  (default) | nonnegative scalar

Rotor damping. The value can be zero.

**Initial rotor speed — Initial rotor speed**

0 rpm (default)

Rotor speed at the start of the simulation.

**Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

**Resistance temperature coefficient — Resistance temperature coefficient**

$3.93e-3 \text{ 1/K}$  (default)

Parameter  $\alpha$  in the equation defining resistance as a function of temperature, as described in “Thermal Model for Actuator Blocks”. This parameter is visible only if the **Parameterize losses by** parameter on the **Electrical Losses** tab is set to Single efficiency measurement. The default value is for copper.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Single efficiency measurement.

**Measurement temperature — Measurement temperature**

25 degC (default)

The temperature for which motor parameters are defined. If you parameterize electrical losses by tabulated loss data, then this is the temperature for which the **Corresponding losses, P(w,T)** are given on the **Electrical Losses** tab.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

The temperature for which the **Corresponding losses, P(w,T), at second measurement temperature** are given.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Tabulated loss data.

**Corresponding losses, P(w,T), at second measurement temperature – Corresponding losses, P(w,T), at second measurement temperature**

[1.49, 1.74, 2.49, 3.74; .42, .67, 1.42, 2.67; .06, .31, 1.06, 2.31; .42, .67, 1.42, 2.67; 1.49, 1.74, 2.49, 3.74] W (default)

Iron losses at the second measurement temperature, corresponding to the speed and torque tabulated values on the **Electrical Losses** tab.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Tabulated loss data.

**Corresponding efficiency (percent), E(w,T), at second measurement temperature – Corresponding efficiency (percent), E(w,T), at second measurement temperature**

[95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95; 95, 95, 95, 95, 95, 95] (default)

Tabulated efficiency values, in percent, at the second measurement temperature, corresponding to the speed and torque tabulated values on the **Electrical Losses** tab.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Tabulated efficiency data as a function of speed and torque.

**Corresponding losses, P(w,T,v), at second measurement temperature – Corresponding losses, P(w,T,v), at second measurement temperature**

ones(5, 4, 3) W (default)

Iron losses at the second measurement temperature, corresponding to the speed, torque, and DC supply voltage tabulated values on the **Electrical Losses** tab.

**Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Tabulated loss data as a function of speed, torque, and DC supply voltage.

**Corresponding efficiency (percent), E(w,T,v), at second measurement temperature – Corresponding efficiency (percent), E(w,T,v), at second measurement temperature**

95 \* ones(5, 4, 3) (default)

Tabulated efficiency values, in percent, at the second measurement temperature, corresponding to the speed, torque, and DC supply voltage tabulated values on the **Electrical Losses** tab.

### **Dependencies**

This parameter is visible only when you set the **Parameterize losses by** parameter on the **Electrical Losses** setting to Tabulated efficiency data as a function of speed, torque, and DC supply voltage.

### **Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Thermal mass — Thermal mass**

100 J/K (default)

Thermal mass of the electrical winding, defined as the energy required to raise the temperature by one degree.

#### **Initial Temperature — Initial temperature**

25 (default)

The temperature of the thermal port at the start of simulation. The default value is 25°C.

## **Version History**

### **Introduced in R2008a**

#### **Library Path and Name Change**

*Behavior changed in R2020b*

The Simplified PMSM Drive block has been renamed the Motor & Drive (System Level) block and moved from the **Simscape > Electrical > Electromechanical > Permanent Magnet** library to the **Simscape > Electrical > Electromechanical > System-Level Modeling** library.

If you have a script that depends on the library path or the Simscape component path of this block, you need to update it to reflect these changes.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

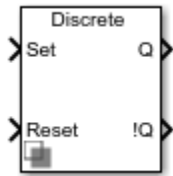
## **See Also**

DC Motor | Induction Machine (Single-Phase) | Shunt Motor | Universal Motor

# Set-Reset Flip-Flop

Set-reset flip-flop or bistable multivibrator

**Library:** Simscape / Electrical / Control / General Control



## Description

The Set-Reset Flip-Flop block implements a set-reset flip-flop or bistable multivibrator.

The block maintains the output signals, Q and !Q, unless an external trigger is applied. An external trigger (*Set*) produces a change of state, which is maintained until a second external trigger (*Reset*) is applied.

The table shows the relationship between the block input and output signals.

Set	Reset	Q	!Q
0	0	Last Q	Last !Q
0	1	0	1
1	0	1	0
1	1	Undefined	Undefined

When the state is undefined, the priority is provided as an external parameter.

## Ports

### Input

#### Set — Set input signal

0 | 1

Input signal that triggers a state change.

Data Types: Boolean

#### Reset — Reset input signal

0 | 1

Input signal that resets a state change.

Data Types: Boolean

## Output

### **Q – Output signal**

0 | 1

Output signal **Q**, with the same dimensions and data type as the input **signal**.

Data Types: Boolean

### **!Q – Complement output signal**

0 | 1

Output signal **!Q**, with the same dimensions and data type as the input **signal**.

Data Types: Boolean

## Parameters

### **Priority when undefined state – State priority**

Set (default) | Reset

Priority for the undefined state, that is, when both *Set* and *Reset* are true.

### **Initial condition for Q state – Initial value of the output Q**

0 (default) | 1

Initial condition for *Q* state.

### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2018b

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

# SFFM Current Source

Single-frequency frequency-modulated current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The SFFM Current Source block represents a frequency-modulated AC current source whose output current value is independent of the voltage across its terminals. The following equation describes the current through the source as a function of time:

$$I_{out} = IO + IA * \sin((2\pi * FC * Time) + MI * \sin(2\pi * FS * Time))$$

where:

- $IO$  is the **Current offset**, **IO** parameter value.
- $IA$  is the **Current amplitude**, **IA** parameter value.
- $FC$  is the **Carrier frequency**, **FC** parameter value.
- $MI$  is the **Modulation index**, **MI** parameter value.
- $FS$  is the **Signal frequency**, **FS** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is 1e-12.
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the SFFM current source positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the SFFM current source negative terminal.

## Parameters

### Current offset, IO — Current offset

0 A (default)

Magnitude of the time-independent part of the output current.

### Current amplitude, IA — Current amplitude

0 A (default)

Magnitude of the sinusoidal part of the output current.

### Carrier frequency, FC — Carrier frequency

0 Hz (default) | nonnegative scalar

Frequency of the carrier wave. The value must be greater than or equal to 0.

### Modulation index, MI — Modulation index

0 (default) | nonnegative scalar

Amount by which the modulated signal varies around its unmodulated level. The value must be greater than or equal to 0.

### Signal frequency, FS — Signal frequency

0 Hz (default) | nonnegative scalar

Frequency of the modulated signal. The value must be greater than or equal to 0.

### Include parallel conductance, GMIN — Whether to include parallel conductance

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance, *GMIN*:

#### Values of parallel conductance, GMIN

Include parallel conductance, GMIN		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, <i>GMIN</i> , is not included and it does not affect the current source.
<i>GMIN</i> is equal to the value of the <b>GMIN</b> parameter in the Environment Parameters block.	<i>GMIN</i> = 1e-12 Ohm	

## Version History

Introduced in R2008a



## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

DC Current Source | Environment Parameters | Exponential Current Source | Pulse Current Source | Piecewise Linear Current Source | SFFM Current Source | SFFM Voltage Source | Sinusoidal Current Source

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# SFFM Voltage Source

Single-frequency frequency-modulated voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The SFFM Voltage Source block represents a frequency-modulated AC voltage source whose output voltage value is independent of the current through the source. The following equation describes the output voltage as a function of time:

$$V_{out} = VO + VA * \sin((2\pi * FC * Time) + MI * \sin(2\pi * FS * Time))$$

where:

- $VO$  is the **Voltage offset**, **VO** parameter value.
- $VA$  is the **Voltage amplitude**, **VA** parameter value.
- $FC$  is the **Carrier frequency**, **FC** parameter value.
- $MI$  is the **Modulation index**, **MI** parameter value.
- $FS$  is the **Signal frequency**, **FS** parameter value.

## Ports

### Conserving

#### + – Positive voltage

electrical

Electrical conserving port associated with the SFFM voltage source positive voltage

#### - – Negative voltage

electrical

Electrical conserving port associated with the SFFM voltage source negative voltage

## Parameters

### Voltage offset, $VO$ – Voltage offset

0 V (default)

Magnitude of the time-independent part of the output voltage.

### Voltage amplitude, $VA$ – Voltage amplitude

0 V (default)

Magnitude of the sinusoidal part of the output voltage.

**Carrier frequency, FC — Carrier frequency**

0 Hz (default) | nonnegative scalar

Frequency of the carrier wave. The value must be greater than or equal to 0.

**Modulation index, MI — Modulation index**

0 (default) | nonnegative scalar

Amount by which the modulated signal varies around its unmodulated level. The value must be greater than or equal to 0.

**Signal frequency, FS — Signal frequency**

0 Hz (default) | nonnegative scalar

Frequency of the modulated signal. The value must be greater than or equal to 0.

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

DC Voltage Source | Environment Parameters | Exponential Voltage Source | Pulse Voltage Source | Piecewise Linear Voltage Source | SFFM Current Source | SFFM Voltage Source | Sinusoidal Voltage Source

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

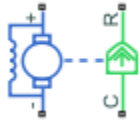
“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Shunt Motor

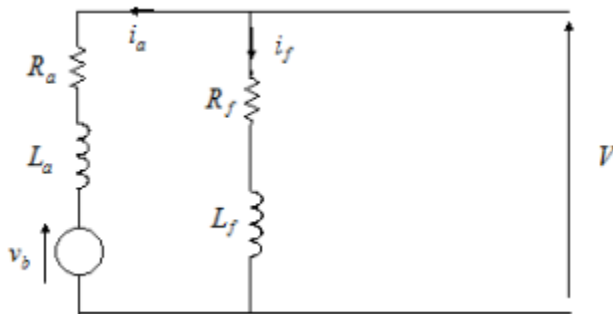
Shunt motor with electrical and torque characteristics

**Library:** Simscape / Electrical / Electromechanical / Brushed Motors



## Description

The Shunt Motor block represents the electrical and torque characteristics of a shunt motor using the following equivalent circuit model.



When you set the **Model parameterization** parameter to **By equivalent circuit parameters**, you specify the equivalent circuit parameters for this model:

- $R_a$  — **Armature resistance**
- $L_a$  — **Armature inductance**
- $R_f$  — **Field winding resistance**
- $L_f$  — **Field winding inductance**

The Shunt Motor block computes the motor torque as follows:

- 1 The magnetic field in the motor induces the following back emf  $v_b$  in the armature:

$$v_b = L_{af} i_f \omega$$

where  $L_{af}$  is a constant of proportionality and  $\omega$  is the angular velocity.

- 2 The mechanical power is equal to the power reacted by the back emf:

$$P = v_b i_a = L_{af} i_f i_a \omega$$

- 3 The motor torque is:

$$T = P/\omega = L_{af} i_f i_a$$

The torque-speed characteristic for the Shunt Motor block model is related to the parameters in the preceding figure. When you set the **Model parameterization** parameter to `By rated power, rated speed & no-load speed`, the block solves for the equivalent circuit parameters as follows:

- 1 For the steady-state torque-speed relationship,  $L$  has no effect.
- 2 Sum the voltages around the loop:

$$V = i_a R_a + L_{af} i_f \omega$$

$$V = i_f R_f$$

- 3 Solve the preceding equations for  $i_a$  and  $i_f$ :

$$i_f = \frac{V}{R_f}$$

$$i_a = \frac{V}{R_a} \left( 1 - \frac{L_{af} \omega}{R_f} \right)$$

- 4 Substitute these values of  $i_a$  and  $i_f$  into the equation for torque:

$$T = \frac{L_{af}}{R_a R_f} \left( 1 - \frac{L_{af} \omega}{R_f} \right) V^2$$

The block uses the rated speed and power to calculate the rated torque. The block uses the rated torque and no-load speed values to get one equation that relates  $R_a$  and  $L_{af}/R_f$ . It uses the no-load speed at zero torque to get a second equation that relates these two quantities. Then, it solves for  $R_a$  and  $L_{af}/R_f$ .

The block models motor inertia  $J$  and damping  $B$  for all values of the **Model parameterization** parameter. The output torque is:

$$T_{load} = \frac{L_{af}}{R_a R_f} \left( 1 - \frac{L_{af} \omega}{R_f} \right) V^2 - J \dot{\omega} - B \omega$$

The block produces a positive torque acting from the mechanical C to R ports.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain thermal ports.
- `Show thermal port` — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the shunt motor positive terminal.

**- – Negative terminal**

electrical

Electrical conserving port associated with the shunt motor negative terminal.

**C – Motor case**

mechanical

Mechanical rotational conserving port associated with the shunt motor case.

**R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the shunt motor rotor.

**Hf – Field winding thermal port**

thermal

Field winding thermal port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Ha – Armature winding thermal port**

thermal

Armature winding thermal port.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Electrical Torque****Model parameterization – Block parameterization**

By equivalent circuit parameters (default) | By rated power, rated speed &amp; no-load speed

Select one of the following methods for block parameterization:

- By equivalent circuit parameters — Provide electrical parameters for an equivalent circuit model of the motor.
- By rated power, rated speed & no-load speed — Provide power and speed parameters that the block converts to an equivalent circuit model of the motor.

**Armature resistance – Armature resistance**

110 Ohm (default)

Resistance of the armature.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Field winding resistance — Field winding resistance**

2500 Ohm (default)

Resistance of the field winding.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Back-emf constant — Back-emf constant**

5.11 s\*V/rad/A (default)

The ratio of the voltage generated by the motor to the motor speed.

**Armature inductance — Armature inductance**

0.1 H (default)

Inductance of the armature. If you do not have information about this inductance, set the value of this parameter to a small, nonzero number. The value can be zero.

**Field winding inductance — Field winding inductance**

0.1 H (default)

Inductance of the field winding. If you do not have information about this inductance, set the value of this parameter to a small, nonzero number. The value can be zero.

**No-load speed — No-load speed**

4600 rpm (default)

Speed of the motor when no load is applied.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Rated speed (at rated load) — Rated speed**

4000 rpm (default)

Motor speed at the rated load.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Rated load (mechanical power) — Rated load**

50 W (default)

The mechanical load for which the motor is rated to operate.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Rated DC supply voltage — Rated DC supply voltage**

220 V (default)

The voltage at which the motor is rated to operate.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Starting current at rated DC supply voltage — Starting current at rated DC supply voltage**

2.09 A (default)

The initial current when starting the motor with the rated DC supply voltage.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By rated power, rated speed & no-load speed.

**Mechanical****Rotor inertia — Rotor inertia**

$2e-4$  kg\*m<sup>2</sup> (default)

Resistance of the rotor to change in motor motion. The value can be zero.

**Rotor damping — Rotor dissipated energy**

$1e-6$  N\*m/(rad/s) (default)

Energy dissipated by the rotor. The value can be zero.

**Initial rotor speed — Initial rotor speed**

0 rpm (default)

Speed of the rotor at the start of the simulation.

**Temperature Dependence**

This tab appears only for blocks with exposed thermal ports.

**Resistance temperature coefficients, [ $\alpha_f$   $\alpha_a$ ] — Stator and rotor resistance temperature coefficients**

[.00393, .00393] 1/K (default)

A 1 by 2 row vector that defines the coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The first element corresponds to the field winding, and the second to the armature. The default value is for copper.

**Measurement temperature — Measurement temperature**

25 C (default)



The temperature for which motor parameters are defined.

### **Thermal Port**

This tab appears only for blocks with exposed thermal ports.

#### **Thermal masses, [Mf Ma] – Thermal masses**

[100, 100] J/K (default)

A 1 by 2 row vector that defines the thermal mass for the field and armature windings. The thermal mass is the energy required to raise the temperature by one degree.

#### **Initial temperatures – Initial temperatures**

[25, 25] C (default)

A 1 by 2 row vector that defines the temperature of the field and armature thermal ports at the start of simulation.

## **Version History**

**Introduced in R2008a**

### **References**

[1] Bolton, W. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 3rd edition Pearson Education, 2004..

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

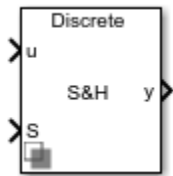
### **See Also**

DC Motor | Induction Machine (Single-Phase) | Motor & Drive (System Level) | Universal Motor

## Signal Sample and Hold

Discrete-time or Continuous-time sample and hold input signal

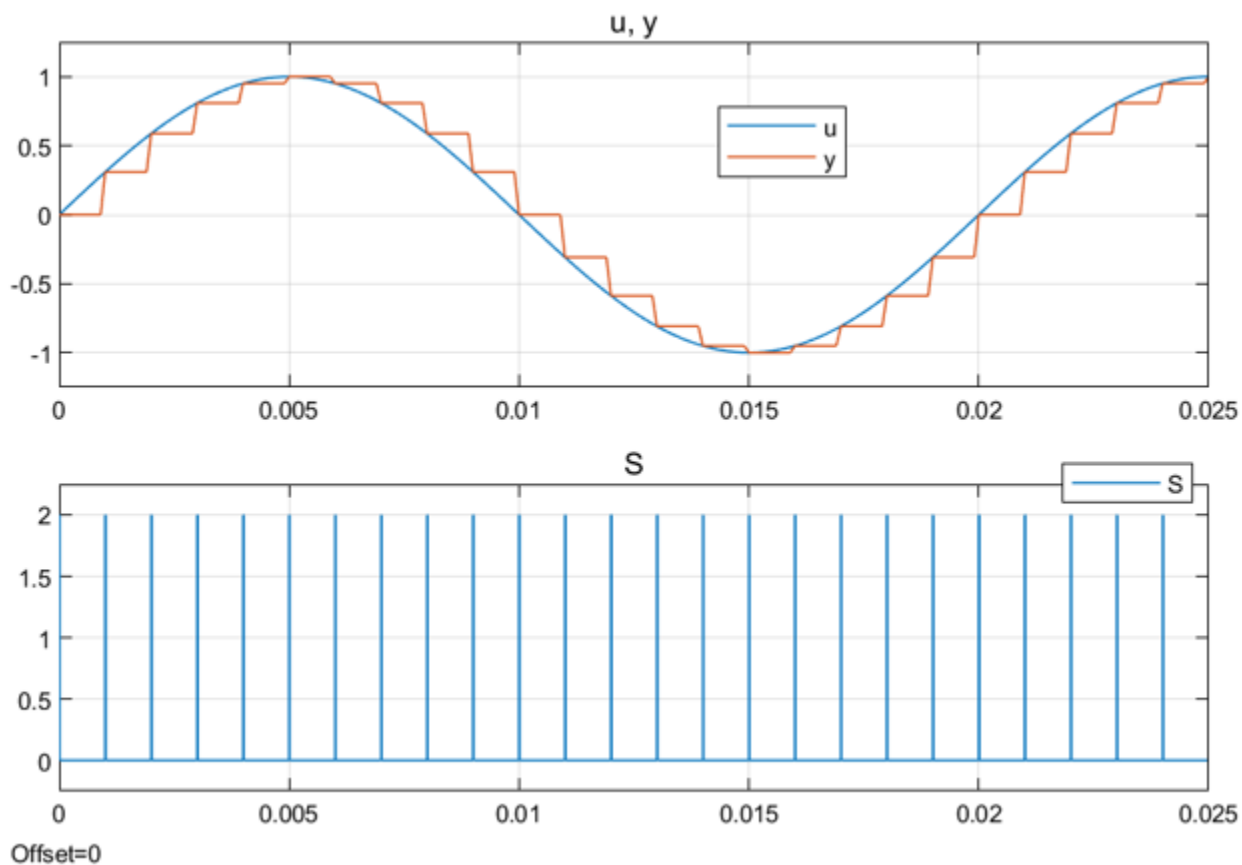
**Library:** Simscape / Electrical / Control / General Control



### Description

The Signal Sample and Hold block implements a signal sample and hold in either discrete or continuous time.

When input  $S$  is true, output  $y$  is equal to input  $u$ . When input  $S$  is false, the block holds the output until  $S$  becomes true again.



## Ports

### Input

**u – Input signal**

scalar | vector

Input signal.

Data Types: single | double

**S – Control signal**

0 | 1

Sample pulse of 0 for false or 1 for true.

Data Types: Boolean

### Output

**y – Output signal**

scalar | vector

Output signal.

Data Types: single | double

## Parameters

**Initial condition – Initial output value**

scalar | vector

Specify initial condition. The value must be a scalar or a vector of the same size as the input signal.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

**Introduced in R2018b**

## **Extended Capabilities**

### **C/C++ Code Generation**

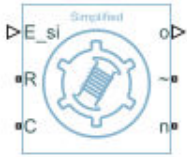
Generate C and C++ code using Simulink® Coder™.

### **See Also**

# Simplified Synchronous Machine

Simplified synchronous machine with electromotive force

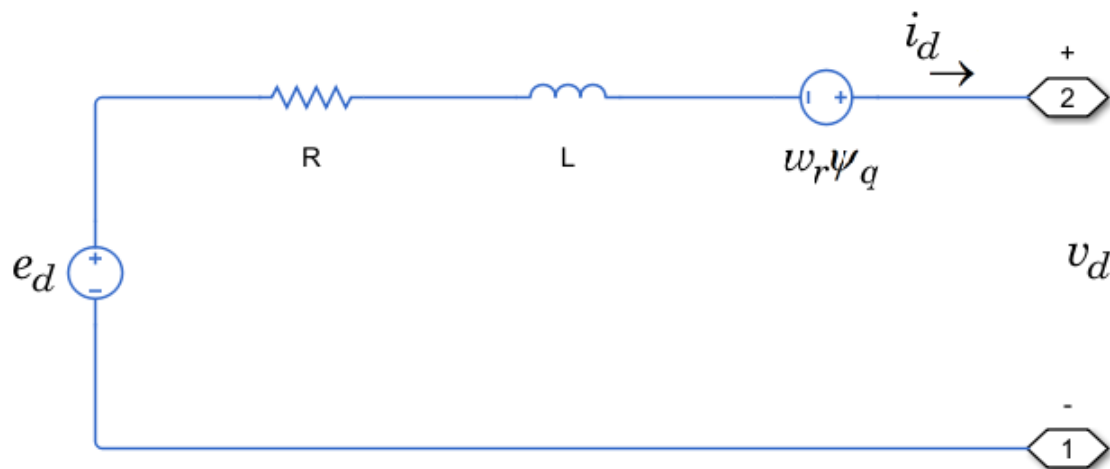
**Library:** Simscape / Electrical / Electromechanical / Synchronous

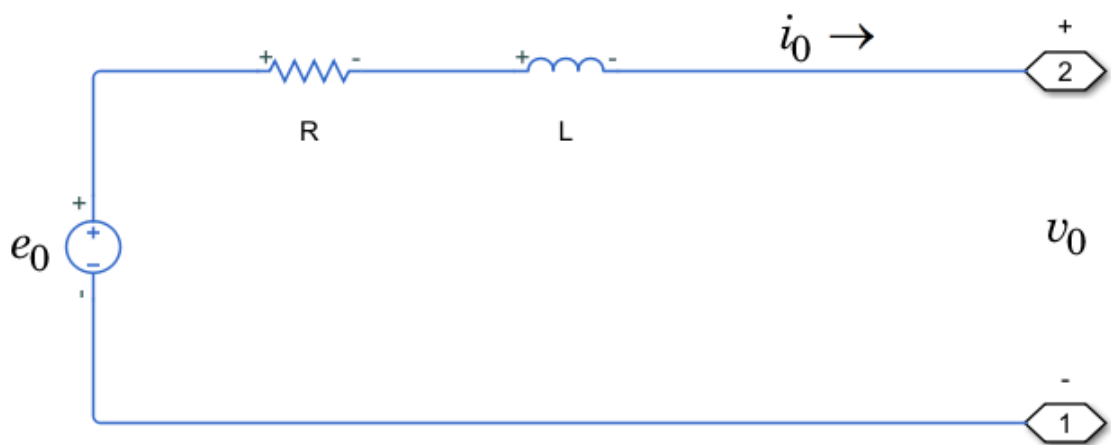
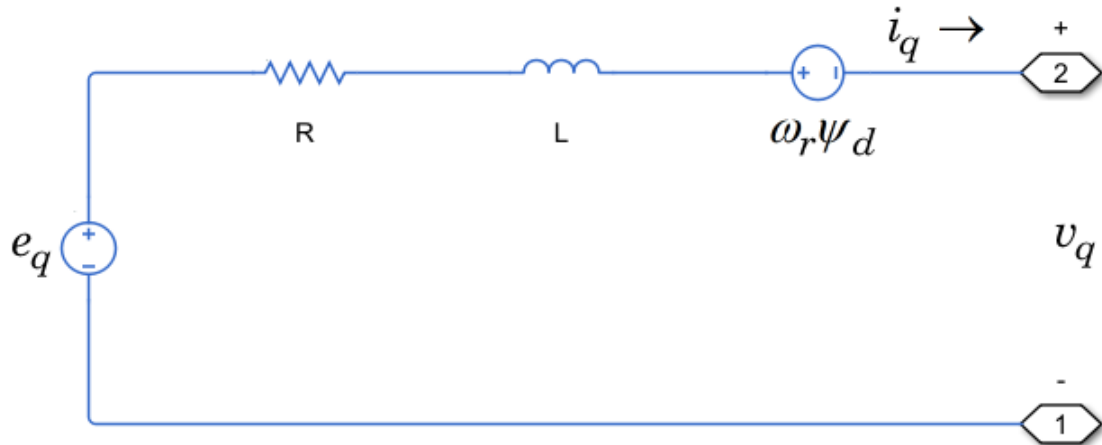


## Description

The Simplified Synchronous Machine block models a simplified synchronous machine with a voltage source that represents electromotive force (EMF). You can specify the internal resistance and inductance with per-unit or SI parameters.

The equivalent circuits of the simplified synchronous machine for the direct axis, the quadrature axis, and the zero sequence are:





**Equations**

The simplified synchronous machine equations are expressed with respect to a rotating reference frame, which is defined by:

$$\theta_e(t) = N\theta_r(t),$$

where:

- $\theta_e$  is the electrical angle.
- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor angle.

The Park transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. The Park transformation is defined by:

$$P_s = \frac{2}{3} \begin{bmatrix} -\cos\theta_e & -\cos(\theta_e - \frac{2\pi}{3}) & -\cos(\theta_e + \frac{2\pi}{3}) \\ \sin\theta_e & \sin(\theta_e - \frac{2\pi}{3}) & \sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformation is used to define the per-unit simplified synchronous machine equations. The voltage equations are defined by:

$$e_d = \frac{1}{\omega_{base}} \frac{d\psi_d}{dt} - \Psi_q \omega_r + Ri_d + v_d$$

$$e_q = \frac{1}{\omega_{base}} \frac{d\psi_q}{dt} + \Psi_d \omega_r + Ri_q + v_q$$

$$e_0 = \frac{1}{\omega_{base}} \frac{d\psi_0}{dt} + Ri_0 + v_0$$

where:

- $e_d$ ,  $e_q$ , and  $e_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence voltages, defined by:

$$\begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = P_s \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}.$$

$e_a$ ,  $e_b$ , and  $e_c$  are the per-unit internal voltage sources, defined by:

$$e_a = E_{pu} \sin\theta_e$$

$$e_b = E_{pu} \sin(\theta_e - 120^\circ)$$

$$e_c = E_{pu} \sin(\theta_e + 120^\circ)$$

$e_{pu}$  is the per-unit amplitude of the internal generated voltage.

- $v_d$ ,  $v_q$ , and  $v_0$  are defined by:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages measured from port  $\sim$  to neutral port  $\mathbf{n}$ .

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_d$ ,  $\psi_q$ , and  $\psi_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence flux linkages.
- $\omega_r$  is the per-unit rotor rotational speed.
- $R$  is the stator resistance.
- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents, defined by:

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing out of port ~.

The stator flux linkage equations are defined by

$$\psi_d = L \cdot i_d$$

$$\psi_q = L \cdot i_q$$

$$\psi_0 = L \cdot i_0$$

where  $L$  is the stator leakage inductance.

The power equation of the simplified synchronous machine in per-unit is defined by:

$$P = e_d i_d + e_q i_q.$$

### Plotting and Display Options

You can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select one of these options:

- **Display Base Values** — Displays the machine per-unit base values in the MATLAB Command Window.
- **Display Associated Initial Conditions** — Displays associated initial conditions in the MATLAB Command Window.

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and machine temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain thermal ports.
- `Show thermal port` — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

For this block, the **Initial Targets** settings are visible only if, in the **Initial Conditions** section, you set the **Initialization option** parameter to `Set targets for rotor angle and Park's transform variables`.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.



## Ports

### Input

#### **E\_si — Amplitude of internal generated voltage (phase-to-neutral)**

vector of physical signals

Physical signal port associated with the amplitude of the internal generated voltage (phase-to-neutral), specified as a vector of physical signals.

#### **Dependencies**

To enable this port, set **Parameterization unit** to SI.

#### **E\_pu — Amplitude of internal generated voltage, in per-unit**

vector of physical signals

Physical signal port associated with the per-unit amplitude of the internal generated voltage, specified as a vector of physical signals.

#### **Dependencies**

To enable this port, set **Parameterization unit** to Per unit.

### Output

#### **o — Machine per-unit measurements**

vector of physical signals

Physical signal port associated with the machine per-unit measurements, returned as a vector of physical signals. The vector elements are:

- Electromotive force associated with phase a,  $pu\_EMF(1)$
- Electromotive force associated with phase b,  $pu\_EMF(2)$
- Electromotive force associated with phase c,  $pu\_EMF(3)$
- Electrical torque,  $pu\_torque$
- Rotor velocity,  $pu\_velocity$
- Stator d-axis voltage,  $pu\_vd$
- Stator q-axis voltage,  $pu\_vq$
- Stator zero-sequence voltage,  $pu\_e0$
- Stator d-axis current,  $pu\_id$
- Stator q-axis current,  $pu\_iq$
- Stator zero-sequence current,  $pu\_i0$
- Rotor electrical angle,  $electrical\_angle\_out$

To connect to this port, use the Synchronous Machine Measurement block.

### Conserving

#### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

**C – Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

**~ – Stator windings**

electrical

Expandable three-phase electrical port associated with the stator windings.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**n – Neutral phase**

electrical

Electrical conserving port associated with the neutral point of the wye winding configuration.

**a – a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b – b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c – c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA – a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters**

**Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and machine temperature.

**Main**

**Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power — Rated apparent power**

555e6 V\*A (default) | scalar

Rated apparent power.

**Rated voltage — RMS rated line-to-line voltage**

24e3 V (default) | scalar

RMS rated line-to-line voltage.

**Rated electrical frequency — Nominal electrical frequency**

60 Hz (default) | scalar

Nominal electrical frequency at which rated apparent power is quoted.

**Number of pole pairs — Number of pole pairs**

2 (default) | scalar

Number of machine pole pairs.

**Parameterization unit — Block parameterization**

SI (default) | Per unit

Block parameterization method. Options are:

- SI
- Per unit

**Internal resistance (pu) — Internal per-unit resistance**

0.05 (default) | scalar

Internal per-unit resistance.

**Dependencies**

To enable this parameter, set **Parameterization unit** to Per unit.

**Internal inductance (pu) — Internal per-unit inductance**

0.3 (default) | scalar

Internal per-unit inductance in per-unit.

**Dependencies**

To enable this parameter, set **Parameterization unit** to Per unit.

**Internal resistance — Internal resistance**

0.0519 Ohm (default) | scalar

Internal resistance.

**Dependencies**

To enable this parameter, set **Parameterization unit** to SI.

**Internal inductance — Internal inductance**

8.2589e-04 H (default) | scalar

Internal inductance.

**Dependencies**

To enable this parameter, set **Parameterization unit** to SI.

**Initial Conditions****Initialization option — Initialization option**

Set real power, reactive power, terminal voltage, and terminal phase (default) |  
Set targets for rotor angle and Park's transform variables | Set targets for  
load flow variables

Model for specifying values for certain parameters and variables at the start of simulation. To:

- Set an operating point regardless of the connected network, select Set real power, reactive power, terminal voltage and terminal phase.

- Specify the priority and initial target values for block variables before simulation using the **Variables** settings, select Set targets for rotor angle and Park's transform variables. For more information, see "Set Priority and Initial Target for Block Variables".
- Select a bus type and specify the related parameters for a load-flow analysis in the **Initial Conditions** settings, select Set targets for load flow variables.

#### Dependencies

If you set this parameter to:

- Set targets for rotor angle and Park's transform variables — The **Variables** settings become visible.
- Set real power, reactive power, terminal voltage, and terminal phase — Related parameters become visible.
- Set targets for load flow variables — Related parameters become visible.

#### Source type — Voltage source model

Swing bus (default) | PV bus | PQ bus

Type of voltage source that the block models.

#### Dependencies

To enable this parameter set **Initialization option** to Set targets for load flow variables and **Source type** to Swing bus or PV bus.

#### Terminal voltage magnitude — Terminal voltage magnitude

24e3 V (default)

Terminal voltage magnitude.

#### Dependencies

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase or **Initialization option** to Set targets for load flow variables and **Source type** to Swing bus or PV bus.

#### Terminal voltage angle — Terminal voltage angle

0 deg (default)

Terminal voltage angle.

#### Dependencies

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase or **Initialization option** to Set targets for load flow variables and **Source type** to Swing bus.

#### Active power generated — Active power generated

500e6 V\*A (default)

Active power generated.

**Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase or **Initialization option** to Set targets for load flow variables and **Source type** to PV bus or PQ bus.

**Reactive power generated — Reactive power generated**

0 V\*A (default)

Reactive power generated.

**Dependencies****Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase or **Initialization option** to Set targets for load flow variables and **Source type** to PQ bus.

**Minimum terminal voltage magnitude (pu) — Minimum terminal voltage magnitude**

0.95 (default)

Per-unit minimum steady-state voltage magnitude.

**Dependencies**

To enable this parameter, set **Initialization option** to Set targets for load flow variables and **Source type** to PQ bus.

**Phase search range at terminals — Phase search range at terminals**

[-inf, inf] deg (default)

Vector that defines the phase angle search range.

**Dependencies**

To enable this parameter, set **Initialization option** to Set targets for load flow variables and **Source type** to PV bus or PQ bus.

**Parasitic conductance to electrical reference — Parasitic conductance to electrical reference**

1e-6 S (default)

Parasitic conductance to the electrical reference.

**Dependencies**

To enable this parameter, set **Initialization option** to Set targets for load flow variables.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature — Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

**Introduced in R2020b****Electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`. Also, the simlog format is identical for `Composite three-phase ports` and `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Simplified Synchronous Machine Measurement

# Simplified Synchronous Machine Measurement

Per-unit measurement from simplified synchronous machine

**Library:** Simscape / Electrical / Electromechanical / Synchronous



## Description

The Simplified Synchronous Machine Measurement block outputs a per-unit measurement associated with a connected Simplified Synchronous Machine block. The input of the Simplified Synchronous Machine Measurement block connects to the pu output port of the Simplified Synchronous Machine block.

You set the **Output** parameter to a per-unit measurement associated with the simplified synchronous machine. Based on the value you select, the Simplified Synchronous Machine Measurement block:

- Directly outputs the value of an element in the input signal vector
- Calculates the per-unit measurement by using values of elements in the input signal vector in mathematical expressions.

The Simplified Synchronous Machine Measurement block outputs a per-unit measurement from the simplified synchronous machine according to the output value expressions in the table. For example, when you set **Output** to **Stator d-axis voltage**, the block directly outputs the value of the pu\_ed element in the input signal vector. However, when you set **Output** to **Reactive power**, the block calculates the value from the pu\_ed, pu\_eq, pu\_id, and pu\_iq elements.

Output Parameter Setting	Output Value Expression
EMF phase A	pu_EMFa
EMF phase B	pu_EMFb
EMF phase C	pu_EMFc
Electrical torque	pu_torque
Rotor velocity	pu_velocity
Stator d-axis voltage	pu_vd
Stator q-axis voltage	pu_vq
Stator zero-sequence voltage	pu_v0
Stator d-axis current	pu_id
Stator q-axis current	pu_iq
Stator zero-sequence current	pu_i0
Apparent power	$\sqrt{pu_{Pt}^2 + pu_{Qt}^2}$
Real power	$pu_{Pt} = (pu_{vd} * pu_{id}) + (pu_{vq} * pu_{iq}) + 2(pu_{v0} * pu_{i0})$



Output Parameter Setting	Output Value Expression
Reactive power	$pu_{Qt} = (pu_{vq} * pu_{id}) - (pu_{vd} * pu_{iq})$
Terminal voltage	$\sqrt{(pu_{vd}^2 + pu_{vq}^2)}$
Terminal current	$\sqrt{(pu_{id}^2 + pu_{iq}^2)}$
Power factor angle	$power\_factor\_angle = \text{atan2}(pu_{Qt}, pu_{Pt})$
Power factor	$\cos(power\_factor\_angle)$
Load angle	$load\_angle(rad) = \text{atan2}(pu_{vd}, pu_{vq})$
Rotor electrical angle	$\theta_e(t) = N\theta_r(t)$

## Ports

### Input

#### pu — Per-unit measurements from a connected synchronous machine

vector of physical signal

Physical signal associated with the per-unit measurements from a connected Simplified Synchronous Machine, specified as:

- pu\_EMFa
- pu\_EMFb
- pu\_EMFc
- pu\_torque
- pu\_velocity
- pu\_vd
- pu\_vq
- pu\_v0
- pu\_id
- pu\_iq
- pu\_i0

### Output

#### o — Per-unit measurement output port

physical signal

Physical signal output port associated with the per-unit measurement, returned as a physical signal.

## Parameters

#### Output — Per-unit measurement from synchronous machine

EMF phase A (default) | EMF phase B | EMF phase C | Electrical torque | Rotor velocity | Stator d-axis voltage | Stator q-axis voltage | Stator zero-sequence voltage | Stator d-axis current | Stator q-axis current | Stator zero-sequence current |

Apparent power | Real power | Reactive power | Terminal voltage | Terminal current | Power factor angle | Power factor | Load angle | Rotor electrical angle

Per-unit measurement from simplified synchronous machine. The default value is EMF phase A.

## **Version History**

**Introduced in R2020b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

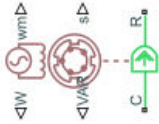
## **See Also**

Simplified Synchronous Machine

# Simplified Induction Motor

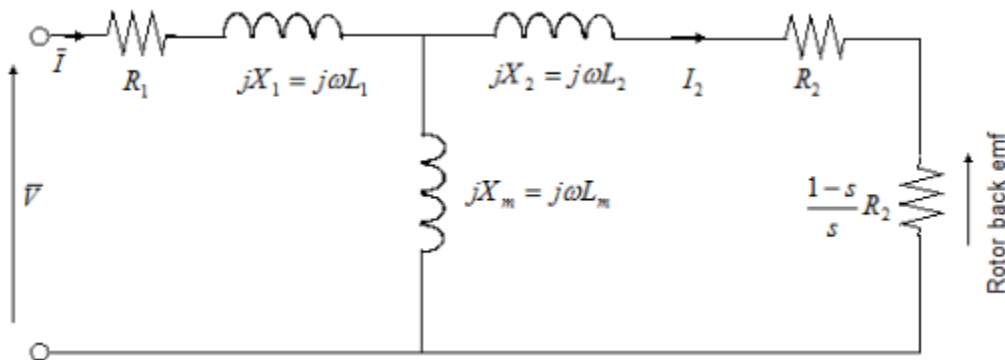
Induction motor powered by ideal AC supply

**Library:** Simscape / Electrical / Electromechanical / Asynchronous



## Description

The Simplified Induction Motor block represents the electrical and torque characteristics of an induction motor powered by an ideal AC supply. The following figure shows the equivalent circuit model of the Simplified Induction Motor block.



In the figure:

- $R_1$  is the stator resistance.
- $R_2$  is the rotor resistance with respect to the stator.
- $L_1$  is the stator inductance.
- $L_2$  is the rotor inductance with respect to the stator.
- $L_m$  is magnetizing inductance.
- $s$  is the rotor slip.
- $\bar{V}$  and  $\bar{I}$  are the sinusoidal supply voltage and current phasors.

Rotor slip  $s$  is defined in terms of the mechanical rotational speed  $\omega_m$ , the number of pole pairs  $p$ , and the electrical supply frequency  $\omega$  by

$$s = 1 - \frac{p\omega_m}{\omega}$$

This means that the slip is one when starting, and zero when running synchronously with the supply frequency.

For an  $n$ -phase induction motor the torque-speed relationship is given by:

$$T = \frac{npR_2}{s\omega} \frac{V_{rms}^2}{\left(R_1 + R_2 + \frac{1-s}{s}R_2\right)^2 + (X_1 + X_2)^2}$$

where:

- $V_{rms}$  is the line-neutral supply voltage for a star-configuration induction motor, and the line-to-line voltage for a delta-configuration induction motor.
- $n$  is the number of phases.

You can parameterize this block in terms of the preceding equivalent circuit model parameters or in terms of the motor ratings the block uses to derive these parameters.

This block produces a positive torque acting from the mechanical C to R ports.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Assumptions and Limitations

- The block does not model the starting mechanism for a single-phase induction motor.
- When you parameterize the block by motor ratings, the block derives the equivalent circuit model parameters by assuming that the effect of the magnetizing inductance  $L_m$  is negligible, and the magnetizing inductance is not included in the simulated component.

### Ports

#### Output

##### W — Real power

physical

Physical signal port associated with the motor real power.

##### wm — Mechanical speed

physical

Physical signal port associated with the motor mechanical speed.

##### VAR — Imaginary power

physical

Physical signal port associated with the motor imaginary power.

##### s — Motor slip

physical

Physical signal port associated with the motor slip.

### Conserving

#### R — Rotor

mechanical

Mechanical rotational conserving port associated with the rotor.

#### C — Case

mechanical

Mechanical rotational conserving port associated with the motor case.

#### H1 — Rotor

thermal

Stator thermal port.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

#### H2 — Rotor

thermal

Rotor thermal port.

#### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Whether to enable thermal ports

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

### Electrical Torque

#### Model parameterization — Block parameterization

By motor ratings (default) | By equivalent circuit parameters

Select one of the following methods for block parameterization:

- **By motor ratings** — Provide electrical torque parameters that the block converts to an equivalent circuit model of the motor assuming that the effect of the magnetizing inductance  $L_m$  is negligible.
- **By equivalent circuit parameters** — Provide electrical parameters for an equivalent circuit model of the motor.

#### Stator resistance R1 — Stator winding resistance

1 Ohm (default)

Resistance of the stator winding. The default value is 1  $\Omega$ .

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Rotor resistance R2 — Rotor resistance**

1 Ohm (default)

Resistance of the rotor, specified with respect to the stator.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Stator inductance L1 — Stator winding inductance**

0.02 H (default)

Inductance of the stator winding.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Rotor inductance L2 — Rotor inductance**

0.02 H (default)

Inductance of the rotor, specified with respect to the stator.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Magnetizing inductance Lm — Stator magnetizing inductance**

0.5 H (default)

Magnetizing inductance of the stator. Its value is hard to estimate from motor parameters, but the effect is usually small. If you do not know its value, use a typical value of 25 times the **Stator inductance L1** value.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Rated mechanical power — Motor mechanical power**

825 W (default)

Mechanical power the motor delivers when running at the rated speed.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**Rated speed — Rated motor speed**

3500 rpm (default)

Speed at which the motor delivers the specified **Rated mechanical power** value.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**Rated RMS line-to-line voltage — Rated line-to-line voltage**

200 V (default)

Line-to-line voltage at which the motor ratings are specified.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**Rated supply frequency — Frequency of the AC supply voltage**

60 Hz (default)

Frequency of the AC supply voltage at which the motor ratings are specified.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**Rated RMS line current — Rated line current**

2.7 A (default)

Line current at which the motor delivers the specified **Rated mechanical power** value.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**R1 parameterization — Equivalent circuit resistance parameterization**

From motor efficiency (default) | From power factor | Use measured stator resistance R1

Select one of the following parameterizations for the equivalent circuit resistance,  $R_1$ , of the motor:

- From motor efficiency — Calculate  $R_1$  from the motor efficiency. This is the default method.
- From power factor — Calculate  $R_1$  from the motor power factor.
- Use measured stator resistance R1 — Measure  $R_1$  directly.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings.

**Motor efficiency (percent) — Motor efficiency**

95 (default)

The percentage of input power to the motor that gets delivered to the mechanical load when running at the **Rated speed** value.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings and the **R1 parameterization** parameter to From motor efficiency.

**Motor power factor — Motor power factor**

0.93 (default)

The cosine of the angle by which the supply current lags the supply voltage when running at the **Rated mechanical power** value.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings and the **R1 parameterization** parameter to From power factor.

**Measured stator resistance R1 — Measured stator resistance**

1 Ohm (default)

The measured stator resistance.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By motor ratings and the **R1 parameterization** parameter to Use measured stator resistance R1.

**Number of pole pairs — Number of motor pole pairs**

1 (default)

Total number of pole pairs for the motor.

**Number of phases — Number of supply phases**

3 (default)

Number of supply phases.

**Stator connections — Motor stator configurations**

Delta configuration (default) | Star configuration

Select one of the following motor configurations:

- Delta configuration — Connect the motor stator windings in delta configuration.
- Star configuration — Connect the motor stator windings in star configuration.

**Power Supply****Supply RMS line-to-line voltage — Supply RMS line-to-line voltage**

200 V (default)

The line-to-line voltage that supplies the motor.

**Supply frequency — AC supply frequency**

60 Hz (default)



Frequency of the AC supply voltage.

## Mechanical

### Rotor inertia – Rotor inertia

1e-3 kg\*m<sup>2</sup> (default)

Rotor inertia. The value can be zero.

### Rotor damping – Rotor damping

1e-4 N\*m/(rad/s) (default)

Rotor damping. The value can be zero.

### Initial rotor speed – Initial rotor speed

0 rpm (default)

Speed of the rotor at the start of the simulation.

## Temperature Dependence

This tab appears only for blocks with exposed thermal ports.

### Stator and rotor resistance temperature coefficients, [alpha\_1 alpha\_2] – Stator and rotor resistance temperature coefficients

[.00393, .00393] 1/K (default)

A 1 by 2 row vector defining the coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The first element corresponds to the stator, and the second to rotor. The default value is for copper.

### Measurement temperature – Measurement temperature

25 C (default)

The temperature for which motor parameters are defined.

## Thermal Port

This tab appears only for blocks with exposed thermal ports.

### Stator and rotor thermal masses, [M\_1 M\_2] – Stator and rotor thermal masses

[100, 100] J/K (default)

A 1 by 2 row vector defining the thermal mass for the stator and rotor windings. The thermal mass is the energy required to raise the temperature by one degree.

### Stator and rotor initial temperatures, [T\_1 T\_2] – Stator and rotor initial temperatures

[25, 25] C (default)

A 1 by 2 row vector defining the temperature of the stator and rotor thermal ports at the start of simulation.

## **Version History**

**Introduced in R2008a**

## **References**

[1] S.E. Lyshevski. *Electromechanical Systems, Electric Machines, and Applied Mechatronics*, CRC, 1999.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

DC Motor | Motor & Drive (System Level) | Shunt Motor | Universal Motor

# Sinusoidal Current Source

Damped sinusoidal current source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Sinusoidal Current Source block represents a damped sinusoidal current source whose output current is independent of the voltage across the terminals of the source. The following equations describe the current through the source as a function of time:

$$I_{out}(Time \leq TD) = IO$$

$$I_{out}(Time > TD) = IO + IA * e^{-(Time - TD) * DF} * \sin(2\pi * FREQ * (Time - TD))$$

where:

- $IO$  is the **Current offset**, **IO** parameter value.
- $IA$  is the **Sinusoidal amplitude**, **IA** parameter value.
- $FREQ$  is the **Sinusoidal frequency**, **FREQ** parameter value.
- $TD$  is the **Time delay**, **TD** parameter value.
- $DF$  is the **Damping factor**, **DF** parameter value.

The block uses a small conductance internally to prevent numerical simulation issues. The conductance connects the + and - ports of the device and has a conductance  $GMIN$ :

- By default,  $GMIN$  matches the **GMIN** parameter of the Environment Parameters block, whose default value is  $1e-12$ .
- To change  $GMIN$ , add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the sinusoidal current source positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the sinusoidal current source negative terminal.

## Parameters

### Current offset, $I_0$ — Current offset

0 A (default)

The magnitude of the time-independent part of the output current.

### Sinusoidal amplitude, $I_A$ — Sinusoidal amplitude

0 A (default)

The magnitude of the sinusoidal part of the output current.

### Sinusoidal frequency, $FREQ$ — Sinusoidal frequency

1e6 Hz (default)

The frequency of the output sine wave. The value can be less than 0.

### Time delay, $TD$ — Time delay

0 s (default)

The time at which the sine wave first starts. A value less than 0 indicates a phase shift of 180 degrees.

### Damping factor, $DF$ — Damping factor

0 1/s (default) | nonnegative scalar

The exponential damping factor for the sine wave to produce the output current. If  $DF$  is greater than or equal to 0, the exponential part of the equation is always damping for time greater than  $TD$ . The value must be greater than or equal to 0.

### Include parallel conductance, $GMIN$ — Whether to include parallel conductance

Yes (default) | No

Specify if the current source includes a parallel conductance. The value of the parallel conductance,  $GMIN$ , depends on whether your model contains an Environment Parameters block.

This table shows the possible values of the parallel conductance,  $GMIN$ :

#### Values of parallel conductance, $GMIN$

Include parallel conductance, $GMIN$		
Yes		No
<b>Model contains Environment Parameters block</b>	<b>Model does not contain Environment Parameters block</b>	Parallel conductance, $GMIN$ , is not included and it does not affect the current source.
$GMIN$ is equal to the value of the <b><math>GMIN</math></b> parameter in the Environment Parameters block.	$GMIN = 1e-12$ 0hm	

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

DC Current Source | Environment Parameters | Exponential Current Source | Pulse Current Source | Piecewise Linear Current Source | SFFM Current Source | Sinusoidal Voltage Source

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Sinusoidal Voltage Source

Damped sinusoidal voltage source

**Library:** Simscape / Electrical / Additional Components / SPICE Sources



## Description

The Sinusoidal Voltage Source block represents a damped sinusoidal voltage source whose output voltage is independent of the current through the source. The following equations describe the output as a function of time:

$$V_{out}(Time \leq TD) = VO$$

$$V_{out}(Time > TD) = VO + VA * e^{-(Time - TD)*DF} * \sin(2\pi * FREQ * (Time - TD))$$

where:

- $VO$  is the **Voltage offset**, **VO** parameter value.
- $VA$  is the **Sinusoidal amplitude**, **VA** parameter value.
- $FREQ$  is the **Sinusoidal frequency**, **FREQ** parameter value.
- $TD$  is the **Time delay**, **TD** parameter value.
- $DF$  is the **Damping factor**, **DF** parameter value.

## Ports

### Conserving

#### + – Positive voltage

electrical

Electrical conserving port associated with the sinusoidal voltage source positive voltage

#### - – Negative voltage

electrical

Electrical conserving port associated with the sinusoidal voltage source negative voltage

## Parameters

### Voltage offset, $VO$ – Voltage offset

0 V (default)

Magnitude of the time-independent part of the output voltage.

**Sinusoidal amplitude, VA — Sinusoidal amplitude** $0$  V (default)

Magnitude of the sinusoidal part of the output voltage.

**Sinusoidal frequency, FREQ — Sinusoidal frequency** $1e6$  Hz (default)

Frequency of the output sine wave. The value can be less than  $0$ .

**Time delay, TD — Time delay** $0$  s (default)

Time at which the sine wave first starts. A value less than  $0$  indicates a phase shift of 180 degrees.

**Damping factor, DF — Damping factor** $0$  1/s (default) | nonnegative scalar

Exponential damping factor for the sine wave to produce the output voltage. If  $DF$  is greater than or equal to  $0$ , the exponential part of the equation is always damping for time greater than  $TD$ . The value must be greater than or equal to  $0$ .

## Version History

Introduced in R2008a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

[DC Voltage Source](#) | [Environment Parameters](#) | [Exponential Voltage Source](#) | [Pulse Voltage Source](#) | [Piecewise Linear Voltage Source](#) | [SFFM Voltage Source](#) | [Sinusoidal Current Source](#)

**Functions**

`subcircuit2ssc`

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## PMSM (Single-Phase)

Single-phase permanent magnet synchronous motor

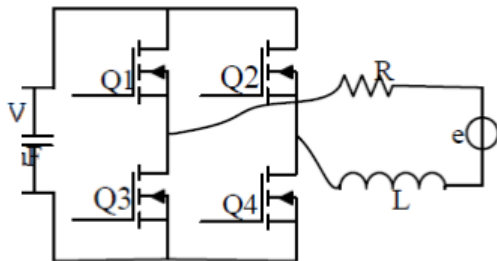
**Library:** Simscape / Electrical / Electromechanical / Permanent Magnet



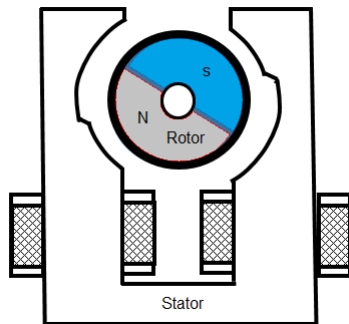
### Description

The PMSM (Single-Phase) represents a single-phase permanent magnet synchronous motor (PMSM), a type of DC motor that is useful for automation applications.

The figure shows the topology of the single-phase PMSM drive.

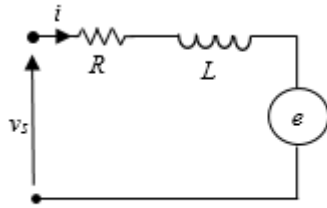


The figure shows the motor construction with a single pole-pair on the rotor. Single-phase PMSMs are not self-starting unless the air gap is asymmetrical.



The figure shows the equivalent circuit for the PMSM (Single-Phase) block.





### Equations

The motor voltage equations are

$$v_s = Ri + L \frac{di}{dt} + e$$

and

$$v_s = V_m \sin(\omega_s t + \varepsilon),$$

where:

- $v_s$  is the supply voltage.
- $i$  is the instantaneous motor current.
- $R$  is the resistance of the windings.
- $L$  is the self-inductance of the windings.
- $e$  is the back-electromotive force (BEMF).
- $\omega_s$  is the angular frequency of the supply voltage.
- $\varepsilon$  is the angle of the supply voltage.

The back electro-motive force (BEMF) is

$$e = k_e \omega_e \sin(\theta_e),$$

where:

- $\omega_e$  is the rotor electrical angular velocity.
- $\theta_e$  is the rotor electrical angle.
- $k_e$  is the BEMF constant.

Due to the large low-permeability gaps between the stator and rotor, the saturation can be neglected. Therefore, the electric torque equations are

$$T_e = p i \psi_m \sin(\theta_e)$$

and

$$\psi_m = \frac{k_e}{p},$$

where:

- $T_e$  is the electric torque.
- $\psi_m$  is the permanent magnet flux linkage.
- $p$  is the number of pole pairs.

The mechanical equation is

$$J_m \frac{d\omega_r}{dt} = T_e - T_L - B_m \omega_r,$$

where:

- $J_m$  is the rotor inertia.
- $T_L$  is the torque load.
- $B_m$  is the friction coefficient.
- $\omega_r$  is the rotor mechanical angular velocity.

### Model Thermal Effects

You can expose thermal ports to simulate the effects of generated heat and motor temperature. To expose the thermal ports and the **Thermal** parameters, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain thermal ports.
- `Show thermal port` — The block contains thermal conserving ports for the stator and the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Limitations and Assumptions

- The machine air gap is free of saliency effects.
- The stator current has negligible effect on the flux distribution under normal operating conditions.
- The hysteresis, saturation effects, and eddy currents are neglected.

### Ports

#### Conserving

##### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

### **C — Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

### **+ — Negative**

electrical

Electrical conserving port associated with the supply positive terminal.

### **- — Positive**

electrical

Electrical conserving port associated with the supply negative terminal.

### **HS — Stator thermal port**

thermal

Thermal conserving port associated with the stator.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

### **HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

## **Parameters**

### **Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and motor temperature.

#### **Main**

### **Number of pole pairs — Rotor pole pairs**

2 (default) | positive integer

Number of permanent magnet pole pairs on the rotor.

### **Permanent magnet flux linkage parameterization — Parameterization method**

Specify flux linkage (default) | Specify back EMF constant

Method for parameterizing the stator.

**Dependencies**

Selecting Specify flux linkage exposes the **Permanent magnet flux linkage** parameter.

Selecting Specify back EMF constant exposes the **Back EMF constant** parameter.

**Permanent magnet flux linkage – Flux linkage**

0.2 Wb (default) | positive scalar

Peak permanent magnet flux linkage.

**Dependencies**

Selecting Specify flux linkage for the **Permanent magnet flux linkage parameterization** parameter exposes the **Permanent magnet flux linkage** parameter.

**Back EMF constant – Back-electromotive force constant**

0.4 V/(rad/s) (default) | positive scalar

Back-electromotive force constant.

**Dependencies**

Selecting Specify back EMF constant for the **Permanent magnet flux linkage parameterization** parameter exposes the **Back EMF constant** parameter.

**Inductance of stator coil – Inductance**

1.15 H (default) | positive scalar

The direct-axis inductance.

**Resistance of stator coil – Resistance**

150 Ohm (default) | positive scalar

Resistance of each of the stator windings.

**Rotor position at standstill due to asymmetric airgap – Rotor angular position at standstill**

45 deg (default) | positive scalar in the interval [0, 360] deg

Rotor angular position at standstill due to asymmetric air gap.

**Mechanical****Rotor inertia – Rotor moment of inertia**

1e-6 kg\*m<sup>2</sup> (default) | positive scalar

Inertia of the rotor attached to mechanical translational port **R**.

**Rotor damping – Rotor damping**

0 N\*m/(rad/s) (default) | positive scalar

Damping of the rotor.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature – Rated temperature**

298.15 K (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Permanent magnet flux temperature coefficient – Permanent magnet flux temperature coefficient**

-0.001 1/K (default)

Fractional rate of change of permanent magnet flux density with temperature. Use this parameter to linearly reduce the torque and the induced back EMF as temperature rises.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

Introduced in R2018b

## References

- [1] Ertugrul, N. and C. Doudle. “Dynamic analysis of a single-phase line-starting permanent magnet synchronous motor.” *Proceedings of International Conference on Power Electronics, Drives and Energy Systems for Industrial Growth*. Vol. 1, 1996, pp. 603-609.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

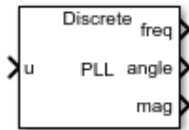
**See Also**

BLDC | PMSM

## Sinusoidal Measurement (PLL)

Estimate sinusoidal characteristics using a phase-locked loop

**Library:** Simscape / Electrical / Control / Measurements



### Description

The Sinusoidal Measurement (PLL) block estimates the frequency, phase angle, and magnitude of a single-phase sinusoidal signal or individual phases of a multiphase sinusoidal signal. The block uses an enhanced phase-locked loop (PLL) strategy to estimate these sinusoidal characteristics of the input signal.

Use this block in control applications when the frequency, phase angle, or magnitude is required and cannot be measured directly. To provide faster phase locking for balanced three-phase input signals, use the Three-Phase Sinusoidal Measurement (PLL) block.

### Equations

The phase-locked loop generates a sinusoid that approximates the input signal  $u(t)$  with the form:

$$y(t) = A(t)\sin\left(\phi_0 + \int 2\pi f(t)dt\right),$$

where:

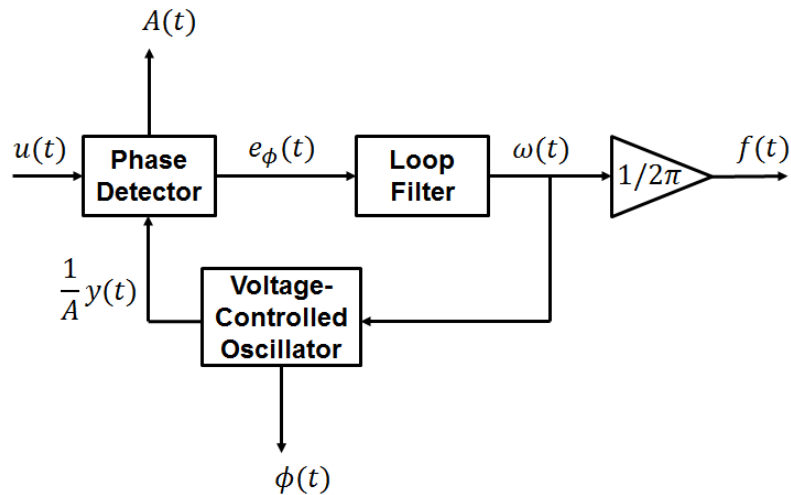
- $y$  is the estimate of the input signal.
- $A$  is the estimate of the amplitude of the input signal.
- $\phi_0$  is the initial phase angle of the input signal.

The estimated phase angle  $\phi$  is the angle of this generated sinusoid:

$$\phi(t) = \phi_0 + \int 2\pi f(t)dt,$$

where  $f$  is the frequency of the sinusoid, and  $\phi_0$  is the initial phase angle.

This diagram shows the overall structure of the phase-locked loop.



In the diagram:

- The phase detector produces an error signal relative to the phase difference  $e_\phi$  between the input sinusoid  $u$  and the synthesized sinusoid  $y$ . It also outputs an estimate of the amplitude  $A$ .
- The loop filter provides an estimate of the input angular frequency  $\omega$  by filtering out the high-frequency components of the phase difference. The block also outputs the converted frequency  $f$  in Hz.
- The voltage-controlled oscillator integrates the angular speed to produce the phase estimate  $\phi$ . The oscillator also generates the normalized synthesized sinusoid  $(1/A)y$  which it sends to the Phase Detector for comparison.

## Ports

### Input

#### **u** – Input signal

scalar or vector

Periodic input signal.

Data Types: `single` | `double`

### Output

#### **freq** – Frequency

scalar or vector

Estimated frequency of the input signal, in Hz.

Data Types: `single` | `double`

#### **angle** – Phase angle

scalar or vector

Estimated phase angle of the input signal, in rad.

Data Types: `single` | `double`

**mag — Magnitude**

scalar or vector

Estimated magnitude of the input signal.

Data Types: `single` | `double`**Parameters****Phase detector integral gain — PD integral gain**

1000 (default) | positive scalar or vector

Integral gain for the phase detector. This determines the aggressiveness of the PLL in tracking and locking to the magnitude.

If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Loop filter proportional gain — LF proportional gain**

400 (default) | positive scalar or vector

Proportional gain for the loop filter. This determines the aggressiveness of the PLL in tracking and locking to the phase angle. Increase this value to improve reaction time of the tracking to step changes in the phase angle.

If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Loop filter integral gain — LF integral gain**

20000 (default) | positive scalar or vector

Integral gain for the loop filter. Increase this value to increase the rate at which steady-state error is eliminated in the phase angle. This value also determines the aggressiveness of the PLL in tracking and locking to the phase.

If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Initial frequency (Hz) — Initial frequency**

60 Hz (default) | scalar or vector

Initial estimate of the input frequency. If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Initial phase angle (rad) — Initial phase**

0 rad (default) | scalar or vector

Initial estimate of the phase angle. If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.

**Initial magnitude — Initial magnitude**

1 (default) | scalar or vector

Initial estimate of the magnitude. If the input signal is a vector, use scalar parameters or use vector parameters that are the same size as the input signal.



**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## References

- [1] Karimi-Ghartemani, M., and M. R. Iravani. "A New Phase-Locked Loop (PLL) System." *IEEE Transactions on Industrial Electronics*. Proceedings of the 44th IEEE Symposium on Circuits and Systems, vol. 1, pp. 421-424. IEEE, 2001..

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

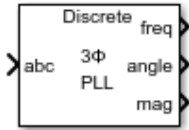
### Blocks

Three-Phase Sinusoidal Measurement (PLL) | RMS Measurement

## Sinusoidal Measurement (PLL, Three-Phase)

Estimate three-phase sinusoidal characteristics using a phase-locked loop

**Library:** Simscape / Electrical / Control / Measurements



### Description

The Sinusoidal Measurement (PLL, Three-Phase) block estimates the frequency characteristics of a balanced three-phase sinusoidal signal. The block uses a standard phase-locked loop (PLL) strategy to estimate the frequency and phase angle of the input signal. It also outputs the magnitude of the input signal.

Use this block in control applications when the frequency, phase angle, or magnitude are required and cannot be measured directly. To estimate the frequency characteristics of a non-three-phase or unbalanced sinusoidal signal, use the Sinusoidal Measurement (PLL) block instead.

### Equations

The phase-locked loop generates a sinusoid that approximates the input signal  $u(t)$  with the form:

$$y(t) = A(t)\sin\left(\phi_0 + \int 2\pi f(t)dt\right),$$

where:

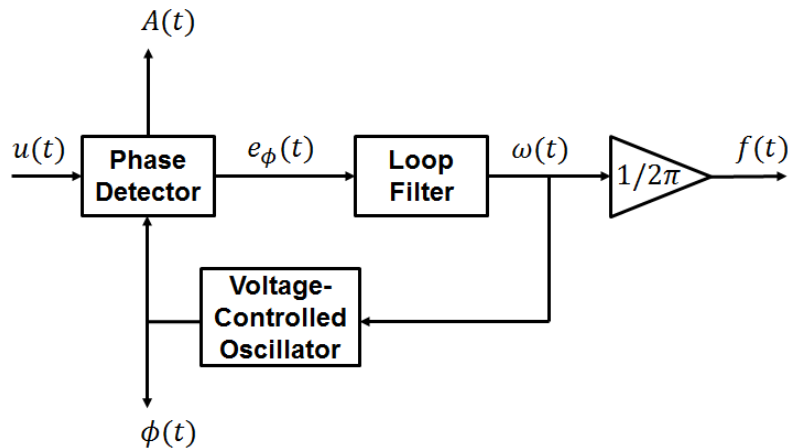
- $y$  is the estimate of the input signal.
- $A$  is the amplitude of the input signal.
- $\phi_0$  is the initial phase angle of the input signal.

Because the input signal is assumed to be balanced, the block calculates the amplitude directly from the instantaneous amplitude of the three phases. The estimated phase angle  $\phi$  is the angle of this generated sinusoid:

$$\phi(t) = \phi_0 + \int 2\pi f(t)dt,$$

where  $f$  is the frequency of the sinusoid, and  $\phi_0$  is the initial phase angle.

This diagram shows the overall structure of the phase-locked loop.



In the diagram:

- The phase detector produces an error signal relative to the phase difference  $e_\phi$  between the input sinusoid  $u$  and the synthesized sinusoid  $y$ . It also outputs the amplitude  $A$ .
- The loop filter provides an estimate of the input angular frequency  $\omega$  by filtering out the high-frequency components of the phase difference. The block also outputs the converted frequency  $f$  in Hz.
- The voltage-controlled oscillator integrates the angular speed to produce the phase estimate  $\phi$  which it sends to the Phase Detector for comparison.

## Ports

### Input

#### abc — Input signal

vector

Three-phase input signal.

Data Types: `single` | `double`

### Output

#### freq — Frequency

scalar

Estimated frequency of the input signal, in Hz.

Data Types: `single` | `double`

#### angle — Phase angle

scalar

Estimated phase angle of the first phase of the input signal, in rad.

Data Types: `single` | `double`

#### mag — Magnitude

scalar

Magnitude of the input signal.

Data Types: `single` | `double`

## Parameters

### Loop filter proportional gain — LF proportional gain

200 (default) | positive scalar

Proportional gain for the loop filter. Increase this value to increase the rate at which steady-state error is eliminated in the phase angle. This value also determines the aggressiveness of the PLL in tracking and locking to the phase angle.

### Loop filter integral gain — LF integral gain

2000 (default) | positive scalar

Integral gain for the loop filter. This determines the aggressiveness of the PLL in tracking and locking to the phase. Increase this value to reduce and eliminate steady-state error in the phase angle.

### Initial frequency (Hz) — Initial frequency

60 Hz (default) | scalar

Initial estimate of the input frequency.

### Initial phase angle (rad) — Initial phase

0 rad (default) | scalar or vector

Initial estimate of the phase angle.

### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Sinusoidal Measurement (PLL) | RMS Measurement

# Sine Wave Generator (Three-Phase)

Discrete-time or continuous-time three-phase sine wave generator

**Library:** Simscape / Electrical / Control / General Control



## Description

The Sine Wave Generator (Three-Phase) block generates a balanced three-phase sine wave from magnitude, phase shift, and frequency inputs.

## Ports

### Input

**mag — Magnitude of generated three-phase sine wave**

scalar

Magnitude of the generated three-phase sine wave, specified as a scalar.

**ang — Phase shift of the generated three-phase sine wave, deg**

scalar

Phase shift for phase *a* of the generated three-phase sine wave, specified as a scalar in degrees.

**freq — Frequency of generated three-phase sine wave, Hz**

scalar

Frequency of the generated three-phase sine wave, specified as a scalar in Hz.

### Output

**abc — Three-phase sine wave**

vector

Three-phase sine wave, returned as a Simulink signal vector.

**wt — Angle of sine wave of all three phases, rad**

vector

Angle of all three phases of the sine wave, returned as a Simulink signal vector in rad.

## Parameters

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## **Version History**

**Introduced in R2020a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

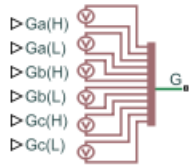
## **See Also**

Stair Generator

# Six-Pulse Gate Multiplexer

Multiplex gate input signals to Converter block

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Six-Pulse Gate Multiplexer block routes gate voltage signals to the six switching devices in a Converter (Three-Phase) block. The block multiplexes the six separate gate signals into a single vector.

When you switch the block inputs to electrical ports, the block shows additional electrical reference input ports. The additional electrical reference ports are associated with the individual phase voltages that connect to the high-side switching devices in the Converter (Three-Phase) block and the negative DC voltage common to each low-side switching device in the Converter (Three-Phase) block.

## Ports

### Input

**Ga(H) , Gb(H) , Gc(H) — Ga(H),Gb(H),Gc(H)**

physical signal

Physical signal ports associated with the gate terminal of the Converter (Three-Phase) block high-side switching devices.

### Dependencies

To enable this port, set **Gate-control port** to PS.

**Ga(L) , Gb(L) , Gc(L) — Ga(L),Gb(L),Gc(L)**

physical signal

Physical signal ports associated with the gate terminal of the Converter (Three-Phase) block low-side switching devices.

### Dependencies

To enable this port, set **Gate-control port** to PS.

### Conserving

**Ga(H) , Gb(H) , Gc(H) — Ga(H),Gb(H),Gc(H)**

electrical



Electrical conserving ports associated with the gate terminals of the Converter (Three-Phase) block high-side switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Ga(L) , Gb(L) , Gc(L) — Ga(L),Gb(L),Gc(L)**

electrical

Electrical conserving ports associated with the gate terminals of the Converter (Three-Phase) block low-side switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G — G**

electrical

Vector output port associated with the multiplexed gate signals. Connect this port to the G port of the Converter (Three-Phase) block.

**a — A-phase AC reference point**

electrical

Electrical conserving port associated with the A-phase for the high-side switching device of the Converter (Three-Phase) block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**b — B-phase AC reference point**

electrical

Electrical conserving port associated with the B-phase for the high-side switching device of the Converter (Three-Phase) block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**c — C-phase AC reference point**

electrical

Electrical conserving port associated with the C-phase for the high-side switching device of the Converter (Three-Phase) block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**L — L**

electrical

Electrical conserving port associated with the negative DC voltage common to each low-side switching device in the Converter (Three-Phase) block.

### Dependencies

To enable this port, set **Gate-control port** to `Electrical`.

### Parameters

#### **Gate-control port — Gate control port**

`PS (default) | Electrical`

Whether to expose PS or electrical control ports.

### Version History

#### **Introduced in R2013b**

#### **Physical and electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between physical and electrical ports, set the **Gate-control port** parameter to either `PS` or `Electrical`.

As a result of these changes, inside a model saved in an earlier release, review the **Gate-control port** parameter of this block.

### Extended Capabilities

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### See Also

Converter (Three-Phase)

### Topics

“Switch Between Physical Signal and Electrical Ports”

# Sliding Mode Controller

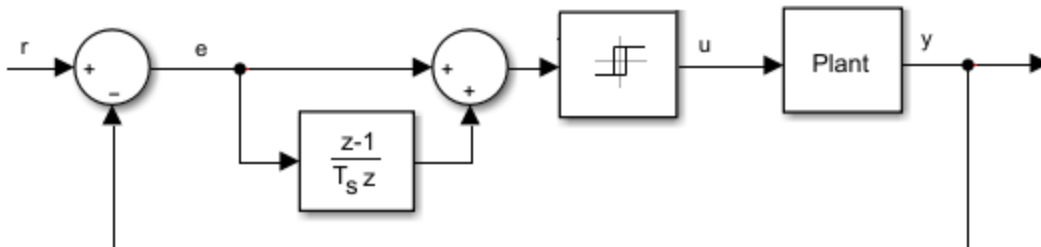
Hysteresis-based sliding mode control

**Library:** Simscape / Electrical / Control / General Control



## Description

The Sliding Mode Controller block implements hysteresis-based sliding mode control (SMC).



## Ports

### Input

#### **r – Plant reference**

scalar

Plant system reference signal.

Data Types: single | double

#### **y – Plant output**

scalar

Plant system output signal.

Data Types: single | double

### Output

#### **u – Controller output**

scalar

Control system output signal.

Data Types: single | double

## Parameters

### **Hysteresis band — Hysteresis bandwidth**

0.2 (default)

Total hysteresis bandwidth, distributed symmetrically about the set point.

### **Control action upper limit — Control signal upper limit, $u_{max}$**

10 (default) | scalar greater than the value of the **Control action lower limit** parameter

Upper limit for the control output signal.

### **Control action lower limit — Control signal lower limit, $u_{min}$**

-10 (default) | scalar

Lower limit for the control output signal.

### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

**Introduced in R2018a**

## Extended Capabilities

### **C/C++ Code Generation**

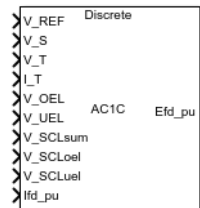
Generate C and C++ code using Simulink® Coder™.

## See Also

## SM AC1C

Discrete-time or continuous-time synchronous machine AC1C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC1C block implements a synchronous machine type AC1C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

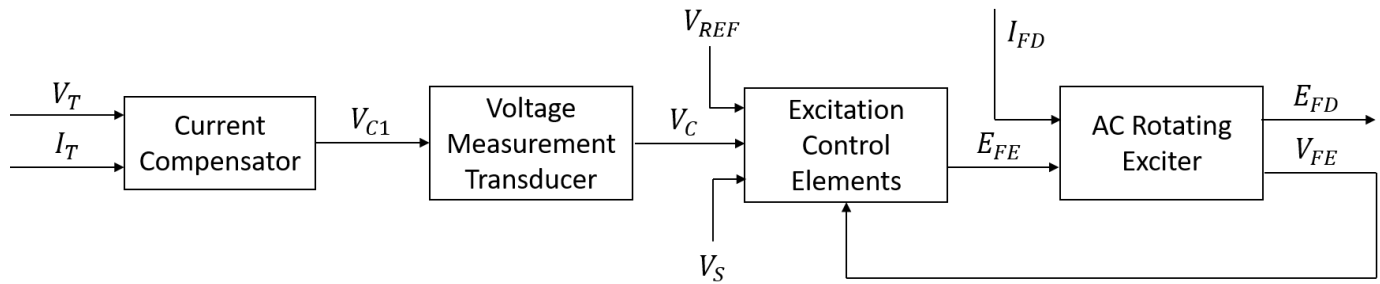
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC1C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC1C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

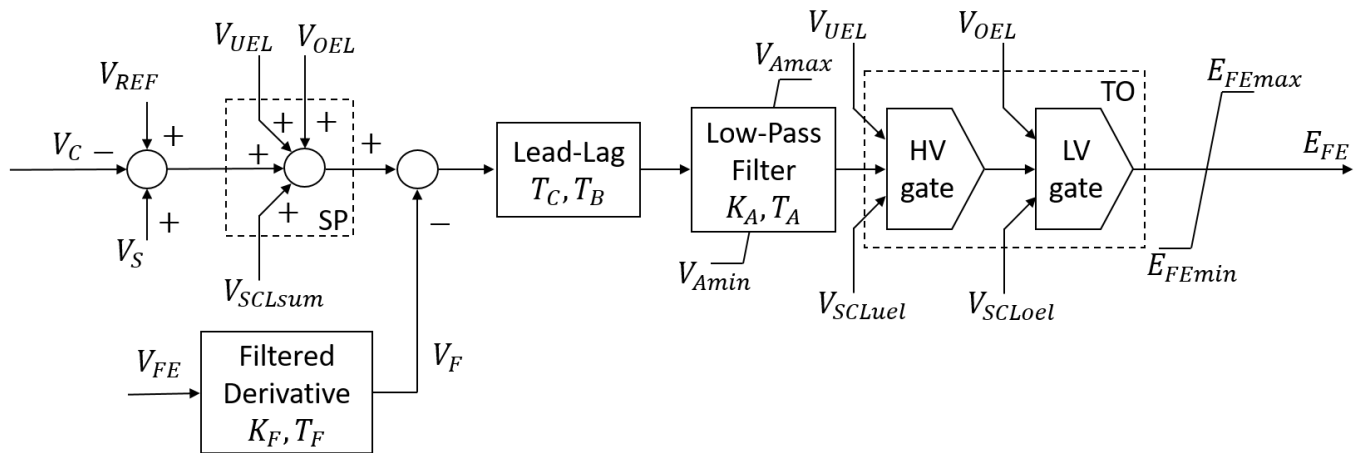
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- SP is the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1793.
- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- TO is the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1793.
- The Filtered Derivative block models the rate feedback path for stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constants of this system, respectively. Refer to the documentation for the Filtered Derivative block for the discrete and continuous implementations.
- $E_{FEmin}$  and  $E_{FEmax}$  are the minimum and maximum saturation limits for the output exciter field voltage  $E_{FE}$ .

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

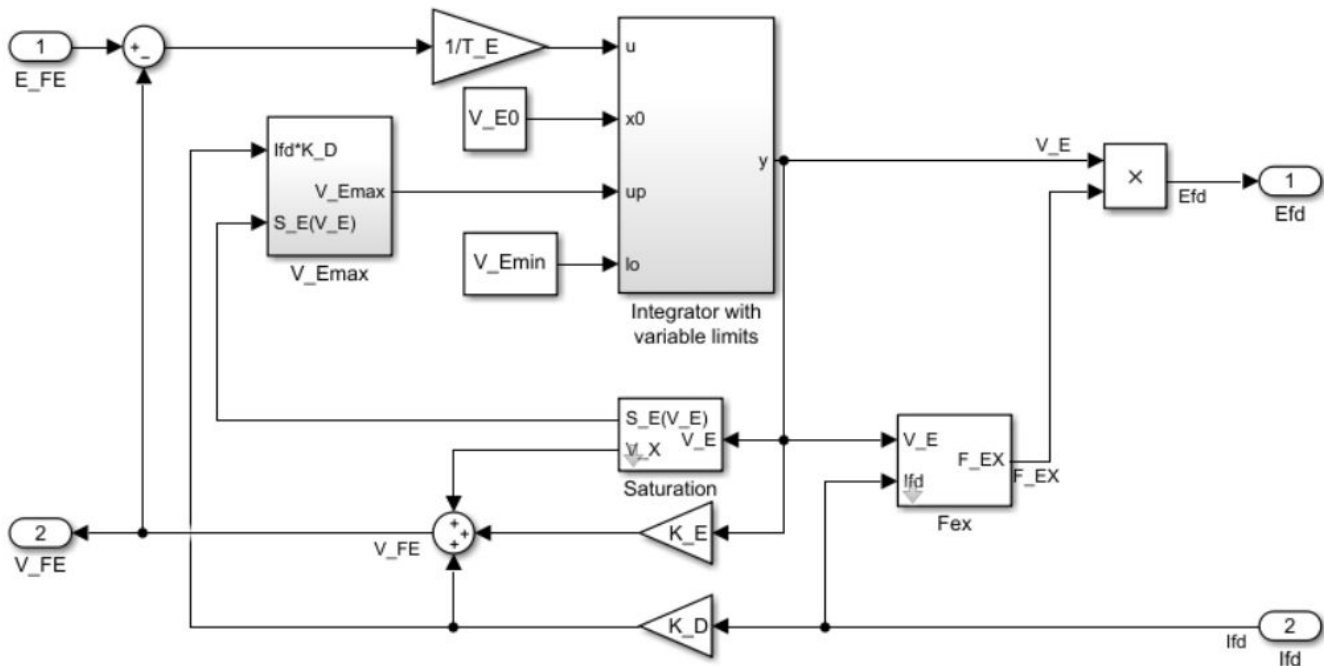
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

## AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

## Ports

### Input

#### **V\_REF** – Voltage reference

scalar



Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_S – Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_T – Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **I\_T – Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_OEL – Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

**V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_SCLoel — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_SCLuel — Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Ifd\_pu — Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

**Output****Efd\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: `single` | `double`

## Parameters

### General

#### **Initial field voltage, Efd0 (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### **Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

#### **Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

#### **Regulator input filter time constant, T\_R (s) — Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

### Control

#### **Regulator output gain, K\_A (pu) — Regulator gain**

400 (default) | positive number

Gain associated with the voltage regulator.

#### **Regulator output time constant, T\_A (s) — Regulator time constant**

0.02 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant,  $T_B$  (s) — Regulator lag time constant**  
0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant,  $T_C$  (s) — Regulator lead time constant**  
0 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Rate feedback excitation system stabilizer gain,  $K_F$  (pu) — Rate feedback gain**  
0.03 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

**Rate feedback time constant,  $T_F$  (s) — Rate feedback time constant**  
1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Maximum regulator output,  $V_{Amax}$  (pu) — Upper limit of the regulator output**  
14.5 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output,  $V_{Amin}$  (pu) — Lower limit of the regulator output**  
-14.5 (default) | real number

Minimum per-unit output voltage of the regulator.

**Maximum exciter field voltage,  $E_{FEmax}$  (pu) — Upper limit of the exciter voltage**  
6.03 (default) | real number

Maximum per-unit field voltage to apply to the exciter.

**Minimum exciter field voltage,  $E_{FEmin}$  (pu) — Lower limit of the exciter voltage**  
-5.43 (default) | real number

Minimum per unit field voltage to apply to the exciter.

**Alternate OEL input locations ( $V_{OEL}$ ) — OEL input location**  
Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

**Alternate UEL input locations ( $V_{UEL}$ ) — UEL input location**  
Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

**Alternate SCL input locations ( $V_{SCL}$ ) — SCL input location**  
Unused (default) | Summation point | Take-over

Location of the stator current limiter input:

- If you select **Summation point**, use the **V\_SCLsum** input port.
- If you select **Take-over**, use the **V\_SCLoel** and **V\_SCLuel** input ports.

## Exciter

### **Exciter field proportional constant, K\_E (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

### **Exciter field time constant, T\_E (s) – Exciter field time constant**

0.8 (default) | positive number

Time constant for the exciter field.

### **Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0.2 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

### **Demagnetizing factor, function of exciter alternator reactances, K\_D (pu) – Demagnetization factor**

0.38 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

### **Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

4.18 (default) | positive number

Exciter output voltage for the first saturation factor.

### **Exciter saturation factor at exciter output voltage E\_1, S\_E(E\_1) (1) – First saturation lookup voltage**

0.1 (default) | positive number

Saturation factor for the first exciter.

### **Exciter output voltage for saturation factor S\_E(E\_2), E\_2 (pu) – Second saturation output voltage**

3.14 (default) | positive number

Exciter output voltage for the second saturation factor.

### **Exciter saturation factor at exciter output voltage E\_2, S\_E(E\_2) (1) – Second saturation lookup voltage**

0.03 (default) | positive number

Saturation factor for the second exciter

### **Maximum exciter field current limit reference, V\_FEmax (pu) – Exciter upper limit**

99 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output, V\_Emin (pu) — Exciter lower limit**

0 (default) | real number

Minimum per-unit exciter voltage output.

## **Version History**

**Introduced in R2017b**

## **References**

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

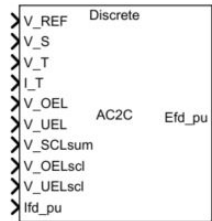
### **Blocks**

Lead-Lag (Discrete or Continuous) | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator | SM AC2C | SM AC3C | SM AC4C | SM AC5C | SM AC6C | SM AC7C | SM AC8C

## SM AC2C

Discrete-time or continuous-time synchronous machine AC2C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC2C block implements a synchronous machine type AC2C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

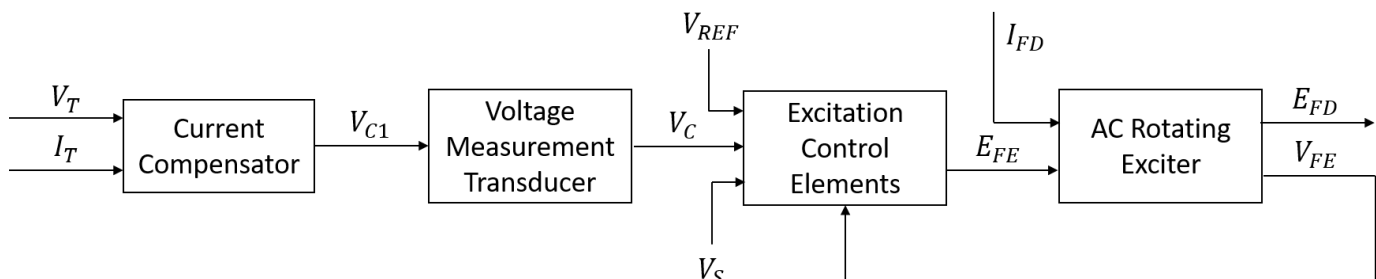
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC2C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC2C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2}$$

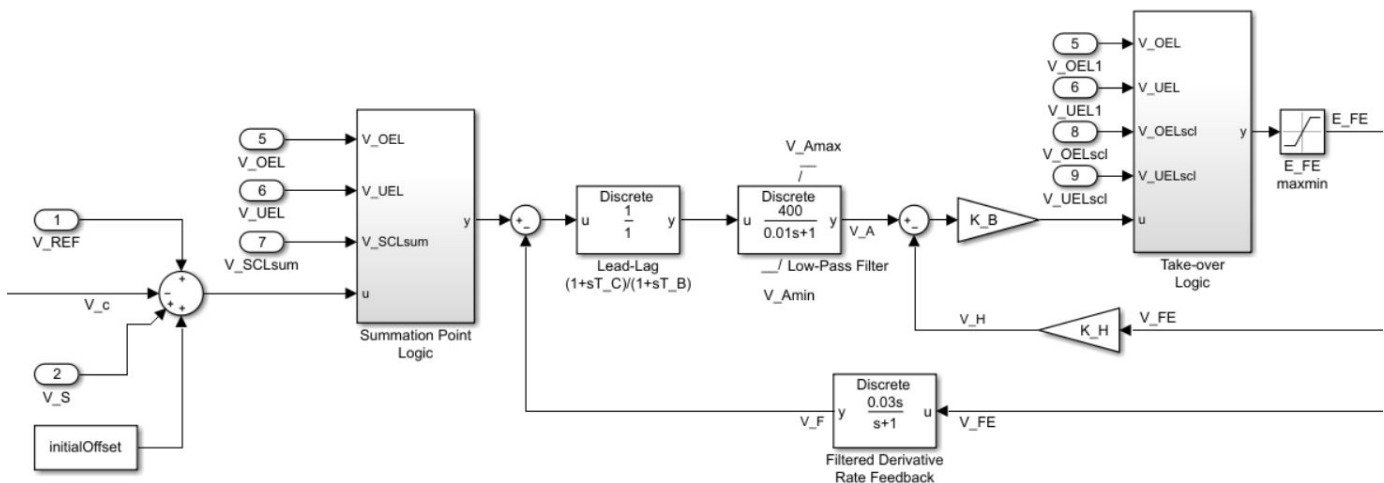
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1803.



- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1803.
- The Filtered Derivative block models the rate feedback path for the stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constants of this system, respectively. Refer to the documentation for the Filtered Derivative block for the discrete and continuous implementations.
- To increase the small signal response bandwidth of the excitation system, a direct negative feedback,  $V_H$ , models the implementation of an exciter time constant compensation. The time constant is reduced by a factor proportional to the **Second stage regulator gain, K\_B (pu)** and **Exciter field current regulator feedback gain, K\_H (pu)** parameters.
- $E_{FEmin}$  and  $E_{FEmax}$  are the minimum and maximum saturation limits for the output exciter field voltage  $E_{FE}$ .

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

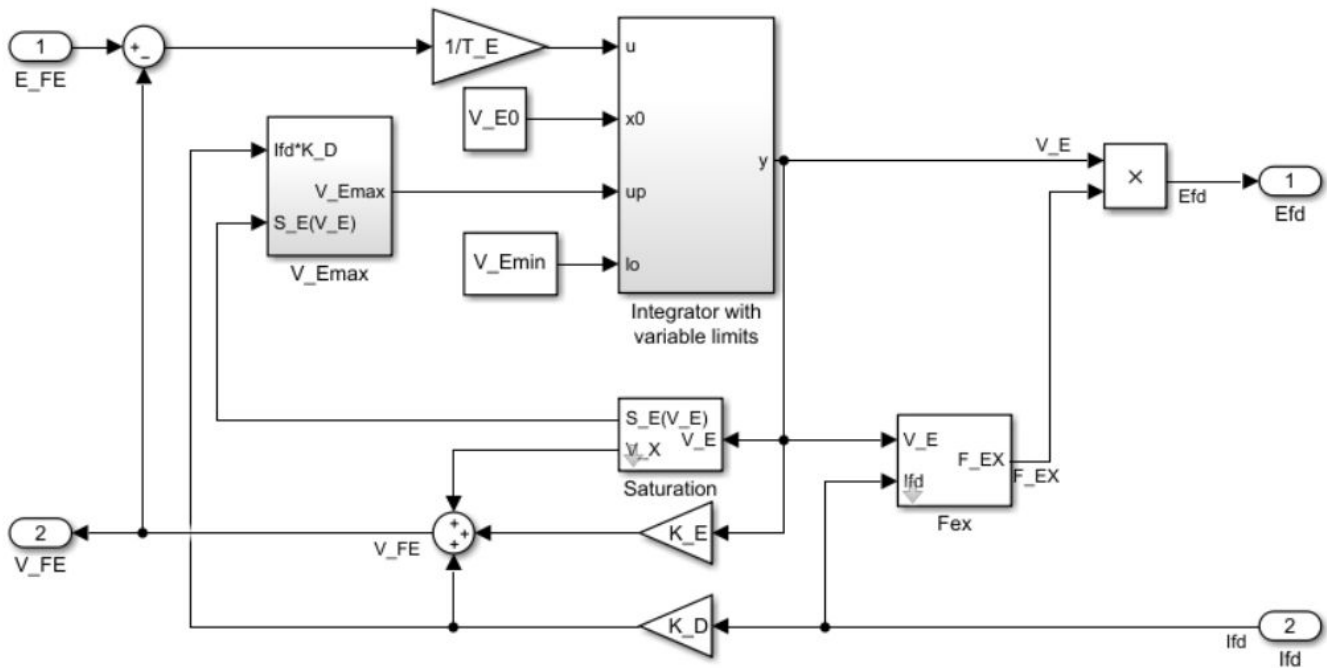
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{OELscl}$ , and the underexcitation input,  $V_{UELscl}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

## Ports

### Input

#### V\_REF – Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_S – Input from stabilizer

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_0ELscl – Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_UELscl – Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Ifd\_pu – Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

**Output****Efd\_pu – Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### Initial field voltage, $E_{fd0}$ (pu) — Initial output voltage

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### Resistive component of load compensation, $R_C$ (pu) — Compensation resistance

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation,  $X_C$  (pu)** to 0 to disable current compensation.

#### Reactance component of load compensation, $X_C$ (pu) — Compensation reactance

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation,  $R_C$  (pu)** to 0 to disable current compensation.

#### Regulator input filter time constant, $T_R$ (s) — Regulator time constant

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

### Control

#### Regulator output gain, $K_A$ (pu) — Regulator gain

400 (default) | positive number

Gain associated with the voltage regulator.

#### Regulator output time constant, $T_A$ (s) — Regulator time constant

0.01 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant, T<sub>B</sub> (s) — Regulator lag time constant**  
0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set parameter this to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant, T<sub>C</sub> (s) — Regulator lead time constant**  
0 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Second stage regulator gain, K<sub>B</sub> (pu) — Second stage regulator gain**  
25 (default) | positive number

Per-unit gain of the second stage regulator.

**Exciter field current regulator feedback gain, K<sub>H</sub> (pu) — Exciter field current regulator feedback gain**

1 (default) | positive number

Per-unit feedback gain for the exciter field current regulator.

**Rate feedback excitation system stabilizer gain, K<sub>F</sub> (pu) — Rate feedback gain**  
0.03 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

**Rate feedback time constant, T<sub>F</sub> (s) — Rate feedback time constant**  
1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Maximum regulator output, V<sub>Amax</sub> (pu) — Upper limit of the regulator output**  
8 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V<sub>Amin</sub> (pu) — Lower limit of the regulator output**  
-8 (default) | real number

Minimum per-unit output voltage of the regulator.

**Maximum exciter field voltage, E<sub>Fmax</sub> (pu) — Upper limit of the exciter voltage**  
105 (default) | real number

Maximum per-unit field voltage to apply to the exciter.

**Minimum exciter field voltage, E<sub>Fmin</sub> (pu) — Lower limit of the exciter voltage**  
-95 (default) | real number

Minimum per-unit field voltage to apply to the exciter.

**Alternate OEL input locations (V<sub>OEL</sub>) — OEL input location**  
Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

**Alternate UEL input locations (V\_UEL) – UEL input location**

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input:

- If you select *Summation point*, use the **V\_SCLsum** input port.
- If you select *Take-over*, use the **V\_OELscl** and **V\_UELscl** input ports.

**Exciter**

**Exciter field proportional constant, K\_E (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant, T\_E (s) – Exciter field time constant**

0.6 (default) | positive number

Time constant for the exciter field.

**Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0.28 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Demagnetizing factor, function of exciter alternator reactances, K\_D (pu) – Demagnetization factor**

0.35 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

**Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

4.4 (default) | positive number

Exciter output voltage for the first saturation factor.

**Exciter saturation factor at exciter output voltage E\_1, S\_E(E\_1) (1) – First saturation lookup voltage**

0.037 (default) | positive number

Saturation factor for the first exciter.

**Exciter output voltage for saturation factor S\_E(E\_2), E\_2 (pu) – Second saturation output voltage**

3.3 (default) | positive number

Exciter output voltage for the second saturation factor.

**Exciter saturation factor at exciter output voltage E<sub>2</sub>, S<sub>E</sub>(E<sub>2</sub>) (1) – Second saturation lookup voltage**

0.012 (default) | positive number

Saturation factor for the second exciter.

**Maximum exciter field current limit reference, V<sub>F</sub>Emax (pu) – Exciter upper limit**

4.4 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output, V<sub>E</sub>min (pu) – Exciter lower limit**

0 (default) | real number

Minimum per-unit exciter voltage output.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

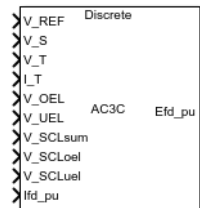
SM AC1C | SM AC3C | SM AC4C | SM AC5C | SM AC6C | SM AC7C | SM AC8C



## SM AC3C

Discrete-time or continuous-time synchronous machine AC3C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC3C block implements a synchronous machine type AC3C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

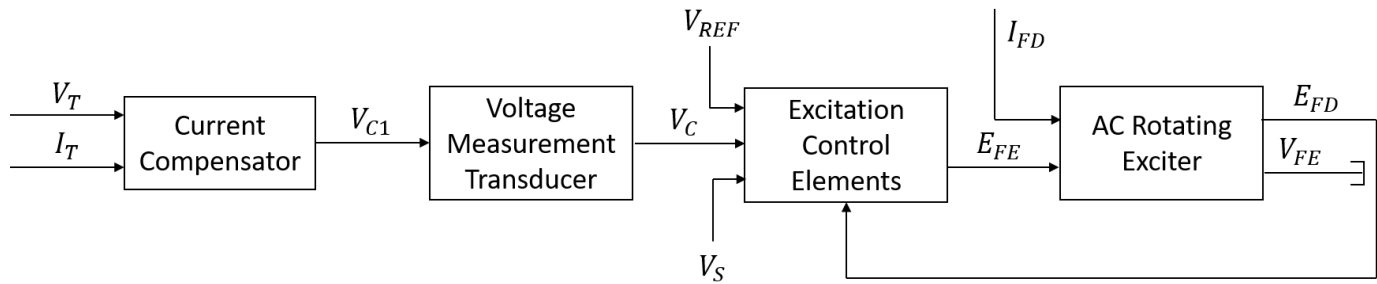
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC3C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC3C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

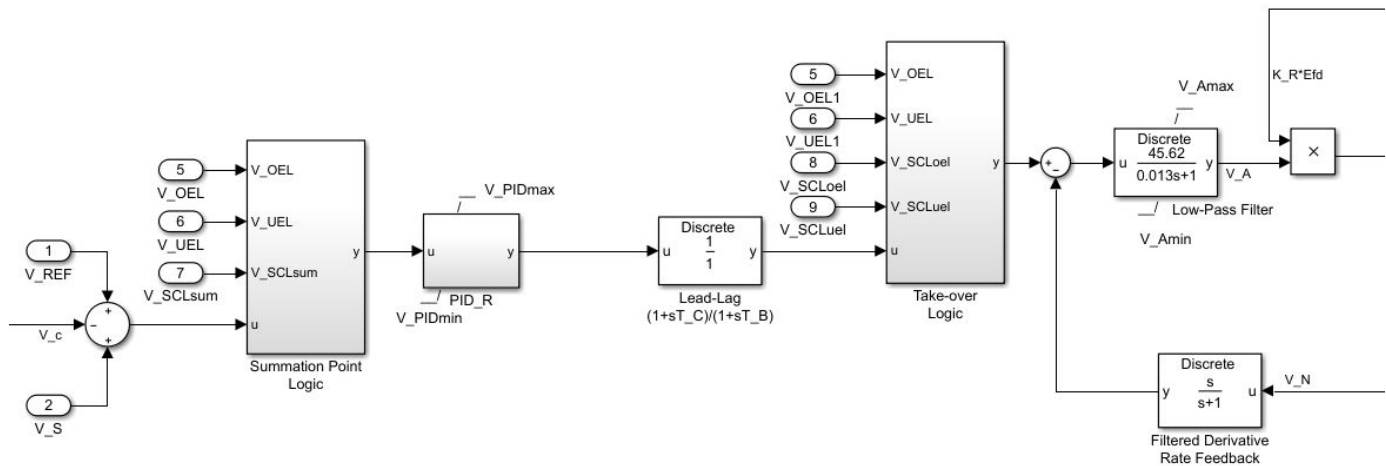
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1814.
- The PID\_R subsystem models a PID controller and it allows the representation of retrofit projects, where a modern digital exciter is added to the exciter. It is possible to model a SM AC3A excitation system model by tuning the PID parameters.
- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1814.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- The voltage regulator power is derived from the exciter output voltage. The voltage regulator command signal,  $V_A$ , is multiplied by the exciter output voltage,  $E_{FD}$ , and multiplied by  $K_R$ . This adds a level of nonlinearity to the system.
- The Filtered Derivative block models the rate feedback path for stabilization of the excitation system. In this case, the stabilizer has a nonlinear characteristic. If the exciter output voltage,  $E_{FD}$ , is less than the **Value of EFD at which feedback gain changes, E\_FDN (pu)** parameter, the feedback gain is  $K_F$ . If the exciter output voltage is greater than the **Value of EFD at which feedback gain changes, E\_FDN (pu)** parameter, the feedback gain is  $K_N$ . Refer to the documentation for the Filtered Derivative block for the exact discrete and continuous implementations.
- $E_{FEmin}$  and  $E_{FEmax}$  are the minimum and maximum saturation limits for the output exciter field voltage  $E_{FE}$ .

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

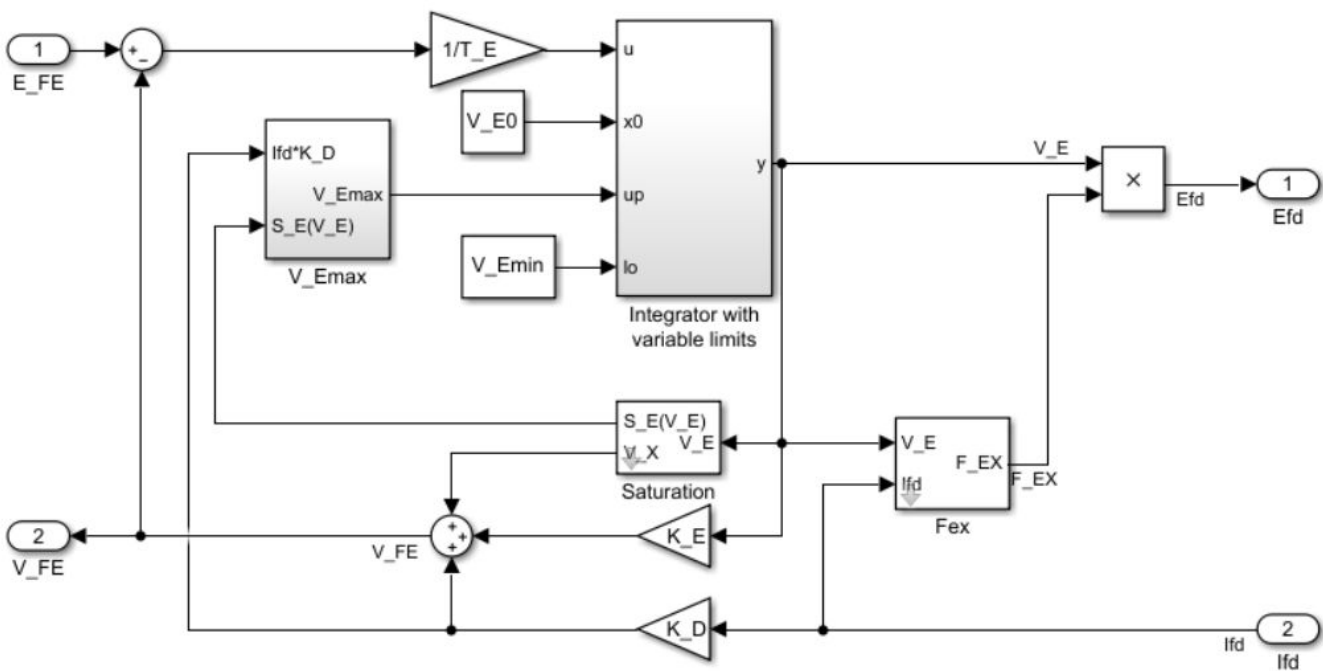
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.

- The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
- The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

### Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum – Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_SCLoe1 – Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_SCLue1 – Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

## Dependencies

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

## I<sub>fd</sub>\_pu — Measured field current

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

## Output

### E<sub>fd</sub>\_pu — Field voltage

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### Initial field voltage, E<sub>fd0</sub> (pu) — Initial output voltage

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### Resistive component of load compensation, R<sub>C</sub> (pu) — Compensation resistance

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X<sub>C</sub> (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X<sub>C</sub> (pu) – Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R<sub>C</sub> (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T<sub>R</sub> (s) – Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control****Voltage regulator proportional gain, K<sub>PR</sub> (pu) – Proportional gain of the voltage regulator**

1 (default)

Per-unit proportional gain of the voltage regulator.

**Voltage regulator integral gain, K<sub>IR</sub> (pu/s) – Integral gain of voltage regulator**

0 (default)

Per-unit over second integral gain of the voltage regulator.

**Voltage regulator derivative gain, K<sub>DR</sub> (pu.s) – Derivative gain of voltage regulator**

0 (default)

Derivative gain of the voltage regulator.

**Lag time constant for derivative channel of PID controller, T<sub>DR</sub> (s) – Lag time constant for PID derivative channel**

1 (default) | positive number

Equivalent lag time constant for the derivative channel of the PID controller.

**Maximum PID regulator output, V<sub>PIDmax</sub> (pu) – Maximum output of the PID regulator**

3.2 (default) | positive number

Maximum admissible per-unit output of the PID regulator.

**Minimum PID regulator output, V<sub>PIDmin</sub> (pu) – Minimum output of the PID regulator**

-3.2 (default) | positive number

Minimum admissible per-unit output of the PID regulator.

**Regulator denominator (lag) time constant, T<sub>B</sub> (s) – Regulator lag time constant**

0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant, T<sub>C</sub> (s) – Regulator lead time constant**

0 (default) | positive number



Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Regulator output gain,  $K_A$  (pu) — Regulator gain**

45.62 (default) | positive number

Gain associated with the voltage regulator.

**Regulator output time constant,  $T_A$  (s) — Regulator time constant**

0.013 (default) | positive number

Major time constant of the voltage regulator.

**Maximum regulator output,  $V_{Amax}$  (pu) — Upper limit of the regulator output**

1 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output,  $V_{Amin}$  (pu) — Lower limit of the regulator output**

-0.95 (default) | real number

Minimum per-unit output voltage of the regulator.

**Rate feedback time constant,  $T_F$  (s) — Rate feedback time constant**

1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Value of EFD at which feedback gain changes,  $E_{FDN}$  (pu) — Value of EFD at which feedback gain changes**

2.36 (default) | positive number

Per-unit value of EFD at which the feedback gain changes.

**Rate feedback excitation system stabilizer gain,  $K_F$  (pu) — Rate feedback gain**

0.143 (default) | positive number

Rate feedback block gain for stabilization of excitation system.

**Rate feedback excitation system stabilizer gain,  $K_N$  (pu) — Rate feedback gain**

0.05 (default) | positive number

Rate feedback block gain for stabilization of excitation system.

**Alternate OEL input locations ( $V_{OEL}$ ) — OEL input location**

Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

**Alternate UEL input locations ( $V_{UEL}$ ) — UEL input location**

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

**Alternate SCL input locations ( $V_{SCL}$ ) — SCL input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input :

- If you select **Summation point**, use the **V\_SCLsum** input port.
- If you select **Take-over**, use the **V\_SCLoel** and **V\_SCLuel** input ports.

### **Exciter**

#### **Exciter field proportional constant, K\_E (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

#### **Exciter field time constant, T\_E (s) – Exciter field time constant**

0.17 (default) | positive number

Time constant for the exciter field.

#### **Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0.104 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

#### **Demagnetizing factor, function of exciter alternator reactances, K\_D (pu) – Demagnetization factor**

0.499 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

#### **Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

6.24 (default) | positive number

Exciter output voltage for the first saturation factor.

#### **Exciter saturation factor at exciter output voltage E\_1, S\_E(E\_1) (1) – First saturation lookup voltage**

1.143 (default) | positive number

Saturation factor for the first exciter.

#### **Exciter output voltage for saturation factor S\_E(E\_2), E\_2 (pu) – Second saturation output voltage**

4.68 (default) | positive number

Exciter output voltage for the second saturation factor.

#### **Exciter saturation factor at exciter output voltage E\_2, S\_E(E\_2) (1) – Second saturation lookup voltage**

0.1 (default) | positive number

Saturation factor for the second exciter.

#### **Gain associated with regulator and alternator field power supply, K\_R (pu) – Regulator and alternator field power supply gain**

3.77 (default) | real number

Per-unit regulator and alternator field power supply gain.

**Minimum exciter voltage output, V\_Emin (pu) – Exciter lower limit**

0.1 (default) | real number

Minimum per-unit exciter voltage output.

**Exciter field current limit, V\_FEmax (pu) – Exciter upper limit**

16 (default) | real number

Per-unit exciter field current upper limit.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

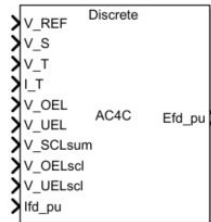
### Blocks

SM AC1C | SM AC2C | SM AC4C | SM AC5C | SM AC6C | SM AC7C | SM AC8C

## SM AC4C

Discrete-time or continuous-time synchronous machine AC4C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC4C block implements a synchronous machine type AC4C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

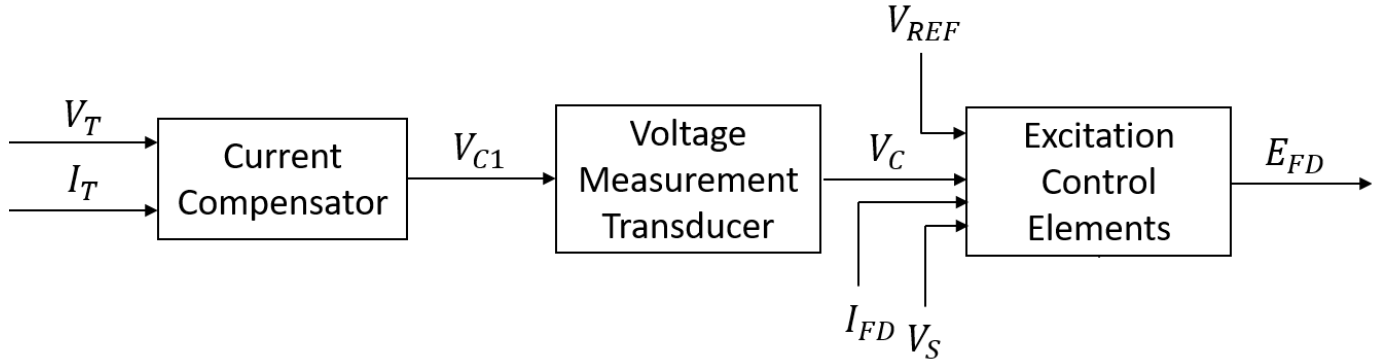
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC4C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC4C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

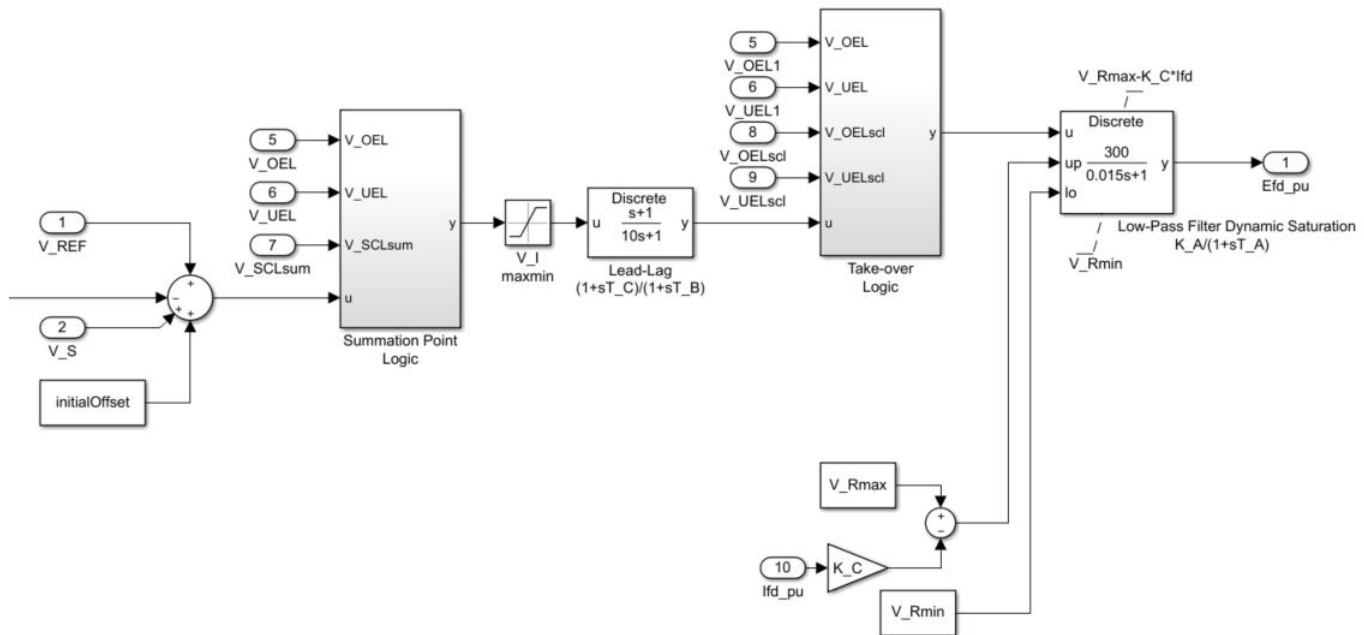
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



Contrary to the other AC excitation models, the SM AC4C block includes a full thyristor bridge that models the exciter output circuit and a voltage regulator that controls the firing of the thyristor bridges. Rather than having a rate feedback, the excitation system stabilization is implemented through a lead-lag network and low-pass filter.

In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1824.
- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1824.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The bottom part of the diagram models the  $I_{fd}$ -based limitations.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.

- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{OELscl}$ , and the underexcitation input,  $V_{UELscl}$ .

## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

### Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.

- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_OELscl — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_UELscl — Take-over stator current limit for underexcitation limiter**

scalar



Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### Dependencies

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

#### I<sub>fd\_pu</sub> — Measured field current

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

#### Output

#### E<sub>fd\_pu</sub> — Field voltage

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### Initial field voltage, E<sub>fd0</sub> (pu) — Initial output voltage

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### Resistive component of load compensation, R<sub>C</sub> (pu) — Compensation resistance

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) – Compensation reactance**  
0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) – Regulator time constant**  
0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

### **Control**

**Regulator output gain, K\_A (pu) – Regulator gain**  
300 (default) | positive number

Gain associated with the voltage regulator.

**Regulator output time constant, T\_A (s) – Regulator time constant**  
0.015 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant, T\_B (s) – Regulator lag time constant**  
10 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant, T\_C (s) – Regulator lead time constant**  
1 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Maximum regulator output, V\_Rmax (pu) – Upper limit of the regulator output**  
5.64 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V\_Rmin (pu) – Lower limit of the regulator output**  
-4.53 (default) | real number

Minimum per-unit output voltage of the regulator.

**Voltage regulator input (voltage error) maximum limit, V\_Imax (pu) – Maximum limit of the voltage error**  
10 (default) | real number

Maximum per-unit limit of the voltage regulator input.

**Voltage regulator input (voltage error) minimum limit, V\_Imin (pu) – Minimum limit of the voltage error**

-10 (default) | real number

Minimum per-unit limit of the voltage regulator input.

**Alternate OEL input locations (V\_OEL) – OEL input location**

Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

**Alternate UEL input locations (V\_UEL) – UEL input location**

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input:

- If you select `Summation point`, use the `V_SCLsum` input port.
- If you select `Take-over`, use the `V_OELscl` and `V_UELscl` input ports.

**Exciter****Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

## Version History

**Introduced in R2020a**

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

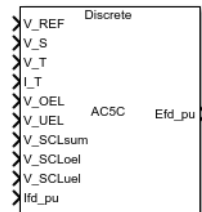
**Blocks**

SM AC1C | SM AC2C | SM AC3C | SM AC5C | SM AC6C | SM AC7C | SM AC8C

## SM AC5C

Discrete-time or continuous-time synchronous machine AC5C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC5C block implements a synchronous machine type AC5C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

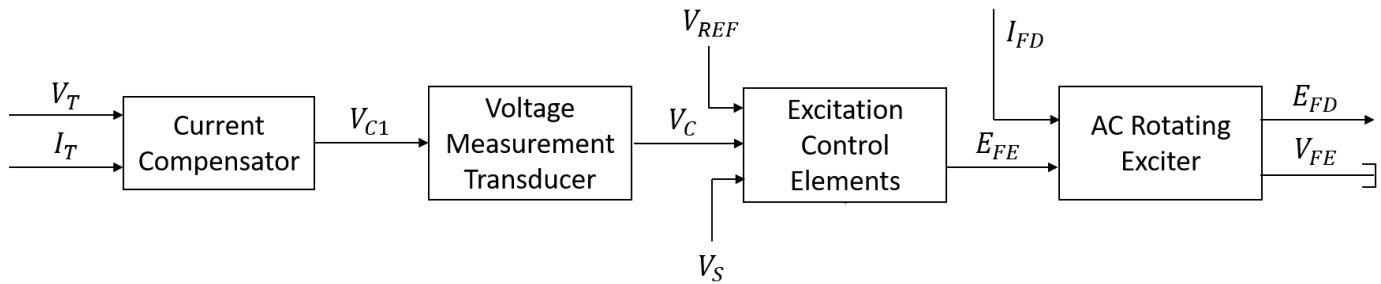
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC5C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC5C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

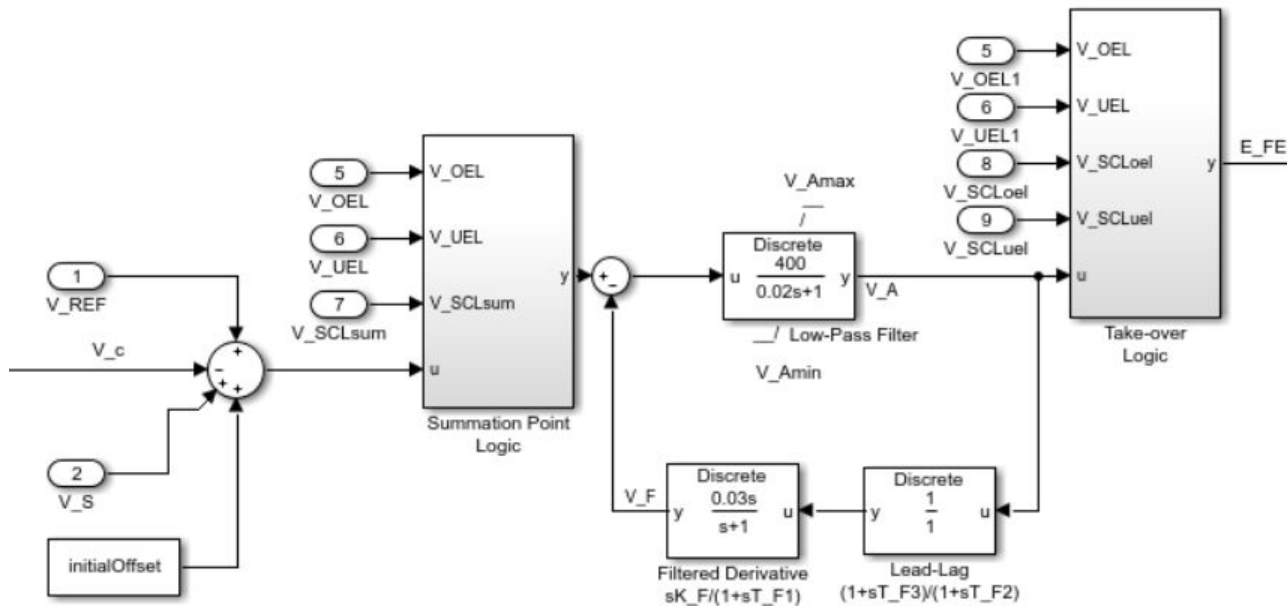
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1832.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- The Filtered Derivative and the Lead-Lag blocks model the rate feedback path for stabilization of the excitation system. Here,  $K_F$  is the gain of this system and  $T_{F1}$ ,  $T_{F2}$ , and  $T_{F3}$  are its time constants. Refer to the documentation for the Filtered Derivative block for the discrete and continuous implementations.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1832.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

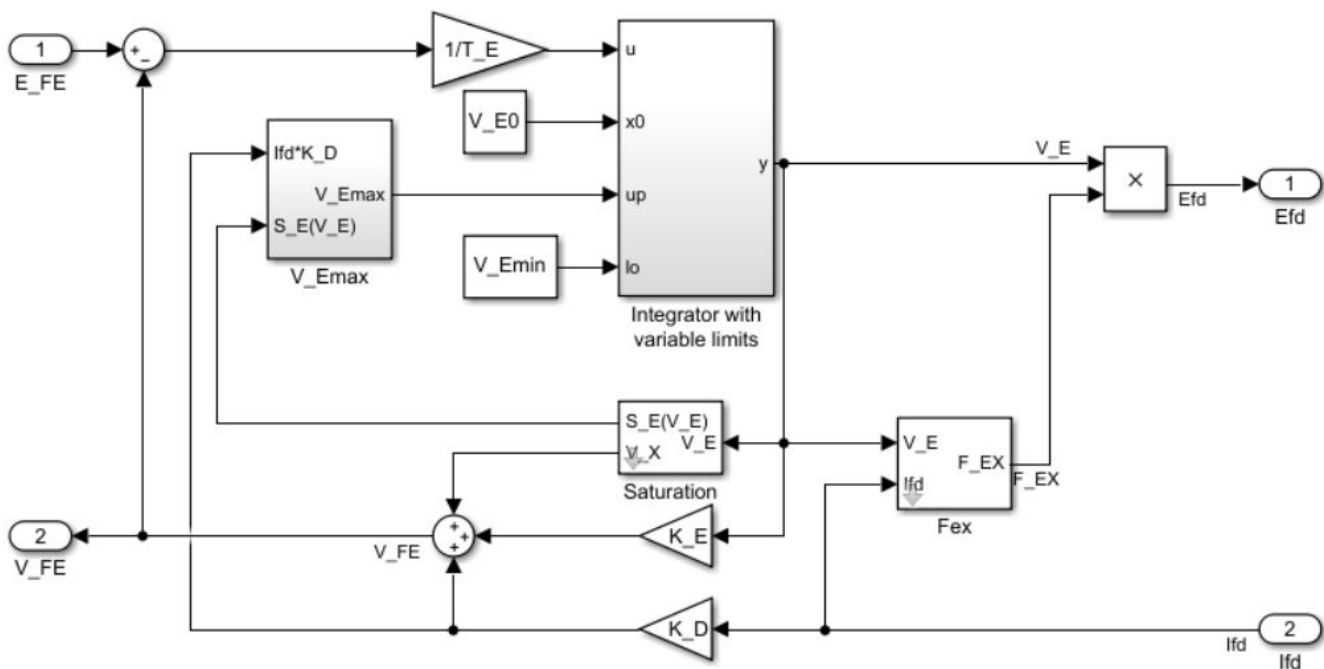
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$  and the underexcitation input,  $V_{SCLuel}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_X$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

## Ports

### Input

#### **V\_REF – Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S – Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T – Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T – Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_OEL – Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to `Unused`.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to `Summation point`.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to `Take-over`.

Data Types: `single` | `double`

#### **V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### Dependencies

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to `Unused`.



- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_SCLoel — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_SCLuel — Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **I<sub>fd</sub>\_pu — Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

## Output

### **Efd\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### **Initial field voltage, Efd0 (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### **Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

#### **Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

#### **Regulator input filter time constant, T\_R (s) — Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

## Control

### Regulator output gain, $K_A$ (pu) – Regulator gain

400 (default) | positive number

Gain associated with the voltage regulator.

### Regulator output time constant, $T_A$ (s) – Regulator time constant

0.02 (default) | positive number

Major time constant of the voltage regulator.

### Maximum regulator output, $V_{Amax}$ (pu) – Upper limit of the regulator output

7.3 (default) | real number

Maximum per-unit output voltage of the regulator.

### Minimum regulator output, $V_{Amin}$ (pu) – Lower limit of the regulator output

-7.3 (default) | real number

Minimum per-unit output voltage of the regulator.

### Rate feedback excitation system stabilizer gain, $K_F$ (pu) – Rate feedback gain

0.03 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

### Rate feedback excitation system stabilizer time constant, $T_{F1}$ (s) – Rate feedback time constant

1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

### Rate feedback excitation system stabilizer time constant, $T_{F2}$ (s) – Rate feedback time constant

0 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

### Rate feedback excitation system stabilizer time constant, $T_{F3}$ (s) – Rate feedback time constant

0 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

### Alternate OEL input locations ( $V_{OEL}$ ) – OEL input location

Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

### Alternate UEL input locations ( $V_{UEL}$ ) – UEL input location

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input:

- If you select Summation point, use the **V\_SCLsum** input port.
- If you select Take-over, use the **V\_SCLoel** and **V\_SCLuel** input ports.

**Exciter****Exciter field proportional constant, K\_E (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant, T\_E (s) – Exciter field time constant**

0.8 (default) | positive number

Time constant for the exciter field.

**Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Demagnetizing factor, function of exciter alternator reactances, K\_D (pu) – Demagnetization factor**

0 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

**Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

5.6 (default) | positive number

Exciter output voltage for first saturation factor.

**Exciter saturation factor at exciter output voltage E\_1, S\_E(E\_1) (1) – First saturation lookup voltage**

0.86 (default) | positive number

Saturation factor for the first exciter.

**Exciter output voltage for saturation factor S\_E(E\_2), E\_2 (pu) – Second saturation output voltage**

4.2 (default) | positive number

Exciter output voltage for second saturation factor.

**Exciter saturation factor at exciter output voltage E\_2, S\_E(E\_2) (1) – Second saturation lookup voltage**

0.5 (default) | positive number

Saturation factor for the second exciter.

**Maximum exciter field current limit reference, V\_FEmax (pu) — Exciter upper limit**  
99 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output, V\_Emin (pu) — Exciter lower limit**  
0 (default) | real number

Minimum per-unit exciter voltage output.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

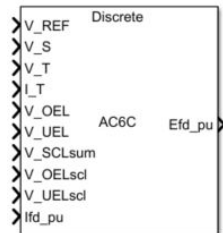
### Blocks

SM AC1C | SM AC2C | SM AC3C | SM AC4C | SM AC6C | SM AC7C | SM AC8C

## SM AC6C

Discrete-time or continuous-time synchronous machine AC6C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC6C block implements a synchronous machine type AC6C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

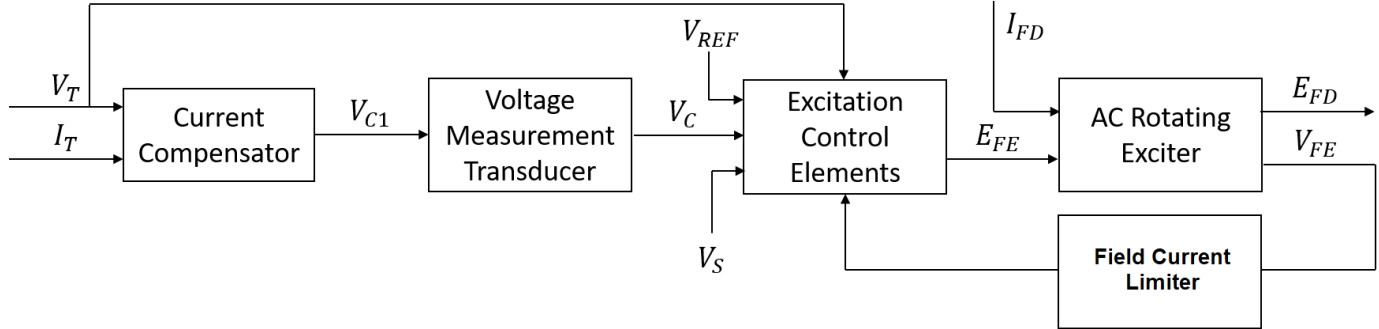
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC6C block is made up of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.

This diagram shows the overall structure of the AC6C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

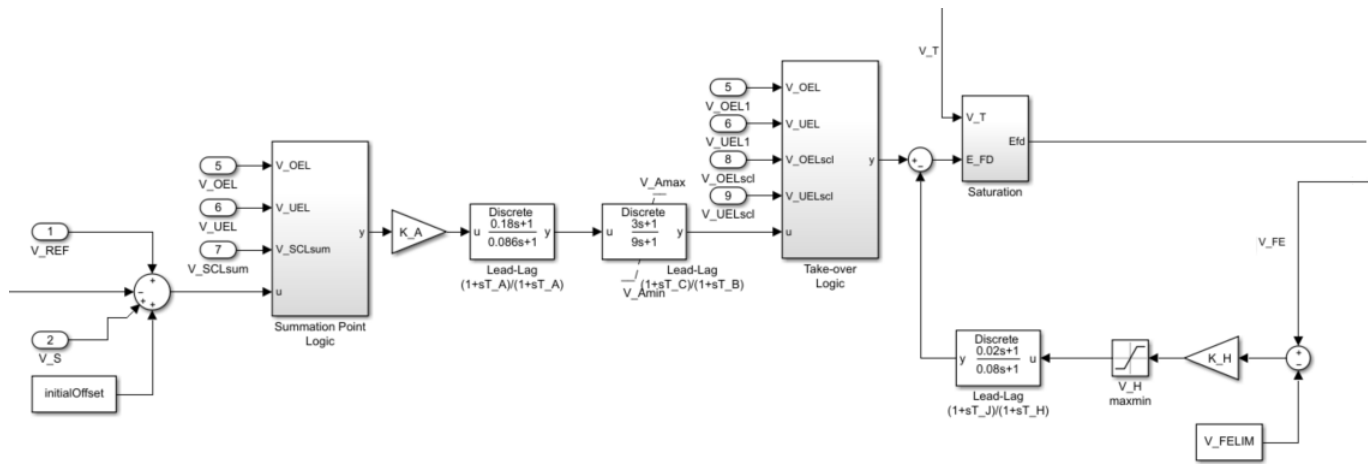
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1842.
- A network of two Lead-Lag blocks models the dynamics of the voltage regulator.  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The second block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1842.
- Another Lead-Lag block models the rate feedback path for the stabilization of the excitation system. Here,  $K_H$  is the gain of this system and  $T_J$  and  $T_H$  are the lead and lag time constants, respectively.
- The Saturation subsystem models the minimum,  $V_T \cdot E_{FEmin}$ , and the maximum,  $V_T \cdot E_{FEmax}$ , saturation limits for the output exciter field voltage  $E_{FE}$ .

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop

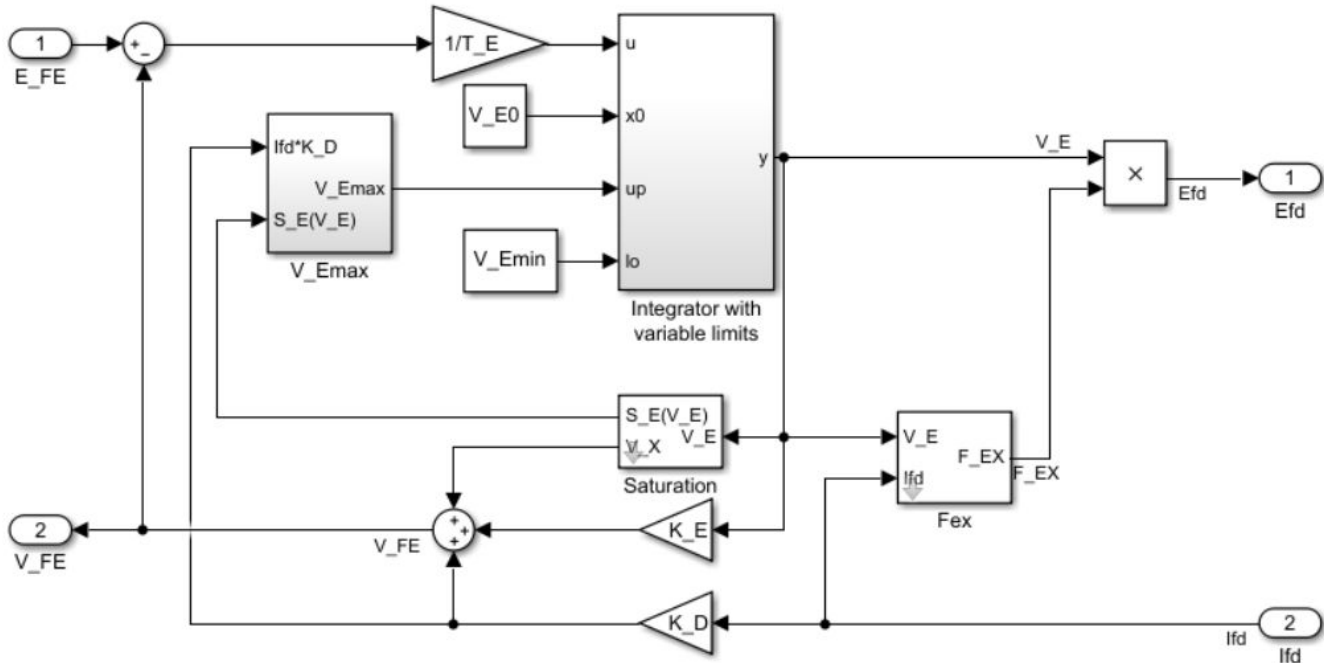


- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{OELscl}$ , and the underexcitation input,  $V_{UELscl}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

### Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to **Unused**.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to **Summation point**.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to **Take-over**.

Data Types: `single` | `double`

#### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

### Dependencies

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to **Unused**.

- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_OELscl — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_UELscl — Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **I<sub>fd</sub>\_pu — Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

## Output

### **Efd\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### **Initial field voltage, Efd0 (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### **Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

#### **Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

#### **Regulator input filter time constant, T\_R (s) — Regulator time constant**

0.02 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

## Control

### Regulator output gain, $K_A$ (pu) – Regulator gain

536 (default) | positive number

Gain associated with the voltage regulator.

### Regulator output time constant, $T_A$ (s) – Regulator time constant

0.086 (default) | positive number

Major time constant of the voltage regulator.

### Regulator denominator (lag) time constant, $T_B$ (s) – Regulator lag time constant

9 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

### Regulator numerator (lead) time constant, $T_C$ (s) – Regulator lead time constant

3 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

### Regulator numerator (lead) time constant, $T_K$ (s) – Regulator lead time constant

0.18 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

### Exciter field current limiter gain, $K_H$ (pu) – Exciter field current limiter gain

92 (default) | positive number

Per-unit gain of the field current limiter of the excitation system.

### Exciter field current limiter maximum output, $V_{Hmax}$ (pu) – Maximum output of the exciter field current limiter

75 (default) | positive number

Maximum per-unit output of the field current limiter of the excitation system.

### Exciter field current limiter reference, $V_{FELIM}$ (pu) – Exciter field current limiter reference

19 (default) | positive number

Per-unit reference of the field current limiter of the excitation system.

### Exciter field current limiter denominator (lag) time constant, $T_H$ (s) – Lag time constant for the exciter field current limiter

0.08 (default) | positive number

Equivalent lag time constant in the exciter field current limiter. Set this parameter to 0 when the additional lag dynamics are negligible.

**Exciter field current limiter numerator (lead) time constant, T<sub>J</sub> (s) – Lead time constant for the exciter field current limiter**

0.02 (default) | positive number

Equivalent lead time constant in the exciter field current limiter. Set this parameter to 0 when the additional lag dynamics are negligible.

**Maximum regulator output, V<sub>Amax</sub> (pu) – Upper limit of the regulator output**

75 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V<sub>Amin</sub> (pu) – Lower limit of the regulator output**

-75 (default) | real number

Minimum per-unit output voltage of the regulator.

**Maximum exciter field voltage, E<sub>Fmax</sub> (pu) – Upper limit of the exciter voltage**

44 (default) | real number

Maximum per-unit field voltage to be applied to exciter.

**Minimum exciter field voltage, E<sub>Fmin</sub> (pu) – Lower limit of the exciter voltage**

-36 (default) | real number

Minimum per-unit field voltage to be applied to exciter.

**Alternate OEL input locations (V<sub>OEL</sub>) – OEL input location**

Unused (default) | Summation point | Take-over

Location of the overexcitation limiter.

**Alternate UEL input locations (V<sub>UEL</sub>) – UEL input location**

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter.

**Alternate SCL input locations (V<sub>SCL</sub>) – SCL input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input:

- If you select *Summation point*, use the **V<sub>SCLsum</sub>** input port.
- If you select *Take-over*, use the **V<sub>OELscl</sub>** and **V<sub>UELscl</sub>** input ports.

**Exciter****Exciter field proportional constant, K<sub>E</sub> (pu) – Exciter field gain**

1.6 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant, T<sub>E</sub> (s) – Exciter field time constant**

1 (default) | positive number

Time constant for the exciter field.

**Rectifier loading factor proportional to commutating reactance,  $K_C$  (pu) – Rectifier loading factor**

0.173 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Demagnetizing factor, function of exciter alternator reactances,  $K_D$  (pu) – Demagnetization factor**

1.91 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

**Exciter output voltage for saturation factor  $S_E(E_1)$ ,  $E_1$  (pu) – First saturation output voltage**

7.4 (default) | positive number

Exciter output voltage for the first saturation factor.

**Exciter saturation factor at exciter output voltage  $E_1$ ,  $S_E(E_1)$  (1) – First saturation lookup voltage**

0.214 (default) | positive number

Saturation factor for the first exciter.

**Exciter output voltage for saturation factor  $S_E(E_2)$ ,  $E_2$  (pu) – Second saturation output voltage**

5.55 (default) | positive number

Exciter output voltage for the second saturation factor.

**Exciter saturation factor at exciter output voltage  $E_2$ ,  $S_E(E_2)$  (1) – Second saturation lookup voltage**

0.044 (default) | positive number

Saturation factor for the second exciter.

**Maximum exciter field current limit reference,  $V_{Fmax}$  (pu) – Exciter upper limit**

999 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output,  $V_{Emin}$  (pu) – Exciter lower limit**

0 (default) | real number

Minimum per-unit exciter voltage output.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

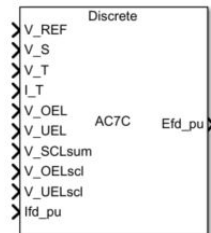
SM AC1C | SM AC2C | SM AC3C | SM AC4C | SM AC5C | SM AC7C | SM AC8C



## SM AC7C

Discrete-time or continuous-time synchronous machine AC7C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC7C block implements a synchronous machine type AC7C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

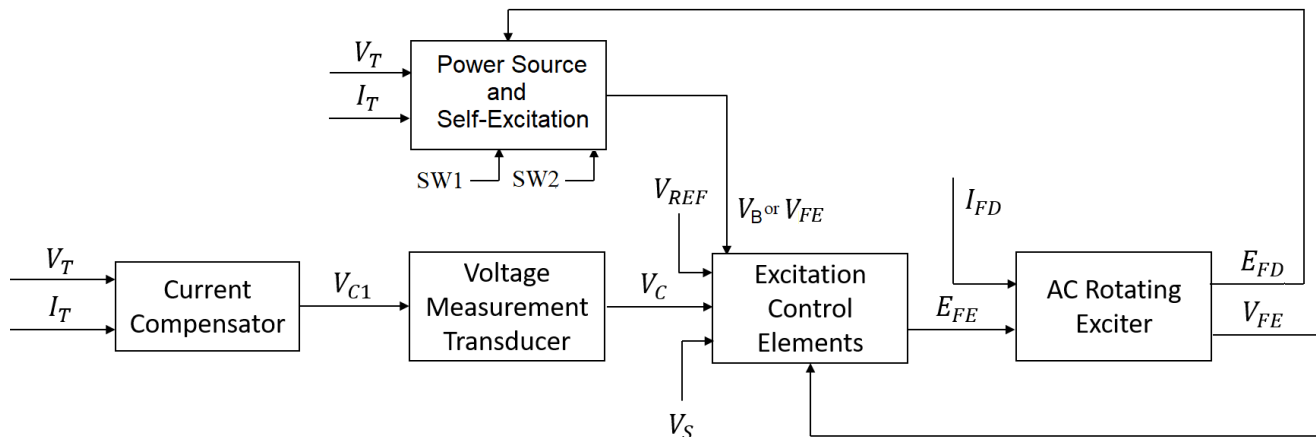
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC7C block is made up of five major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.
- The Power Source and Self-excitation models the power source for the controlled rectifier when it is independent from the terminal voltage. It also allows the representation of self-excitation for the exciter.

This diagram shows the overall structure of the AC7C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $SW_1$  and  $SW_2$  are the user-selected power source switches for the controlled rectifier.
- $V_B$  is the exciter field voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2}$$

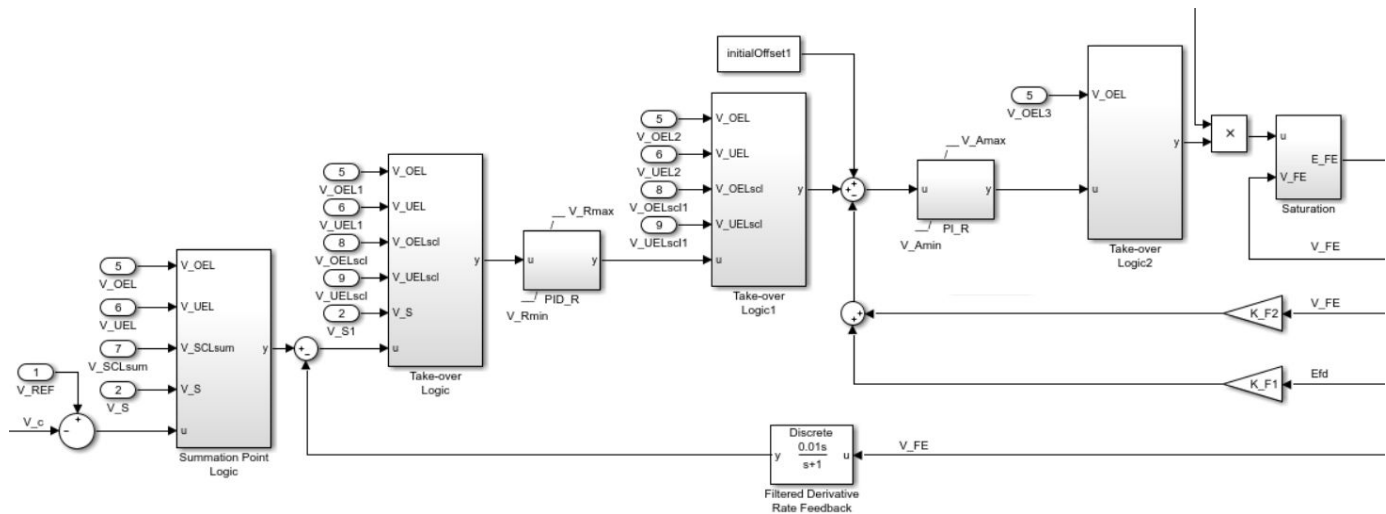
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1853.
- There are three Take-over Logic subsystems. They model the take-over point input locations for the OEL, UEL, SCL and PSS voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1853.
- The PID\_R subsystem models a PID controller that functions as a control structure for the automatic voltage regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The PI\_R subsystem models a PI controller that functions as a control structure for the field current regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- In case of self-excitation, with a nonzero  $K_R$  and the **Power source selector SW2** parameter set to **Position B**: do not use variable  $V_B$ , the voltage regulator power is derived from the exciter output voltage. The voltage regulator command signal is multiplied by the exciter output voltage,  $E_{FD}$ , times  $K_R$ . For more information about the user-selected logical switches for the power source of the controlled rectifier, see “Power Source and Self-Excitation” on page 1-1855.
- $K_{F1}$  and  $K_{F2}$  model a high-bandwidth regulator inner loop that regulates the generator field or exciting current.
- The Filtered Derivative block models the rate feedback path for the stabilization of the excitation system if the automatic voltage regulator does not have a derivative term. Here,  $K_{F3}$  and  $T_F$  are the gain and time constants of this system, respectively. Refer to the documentation for the Filtered Derivative block for the discrete and continuous implementations.
- $-K_L V_{FE}$  and  $inf$  are the minimum and maximum saturation limits for the output exciter field voltage  $E_{FE}$ .

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

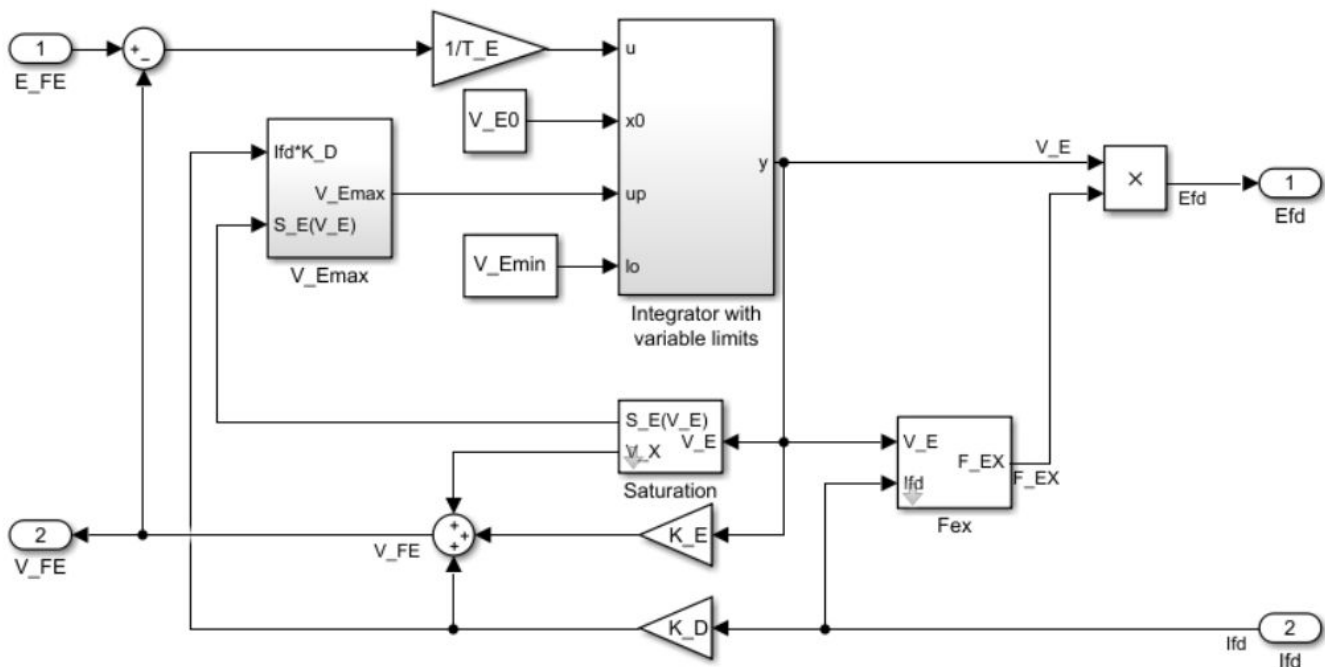
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{OELscl}$ , and the underexcitation input,  $V_{UELscl}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



In the diagram:

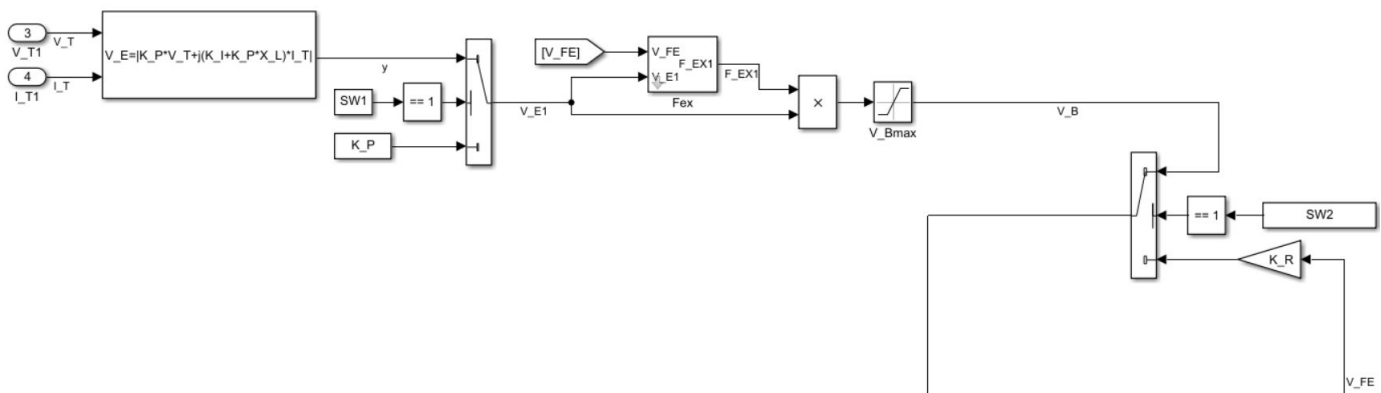
- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.

- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

### Power Source and Self-Excitation

It is possible to use different power source representations for the controlled rectifier by selecting the relevant option in the **Power source selector SW1** parameter. The power source for the controlled rectifier can be either derived from the terminal voltage (Position A: power source derived from terminal voltage) or it can be independent of the terminal voltage (Position B: power source independent from the terminal conditions). Note that this switch is dependent on the state of the second logical switch.

The second logical switch, **Power source selector SW2**, determines whether the exciter employs self-excitation or not. The exciter is self-excited if the **Power source selector SW2** parameter is set to Position B: do not use variable  $V_B$ , as the model parameters associated with the variable  $V_B$  do not impact the simulation.



## Ports

### Input

#### **V\_REF** – Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_S** – Input from stabilizer

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

#### **V\_T** – Terminal voltage

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to `Unused`.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to `Summation point`.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to `Take-over`.

Data Types: `single` | `double`

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to `Unused`.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to `Summation point`.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to `Take-over`.

Data Types: `single` | `double`

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to `Unused`.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to `Summation point`.

Data Types: single | double

### **V\_OELsc1 — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_UELsc1 — Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **I<sub>fd</sub>\_pu — Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

#### **Output**

### **E<sub>fd</sub>\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## **Parameters**

### **General**

#### **Initial field voltage, E<sub>fd0</sub> (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control****Resistive component of load compensation, R\_C (pu) – Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) – Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) – Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control****Voltage regulator proportional gain, K\_PR (pu) – Proportional gain of the voltage regulator**

40 (default)

Per-unit proportional gain of the voltage regulator.

**Voltage regulator integral gain, K\_IR (pu/s) – Integral gain of the voltage regulator**

5.6 (default)

Per-unit integral gain of the voltage regulator.

**Voltage regulator derivative gain, K\_DR (pu.s) – Derivative gain of the voltage regulator**

0 (default)

Derivative gain of the voltage regulator.



**Lag time constant for derivative channel of PID controller, T\_DR (s) – Lag time constant for PID derivative channel**

1 (default) | positive number

Equivalent lag time constant for the derivative channel of the PID controller.

**Maximum regulator output, V\_Rmax (pu) – Upper limit of the regulator output**

3.2 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V\_Rmin (pu) – Lower limit of the regulator output**

-3.2 (default) | real number

Minimum per-unit output voltage of the regulator.

**Field current regulator proportional gain, K\_PA (pu) – Proportional gain of the field current regulator**

112 (default) | real number

Per-unit proportional gain associated with the field current regulator.

**Field current regulator integral gain, K\_IA (pu/s) – Integral gain of the field current regulator**

0 (default) | real number

Per-unit integral gain associated with the field current regulator.

**Maximum field current regulator output, V\_Amax (pu) – Maximum field current regulator output**

65.2 (default) | real number

Maximum per-unit field current regulator output.

**Minimum field current regulator output, V\_Amin (pu) – Minimum field current regulator output**

-54 (default) | real number

Minimum per-unit field current regulator output.

**Generator field voltage feedback gain, K\_F1 (pu) – Generator field voltage feedback gain**

0 (default) | real number

Per-unit field voltage feedback gain associated with the generator.

**Exciter field current feedback gain, K\_F2 (pu) – Exciter field current feedback gain**

0.08 (default) | real number

Per-unit field current feedback gain associated with the exciter.

**Rate feedback gain, K\_F3 (pu) – Rate feedback gain**

0.01 (default) | real number

Per-unit rate feedback block gain.

**Rate feedback time constant, T<sub>F</sub> (s) – Rate feedback time constant**

1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Alternate PSS input locations (V<sub>S</sub>) – PSS input location**

Voltage error calculation (default) | After take-over UEL

Location of the power system stabilizer input.

**Alternate OEL input locations (V<sub>OEL</sub>) – OEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output | Take-over at output of inner loop regulator

Location of the overexcitation limiter input:

- If you select Summation point at voltage error, **V<sub>OEL</sub>** is an input of the **Summation Point Logic** subsystem.
- If you select Take-over at voltage error, **V<sub>OEL</sub>** is an input of the **Take-over Logic** subsystem.
- If you select Take-over at voltage regulator output, **V<sub>OEL</sub>** is an input of the **Take-over Logic 1** subsystem.
- If you select Take-over at output of inner loop regulator, **V<sub>OEL</sub>** is an input of the **Take-over Logic 2** subsystem.

**Alternate UEL input locations (V<sub>UEL</sub>) – UEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Location of the underexcitation limiter input:

- If you select Summation point at voltage error, **V<sub>UEL</sub>** is an input of the **Summation Point Logic** subsystem.
- If you select Take-over at voltage error, **V<sub>UEL</sub>** is an input of the **Take-over Logic** subsystem.
- If you select Take-over at voltage regulator output, **V<sub>UEL</sub>** is an input of the **Take-over Logic 1** subsystem.

**Alternate SCL input locations (V<sub>SCL</sub>) – SCL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Location of the stator current limiter input:

- If you select Summation point, use the **V<sub>SCLsum</sub>** input port.
- If you select any of the Take-over options, use the **V<sub>OELscl</sub>** and **V<sub>UElscl</sub>** input ports.

**Exciter****Exciter field proportional constant, K<sub>E</sub> (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant,  $T_E$  (s) — Exciter field time constant**

1.2 (default) | positive number

Time constant for the exciter field.

**Rectifier loading factor proportional to commutating reactance,  $K_C$  (pu) — Rectifier loading factor**

0.12 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Demagnetizing factor, function of exciter alternator reactances,  $K_D$  (pu) — Demagnetization factor**

3.3 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

**Exciter output voltage for saturation factor  $S_E(E_1)$ ,  $E_1$  (pu) — First saturation output voltage**

13.6 (default) | positive number

Exciter output voltage for the first saturation factor.

**Exciter saturation factor at exciter output voltage  $E_1$ ,  $S_E(E_1)$  (1) — First saturation lookup voltage**

3.74 (default) | positive number

Saturation factor for the first exciter.

**Exciter output voltage for saturation factor  $S_E(E_2)$ ,  $E_2$  (pu) — Second saturation output voltage**

10.2 (default) | positive number

Exciter output voltage for the second saturation factor.

**Exciter saturation factor at exciter output voltage  $E_2$ ,  $S_E(E_2)$  (1) — Second saturation lookup voltage**

0.32 (default) | positive number

Saturation factor for the second exciter.

**Maximum exciter field current limit reference,  $V_{Fmax}$  (pu) — Exciter upper limit**

23.2 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output,  $V_{Emin}$  (pu) — Exciter lower limit**

0 (default) | real number

Minimum per-unit exciter voltage output.

**Potential circuit gain coefficient,  $K_P$  (pu) — Potential circuit gain coefficient**

1 (default) | real number

Per-unit potential circuit gain coefficient.

**Dependencies**

This parameter is visible only when you set the **Power source selector SW1** parameter to Position A: power source derived from terminal voltage and the **Power source selector SW2** parameter to Position A: use variable V\_B.

**Potential circuit phase angle (degrees) – Potential circuit phase angle**

0 (default) | real number

Potential circuit phase angle, in degrees.

**Dependencies**

To enable this parameter, set **Power source selector SW1** to Position A: power source derived from terminal voltage and **Power source selector SW2** to Position A: use variable V\_B.

**Potential circuit (current) gain coefficient, K\_I (pu) – Potential circuit current gain coefficient**

0 (default) | real number

Per-unit potential circuit current gain coefficient.

**Dependencies**

To enable this parameter, set **Power source selector SW1** to Position A: power source derived from terminal voltage and **Power source selector SW2** to Position A: use variable V\_B.

**Reactance associated with potential source, X\_L (pu) – Reactance associated with potential source**

0 (default) | real number

Per-unit reactance associated with the potential source.

**Dependencies**

To enable this parameter, set **Power source selector SW1** to Position A: power source derived from terminal voltage and **Power source selector SW2** to Position A: use variable V\_B.

**Gain related to regulator and alternator field power supply, K\_R (pu) – Gain related to regulator and alternator field power supply**

0 (default) | real number

Per-unit gain related to the regulator and alternator field power supply.

**Dependencies**

To enable this parameter, set **Power source selector SW2** to Position B: do not use variable V\_B.

**Gain related to negative exciter field current capability, K\_L (pu) – Gain related to negative exciter field current capability**

0 (default) | real number

Per-unit gain related to the negative exciter field current capability.

**Power source selector SW1 — Power source selector**

Position A: power source derived from terminal voltage (default) | Position B: power source independent of terminal conditions

Position of the power source selector SW1.

**Dependencies**

To enable this parameter, set **Power source selector SW2** to Position A: use variable V\_B.

**Power source selector SW2 — Power source selector**

Position A: use variable V\_B (default) | Position B: do not use variable V\_B

Position of the power source selector SW2.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

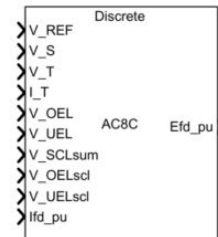
**Blocks**

SM AC1C | SM AC2C | SM AC3C | SM AC4C | SM AC5C | SM AC6C | SM AC8C

## SM AC8C

Discrete-time or continuous-time synchronous machine AC8C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM AC8C block implements a synchronous machine type AC8C excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

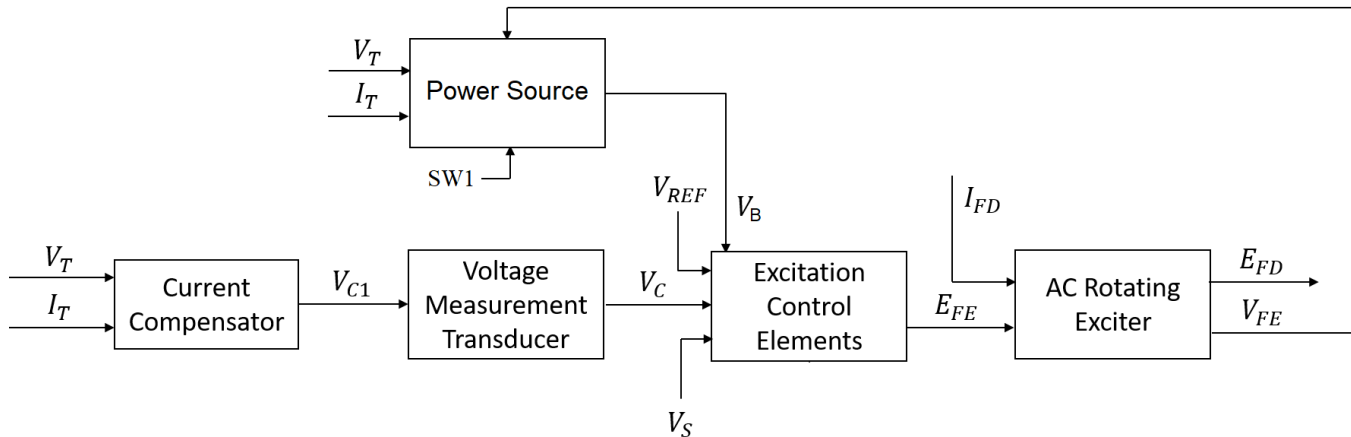
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using an AC rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM AC8C block is made up of five major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The AC Rotating Exciter models the AC rotating exciter, which produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the exciter field current (which is given the standard symbol  $V_{FE}$ ) back to the excitation system.
- The Power Source models the dependency of the power source for the controlled rectifier from the terminal voltage.

This diagram shows the overall structure of the AC8C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $SW_1$  is the user-selected power source switch for the controlled rectifier.
- $V_B$  is the exciter field voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2}$$

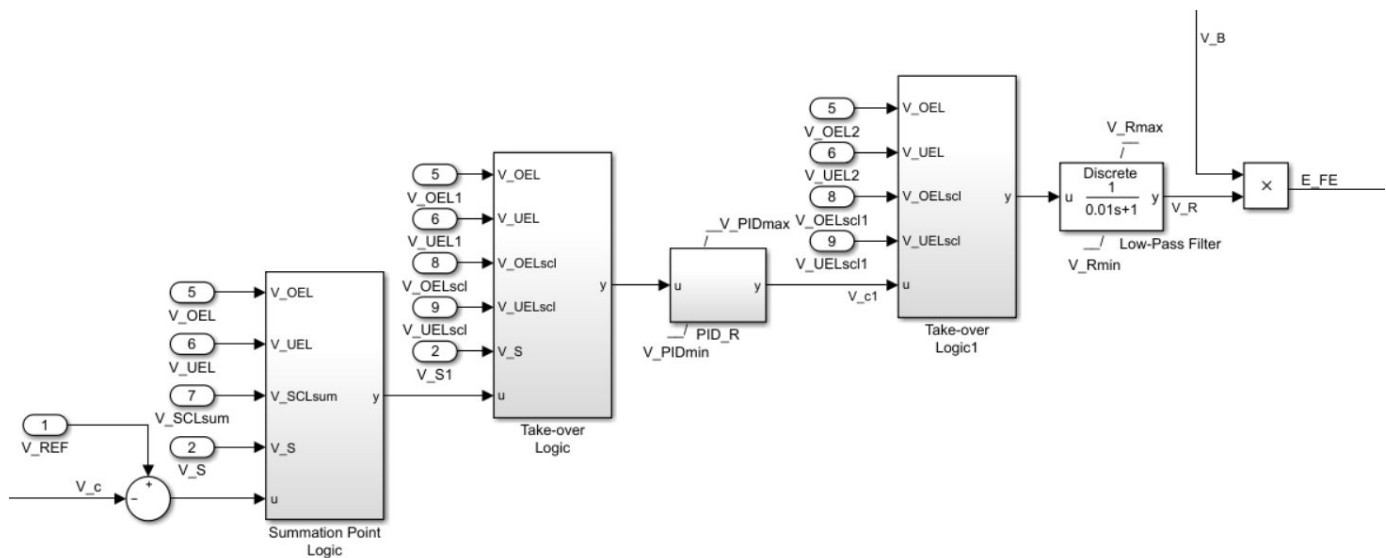
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with time constant  $T_R$ . Refer to the documentation for this block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1866.
- There are two Take-over Logic subsystems. They model the take-over point input location for the OEL, UEL, SCL and PSS voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1793.
- The PID\_R subsystem models a PID controller that functions as a control structure for the automatic voltage regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{PIDmin}$  and  $V_{PIDmax}$ , respectively.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The **Logical switch 1** parameter controls the origin of the power source for the controlled rectifier. The voltage regulator command signal  $V_R$  is multiplied by the exciter field voltage,  $V_B$ . For more information about the user-selected logical switch for the power source of the controlled rectifier, see “Power Source” on page 1-1867.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

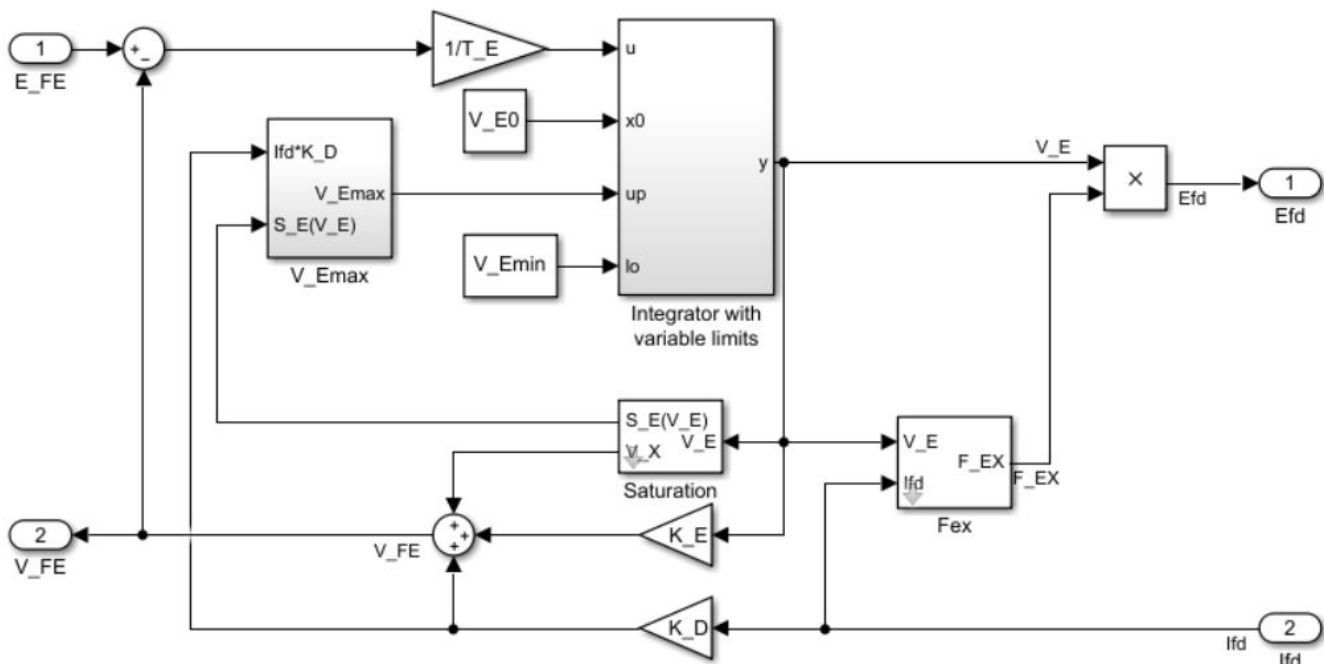


- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{OELscl}$ , and the underexcitation input,  $V_{UELscl}$ .

### AC Rotating Exciter

This diagram illustrates the overall structure of the AC rotating exciter:



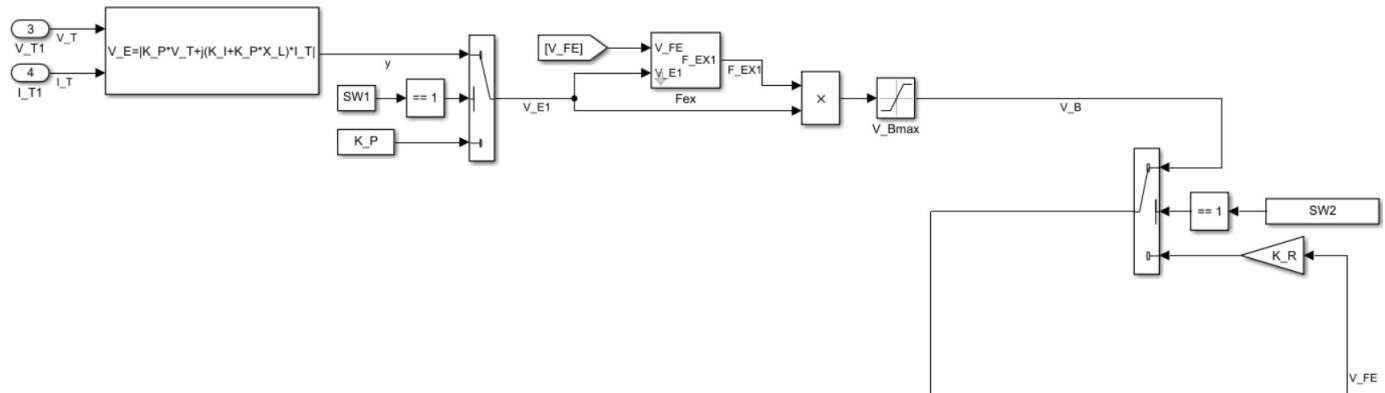
In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of three signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
  - The demagnetizing effect of the load current on the exciter output voltage is modeled using the demagnetization constant  $K_D$  in the feedback loop.
- The Integrator with variable limits subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the exciter alternator output voltage  $V_E$ .  $T_E$  is the time constant for this process.
- The nonlinear function  $F_{EX}$  models the exciter output voltage drop from the rectifier regulation. This function depends on the constant  $K_C$ , which itself is a function of commutating reactance.
- The parameters  $V_{Emin}$  and  $V_{FEmax}$  model the lower and upper limits of the rotating exciter.

### Power Source

It is possible to use different power source representations for the controlled rectifier by selecting the relevant option in the **Logical switch 1** parameter. The power source for the controlled rectifier can

be either derived from the terminal voltage (Position A: power source derived from terminal voltage) or it can be independent of the terminal voltage (Position B: power source independent from the terminal conditions).



## Ports

### Input

#### V\_REF – Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_S – Input from stabilizer

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_T – Terminal voltage

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### I\_T – Terminal current

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_OEL – Overexcitation limit signal

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

**V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

**V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_OELsc1 — Take-over stator current limit for overexcitation limiter**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_UELscl – Take-over stator current limit for underexcitation limiter**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **Ifd\_pu – Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

#### **Output**

### **Efd\_pu – Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## **Parameters**

### **General**

#### **Initial field voltage, Efd0 (pu) – Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### **Initial terminal voltage, Vt0 (pu) – Initial terminal voltage**

1 (default) | real number

Initial per-unit voltage to apply to the terminal.

#### **Initial terminal current, It0 (pu) – Initial terminal current**

1 (default) | real number

Initial per-unit voltage to apply to the terminal.

#### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

**Resistive component of load compensation, R\_C (pu) — Compensation resistance**  
0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) — Compensation reactance**  
0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) — Regulator time constant**  
0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

### Control

**Voltage regulator proportional gain, K\_PR (pu) — Proportional gain of the voltage regulator**  
80 (default)

Per-unit proportional gain of the voltage regulator.

**Voltage regulator integral gain, K\_IR (pu/s) — Integral gain of the voltage regulator**  
5 (default)

Per-unit integral gain of the voltage regulator.

**Voltage regulator derivative gain, K\_DR (pu.s) — Derivative gain of the voltage regulator**  
10 (default)

Derivative gain of the voltage regulator.

**Lag time constant for derivative channel of PID controller, T\_DR (s) — Lag time constant for PID derivative channel**  
0.02 (default) | positive number

Equivalent lag time constant for the derivative channel of the PID controller.

**Maximum voltage regulator output, V\_PIDmax (pu) – Maximum output of PID regulator**

99 (default) | positive number

Maximum admissible per-unit output of the PID regulator.

**Minimum voltage regulator output, V\_PIDmin (pu) – Minimum output of the regulator**

-99 (default) | positive number

Minimum admissible per-unit output of the PID regulator.

**Rectifier bridge gain, K\_A (pu) – Rectifier bridge gain**

1 (default) | positive number

Gain associated with the rectifier.

**Rectifier bridge time constant, T\_A (s) – Rectifier bridge time constant**

0.01 (default) | positive number

Time constant of the rectifier.

**Maximum regulator output, V\_Rmax (pu) – Upper limit of the regulator output**

35 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V\_Rmin (pu) – Lower limit of the regulator output**

0 (default) | real number

Minimum per-unit output voltage of the regulator.

**Alternate PSS input locations (V\_S) – PSS input location**

Voltage error calculation (default) | After take-over UEL

Location of the power system stabilizer input.

**Alternate OEL input locations (V\_OEL) – OEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Location of the overexcitation limiter input.

**Alternate UEL input locations (V\_UEL) – UEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Location of the underexcitation limiter input.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Location of the stator current limiter input:

- If you select Summation point, use the **V\_SCLsum** input port.

- If you select any of the Take-over options, use the **V\_OELscl** and **V\_UELscl** input ports.

## Exciter

### **Exciter field proportional constant, $K_E$ (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

### **Exciter field time constant, $T_E$ (s) – Exciter field time constant**

1.2 (default) | positive number

Time constant for the exciter field.

### **Rectifier loading factor proportional to commutating reactance, $K_C$ (pu) – Rectifier loading factor**

0.55 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

### **Demagnetizing factor, function of exciter alternator reactances, $K_D$ (pu) – Demagnetization factor**

1.1 (default) | positive number

Demagnetization factor related to the exciter alternator reactances.

### **Exciter output voltage for saturation factor $S_E(E_1)$ , $E_1$ (pu) – First saturation output voltage**

9 (default) | positive number

Exciter output voltage for the first saturation factor.

### **Exciter saturation factor at exciter output voltage $E_1$ , $S_E(E_1)$ (1) – First saturation lookup voltage**

3 (default) | positive number

Saturation factor for the first exciter.

### **Exciter output voltage for saturation factor $S_E(E_2)$ , $E_2$ (pu) – Second saturation output voltage**

6.5 (default) | positive number

Exciter output voltage for the second saturation factor.

### **Exciter saturation factor at exciter output voltage $E_2$ , $S_E(E_2)$ (1) – Second saturation lookup voltage**

0.3 (default) | positive number

Saturation factor for the second exciter.

### **Maximum exciter field current, $V_{FEmax}$ (pu) – Exciter upper limit**

6 (default) | real number

Maximum per-unit field current limit reference.

**Minimum exciter voltage output limit, V\_Emin (pu) – Exciter lower limit**

0 (default) | real number

Minimum per-unit exciter voltage output.

**Potential circuit gain coefficient, K\_P (pu) – Potential circuit gain coefficient**

1 (default) | real number

Per-unit potential circuit gain coefficient.

**Dependencies**

To enable this parameter, set **Logical switch 1** to Position A: power source derived from terminal voltage.

**Potential circuit phase angle (degrees) – Potential circuit phase angle**

0 (default) | real number

Potential circuit phase angle, in degrees.

**Dependencies**

To enable this parameter, set **Logical switch 1** to Position A: power source derived from terminal voltage.

**Potential circuit (current) gain coefficient, K\_I (pu) – Potential circuit current gain coefficient**

0 (default) | real number

Per-unit potential circuit current gain coefficient.

**Dependencies**

To enable this parameter, set **Logical switch 1** to Position A: power source derived from terminal voltage.

**Reactance associated with potential source, X\_L (pu) – Reactance associated with potential source**

0 (default) | real number

Per-unit reactance associated with the potential source.

**Dependencies**

To enable this parameter, set **Logical switch 1** to Position A: power source derived from terminal voltage.

**Rectifier loading factor proportional to commutating reactance, K\_C1 (pu) – Rectifier loading factor proportional to commutating reactance**

0 (default) | real number

Per-unit loading factor of the rectifier that is proportional to the commutating reactance.

**Logical switch 1 – Logical switch**

Position A: power source derived from terminal voltage (default) | Position B: power source independent of the terminal conditions



Position of logical switch 1.

**Maximum available exciter field voltage, V\_Bmax (pu) — Maximum available exciter field voltage**

1.25 (default) | real number

Maximum per-unit available field voltage for the exciter.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

SM AC1C | SM AC2C | SM AC3C | SM AC4C | SM AC5C | SM AC6C | SM AC7C

# SM Current Controller

Discrete-time synchronous machine current PI controller

**Library:** Simscape / Electrical / Control / SM Control



## Description

The SM Current Controller block implements a discrete time PI-based synchronous machine (SM) current controller in the rotor  $d$ - $q$  reference frame.

## Defining Equations

The block is discretized using the backward Euler method due to its first-order simplicity and its stability.

Three PI current controllers implemented in the rotor reference frame produce the reference voltage vector:

$$v_d^{ref} = \left( K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) (i_d^{ref} - i_d) + v_{d\_FF},$$

$$v_q^{ref} = \left( K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) (i_q^{ref} - i_q) + v_{q\_FF},$$

and

$$v_f^{ref} = \left( K_{p\_if} + K_{i\_if} \frac{T_s z}{z-1} \right) (i_f^{ref} - i_f),$$

where:

- $v_d^{ref}$ ,  $v_q^{ref}$ , and  $v_f^{ref}$  are the  $d$ -axis,  $q$ -axis, and field reference voltages, respectively.
- $i_d^{ref}$ ,  $v_q^{ref}$ , and  $i_f^{ref}$  are the  $d$ -axis,  $q$ -axis, and field reference currents, respectively.
- $i_d$ ,  $i_q$ , and  $i_f$  are the  $d$ -axis,  $q$ -axis, and field currents, respectively.
- $K_{p\_id}$ ,  $K_{p\_iq}$ , and  $K_{p\_if}$  are the proportional gains for the  $d$ -axis,  $q$ -axis and field controllers, respectively.
- $K_{i\_id}$ ,  $K_{i\_iq}$ , and  $K_{i\_if}$  are the integral gains for the  $d$ -axis,  $q$ -axis and field controllers, respectively.
- $v_{d\_FF}$ , and  $v_{q\_FF}$  are the feedforward voltages for the  $d$ -axis and  $q$ -axis, respectively, obtained from the machine mathematical equations and provided as inputs.

- $T_s$ , is the sample time of the discrete controller.

Using PI control results in a zero in the closed-loop transfer function which can be canceled by introducing a zero-cancellation block in the feedforward path. The zero cancellation transfer functions in discrete time are:

$$G_{ZC\_id}(z) = \frac{\frac{T_s K_{i\_id}}{K_{p\_id}}}{z + \left( \frac{T_s - \frac{K_{p\_id}}{K_{i\_id}}}{\frac{K_{p\_id}}{K_{i\_id}}} \right)},$$

$$G_{ZC\_iq}(z) = \frac{\frac{T_s K_{i\_iq}}{K_{p\_iq}}}{z + \left( \frac{T_s - \frac{K_{p\_iq}}{K_{i\_iq}}}{\frac{K_{p\_iq}}{K_{i\_iq}}} \right)},$$

and

$$G_{ZC\_if}(z) = \frac{\frac{T_s K_{i\_if}}{K_{p\_if}}}{z + \left( \frac{T_s - \frac{K_{p\_if}}{K_{i\_if}}}{\frac{K_{p\_if}}{K_{i\_if}}} \right)}.$$

Saturation must be imposed when the stator voltage vector exceeds the voltage phase limit  $V_{ph\_max}$ :

$$\sqrt{v_d^2 + v_q^2} \leq V_{ph\_max},$$

where  $v_d$ , and  $v_q$  are the  $d$ -axis and  $q$ -axis voltages, respectively.

In the case of axis prioritization, the voltages  $v_1$  and  $v_2$  are introduced, where:

- $v_1 = v_d$  and  $v_2 = v_q$  for  $d$ -axis prioritization.
- $v_1 = v_q$  and  $v_2 = v_d$  for  $q$ -axis prioritization.

The constrained (saturated) voltages  $v_1^{sat}$  and  $v_2^{sat}$  are obtained as follows:

$$v_1^{sat} = \min(\max(v_1^{unsat}, -V_{ph\_max}), V_{ph\_max}),$$

and

$$v_2^{sat} = \min(\max(v_2^{unsat}, -V_{2\_max}), V_{2\_max}),$$

where:

- $v_1^{unsat}$  and  $v_2^{unsat}$  are the unconstrained (unsaturated) voltages.
- $v_{2\_max}$  is the maximum value of  $v_2$  that does not exceed the voltage phase limit, given by  $v_{2\_max} = \sqrt{(V_{ph\_max})^2 - (v_1^{sat})^2}$ .

In the case that the direct and quadrature axes have the same priority (d-q equivalence) the constrained voltages are obtained as follows:

$$v_d^{sat} = \min(\max(v_d^{unsat}, -V_{d\_max}), V_{d\_max}),$$

and

$$v_q^{sat} = \min(\max(v_q^{unsat}, -V_{q\_max}), V_{q\_max}),$$

where

$$V_{d\_max} = \frac{V_{ph\_max}|v_d^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}},$$

and

$$V_{q\_max} = \frac{V_{ph\_max}|v_q^{unsat}|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

The constrained (saturated) field voltage  $v_f^{sat}$  is limited according to the maximum admissible value:

$$v_f^{sat} = \min(\max(v_f^{unsat}, -V_{f\_max}), V_{f\_max}),$$

where:

- $v_f^{unsat}$  is the unconstrained (unsaturated) field voltage.
- $V_{f\_max}$  is the maximum allowable field voltage.

An anti-windup mechanism is employed to avoid saturation of integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}(v_d^{sat} - v_d^{unsat}),$$

$$K_{i\_iq} + K_{aw\_iq}(v_q^{sat} - v_q^{unsat}),$$

and

$$K_{i\_if} + K_{aw\_if}(v_f^{sat} - v_f^{unsat}),$$

where  $K_{aw\_id}$ ,  $K_{aw\_iq}$ , and  $K_{aw\_if}$  are the anti-windup gains for the  $d$ -axis,  $q$ -axis and field controllers, respectively.

### Assumptions

- The plant model for direct and quadrature axis can be approximated with a first order system.
- This control solution is used only for synchronous motors with sinusoidal flux distribution and field windings.

## Ports

### Input

#### **idqfRef — Reference currents, A**

vector

Reference d-q and field currents for control of synchronous motor.

Data Types: single | double

#### **idqf — Measured currents, A**

vector

Actual d-q and field axis currents of controlled synchronous motor.

Data Types: single | double

#### **vdqFF — Purpose, V**

vector

Feedforward pre-control voltages.

Data Types: single | double

#### **VphMax — Maximum phase voltage, V**

scalar

Maximum allowable voltage in each phase.

Data Types: single | double

#### **VfMax — Maximum field voltage, V**

scalar

Maximum allowable field voltage.

Data Types: single | double

#### **Reset — External reset**

scalar

External reset signal (rising edge) for integrators.

Data Types: single | double

### Output

#### **vdqfRef — Reference voltages, V**

vector

Reference d-q and field voltages for control of synchronous motor.

Data Types: single | double

## Parameters

### General

#### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

#### **Discretization sample time – Block discretization sample time**

0.001 (default) | -1 or positive number

Specify the discretization sample time when zero-cancellation is active and sample time is set to -1 (e.g., when the block is used inside a triggered subsystem).

#### **Axis prioritization – Axis prioritization for voltage limiter**

q-axis (default) | d-axis | d-q equivalence

Prioritize or maintain ratio between d and q axes when block limits voltage.

#### **Enable zero cancellation – Feedforward zero-cancellation**

off (default) | on

Enable or disable zero-cancellation on the feedforward path.

#### **Enable pre-control voltage – Pre-control voltage**

on (default) | off

Enable or disable pre-control voltage.

### d-q Control

#### **D-axis current proportional gain – D-axis proportional gain**

1 (default) | positive number

Proportional gain of PI controller used for direct-axis current control.

#### **D-axis current integral gain – D-axis integral gain**

100 (default) | positive number

Integrator gain of PI controller used for direct-axis current control.

#### **D-axis current anti-windup gain – D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of PI controller used for direct-axis current control.

#### **Q-axis current proportional gain – Q-axis proportional gain**

1 (default) | positive number

Proportional gain of PI controller used for quadrature-axis current control.

**Q-axis current integral gain – Q-axis integral gain**

100 (default) | positive number

Integrator gain of PI controller used for quadrature-axis current control.

**Q-axis current anti-windup gain – Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of PI controller used for quadrature-axis current control.

**Field Control****Field current proportional gain – Field current proportional gain**

9 (default) | positive number

Proportional gain of PI controller used for field current control.

**Field current integral gain – Field current integral gain**

350 (default) | positive number

Integrator gain of PI controller used for field current control.

**Field current anti-windup gain – Field current anti-windup gain**

1 (default) | positive number

Anti-windup gain of PI controller used for field current control.

## Version History

Introduced in R2017b

## References

- [1] Märgner, M., and W. Hackmann. "Control challenges of an externally excited synchronous machine in an automotive traction drive application." *Emobility-Electrical Power Train, 2006*, pp. 1-6.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

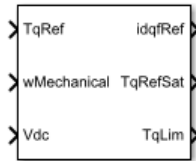
**Blocks**

SM Current Reference Generator | SM AC1C

# SM Current Reference Generator

Synchronous machine current reference generator

**Library:** Simscape / Electrical / Control / SM Control



## Description

The SM Current Reference Generator block implements a current reference generator for synchronous machine (SM) current control in the rotor  $d$ - $q$  reference frame.

## Defining Equations

The SM Current Reference Generator block can obtain the current reference using one of these methods:

- Zero  $d$ -axis control (ZDAC).
- Lookup tables.

For the ZDAC method, the block sets:

- The  $d$ -axis current reference  $i_d^{ref}$  to zero:

$$i_d^{ref} = 0,$$

- The field current reference  $i_f^{ref}$  using the torque reference:

$$i_f^{ref} = \frac{|T_{ref}|_{max}}{T_{max}},$$

where  $i_{f,max}$  is the maximum field current and  $T_{max}$  is the maximum torque.

- The  $q$ -axis current reference  $i_q^{ref}$  using the torque equation:

$$i_q^{ref} = \frac{T_{ref}}{K_t i_f^{ref}},$$

where  $T_{ref}$  is the reference torque input and  $K_t$  is the torque constant of the synchronous machine expressed by the simplified torque equation  $T = K_t i_f i_q$ .

For operation below the base speed of the synchronous machine, ZDAC is a suitable method. Above base speed, a field weakening controller is required to adjust the  $d$ -axis reference.

To pregenerate the current references for several operating points, define three lookup tables using the lookup tables approach:



$$i_d^{ref} = f(n_m, T_{ref}, v_{dc}),$$

$$i_q^{ref} = g(n_m, T_{ref}, v_{dc}),$$

and

$$i_f^{ref} = h(n_m, T_{ref}, v_{dc}).$$

## Ports

### Input

#### **TqRef** — Reference torque, N\*m

scalar

Desired mechanical torque produced by the synchronous machine.

Data Types: `single` | `double`

#### **wMechanical** — Rotor mechanical speed, rad/s

scalar

Mechanical angular velocity of the synchronous machine rotor, obtained via direct measurement from the synchronous machine.

Data Types: `single` | `double`

#### **Vdc** — DC-link voltage, V

scalar

DC-link voltage of the converter. For the ZDAC method, this value is used to limit the output reference torque and torque limit. For the lookup table method, this value is used as an input to the lookup tables.

Data Types: `single` | `double`

### Output

#### **idqfRef** — Reference currents, A

vector

Reference d-q and field currents to be given as inputs to a current controller.

Data Types: `single` | `double`

#### **TqRefSat** — Reference torque, N\*m

scalar

Reference torque saturated by the calculated torque limit **TqLim**.

Data Types: `single` | `double`

#### **TqLim** — Torque limit, N\*m

scalar

Torque limit imposed by both the electrical and mechanical constraints of the system.

Data Types: single | double

## Parameters

### General Parameters

#### **Nominal dc-link voltage (V) — Rated DC voltage**

300V (default) | positive number

Nominal DC-link voltage of the electrical source.

#### **Maximum power (W) — Maximum power**

30000W (default) | positive number

Maximum synchronous machine power.

#### **Maximum torque (N\*m) — Maximum torque**

250N\*m (default) | positive number

Maximum synchronous machine torque.

#### **Maximum field current (A) — Maximum field current**

25A (default) | positive number

Maximum field current of the synchronous machine.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | -1 or positive number

Sample time for the block (-1 for inherited). If this block is used inside a triggered subsystem, the sample time should be -1. If this block is used in a continuous variable-step model, then the sample time can be explicitly specified.

### Reference Generation Strategy

#### **Current references — Current reference strategy**

Zero d-axis control (default) | Lookup-table based

Select the strategy for determining current references.

#### **Torque constant (N\*m/A) — Torque constant**

0.0375 (default) | positive number

Torque constant of the synchronous machine.

#### **Mechanical speed vector, wMechanical (rpm) — Rotor speed lookup vector**

[0, 3000] (default) | positive monotonically increasing vector

Speed vector used in the lookup-tables for determining current references.

#### **Torque reference vector, TqRef (N\*m) — Torque reference lookup vector**

[-100, 0, 100] (default) | positive monotonically increasing vector

Torque vector used in the lookup-tables for determining current references.

**DC-link voltage vector, Vdc (V) — DC-link voltage lookup vector**

[300, 350] (default) | positive monotonically increasing vector

DC-link voltage vector used in the lookup-tables for determining current references.

**D-axis current reference matrix, id(wMechanical,TqRef,Vdc), (A) — Reference d-axis current values**

zeros(2,3,2) (default) | real matrix

Direct-axis current reference lookup data.

**Q-axis current reference matrix, iq(wMechanical,TqRef,Vdc), (A) — Reference q-axis current values**

zeros(2,3,2) (default) | real matrix

Quadrature-axis current reference lookup data.

**Field current reference matrix, iq(wMechanical,TqRef,Vdc), (A) — Reference field current values**

zeros(2,3,2) (default) | real matrix

Field current reference lookup data.

## Version History

**Introduced in R2017b**

## References

- [1] Girardin, A., and G. Friedrich. "Optimal control for a wound rotor synchronous starter generator." *Industry Applications Conference, 2006*, pp. 14-19.
- [2] Carpiuc, S., C. Lazar, and D. I. Patrascu. "Optimal Torque Control of the Externally Excited Synchronous Machine." *Control Engineering and Applied Informatics, 14(2), 2012*, pp. 80-88.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

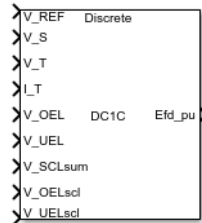
**Blocks**

SM Current Controller | SM AC1C

## SM DC1C

Discrete-time or continuous-time synchronous machine DC1C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM DC1C block models a synchronous machine type DC1C excitation system that conforms to IEEE 421.5-2016<sup>[1]</sup>.

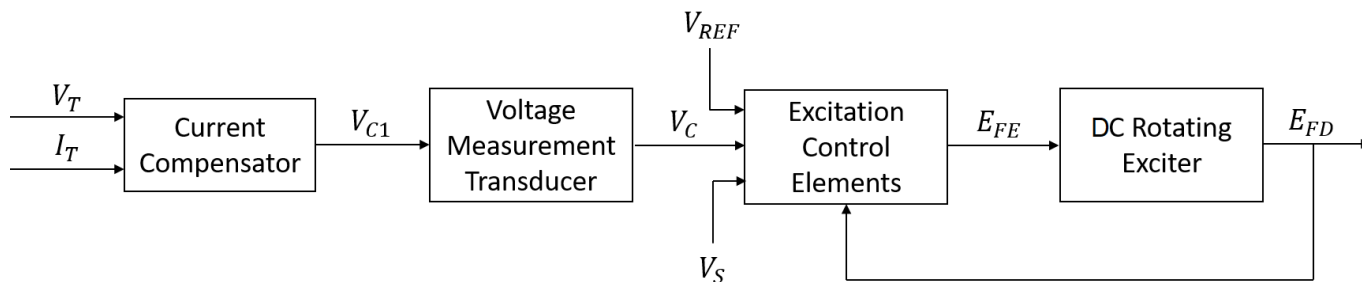
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using a DC commutator rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM DC1C block is made up of four major components:

- The current compensator modifies the measured terminal voltage as a function of terminal current.
- The voltage measurement transducer simulates the dynamics of a terminal voltage transducer by using a low-pass filter.
- The excitation control elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The DC rotating exciter models the DC commutator rotating exciter and produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the field voltage back to the excitation system.

This diagram shows the overall structure of the DC1C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  is the exciter field voltage.
- $E_{FD}$  is the field voltage.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2}$$

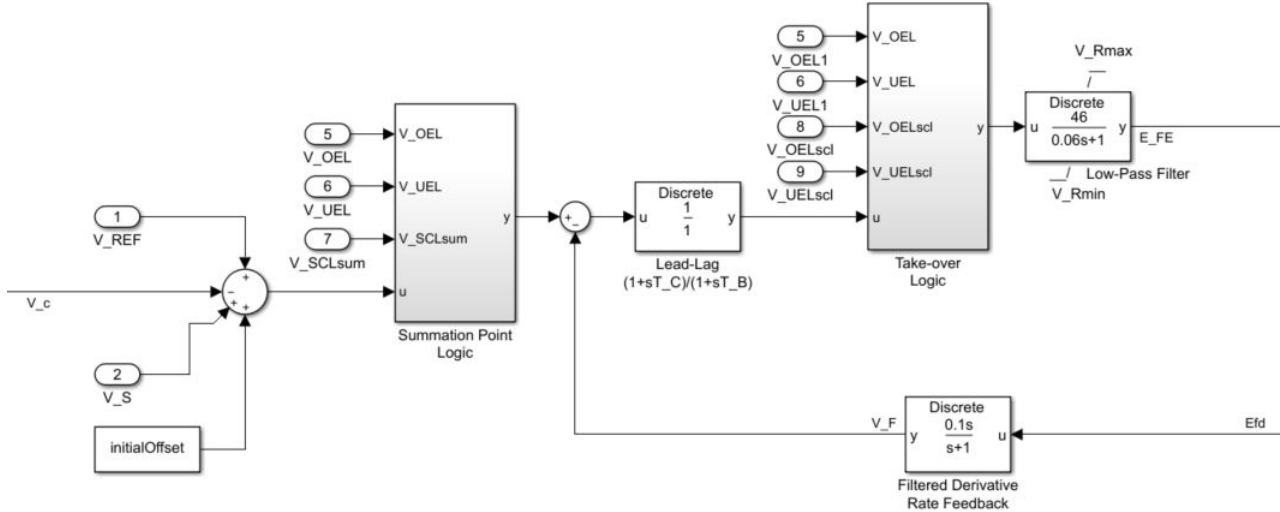
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for this block for the exact discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1888.
- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for this block for the exact discrete and continuous implementations.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1888.
- The Filtered Derivative block models the rate feedback path for the stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constant of this system, respectively. Refer to the documentation for the Filtered Derivative block for the exact discrete and continuous implementations.

## Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which can risk desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

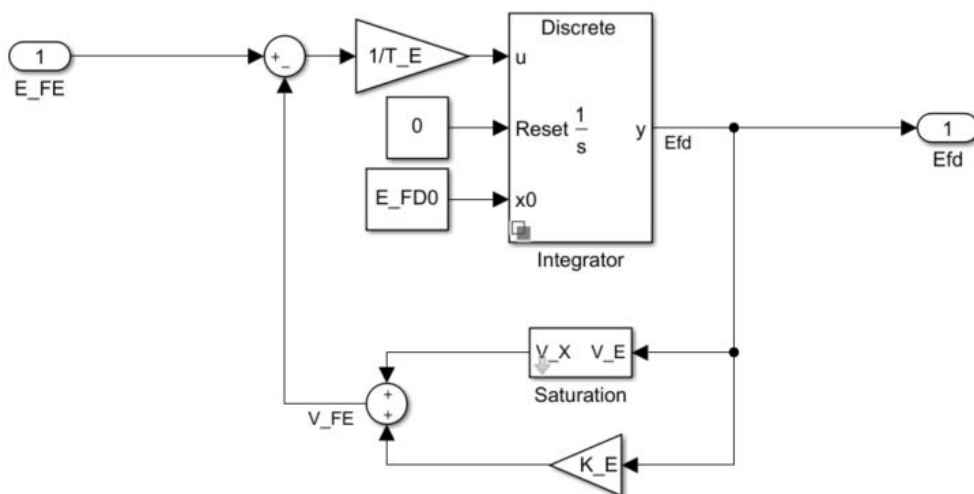
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both an overexcitation input  $V_{OELscl}$  and an underexcitation input  $V_{UELscl}$ .

## DC Rotating Exciter

This diagram illustrates the overall structure of the DC commutator rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of two signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
- The **Integrator** subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the output field voltage  $E_{fd}$ .  $T_E$  is the time constant for this process.

## Ports

### Input

#### V\_REF — Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_S — Input from stabilizer

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

**V\_T – Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

**I\_T – Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

**V\_OEL – Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

**V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

**V\_SCLsum – Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.



**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_OELscl — Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_UELscl — Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Output****Efd\_pu — Field voltage**

scalar

Per-unit field voltage to be applied to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

**Parameters****General****Initial field voltage, Efd0 (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to be applied to the field circuit of the synchronous machine.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control****Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) — Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control****Regulator output gain, K\_A (pu) — Regulator gain**

46 (default) | positive number

Gain associated with the voltage regulator.

**Regulator time constant, T\_A (s) — Regulator time constant**

0.06 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant, T\_B (s) — Regulator lag time constant**

0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant, T\_C (s) — Regulator lead time constant**

0 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this to 0 when the additional lead dynamics are negligible.

**Rate feedback gain,  $K_F$  (pu) — Rate feedback gain**

0.1 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

**Rate feedback time constant,  $T_F$  (s) — Rate feedback time constant**

1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Maximum controller output,  $V_{Rmax}$  (pu) — Controller output upper limit**

1 (default) | real number

Maximum per-unit output voltage of the controller.

**Minimum controller output,  $V_{Rmin}$  (pu) — Controller output lower limit**

-0.9 (default) | real number

Minimum per-unit output voltage of the controller.

**Alternate OEL input locations ( $V_{OEL}$ ) — Overexcitation limiter input location**

Unused (default) | Summation point | Take-over

Select the overexcitation limiter input location.

**Alternate UEL input locations ( $V_{UEL}$ ) — Underexcitation limiter input location**

Unused (default) | Summation point | Take-over

Select the underexcitation limiter input location.

**Alternate SCL input locations ( $V_{SCL}$ ) — Stator current limiter input location**

Unused (default) | Summation point | Take-over

Select stator current limiter input location. To specify the SCL input:

- If you select *Summation point*, use the  **$V_{SCLsum}$**  input port.
- If you select *Take-over*, use the  **$V_{OELscl}$**  and  **$V_{UELscl}$**  input ports.

**Exciter**

**Exciter field proportional constant,  $K_E$  (pu) — Exciter field gain**

0.05 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant,  $T_E$  (s) — Exciter field time constant**

0.46 (default) | positive number

Time constant for the exciter field.

**Lower limit,  $E_{FDmin}$  (pu) — Field voltage lower limit**

-99 (default) | positive number

Lower limit for the field voltage.

**Exciter output voltage for saturation factor  $S_E(E_1)$ ,  $E_1$  (pu) – First saturation output voltage**

3.1 (default) | positive number

Exciter output voltage for the first saturation factor.

**Exciter saturation factor at exciter output voltage  $E_1$ ,  $S_E(E_1)$  (1) – First saturation lookup voltage**

0.33 (default) | positive number

First exciter saturation factor.

**Exciter output voltage for saturation factor  $S_E(E_2)$ ,  $E_2$  (pu) – Second saturation output voltage**

2.3 (default) | positive number

Exciter output voltage for second saturation factor.

**Exciter saturation factor at exciter output voltage  $E_2$ ,  $S_E(E_2)$  (1) – Second saturation lookup voltage**

0.10 (default) | positive number

Second exciter saturation factor.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

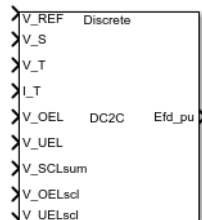
## See Also

SM DC2C

## SM DC2C

Discrete-time or continuous-time synchronous machine DC2C excitation system including an automatic voltage regulator and an exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM DC2C block models a synchronous machine type DC2C excitation system that conforms to IEEE 421.5-2016<sup>[1]</sup>.

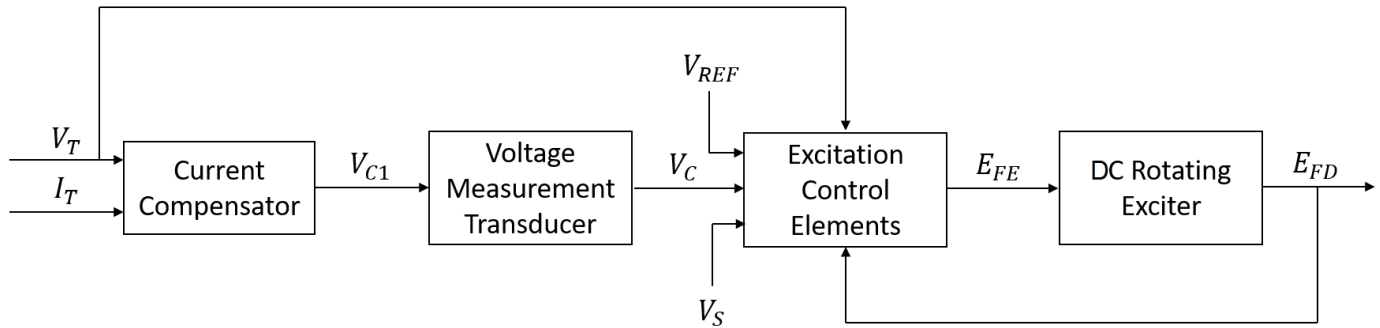
Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using a DC commutator rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM DC2C block is made up of four major components:

- The current compensator modifies the measured terminal voltage as a function of terminal current.
- The voltage measurement transducer simulates the dynamics of a terminal voltage transducer by using a low-pass filter.
- The excitation control elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The DC rotating exciter models the DC commutator rotating exciter and produces a field voltage that is applied to the controlled synchronous machine. The block also feeds the field voltage back to the excitation system.

This diagram shows the overall structure of the DC2C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FE}$  is the exciter field voltage.
- $E_{FD}$  is the field voltage.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

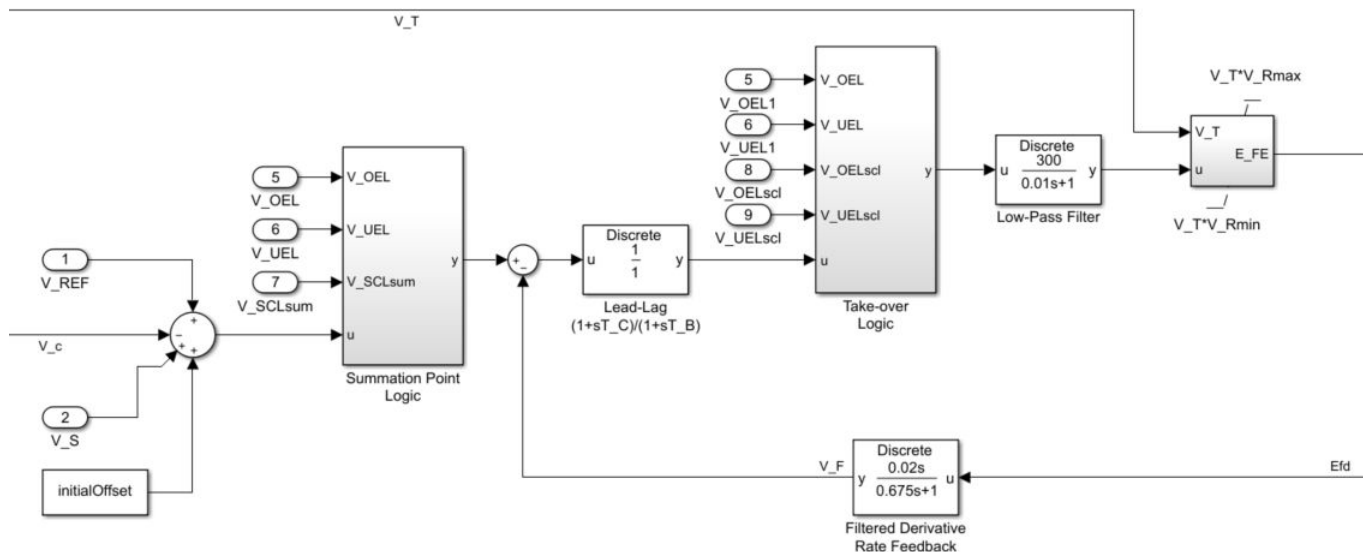
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for this block for the exact discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1897.
- The Lead-Lag block models additional dynamics associated with the voltage regulator. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the Lead-Lag block documentation for the exact discrete and continuous implementations.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1897.
- The Filtered Derivative block models the rate feedback path for stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constant of this system, respectively. Refer to the documentation for the Filtered Derivative block for the exact discrete and continuous implementations.
- $V_T*V_{Rmax}$  and  $V_T*V_{Rmin}$  are the minimum and maximum saturation limits for the output exciter field voltage  $E_{FE}$ .

## Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which can risk desynchronization.

- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

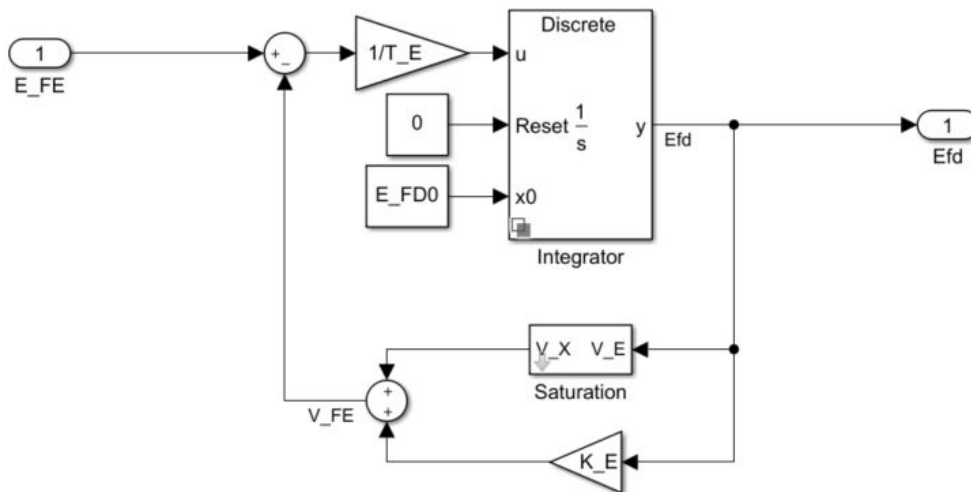
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both an overexcitation input  $V_{OELscl}$  and an underexcitation input  $V_{UELscl}$ .

### DC Rotating Exciter

This diagram illustrates the overall structure of the DC commutator rotating exciter:



In the diagram:

- The exciter field current  $V_{FE}$  is modeled as the summation of two signals:
  - The nonlinear function  $V_x$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
- The **Integrator** subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the output field voltage  $E_{fd}$ .  $T_E$  is the time constant for this process.

### Ports

#### Input

#### V\_REF — Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double



**V\_S – Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

**V\_T – Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

**I\_T – Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

**V\_OEL – Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

**V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

**V\_SCLsum – Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_OELscl – Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_UELscl – Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Output****Efd\_pu – Field voltage**

scalar

Per-unit field voltage to be applied to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

## Parameters

### General

#### Initial field voltage, $E_{fd0}$ (pu) — Initial output voltage

1 (default) | real number

Initial per-unit voltage to be applied to the field circuit of the synchronous machine.

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### Resistive component of load compensation, $R_C$ (pu) — Compensation resistance

0 (default) | positive number

Resistance used in the current compensation system. Set this and **Reactance component of load compensation,  $X_C$  (pu)** to 0 to disable current compensation.

#### Reactance component of load compensation, $X_C$ (pu) — Compensation reactance

0 (default) | positive number

Reactance used in the current compensation system. Set this and **Resistive component of load compensation,  $R_C$  (pu)** to 0 to disable current compensation.

#### Regulator input filter time constant, $T_R$ (s) — Regulator time constant

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

### Control

#### Regulator output gain, $K_A$ (pu) — Regulator gain

300 (default) | positive number

Gain associated with the voltage regulator.

#### Regulator time constant, $T_A$ (s) — Regulator time constant

0.01 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant,  $T_B$  (s) — Regulator lag time constant**  
0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant,  $T_C$  (s) — Regulator lead time constant**  
0 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this to 0 when the additional lead dynamics are negligible.

**Rate feedback gain,  $K_F$  (pu) — Rate feedback gain**  
0.02 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

**Rate feedback time constant,  $T_F$  (s) — Rate feedback time constant**  
0.675 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system.

**Maximum controller output,  $V_{Rmax}$  (pu) — Controller output upper limit**  
4.95 (default) | real number

Maximum per-unit output voltage of the controller.

**Minimum controller output,  $V_{Rmin}$  (pu) — Controller output lower limit**  
-4.90 (default) | real number

Minimum per-unit output voltage of the controller.

**Alternate OEL input locations ( $V_{OEL}$ ) — Overexcitation limiter input location**  
Unused (default) | Summation point | Take-over

Select the overexcitation limiter input location.

**Alternate UEL input locations ( $V_{UEL}$ ) — Underexcitation limiter input location**  
Unused (default) | Summation point | Take-over

Select the underexcitation limiter input location.

**Alternate SCL input locations ( $V_{SCL}$ ) — Stator current limiter input location**  
Unused (default) | Summation point | Take-over

Select stator current limiter input location. To specify the SCL input:

- If you select *Summation point*, use the  **$V_{SCLsum}$**  input port.
- If you select *Take-over*, use the  **$V_{OELscl}$**  and  **$V_{UELscl}$**  input ports.

### **Exciter**

**Exciter field proportional constant,  $K_E$  (pu) — Exciter field gain**  
1 (default) | positive number

Proportional constant for exciter field.

**Exciter field time constant, T\_E (s) – Exciter field time constant**

1.33 (default) | positive number

Time constant for exciter field.

**Lower limit, E\_FDmin (pu) – Field voltage lower limit**

-99 (default) | positive number

Lower limit for the field voltage.

**Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

3.05 (default) | positive number

Exciter output voltage for first saturation factor.

**Exciter saturation factor at exciter output voltage E\_1, S\_E(E\_1) (1) – First saturation lookup voltage**

0.279 (default) | positive number

First exciter saturation factor.

**Exciter output voltage for saturation factor S\_E(E\_2), E\_2 (pu) – Second saturation output voltage**

2.29 (default) | positive number

Exciter output voltage for second saturation factor.

**Exciter saturation factor at exciter output voltage E\_2, S\_E(E\_2) (1) – Second saturation lookup voltage**

0.117 (default) | positive number

Second exciter saturation factor.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

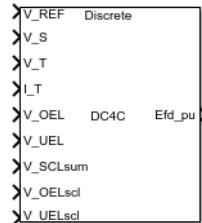
**See Also**

SM DC1C

## SM DC4C

Discrete-time or continuous-time synchronous machine DC4C excitation system including an automatic voltage regulator and exciter

**Library:** Simscape / Electrical / Control / SM Control



### Description

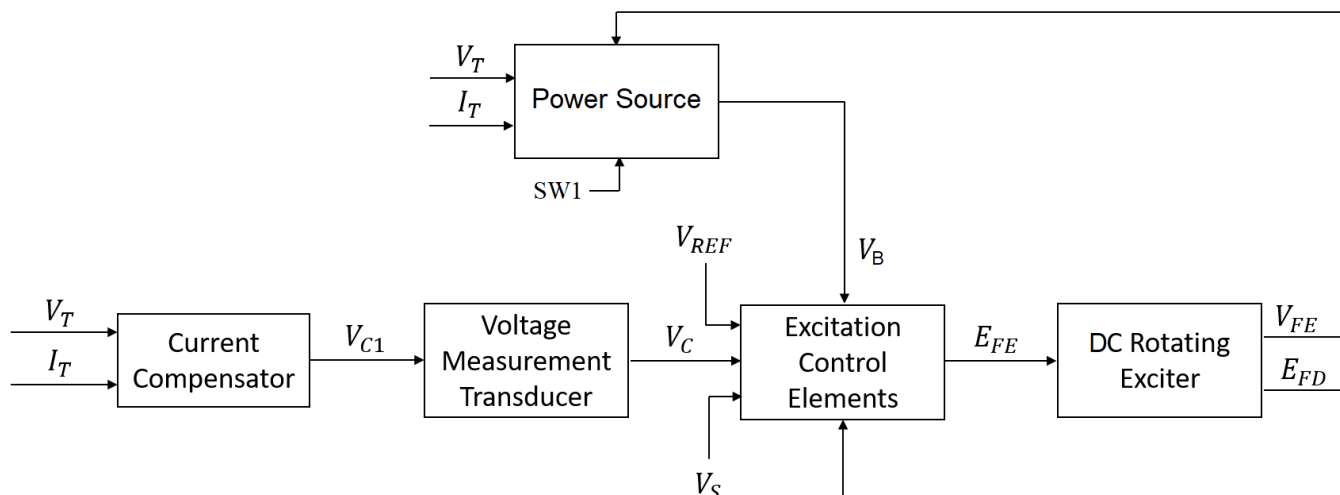
The SM DC4C block models a synchronous machine type DC4C excitation system that conforms to IEEE 421.5-2016<sup>[1]</sup>. Use this block to model the control and regulation of the field voltage of a synchronous machine that operates as a generator using a DC commutator rotating exciter.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the block for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the block for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM DC4C block is made up of five major components:

- The current compensator modifies the measured terminal voltage as a function of terminal current.
- The voltage measurement transducer simulates the dynamics of a terminal voltage transducer by using a low-pass filter.
- The excitation control elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the exciter field voltage.
- The DC rotating exciter models the DC commutator rotating exciter which produces and applies a field voltage to the controlled synchronous machine. The block also feeds the field voltage back to the excitation system.
- The power source models the dependency of the power source for the controlled rectifier from the terminal voltage.

This diagram shows the overall structure of the DC4C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $SW_1$  is the user-selected power source switch for the controlled rectifier.
- $V_B$  is the exciter field voltage.
- $E_{FE}$  and  $V_{FE}$  are the exciter field voltage and current, respectively.
- $E_{FD}$  is the field voltage.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

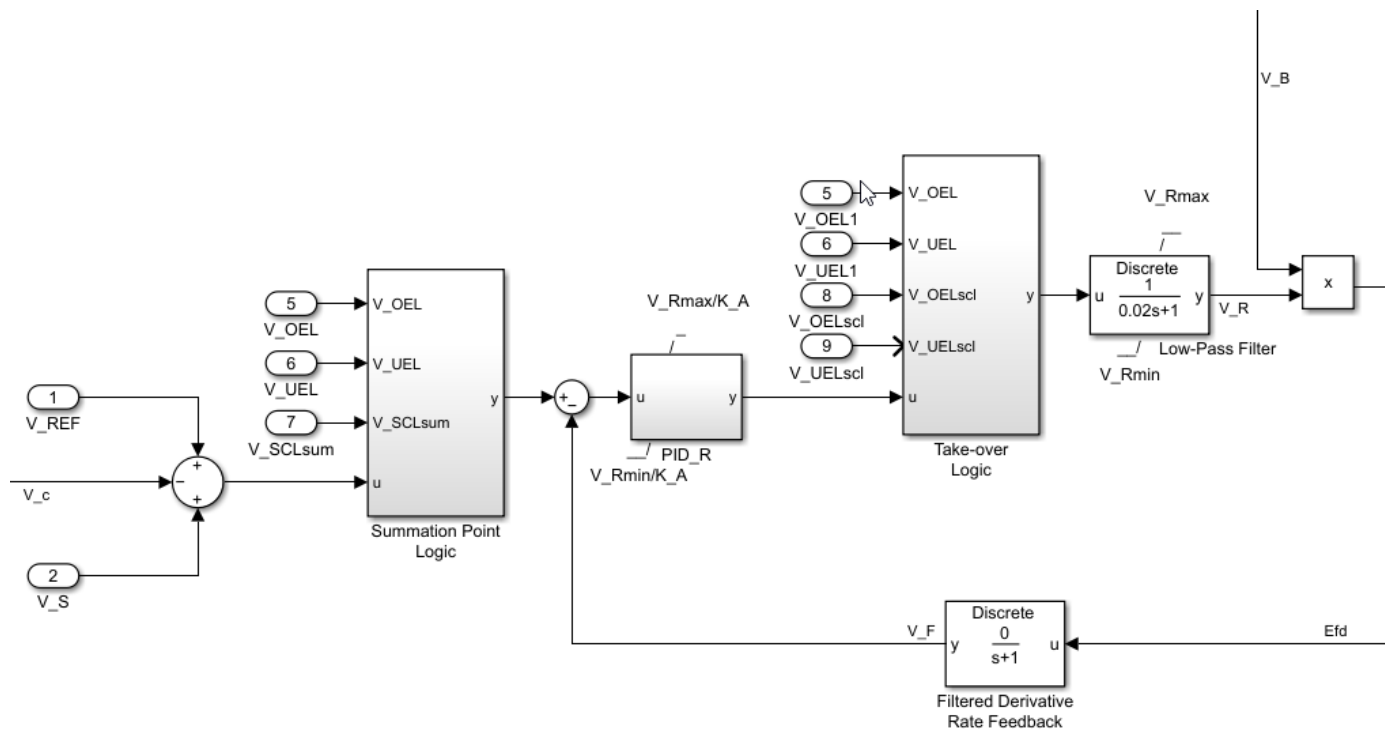
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . To see the exact discrete and continuous implementations, see Low-Pass Filter.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), and stator current limiter (SCL) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1897.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The PID\_R subsystem models a PID controller that functions as a control structure for the automatic voltage regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmax} / K_A$  and  $V_{Rmin} / K_A$ , respectively.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1897.
- The Filtered Derivative block models the rate feedback path for stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constant of this system, respectively. To see the exact discrete and continuous implementations, see Filtered Derivative.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which can risk desynchronization.



- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

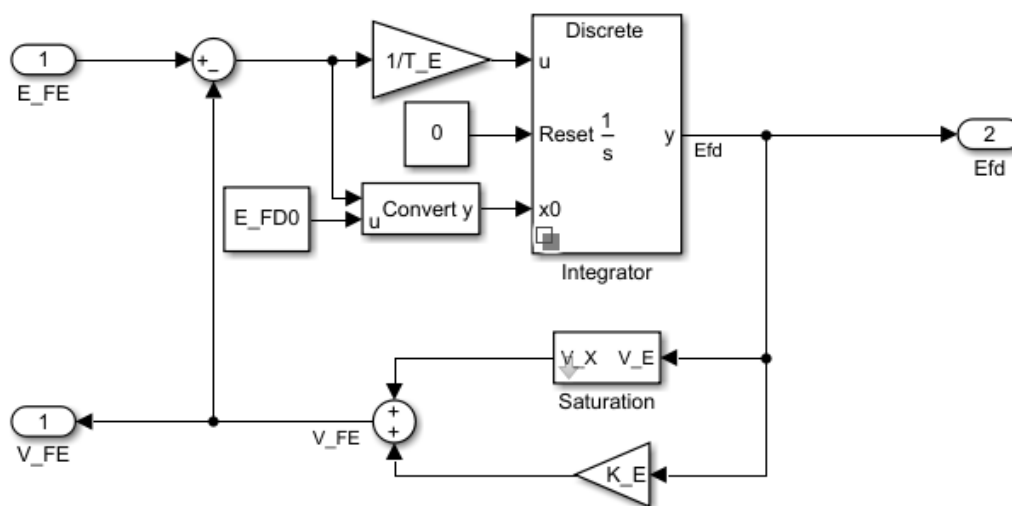
Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input **V\_SCLsum**. If you are using the stator current limiter at the take-over point, use both an overexcitation input **V\_OELscl** and an underexcitation input **V\_UELscl**.

### DC Rotating Exciter

This diagram illustrates the overall structure of the DC commutator rotating exciter:

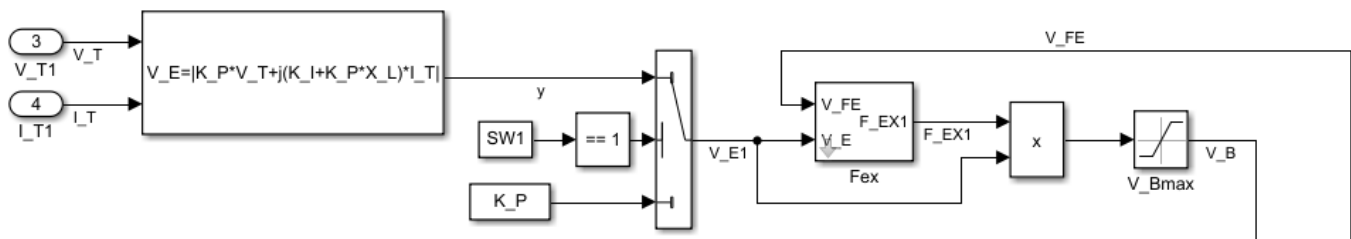


In the diagram:

- The exciter field current  $V_{FE}$  is the sum of two signals:
  - The nonlinear function  $V_X$  models the saturation of the exciter output voltage.
  - The proportional term  $K_E$  models the linear relationship between exciter output voltage and the exciter field current.
- The Integrator subsystem integrates the difference between  $E_{FE}$  and  $V_{FE}$  to generate the output field voltage  $E_{fd}$ .  $T_E$  is the time constant for this process.

### Power Source

You can change the power source representations for the controlled rectifier with the **Logical switch 1** parameter. The power source for the controlled rectifier can be either derived from the terminal voltage (Position A: power source derived from terminal voltage) or it can be independent of the terminal voltage (Position B: power source independent from the terminal conditions).



## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T — Terminal current**

scalar

Terminal current magnitude, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: `single` | `double`

**V\_UEL – Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

**V\_SCLsum – Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

**V\_OELscL – Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter to prevent field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_UELscL – Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter to prevent field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

## Output

### **Efd\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: `single` | `double`

## Parameters

### General

#### **Initial field voltage, Efd0 (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

#### **Initial terminal voltage, Vt0 (pu) — Initial terminal voltage**

1 (default) | real number

Initial per-unit terminal voltage.

#### **Initial terminal current, It0 (pu) — Initial terminal current**

1 (default) | real number

Initial per-unit terminal current.

#### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Pre-Control

#### **Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system, in per-unit. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

#### **Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system, in per-unit. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant,  $T_R$  (s) – Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering, in seconds.

**Control****Regulator proportional gain,  $K_{PR}$  (pu) – Proportional gain of the regulator**

80 (default) | scalar

Per-unit proportional gain of the regulator.

**Regulator integral gain,  $K_{IR}$  (pu/s) – Integral gain of the regulator**

20 (default) | scalar

Per-unit integral gain of the regulator.

**Regulator derivative gain,  $K_{DR}$  (pu.s) – Derivative gain of the regulator**

20 (default) | scalar

Derivative gain of the regulator.

**Regulator derivative filter time constant,  $T_{DR}$  (s) – Filter time constant for PID derivative channel**

0.01 (default) | positive number

Equivalent filter time constant for the derivative channel of the PID controller, in seconds.

**Regulator output gain,  $K_A$  (pu) – Regulator gain**

1 (default) | positive number

Gain associated with the voltage regulator.

**Regulator time constant,  $T_A$  (s) – Regulator time constant**

0.02 (default) | positive number

Major time constant of the voltage regulator, in seconds.

**Rate feedback gain,  $K_F$  (pu) – Rate feedback gain**

0 (default) | positive number

Rate feedback block gain for the stabilization of the excitation system.

**Rate feedback time constant,  $T_F$  (s) – Rate feedback time constant**

1 (default) | positive number

Rate feedback block time constant for the stabilization of the excitation system, in seconds.

**Maximum controller output,  $V_{Rmax}$  (pu) – Controller output upper limit**

2.7 (default) | real number

Maximum per-unit output voltage of the controller.

**Minimum controller output,  $V_{Rmin}$  (pu) – Controller output lower limit**

-2.7 (default) | real number

Minimum per-unit output voltage of the controller.

**Alternate OEL input locations (V\_OEL) – Overexcitation limiter input location**

Unused (default) | Summation point | Take-over

Location of the overexcitation limiter input.

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

**Alternate UEL input locations (V\_UEL) – Underexcitation limiter input location**

Unused (default) | Summation point | Take-over

Location of the underexcitation limiter input.

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

**Alternate SCL input locations (V\_SCL) – Stator current limiter input location**

Unused (default) | Summation point | Take-over

Location of the stator current limiter input. To specify the stator current limiter input:

- If you select Summation point, use the **V\_SCLsum** input port.
- If you select Take-over, use the **V\_OELscl** and **V\_UELscl** input ports.

**Exciter****Exciter field proportional constant, K\_E (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

**Exciter field time constant, T\_E (s) – Exciter field time constant**

0.8 (default) | positive number

Time constant for the exciter field, in seconds.

**Lower limit, E\_FDmin (pu) – Field voltage lower limit**

-99 (default) | positive number

Lower limit for the field voltage.

**Exciter output voltage for saturation factor S\_E(E\_1), E\_1 (pu) – First saturation output voltage**

4.8 (default) | positive number

Exciter output voltage for the first saturation factor.

**Exciter saturation factor at exciter output voltage  $E_1$ ,  $S_E(E_1)$  (1) – First saturation lookup voltage**

1.54 (default) | positive number

First exciter saturation factor.

**Exciter output voltage for saturation factor  $S_E(E_2)$ ,  $E_2$  (pu) – Second saturation output voltage**

3.6 (default) | positive number

Exciter output voltage for the second saturation factor.

**Exciter saturation factor at exciter output voltage  $E_2$ ,  $S_E(E_2)$  (1) – Second saturation lookup voltage**

1.26 (default) | positive number

Second exciter saturation factor.

**Potential circuit gain coefficient,  $K_P$  (pu) – Potential circuit gain coefficient**

1 (default) | real number

Per-unit potential circuit gain coefficient.

**Potential circuit phase angle (degrees) – Potential circuit phase angle**

0 (default) | real number

Potential circuit phase angle, in degrees.

**Potential circuit (current) gain coefficient,  $K_I$  (pu) – Potential circuit current gain coefficient**

0 (default) | real number

Per-unit potential circuit current gain coefficient.

**Reactance associated with potential source,  $X_L$  (pu) – Reactance associated with potential source**

0 (default) | real number

Per-unit reactance associated with the potential source.

**Rectifier loading factor proportional to commutating reactance,  $K_{C1}$  (pu) – Rectifier loading factor proportional to commutating reactance**

0 (default) | real number

Per-unit loading factor of the rectifier that is proportional to the commutating reactance.

**Logical switch 1 – Logical switch position**

Position A: power source derived from terminal voltage (default) | Position B: power source independent of the terminal conditions

Position of logical switch 1.

**Maximum available exciter field voltage, V\_Bmax (pu) — Maximum available exciter field voltage**

1.5 (default) | real number

Maximum per-unit available field voltage for the exciter.

## **Version History**

**Introduced in R2021b**

## **References**

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

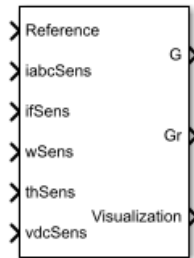
SM DC1C | SM DC2C



# SM Field-Oriented Control

Synchronous machine field-oriented control

**Library:** Simscape / Electrical / Control / SM Control



## Description

The SM Field-Oriented Control block implements a synchronous machine (SM) field-oriented control structure. Field Oriented Control (FOC) is a performant AC motor control strategy that decouples torque and flux by transforming the stationary phase currents to a rotating frame. Use FOC when rotor speed and position are known and your application requires:

- High torque and low current at startup
- High efficiency

## Equations

The SM FOC is made up of several control blocks from the Control library. To see and modify these blocks and the FOC's internal structure, right-click the block in Simulink and select **Mask > Look Under Mask**. The overall control structure is made up of several parts:

- The outer loop controller converts the reference signal you supply to the reference  $d$ -axis,  $q$ -axis, and field currents.

You can choose the type of reference signal you provide using the `Control mode` parameter:

- `Velocity control` — Control or regulate the rotation speed of the synchronous machine. An internal Velocity Controller block generates a reference torque from the rotor speed error.
- `Torque control` — Control or regulate the mechanical torque of the SM.

An internal SM Current Reference Generator block generates the reference currents using a proportional-integral (PI) controller, minimizing the torque error.

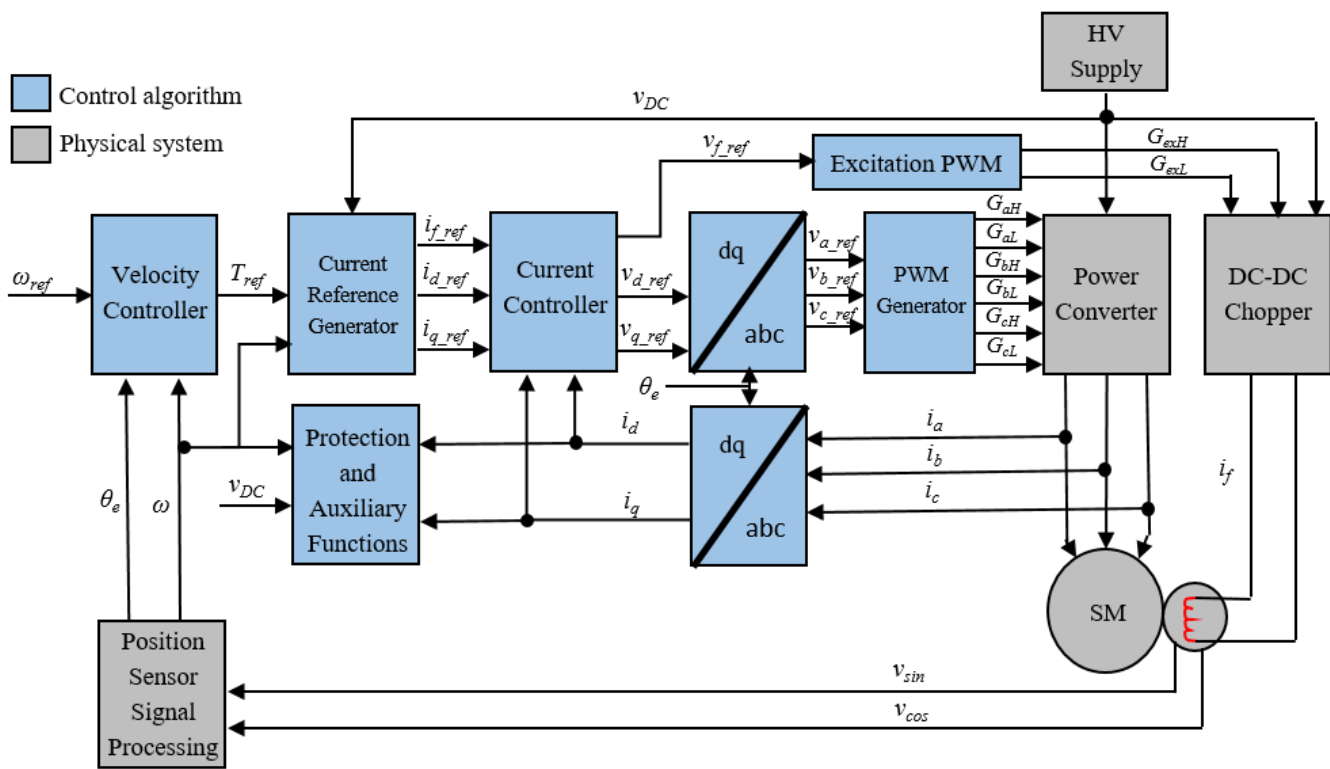
- The inner loop controller converts the current references into voltage references. An internal SM Current Controller generates the voltage references using a PI controller minimizing the current error, and the feedforward terms:

$$\begin{aligned}
 v_{d\_FF} &= -\omega_e L_{qi} i_q \\
 v_{q\_FF} &= \omega_e (L_d i_d + L_m f i_f) \\
 v_{f\_FF} &= 0
 \end{aligned}$$

where:

- $\omega_e$  is the rotor electrical angular velocity.
- $L_d$  and  $L_q$  are the  $d$ - and  $q$ -axis stator inductances.
- $L_{mf}$  is the mutual field armature inductance.
- $i_d$ ,  $i_q$ , and  $i_f$  are the stator  $d$ - $q$  and field excitation currents, respectively.
- The PWM Generator converts the reference stator voltages into gate pulses to be passed to a Power Converter that is powering the stator windings of the synchronous machine.
- The Excitation PWM Generator converts the reference field voltage into gate pulses to be passed to a DC-DC Chopper powering the SM field winding.

This diagram shows the overall architecture of the block.



In the diagram:

- $\omega$  and  $\omega_{ref}$  are the measured and reference angular velocities, respectively.
- $T_{ref}$  is the reference electromagnetic torque. If you configure the block for speed control, a Velocity Controller generates this reference torque.
- $i$  and  $v$  are stator currents and voltages. Subscripts  $d$ ,  $q$ , and  $f$ , represent the  $d$ -axis,  $q$ -axis, and field winding. Subscripts  $a$ ,  $b$ , and  $c$ , represent the three stator windings.
- $\theta_e$  is the rotor electrical angle.
- $G$  is a gate pulse, subscripts  $H$  and  $L$  represent high and low, and subscripts  $a$ ,  $b$ , and  $c$ , represent the three stator windings. Subscript  $ex$  represents the field excitation pulses.

You can choose to implement either velocity or torque control with the `Control mode` parameter. The block implements velocity control exactly as shown in the diagram. The block implements torque control by removing the Velocity Controller block and accepting the reference torque directly.

### Assumptions

The machine parameters are known.

### Limitations

The control structure is implemented with a single sample rate.

## Ports

### Input

#### Reference — System reference

scalar

System reference specified as torque reference in N\*m or velocity reference in rad/s, depending on the control mode selected.

Data Types: `single` | `double`

#### iabcSens — Measured phase currents

vector

Measured stator phase currents, in A.

Data Types: `single` | `double`

#### ifSens — Measured field current

scalar

Measured rotor field current, in A.

Data Types: `single` | `double`

#### wSens — Rotor speed

scalar

Measured mechanical angular velocity of rotor, in rad/s.

Data Types: `single` | `double`

#### thSens — Rotor angle

scalar

Measured mechanical angle of rotor, in rad.

Data Types: `single` | `double`

#### vdcSens — DC-link voltage

scalar

Measured DC-link voltage, in V.

Data Types: `single` | `double`

## Output

### **G — Stator converter gate pulses**

vector

Six pulse waveforms that determine switching behavior in the attached power converter.

Data Types: `single` | `double`

### **Gr — Excitation chopper gate pulses**

vector

Waveforms that determine switching behavior in the attached excitation chopper. The size of the waveform depends on the selected chopper type. To specify the chopper type, use the **Chopper type** parameter:

- First and fourth quadrant chopper — The output waveform has two pulses.
- Four-quadrant chopper — The output waveform has four pulses.

Data Types: `single` | `double`

### **Visualization — Visualization signals**

bus

Bus containing signals for visualization, including:

- Reference
- `wElectrical`
- `iabc`
- `theta`
- `Vdc`
- `PwmEnable`
- `TqRef`
- `TqLim`
- `idqRef`
- `idqf`
- `vdqRef`
- `modWave`
- `DCexcit`

Data Types: `single` | `double`

## Parameters

### General

#### **Control mode — Control mode strategy**

Velocity control (default) | Torque control

Specify either a torque control or velocity control strategy.

**Nominal dc-link voltage (V) – Rated DC voltage**

300 V (default) | positive number

Nominal DC-link voltage of the electrical source.

**Maximum power (W) – Maximum power**

60000 W (default) | positive number

Maximum machine power.

**Maximum torque (N\*m) – Maximum torque**

250 N\*m (default) | positive number

Maximum machine torque.

**Maximum field current (A) – Maximum field current**

25 A (default) | positive number

Maximum current in the field winding.

**Inverter dc-link voltage threshold (V) – DC-link voltage threshold**

100 V (default) | positive number

Voltage threshold to activate the power inverter.

**Number of rotor pole pairs – Pole pairs**

4 (default) | positive integer

Number of pole pairs on the rotor.

**Fundamental sample time (s) – Block sample time**

5e-6 (default) | positive number

Fundamental sample time for the block.

**Outer Loop****Control Type – Control type strategy**

PI control (default) | P control | P-PI control

Specify the type of the control strategy.

**Controller proportional gain – Proportional gain of PI controller**

1 (default) | positive number

Proportional gain of the PI controller.

**Controller integral gain – Integral gain of PI controller**

1 (default) | positive number

Integral gain of the PI controller.

**P controller proportional gain – Proportional gain of P controller**

1 (default) | positive number

Proportional gain of P controller.

**Integral anti-windup gain — Anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller.

**Current references — Current reference strategy**

Zero d-axis control (default) | Lookup-table based

Select the current reference strategy.

**Mechanical speed vector, wMechanical (rpm) — Rotor speed lookup vector**

[0, 3000] rpm (default) | positive monotonically increasing vector

Speed vector used in the lookup tables for determining current references.

**Torque reference vector, TqRef (N\*m) — Torque reference lookup vector**

[-100, 0, 100] N\*m (default) | positive monotonically increasing vector

Torque vector used in the lookup tables for determining current references.

**DC-link voltage vector, Vdc (V) — DC-link voltage lookup vector**

[300, 350] V (default) | positive monotonically increasing vector

DC-link voltage vector used in the lookup tables for determining current references.

**D-axis current reference matrix, id(wMechanical,TqRef,Vdc) (A) — Reference d-axis current values**

zeros(2,3,2) A (default) | real matrix

Direct-axis current reference lookup data.

**Q-axis current reference matrix, iq(wMechanical,TqRef,Vdc) (A) — Reference q-axis current values**

zeros(2,3,2) A (default) | real matrix

Quadrature-axis current reference lookup data.

**Field current reference matrix, if(wMechanical,TqRef,Vdc), (A) — Reference field current values**

zeros(2,3,2) A (default) | real matrix

Field current reference lookup data.

**Torque constant — Motor torque constant**

0.04 N\*m/A (default) | positive scalar

Synchronous machine torque constant. This value is numerically equivalent to the back EMF constant expressed in V/(rad/s).

**Inner Loop****D-axis current proportional gain — D-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for direct-axis current control.

**D-axis current integral gain — D-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for direct-axis current control.

**D-axis current anti-windup gain — D-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for direct-axis current control.

**Q-axis current proportional gain — Q-axis proportional gain**

1 (default) | positive number

Proportional gain of the PI controller used for quadrature-axis current control.

**Q-axis current integral gain — Q-axis integral gain**

100 (default) | positive number

Integrator gain of the PI controller used for quadrature-axis current control.

**Q-axis current anti-windup gain — Q-axis anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for quadrature-axis current control.

**Field current proportional gain — Field winding proportional gain**

10 (default) | positive number

Proportional gain of the PI controller used for field current control.

**Field current integral gain — Field winding integral gain**

1000 (default) | positive number

Integral gain of the PI controller used for field current control.

**Field current anti-windup gain — Field winding anti-windup gain**

1 (default) | positive number

Anti-windup gain of the PI controller used for field current control.

**Axis prioritization — Axis prioritization for voltage limiter**

q-axis (default) | d-axis | d-q equivalence

Prioritize or maintain ratio between  $d$ - and  $q$ -axis when the block limits voltage.**Enable zero cancellation — Feedforward zero cancellation**

off (default) | on

Enable or disable zero cancellation on the feedforward path.

**Enable pre-control voltage — pre-control voltage**

on (default) | off

Enable or disable pre-control voltage.

**Machine parameters — Machine parameterization**

Constant parameters (default) | Lookup table based parameters

Specify how to parameterize the machine.

- Constant parameters — Specify machine parameters that are constant throughout the simulation.
- Lookup table based parameters — Specify machine parameters as lookup tables that depend on current.

**Dependencies**

Enabled when the **Enable pre-control voltage** parameter is selected.

**D-axis inductance (H) — Feedforward d-axis inductance**

0.00024 (default) | positive scalar

Direct-axis inductance for feedforward pre-control.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Constant parameters.

**Q-axis inductance (H) — Feedforward q-axis inductance**

0.00029 (default) | positive scalar

Quadrature-axis inductance for feedforward pre-control.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Constant parameters.

**Mutual field armature inductance (H) — Mutual inductance between field and armature**

0.007 (default)

Mutual inductance between the field and armature windings.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Constant parameters.

**D-axis current vector,  $i_d$  (A) — D-axis current breakpoint vector**

[-200, 0, 200] A (default) | monotonically increasing vector

Direct-axis current vector used in the lookup tables for parameters determination.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**Q-axis current vector,  $i_q$  (A) — Q-axis current breakpoint vector**

[-200, 0, 200] A (default) | monotonically increasing vector

Quadrature-axis current vector used in the lookup tables for parameters determination.



**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**Field current vector,  $i_f$  (A) — Field winding current vector**

[0, 20] A (default) | monotonically increasing vector

Field current vector used in the lookup tables for parameters determination.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**Ld matrix,  $L_d(i_d, i_q)$  (H) — D-axis inductance lookup data**

0.0002 \* ones(3, 3) H (default) | positive matrix

$L_d$  matrix used as lookup table data.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**Lq matrix,  $L_q(i_d, i_q)$  (H) — Q-axis inductance lookup data**

0.0002 \* ones(3, 3) H (default) | positive matrix

$L_q$  matrix used as lookup table data.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**Lmf matrix,  $L_{mf}(i_d, i_q, i_f)$ , (H) — Field-armature mutual inductance lookup**

0.0007 \* ones(2, 2, 2) H (default) | positive matrix

$L_{mf}$  matrix used as lookup table data.

**Dependencies**

Enabled when the **Machine parameters** parameter is set to Lookup table based parameters.

**PWM****PWM method — Pulse width modulation method**

SVM: space vector modulation (default) | SPWM: sinusoidal PWM

Specify the waveform technique.

**Sampling mode — Wave-sampling method**

Natural (default) | Asymmetric | Symmetric

Specify whether the block samples the modulation waveform when the waves intersect or when the carrier wave is at one or both of its boundary conditions.

**Switching frequency (Hz) — Switching rate**

1000 Hz (default) | positive integer

Specify the rate at which you want the switches in the power converter to switch.

**Chopper type — DC-DC chopper type**

First and fourth quadrant chopper (default) | Four-quadrant chopper

Specify DC-DC chopper type.

**Switching frequency excitation (Hz) — Excitation system switching frequency**

1000 Hz (default) | positive integer

Specify PWM switching frequency for excitation system.

## **Version History**

**Introduced in R2018a**

## **References**

- [1] Märgner, M., and W. Hackmann. "Control challenges of an externally excited synchronous machine in an automotive traction drive application." *In Emobility-Electrical Power Train*. (2010): 1-6.
- [2] Carpiuc, S., C. Lazar, and D. Patrascu. "Optimal Torque Control of the Externally Excited Synchronous Machine." *Journal of Control Engineering and Applied Informatics*. 14, no 2 (2012): 80-88.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

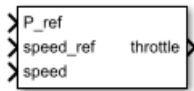
## **See Also**

SM Current Controller | SM Current Reference Generator

# SM Governor with Droop

Synchronous machine governor with droop

**Library:** Simscape / Electrical / Control / SM Control



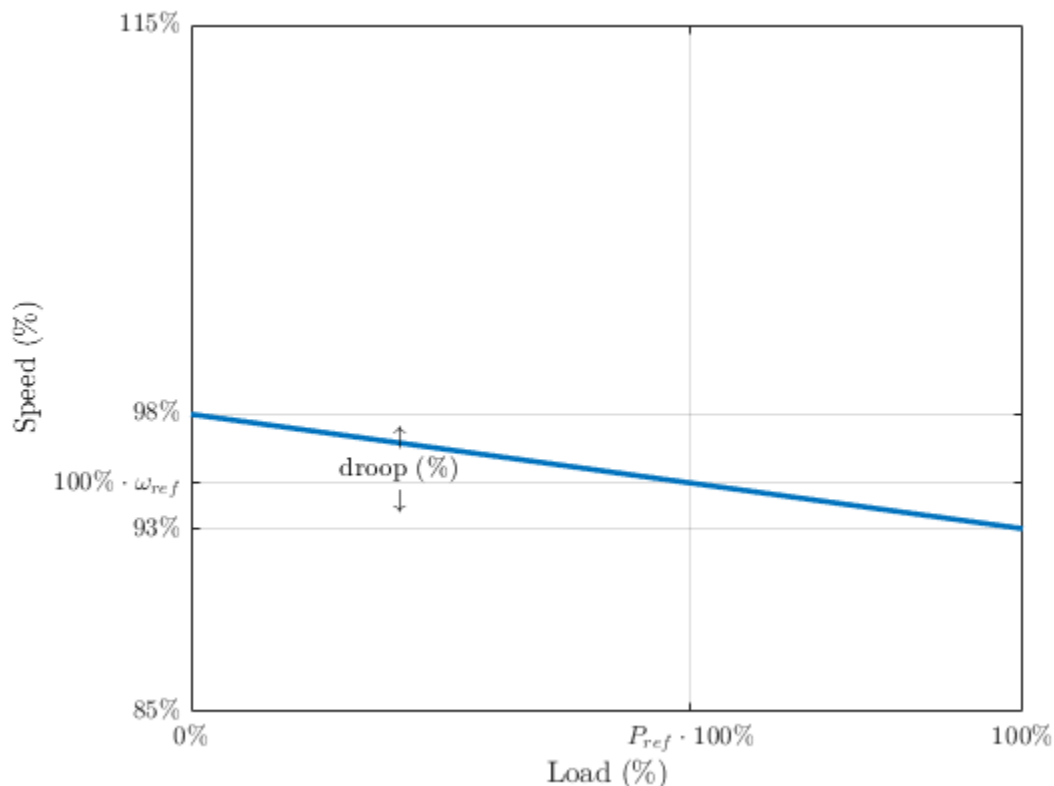
## Description

The SM Governor with Droop block implements a synchronous machine (SM) governor with a droop characteristic. Use this block to regulate or control the throttle input to a prime mover driving a synchronous generator.

The block uses the error between the measured and desired generator speeds to set the prime mover throttle position. For example, when the prime mover is rotating too slowly, the throttle is opened to increase the energy input to the generator and increase its speed.

## Operation

When multiple governor-driven generators are connected in parallel, droop characteristics ensure overall stability of the grid. Droop is defined as percent change in speed from no load to full load of the generator. This figure shows the speed-load relationship for a governor with 5% droop.



Here,

- $\omega_{ref}$  is the reference speed of the governor. Set this value as a per-unit quantity using the **speed\_ref** port.
- $P_{ref}$  is the reference load of the governor. Set this value as a per-unit quantity using the **P\_ref** port.
- **droop** is the droop percentage of the governor. Set this value as a percentage using the **Percentage droop, (%)** parameter.

The block calculates the reference throttle position, expressed as a per-unit quantity, as:

$$u_{throttle,ref} = P_{ref} - \frac{100}{\text{droop}}(\omega - \omega_{ref})$$

where  $\omega$  is the actual, per-unit generator speed.

The inertia of the valve introduces a delay between this reference throttle position and the actual throttle position, which is modeled as a first-order lag:

$$u_{throttle} = \left( \frac{1}{T_s s + 1} \right) u_{throttle,ref}$$

Here,  $T_s$  is the time constant. Set this value using the **Time constant of governor, (s)** parameter.

## Ports

### Input

#### **P\_ref — Desired per-unit generator load**

scalar

Reference generator load expressed as a per-unit quantity. The steady-state output of the governor running at its reference speed is equivalent to this value.

Data Types: `single` | `double`

#### **speed\_ref — Desired per-unit generator speed**

scalar

Reference generator speed expressed as a per-unit quantity.

Data Types: `single` | `double`

#### **speed — Actual per-unit generator speed**

scalar

Measured generator speed expressed as a per-unit quantity.

Data Types: `single` | `double`

### Output

#### **throttle — Throttle position**

scalar

Throttle position of the governor, expressed as a per-unit quantity.

Data Types: single | double

## Parameters

### **Percentage droop, (%) – Droop**

5 % (default)

Percent change in governor speed from 0% to 100% load. If multiple governor-driven generators are connected in parallel, those with lower droop percentages are more sensitive to load changes.

### **Time constant of governor, (s) – First-order lag time constant**

0.2 (default)

Time constant of the governor first-order lag representing the throttle inertia dynamics.

### **Initial throttle position, (pu) – Initial throttle output**

0.5 (default)

Initial state of the throttle.

## Version History

Introduced in R2018a

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

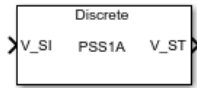
## See Also

Synchronous Machine Round Rotor (fundamental)

## SM PSS1A

Discrete-time or continuous-time single input PSS1A power system stabilizer

**Library:** Simscape / Electrical / Control / SM Control



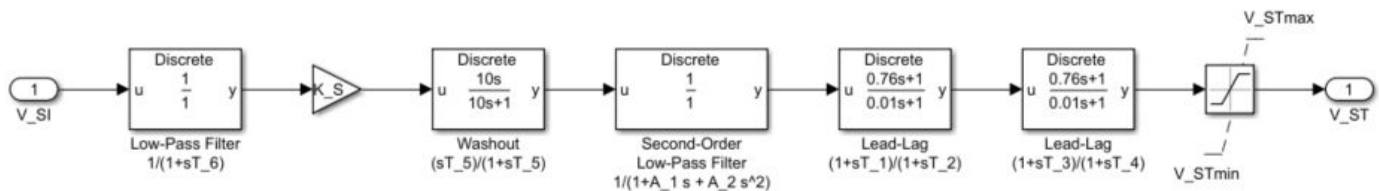
### Description

The SM PSS1A block implements a single-input PSS1A power system stabilizer (PSS) that maintains rotor angle stability in a synchronous machine (SM) in conformance with IEEE 421.5-2016<sup>[1]</sup>.

Typically, you use a PSS to enhance the damping of power system oscillations through excitation control.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the PSS1A power system stabilizer:



In the diagram:

- $V_{SI}$  is the power system stabilizer input. Commonly used inputs are speed, frequency, or power. For more information, see “ $V_{SI}$ ” on page 1-0 .
- The Low-Pass Filter (Discrete or Continuous) block can be used to model a transducer, with a time constant  $T_6$ .
- $K_s$  models the stabilizer gain.
- The Washout (Discrete or Continuous) block models a high-pass filter. Here,  $T_5$  is the time constant.
- The Second-Order Low-Pass Filter (Discrete or Continuous) block takes into account the low-frequency effects of the high-frequency torsional filters. Here,  $A_1$  and  $A_2$  are the stabilizer denominator constants for the second-order block.
- The two Lead-Lag (Discrete or Continuous) blocks models additional dynamics associated with the power system stabilizer, and represent two stages of the lead-lag compensation.

## Ports

### Input

#### **V\_SI — Speed, frequency, or power**

scalar

Per-unit speed, frequency of the terminal bus voltage, compensated frequency, or electrical power, specified as a scalar.

Data Types: single | double

### Output

#### **V\_ST — Stabilization speed, frequency, or power**

scalar

Automatic voltage regulator input stabilization signal, as limited by  $V_{ST\_min}$  and  $V_{ST\_max}$ , returned as a scalar.

Data Types: single | double

## Parameters

#### **Power system stabilizer (PSS) gain, K\_S (pu) — Power system stabilizer gain**

3.15 (default) | scalar

Power system stabilizer forward path gain.

#### **PSS denominator constant for second-order block, A\_1 — Filter constant 1**

0 (default) | scalar

Power system stabilizer signal conditioning filter coefficient 1.

#### **PSS denominator constant for second-order block, A\_2 — Filter constant 2**

0 (default) | scalar

Power system stabilizer signal conditioning filter coefficient 2.

#### **PSS numerator (lead) compensating time constant, T\_1 (s) — Lead time constant 1**

0.76 (default) | scalar

Lead time constant 1, associated with the first Lead-Lag block in the diagram.

#### **PSS denominator (lag) compensating time constant, T\_2 (s) — Lag time constant 2**

0.01 (default) | scalar

Lag time constant 2, associated with the first Lead-Lag block in the diagram.

#### **PSS numerator (lead) compensating time constant, T\_3 (s) — Lead time constant 3**

0.76 (default) | scalar

Lead time constant 3, associated with the second Lead-Lag block in the diagram.

#### **PSS denominator (lag) compensating time constant, T\_4 (s) — Lag time constant 4**

0.01 (default) | scalar

Lag time constant 4, associated with the second Lead-Lag block in the diagram.

**PSS washout time constant, T\_5 (s) – Washout time constant**

10 (default) | scalar

Washout time constant, associated with the Washout block in the diagram.

**PSS transducer time constant, T\_6 (s) – Transducer time constant**

0 (default) | scalar

Transducer time constant, associated with the Low-Pass Filter block in the diagram.

**Maximum PSS output, V\_STmax (pu) – Maximum stabilization**

0.09 (default) | scalar

Maximum power system stabilizer output to the automatic voltage regulator (AVR).

**Minimum PSS output, V\_STmin (pu) – Minimum stabilization**

-0.09 (default) | scalar

Minimum power system stabilizer output to the automatic voltage regulator (AVR).

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, set the sample time to -1. For discrete-time operation, set the sample time to a positive scalar. For continuous-time operation, set the sample time to 0.

## Version History

Introduced in R2018b

## References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

[2] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

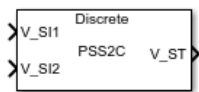
SM PSS2C | SM PSS7C | Synchronous Machine Measurement | Synchronous Machine Field Circuit | Synchronous Machine Model 1.0 | Synchronous Machine Model 2.1 | Synchronous Machine Round Rotor | Synchronous Machine Salient Pole



## SM PSS2C

Discrete-time or continuous-time single input PSS2C power system stabilizer

**Library:** Simscape / Electrical / Control / SM Control



### Description

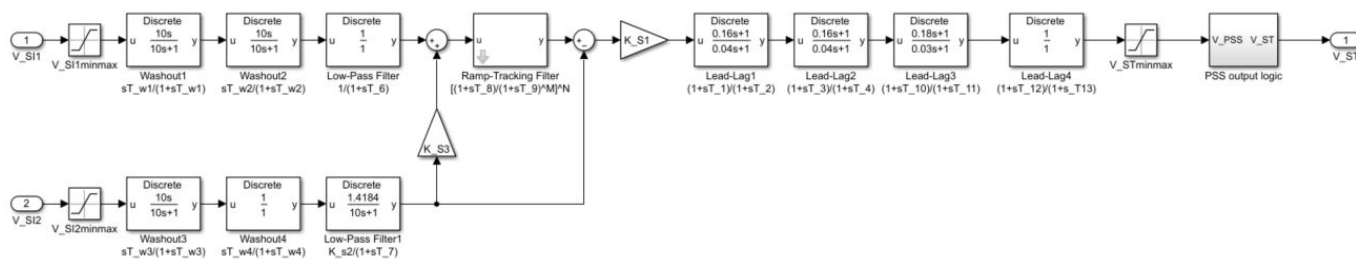
The SM PSS2C block implements a double-input PSS2C power system stabilizer (PSS) that maintains rotor angle stability in a synchronous machine (SM) in conformance with IEEE 421.5-2016<sup>[1]</sup>. Typically, you use a PSS to enhance the damping of power system oscillations through excitation control.

You can represent two different types of dual-input power system stabilizers with this same model:

- A stabilizer that uses electrical power and speed (or frequency) signals to calculate the integral of the accelerating power. This makes the calculated stabilizer signal insensitive to mechanical changes.
- A stabilizer that uses a combination of electrical power and either speed or frequency. To achieve the desired stabilizing signal shaping, the system uses the speed directly, without phase-lead compensation, and adds a signal that is proportional to the electrical power.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the PSS2C power system stabilizer:



In the diagram:

- $V_{SI1}$  and  $V_{SI2}$  are the two power system stabilizer inputs. Commonly used inputs are speed, frequency, or power.
- Two Washout (Discrete or Continuous) blocks are represented for each stabilizer input, with time constants  $T_{W1}$  to  $T_{W4}$ , along with a transducer, represented by a Low-Pass Filter (Discrete or Continuous), with time constants  $T_6$  and  $T_7$ .
- To allow a ramp-tracking filter characteristic, the Ramp Tracking Filter subsystem models a network of lead-lag and low-pass filter blocks in series.

- To provide phase compensation, a Lead-Lag (Discrete or Continuous) network models additional dynamics associated with the power system stabilizer, representing four stages of lead-lag compensation, with time constants  $T_1$  to  $T_4$  and  $T_{10}$  to  $T_{13}$ .
- The PSS output logic subsystem allows the representation of the PSS output logic associated with the generator active power output.  $P_{PSSon}$  and  $P_{PSSoff}$  are the threshold values used to define a hysteresis.

## Ports

### Input

#### **V\_SI1 — Speed, frequency, or power**

scalar

Per-unit speed, frequency of the terminal bus voltage, compensated frequency, or electrical power, specified as a scalar.

Data Types: single | double

#### **V\_SI2 — Electrical power**

scalar

Per-unit electrical power, specified as a scalar.

Data Types: single | double

### Output

#### **V\_ST — Stabilization speed, frequency, or power**

scalar

Automatic voltage regulator input stabilization signal, as limited by  $V_{ST_{min}}$  and  $V_{ST_{max}}$ , returned as a scalar.

Data Types: single | double

## Parameters

#### **PSS gain, K\_S1 (pu) — Power system stabilizer forward path gain**

5 (default) | scalar

Power system stabilizer forward path gain.

#### **PSS gain, K\_S2 (pu) — Power system stabilizer transducer gain**

1.4184 (default) | scalar

Power system stabilizer transducer gain.

#### **PSS gain, K\_S3 (pu) — Power system stabilizer gain**

1 (default) | scalar

Power system stabilizer gain.

#### **PSS transducer time constant, T\_6 (s) — Transducer time constant 6**

0 (default) | scalar

Transducer time constant 6.

**PSS transducer time constant,  $T_7$  (s) – Transducer time constant 7**

10 (default) | scalar

Transducer time constant 7.

**PSS washout time constant,  $T_{w1}$  (s) – Washout time constant 1**

10 (default) | scalar

Washout time constant 1.

**PSS washout time constant,  $T_{w2}$  (s) – Washout time constant 2**

10 (default) | scalar

Washout time constant 2.

**PSS washout time constant,  $T_{w3}$  (s) – Washout time constant 3**

10 (default) | scalar

Washout time constant 3.

**PSS washout time constant,  $T_{w4}$  (s) – Washout time constant 4**

0 (default) | scalar

Washout time constant 4.

**PSS transducer time constant,  $T_8$  (s) – Transducer time constant 8**

0.5 (default) | scalar

Transducer time constant 8.

**PSS washout time constant,  $T_9$  (s) – Washout time constant 9**

0.1 (default) | scalar

Washout time constant 9.

**Denominator exponent for ramp-track filter,  $M$  – Exponent of the denominator of the ramp-track filter**

5 (default) | scalar

Denominator exponent for ramp-track filter.

**Overall exponent for ramp-track filter,  $N$  – Overall exponent of ramp-track filter**

1 (default) | scalar

Overall exponent for ramp-track filter.

**PSS numerator (lead) compensating time constant (first block),  $T_1$  (s) – Lead time constant 1**

0.16 (default) | scalar

Lead time constant 1, associated with the block labeled Lead-Lag1 in the diagram.

**PSS denominator (lag) compensating time constant (first block),  $T_2$  (s) – Lag time constant 2**

0.04 (default) | scalar

Lag time constant 2, associated with the block labeled Lead-Lag1 in the diagram.

**PSS numerator (lead) compensating time constant (second block),  $T_3$  (s) – Lead time constant 3**

0.16 (default) | scalar

Lead time constant 3, associated with the block labeled Lead-Lag2 in the diagram.

**PSS denominator (lag) compensating time constant (second block),  $T_4$  (s) – Lag time constant 4**

0.04 (default) | scalar

Lag time constant 4, associated with the block labeled Lead-Lag2 in the diagram.

**PSS denominator (lead) compensating time constant (third block),  $T_{10}$  (s) – Lag time constant 10**

0.18 (default) | scalar

Lag time constant 10, associated with the block labeled Lead-Lag3 in the diagram.

**PSS denominator (lag) compensating time constant (third block),  $T_{11}$  (s) – Lag time constant 11**

0.03 (default) | scalar

Lag time constant 11, associated with the block labeled Lead-Lag3 in the diagram.

**PSS denominator (lead) compensating time constant (fourth block),  $T_{12}$  (s) – Lag time constant 12**

0 (default) | scalar

Lag time constant 12, associated with the block labeled Lead-Lag4 in the diagram.

**PSS denominator (lag) compensating time constant (fourth block),  $T_{13}$  (s) – Lag time constant 13**

0 (default) | scalar

Lag time constant 13, associated with the block labeled Lead-Lag4 in the diagram.

**Maximum PSS output,  $V_{STmax}$  (pu) – Maximum stabilization**

0.1 (default) | scalar

Maximum power system stabilizer output to the automatic voltage regulator.

**Minimum PSS output,  $V_{STmin}$  (pu) – Minimum stabilization**

-0.1 (default) | scalar

Minimum power system stabilizer output to the automatic voltage regulator.

**Input signal #1 maximum limit,  $V_{S1max}$  (pu) – Maximum limit for first input**

2 (default) | scalar

Maximum signal value for the first stabilizer input.

**Input signal #1 minimum limit, V\_S1min (pu) – Minimum limit for first input**  
-2 (default) | scalar

Minimum signal value for the first stabilizer input.

**Input signal #2 maximum limit, V\_S2max (pu) – Maximum limit for second input**  
2 (default) | scalar

Maximum signal value for the second stabilizer input.

**Input signal #2 minimum limit, V\_S2min (pu) – Minimum limit for second input**  
-2 (default) | scalar

Minimum signal value for the second stabilizer input.

**Generator MW threshold for PSS activation, P\_PSSon (pu) – Generator threshold for power system stabilizer activation**  
0 (default) | scalar

Generator threshold for power system stabilizer activation.

**Generator MW threshold for PSS de-activation, P\_PSSoff (pu) – Generator threshold for power system stabilizer deactivation**  
0 (default) | scalar

Generator threshold for power system stabilizer deactivation.

**Sample time (-1 for inherited) – Block sample time**  
-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, set the sample time to -1. For discrete-time operation, set the sample time to a positive scalar. For continuous-time operation, set the sample time to 0.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

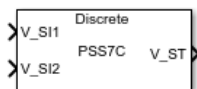
**See Also**

SM PSS1A | SM PSS7C

## SM PSS7C

Discrete-time or continuous-time single input PSS7C power system stabilizer

**Library:** Simscape / Electrical / Control / SM Control



### Description

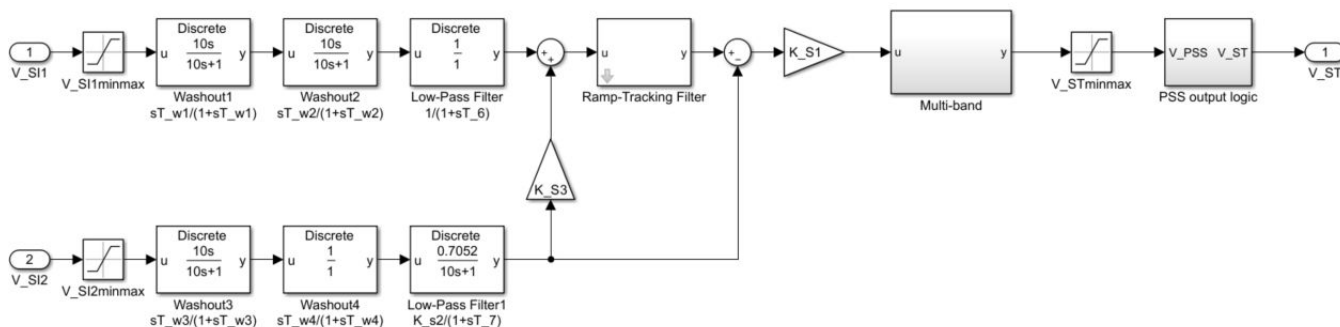
The SM PSS7C block implements a double-input PSS7C power system stabilizer (PSS) that maintains rotor angle stability in a synchronous machine (SM) in conformance with IEEE 421.5-2016<sup>[1]</sup>. Typically, you use a PSS to enhance the damping of power system oscillations through excitation control.

You can represent two different types of dual-input power system stabilizers with this same model:

- A stabilizer that uses electrical power and speed (or frequency) signals to calculate the integral of the accelerating power. This makes the calculated stabilizer signal insensitive to mechanical changes.
- A stabilizer that uses a combination of electrical power and either speed or frequency. To achieve the desired stabilizing signal shaping, the system uses the speed directly, without phase-lead compensation, and adds a signal that is proportional to the electrical power.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

This diagram illustrates the overall structure of the PSS7C power system stabilizer:



In the diagram:

- $V_{SI1}$  and  $V_{SI2}$  are the two power system stabilizer inputs. Commonly used inputs are speed, frequency, or power.
- Two Washout (Discrete or Continuous) blocks are represented for each stabilizer input, with time constants  $T_{W1}$  to  $T_{W4}$ , along with a transducer, represented by a Low-Pass Filter (Discrete or Continuous), with time constants  $T_6$  and  $T_7$ .

- To allow a ramp-tracking filter characteristic, the Ramp Tracking Filter subsystem models a network of lead-lag and low-pass filter blocks in series.
- To provide phase compensation, the Multi-band subsystem implements a canonical state equation, with time constants  $T_{i1}$  to  $T_{i4}$  and gains  $K_0$  to  $K_4$ .
- The PSS output logic subsystem allows the representation of the PSS output logic associated with the generator active power output.  $P_{PSSon}$  and  $P_{PSSoff}$  are the threshold values used to define a hysteresis.

## Ports

### Input

#### **V\_SI1 — Speed, frequency, or power**

scalar

Per-unit speed, frequency of the terminal bus voltage, compensated frequency, or electrical power, specified as a scalar.

Data Types: single | double

#### **V\_SI2 — Electrical power**

scalar

Per-unit electrical power, specified as a scalar.

Data Types: single | double

### Output

#### **V\_ST — Stabilization speed, frequency, or power**

scalar

Automatic voltage regulator input stabilization signal, as limited by  $V_{ST\_min}$  and  $V_{ST\_max}$ , returned as a scalar.

Data Types: single | double

## Parameters

#### **Power system stabilizer main gain, K\_S1 (pu) — Power system stabilizer forward path gain**

50 (default) | scalar

Power system stabilizer forward path gain.

#### **Power system stabilizer gain, K\_S2 (pu) — Power system stabilizer transducer gain**

0.7052 (default) | scalar

Power system stabilizer transducer gain.

#### **Power system stabilizer gain, K\_S3 (pu) — Power system stabilizer gain**

1 (default) | scalar

Power system stabilizer gain.



**PSS transducer time constant,  $T_6$  (s) – Transducer time constant 6**

0 (default) | scalar

Transducer time constant 6, associated with the block labeled Low-Pass Filter in the diagram.

**PSS transducer time constant,  $T_7$  (s) – Transducer time constant 7**

10 (default) | scalar

Transducer time constant 7, associated with the block labeled Low-Pass Filter1 in the diagram.

**PSS washout time constant,  $T_{w1}$  (s) – Washout time constant 1**

10 (default) | scalar

Washout time constant 1, associated with the block labeled Washout1 in the diagram.

**PSS washout time constant,  $T_{w2}$  (s) – Washout time constant 2**

10 (default) | scalar

Washout time constant 2, associated with the block labeled Washout2 in the diagram.

**PSS washout time constant,  $T_{w3}$  (s) – Washout time constant 3**

10 (default) | scalar

Washout time constant 3, associated with the block labeled Washout3 in the diagram.

**PSS washout time constant,  $T_{w4}$  (s) – Washout time constant 4**

0 (default) | scalar

Washout time constant 4, associated with the block labeled Washout4 in the diagram.

**PSS transducer time constant,  $T_8$  (s) – Transducer time constant 8**

0.5 (default) | scalar

Transducer time constant 8.

**PSS washout time constant,  $T_9$  (s) – Washout time constant 9**

0.1 (default) | scalar

Washout time constant 9.

**Denominator exponent for ramp-track filter,  $M$  – Exponent of denominator of the ramp-track filter**

5 (default) | scalar

Denominator exponent for the ramp-track filter.

**Overall exponent for ramp-track filter,  $N$  – Overall exponent of ramp-track filter**

1 (default) | scalar

Overall exponent for the ramp-track filter.

**PSS canonical gain 0,  $K_0$  (pu) – Power system stabilizer canonical gain 0**

0.399 (default) | scalar

Power system stabilizer canonical gain 0 for the Multi-band subsystem.

**PSS canonical gain 1,  $K_1$  (pu) — Power system stabilizer canonical gain 1**  
1.8462 (default) | scalar

Power system stabilizer canonical gain 1 for the Multi-band subsystem.

**PSS canonical gain 2,  $K_2$  (pu) — Power system stabilizer canonical gain 2**  
0.4231 (default) | scalar

Power system stabilizer canonical gain 2 for the Multi-band subsystem.

**PSS canonical gain 3,  $K_3$  (pu) — Power system stabilizer canonical gain 3**  
0.2104 (default) | scalar

Power system stabilizer canonical gain 3 for the Multi-band subsystem.

**PSS canonical gain 4,  $K_4$  (pu) — Power system stabilizer canonical gain 4**  
0 (default) | scalar

Power system stabilizer canonical gain 4 for the Multi-band subsystem.

**PSS third block gain,  $K_{i3}$  (pu) — Power system stabilizer third block gain**  
1 (default) | scalar

Power system stabilizer third block gain for the Multi-band subsystem.

**PSS fourth block gain,  $K_{i4}$  (pu) — Power system stabilizer fourth block gain**  
0 (default) | scalar

Power system stabilizer fourth block gain for the Multi-band subsystem.

**PSS time constant (first block),  $T_{i1}$  (s) — Power system stabilizer time constant for the first block**  
0.03 (default) | scalar

Power system stabilizer time constant for the first block inside the Multi-band subsystem.

**PSS time constant (second block),  $T_{i2}$  (s) — Power system stabilizer time constant for the second block**  
0.0293 (default) | scalar

Power system stabilizer time constant for the second block inside the Multi-band subsystem.

**PSS time constant (third block),  $T_{i3}$  (s) — Power system stabilizer time constant for the third block**  
0.2804 (default) | scalar

Power system stabilizer time constant for the third block inside the Multi-band subsystem.

**PSS time constant (fourth block),  $T_{i4}$  (s) — Power system stabilizer time constant for the fourth block**  
1 (default) | scalar

Power system stabilizer time constant for the fourth block inside the Multi-band subsystem.

**Maximum PSS output,  $V_{STmax}$  (pu) — Maximum stabilization**  
0.05 (default) | scalar

Maximum power system stabilizer output to the automatic voltage regulator (AVR).

**Minimum PSS output, V\_STmin (pu) – Minimum stabilization**

-0.05 (default) | scalar

Minimum power system stabilizer output to the automatic voltage regulator (AVR).

**Input signal #1 maximum limit, V\_S1max (pu) – Maximum limit for the first input**

2 (default) | scalar

Maximum signal value for the first stabilizer input.

**Input signal #1 minimum limit, V\_S1min (pu) – Minimum limit for the first input**

-2 (default) | scalar

Minimum signal value for the first stabilizer input.

**Input signal #2 maximum limit, V\_S2max (pu) – Maximum limit for the second input**

2 (default) | scalar

Maximum signal value for the second stabilizer input.

**Input signal #2 minimum limit, V\_S2min (pu) – Minimum limit for the second input**

-2 (default) | scalar

Minimum signal value for the second stabilizer input.

**Generator MW threshold for PSS activation, P\_PSSon (pu) – Generator threshold for power system stabilizer activation**

0 (default) | scalar

Generator threshold for power system stabilizer activation.

**Generator MW threshold for PSS de-activation, P\_PSSoff (pu) – Generator threshold for power system stabilizer deactivation**

0 (default) | scalar

Generator threshold for power system stabilizer deactivation.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, set the sample time to -1. For discrete-time operation, set the sample time to a positive scalar. For continuous-time operation, set the sample time to 0.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

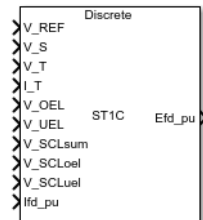
## See Also

SM PSS1A | SM PSS2C

## SM ST1C

Discrete-time or continuous-time synchronous machine ST1C static excitation system with an automatic voltage regulator

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM ST1C block implements a synchronous-machine-type ST1C static excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

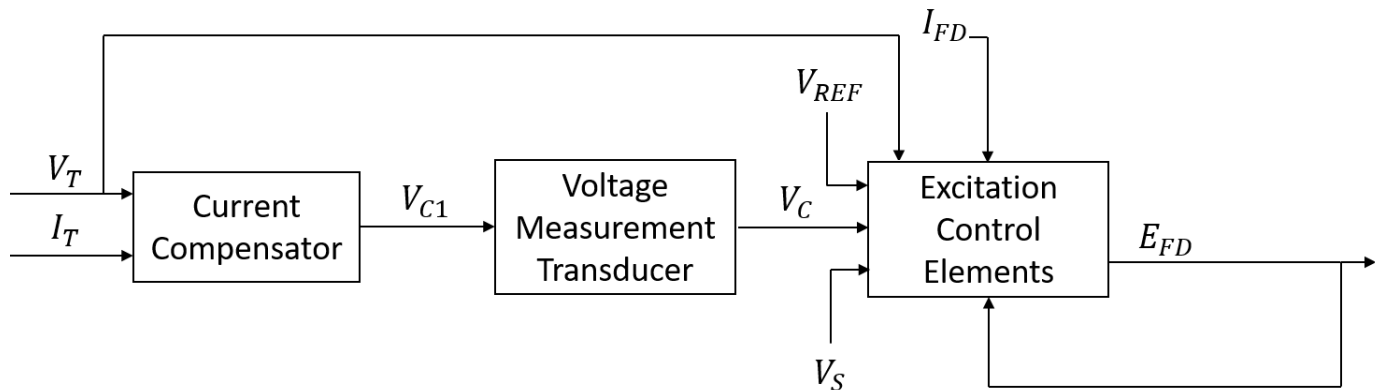
Use this block to model the control and regulation of the field voltage of a synchronous machine.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM ST1C block comprises three major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the field voltage.

This diagram shows the overall structure of the ST1C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

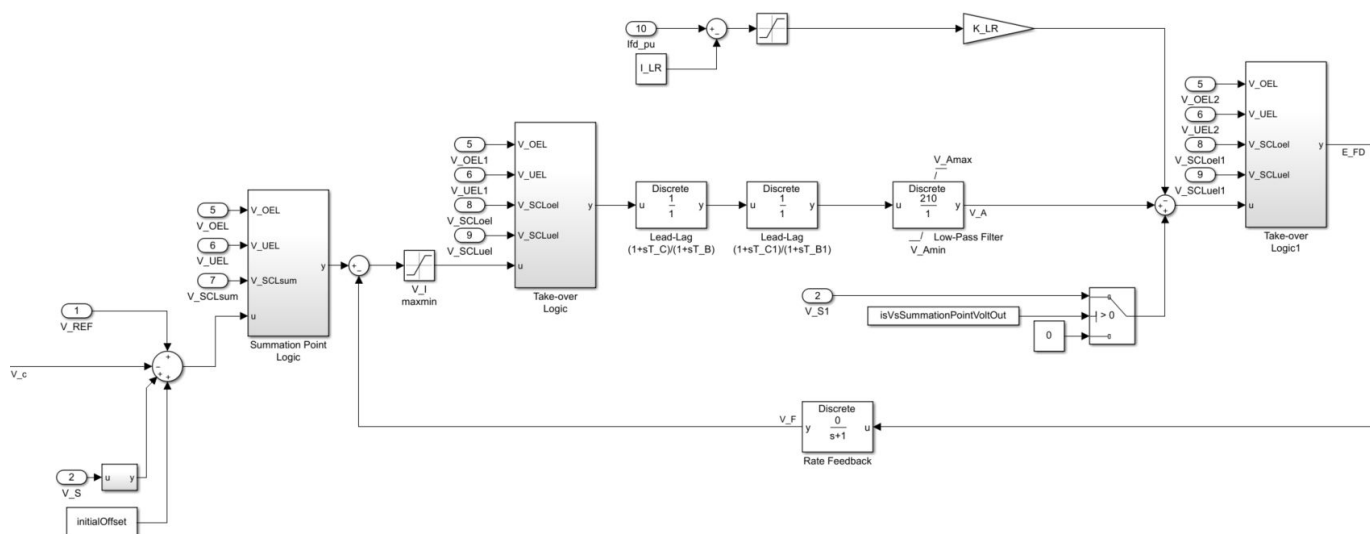
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for the Low-Pass Filter block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1945.
- There are two Take-over Logic subsystems. The subsystems model the take-over point input location for the OEL, UEL, SCL and PSS voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1945.
- The two Lead-Lag blocks model additional dynamics associated with the voltage regulator. The former represents a transient gain reduction, where  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. The latter allows the possibility of representing a transient gain increase, where  $T_{C1}$  is the lead time constant and  $T_{B1}$  is the lag time constant. Refer to the documentation for the Lead-Lag block for the discrete and continuous implementations.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.
- The Filtered Derivative block models the rate feedback path for the stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constants of this system, respectively. Refer to the documentation for the Filtered Derivative block for the discrete and continuous implementations.
- Due to very high forcing capability, the model employs a field current limiter to protect the generator rotor and exciter. The initial threshold and gain are defined by  $I_{LR}$  and  $K_{LR}$ , respectively. If you use an explicit OEL model, disable this field current limiter by setting the gain,  $K_{LR}$ , to 0.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.

- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

### Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to **Unused**.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to **Summation point**.



- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations** to Summation point.

Data Types: single | double

### **V\_SCLoel — Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter that prevents field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

### **V\_SCLuel — Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter that prevents field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**I<sub>fd</sub>\_pu — Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

**Output****E<sub>fd</sub>\_pu — Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

**Parameters****General****Initial field voltage, E<sub>fd0</sub> (pu) — Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control****Resistive component of load compensation, R<sub>C</sub> (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X<sub>C</sub> (pu)** to 0 to disable current compensation.

**Reactance component of load compensation,  $X_C$  (pu) – Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation,  $R_C$  (pu)** to 0 to disable current compensation.

**Regulator input filter time constant,  $T_R$  (s) – Regulator time constant**

0.02 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control****Regulator output gain,  $K_A$  (pu) – Regulator gain**

210 (default) | positive number

Gain associated with the voltage regulator.

**Regulator output time constant,  $T_A$  (s) – Regulator time constant**

0 (default) | positive number

Major time constant of the voltage regulator.

**Regulator denominator (lag) time constant,  $T_B$  (s) – Regulator lag time constant for first lead-lag block**

1 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant,  $T_C$  (s) – Regulator lead time constant for first lead-lag block**

1 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Regulator denominator (lag) time constant,  $T_{B1}$  (s) – Regulator lag time constant for second lead-lag block**

0 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant,  $T_{C1}$  (s) – Regulator lead time constant for second lead-lag block**

0 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Rate feedback gain,  $K_F$  (pu) – Rate feedback gain**

0 (default) | positive number

Rate feedback block gain.

**Rate feedback time constant,  $T_F$  (s) – Rate feedback time constant**

1 (default) | positive number

Time constant for the rate feedback block.

**Maximum regulator output,  $V_{Amax}$  (pu) – Upper limit of the regulator output**

6.43 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output,  $V_{Amin}$  (pu) – Lower limit of the regulator output**

-6 (default) | real number

Minimum per-unit output voltage of the regulator.

**Maximum voltage error (regulator input),  $V_{Imax}$  (pu) – Voltage upper limit**

99 (default) | real number

Maximum per-unit voltage given as input to the regulator.

**Minimum voltage error (regulator input),  $V_{Imin}$  (pu) – Voltage lower limit**

-99 (default) | real number

Minimum per-unit voltage given as input to the regulator.

**Alternate PSS input locations ( $V_S$ ) – PSS input location**

Summation point at voltage error (default) | Summation point at voltage regulator output

Power system stabilizer input location.

**Alternate OEL input locations ( $V_{OEL}$ ) – OEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Overexcitation limiter input location:

- If you select Summation point at voltage error,  $V_{OEL}$  is an input of the Summation Point Logic subsystem.
- If you select Take-over at voltage error,  $V_{OEL}$  is an input of the Take-over Logic subsystem.
- If you select Take-over at voltage regulator output,  $V_{OEL}$  is an input of the Take-over Logic 1 subsystem.

**Alternate UEL input locations ( $V_{UEL}$ ) – UEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Underexcitation limiter input location:

- If you select Summation point at voltage error,  $V_{UEL}$  is an input of the Summation Point Logic subsystem.
- If you select Take-over at voltage error,  $V_{UEL}$  is an input of the Take-over Logic subsystem.

- If you select Take-over at voltage regulator output, **V\_UEL** is an input of the Take-over Logic 1 subsystem.

#### **Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage error | Take-over at voltage regulator output

Stator current limiter input location:

- If you select Summation point, use the **V\_SCLsum** input port.
- If you select any of the Take-over options, use the **V\_SCLoel** and **V\_SCLuel** input ports.

#### **Exciter**

##### **Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0.038 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

##### **Exciter output current limiter gain, K\_LR (pu) – Exciter output current limiter gain**

4.54 (default) | real number

Maximum per-unit exciter current limiter gain.

##### **Exciter output current limit reference, I\_LR (pu) – Exciter output current limit reference**

4.4 (default) | real number

Per-unit exciter current limit reference.

##### **Maximum exciter output, V\_Rmax (pu) – Exciter upper limit**

6.43 (default) | real number

Maximum per-unit exciter voltage output.

##### **Minimum exciter output, V\_Rmin (pu) – Exciter lower limit**

-6 (default) | real number

Minimum per-unit exciter voltage output.

## **Version History**

Introduced in R2020a

## **References**

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

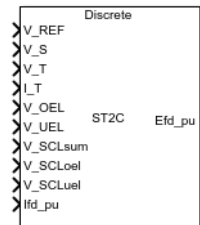
### **See Also**

SM ST2C | SM ST3C | SM ST4C

## SM ST2C

Discrete-time or continuous-time synchronous machine ST2C static excitation system with an automatic voltage regulator

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM ST2C block implements a synchronous-machine-type ST2C static excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

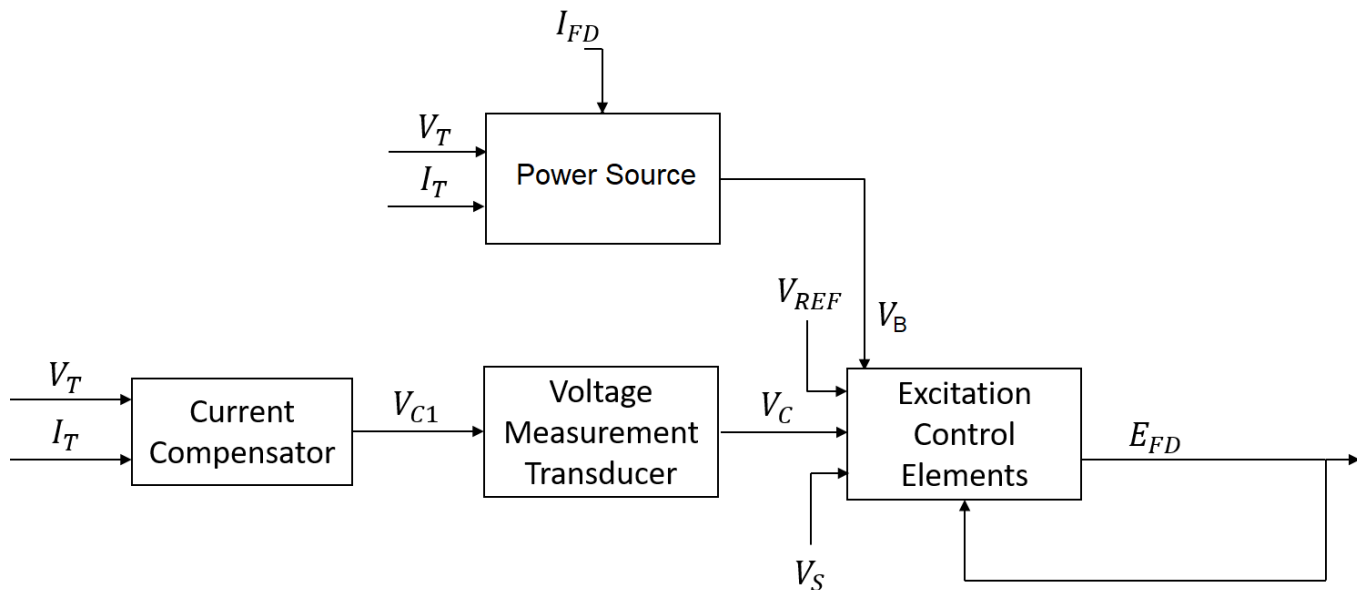
Use this block to model the control and regulation of the field voltage of a synchronous machine.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM ST2C block comprises four major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the field voltage.
- The Power Source models the power source for the controlled rectifier when it is independent from the terminal voltage.

This diagram shows the overall structure of the ST2C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $V_B$  is the exciter field voltage.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

where:

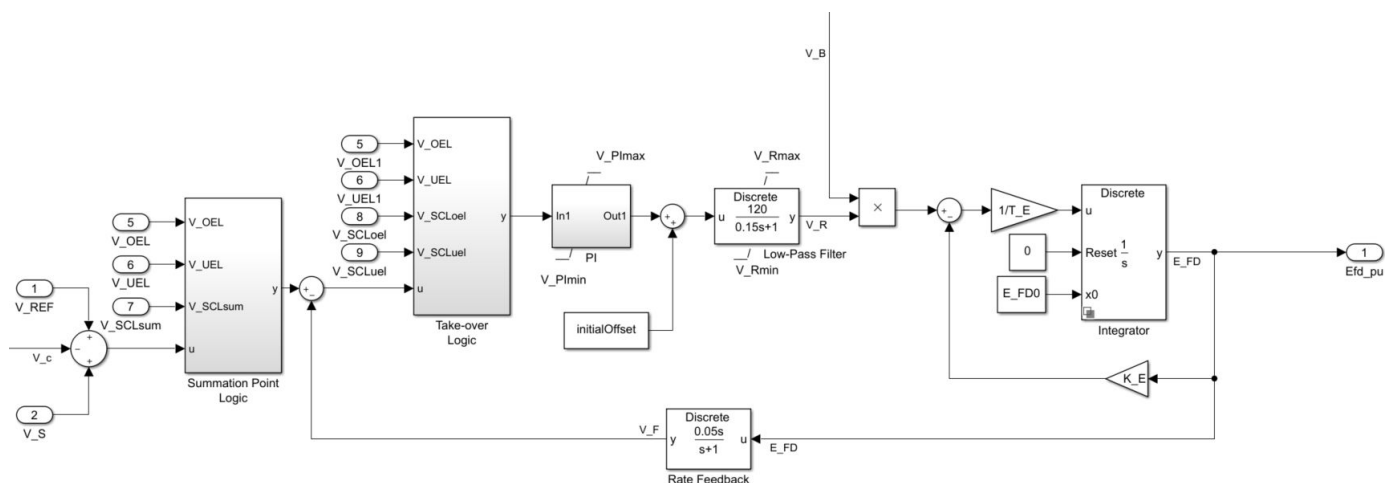
- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for the Low-Pass Filter block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:





In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1955.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1955.
- The PI subsystem models a PI controller that functions as a control structure for the automatic voltage regulator and allows the representation of an equipment retrofit with a modern digital controller. The minimum and maximum anti-windup saturation limits for the block are  $V_{PImin}$  and  $V_{PImax}$ , respectively.
- The Low-Pass Filter block models the major dynamics of the voltage regulator. Here,  $K_A$  is the regulator gain and  $T_A$  is the major time constant of the regulator. The minimum and maximum anti-windup saturation limits for the block are  $V_{Rmin}$  and  $V_{Rmax}$ , respectively.
- The Filtered Derivative block models the rate feedback path for stabilization of the excitation system. Here,  $K_F$  and  $T_F$  are the gain and time constant of this system, respectively. Refer to the documentation for the Filtered Derivative block for the exact discrete and continuous implementations.
- The Integrator block controls the exciter output through controlled saturation. The parameter  $E_{FDmax}$  represents the limit on the exciter voltage. The time constant  $T_E$  is associated with the inductance of the control windings.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

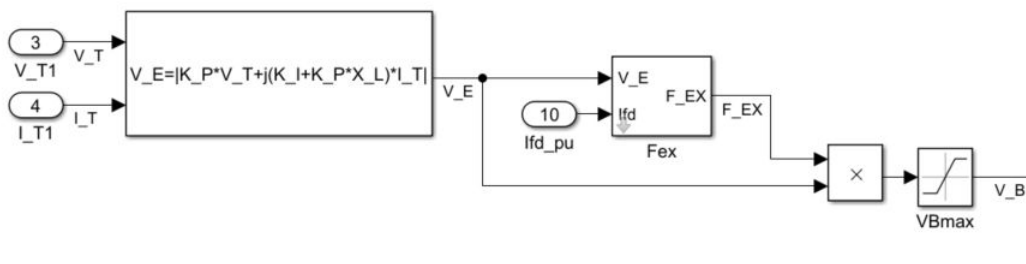
- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

## Power Source

Some static systems utilize both current and voltage sources to generate the power source.

This diagram shows a model of the exciter power source utilizing a phasor combination of the terminal voltage,  $V_T$ , and terminal current,  $I_T$ :



## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_SCLoel — Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter that prevents field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_SCLuel – Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter that prevents field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Ifd\_pu – Measured field current**

scalar

Measured per-unit field current of the synchronous machine.

Data Types: single | double

**Output****Efd\_pu – Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

**Parameters****General****Initial field voltage, Efd0 (pu) – Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

**Initial terminal voltage, Vt0 (pu) – Initial terminal voltage**

1 (default) | real number

Initial per-unit terminal voltage.

**Initial terminal current, It0 (pu) – Initial terminal current**

1 (default) | real number

Initial per-unit terminal current.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control**

**Resistive component of load compensation, R\_C (pu) – Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) – Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) – Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control**

**Voltage regulator proportional gain, K\_PR (pu) – Regulator proportional gain**

1 (default) | positive number

Proportional gain associated with the voltage regulator PI control block.

**Voltage regulator integral gain, K\_IR (pu/s) – Regulator integral gain**

0 (default) | positive number

Integral gain associated with the voltage regulator PI control block.

**Maximum voltage regulator output, V\_PImax (pu) – Upper limit of regulator output for PI control**

99 (default) | real number

Maximum per-unit output voltage of the regulator PI control block.

**Minimum voltage regulator output, V\_PImin (pu) – Lower limit of regulator output for PI control**

-99 (default) | real number

Minimum per-unit output voltage of the regulator PI control block.

**Voltage regulator gain, K\_A (pu) – Regulator gain**

120 (default) | positive number

Gain associated with the voltage regulator.

**Voltage regulator time constant, T\_A (s) – Regulator time constant**

0.15 (default) | positive number

Major time constant of the voltage regulator.

**Rate feedback gain, K\_F (pu) – Rate feedback gain**

0.05 (default) | positive number

Rate feedback block gain.

**Rate feedback time constant, T\_F (s) – Rate feedback time constant**

1 (default) | positive number

Rate feedback block time constant.

**Maximum regulator output, V\_Rmax (pu) – Upper limit of the regulator output**

1 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V\_Rmin (pu) – Lower limit of the regulator output**

0 (default) | real number

Minimum per-unit output voltage of the regulator.

**Alternate OEL input locations (V\_OEL) – OEL input location**

Unused (default) | Summation point | Take-over

Overexcitation limiter input location.

**Alternate UEL input locations (V\_UEL) – UEL input location**

Unused (default) | Summation point | Take-over

Underexcitation limiter input location.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point | Take-over

Stator current limiter input location:

- If you select `Summation point`, use the `V_SCLsum` input port.
- If you select `Take-over`, use the `V_SCLoel` and `V_SCLuel` input ports.

## Exciter

### **Rectifier loading factor proportional to commutating reactance, $K_C$ (pu) – Rectifier loading factor**

0.1 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

### **Exciter field proportional constant, $K_E$ (pu) – Exciter field gain**

1 (default) | positive number

Proportional constant for the exciter field.

### **Exciter field time constant, $T_E$ (s) – Exciter field time constant**

0.5 (default) | positive number

Time constant for exciter field.

### **Maximum generator field voltage, $E_{FDmax}$ (pu) – Upper limit of generator field voltage**

4.4 (default) | positive number

Maximum per-unit generator field voltage value.

### **Potential circuit (voltage) gain coefficient, $K_P$ (pu) – Potential circuit voltage gain coefficient**

4.88 (default) | positive number

Per-unit voltage gain coefficient in the power source circuit.

### **Compound circuit (current) gain coefficient, $K_I$ (pu) – Potential circuit current gain coefficient**

0 (default) | positive number

Per-unit current gain coefficient in the power source circuit.

### **Reactance associated with potential source, $X_L$ (pu) – Potential source reactance**

0 (default) | positive number

Per-unit reactance in the power source circuit.

### **Potential circuit phase angle (degrees) – Potential circuit phase angle**

0 (default) | positive number

Phase angle of the power source circuit, in degrees.

### **Maximum available exciter voltage, $V_{Bmax}$ (pu) – Upper limit of exciter voltage**

5.2 (default) | positive number

Maximum per-unit available exciter voltage value.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*.  
IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

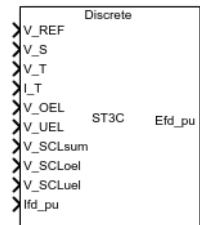
SM ST1C | SM ST3C | SM ST4C



## SM ST3C

Discrete-time or continuous-time synchronous machine ST3C static excitation system with an automatic voltage regulator

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM ST3C block implements a synchronous-machine-type ST3C static excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

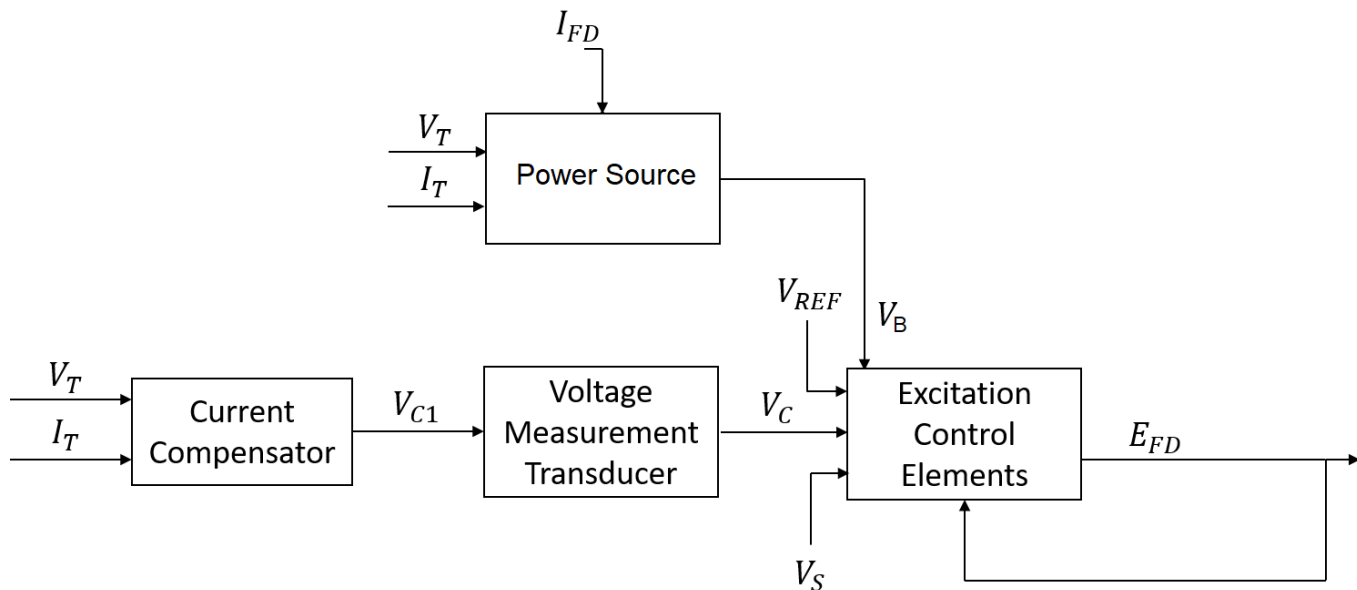
Use this block to model the control and regulation of the field voltage of a synchronous machine.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM ST3C block comprises four major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the field voltage.
- The Power Source models the power source for the controlled rectifier when it is independent from the terminal voltage.

This diagram shows the overall structure of the ST3C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $V_B$  is the exciter field voltage.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

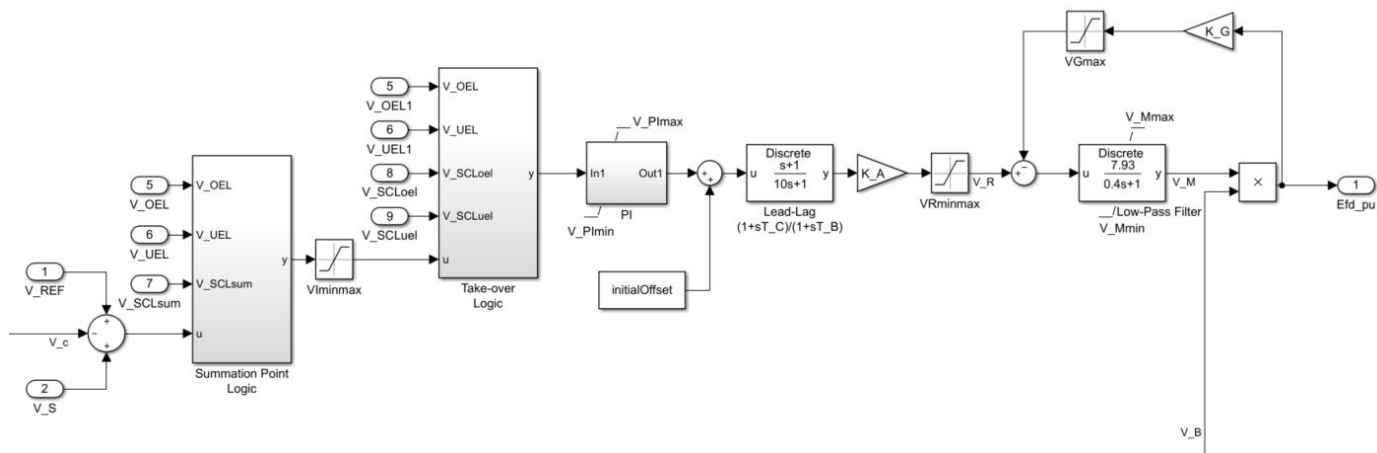
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for the Low-Pass Filter block for the discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1965.
- The Take-over Logic subsystem models the take-over point input location for the OEL, UEL, and SCL voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1965.
- The PI subsystem models a PI controller that functions as a control structure for the automatic voltage regulator and allows the representation of an equipment retrofit with a modern digital controller. The minimum and maximum anti-windup saturation limits for the block are  $V_{PImin}$  and  $V_{PImax}$ , respectively.
- The Lead-Lag block models additional dynamics associated with the voltage regulator and represents the system stabilizer. Here,  $T_C$  is the lead time constant and  $T_B$  is the lag time constant. Refer to the documentation for this block for the exact discrete and continuous implementations.
- An inner field voltage control loop is utilized to linearize the exciter control characteristic and it is composed of the gains  $K_M$  and  $K_G$  and the time constant  $T_M$ . The minimum and maximum anti-windup saturation limits for the block are  $V_{Mmin}$  and  $V_{Mmax}$ , respectively.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

- The summation point as part of the automatic voltage regulator (AVR) feedback loop

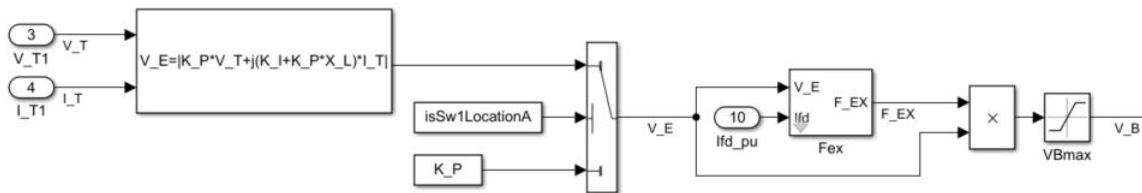
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

## Power Source

It is possible to adopt a different representation of the power source for the controlled rectifier by selecting the relevant option in the **Power source selector** parameter. The power source for the controlled rectifier can be either derived from the terminal voltage (Position A: power source derived from generator terminal voltage) or it can be independent of the terminal voltage (Position B: power source independent of generator terminal conditions).

This diagram shows a model of the exciter power source utilizing a phasor combination of the terminal voltage,  $V_T$ , and terminal current,  $I_T$ :



## Ports

### Input

#### V\_REF — Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_S — Input from stabilizer

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: single | double

#### V\_T — Terminal voltage

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

#### I\_T — Terminal current

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_SCLoel — Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter that prevents field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_SCLuel – Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter that prevents field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**I<sub>fd</sub>\_pu – Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

**Output****E<sub>fd</sub>\_pu – Field voltage**

scalar

Per-unit field voltage to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

**Parameters****General****Initial field voltage, E<sub>fd0</sub> (pu) – Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

**Initial terminal voltage, V<sub>t0</sub> (pu) – Initial terminal voltage**

1 (default) | real number

Initial per-unit terminal voltage.

**Initial terminal current, I<sub>t0</sub> (pu) – Initial terminal current**

1 (default) | real number

Initial per-unit terminal current.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control**

**Resistive component of load compensation, R\_C (pu) – Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) – Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) – Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control**

**Voltage regulator gain, K\_A (pu) – Regulator gain**

200 (default) | positive number

Gain associated with the voltage regulator.

**Regulator denominator (lag) time constant, T\_B (s) – Regulator lag time constant for first lead-lag block**

10 (default) | positive number

Equivalent lag time constant in the voltage regulator. Set this parameter to 0 when the additional lag dynamics are negligible.

**Regulator numerator (lead) time constant, T\_C (s) – Regulator lead time constant for first lead-lag block**

1 (default) | positive number

Equivalent lead time constant in the voltage regulator. Set this parameter to 0 when the additional lead dynamics are negligible.

**Voltage regulator proportional gain,  $K_{PR}$  (pu) — Regulator proportional gain**  
1 (default) | positive number

Proportional gain associated with the voltage regulator PI control block.

**Voltage regulator integral gain,  $K_{IR}$  (pu/s) — Regulator integral gain**  
0 (default) | positive number

Integral gain associated with the voltage regulator PI control block.

**Maximum voltage regulator output,  $V_{PImax}$  (pu) — Upper limit of regulator output for PI control**

99 (default) | real number

Maximum per-unit output voltage of the regulator PI control block.

**Minimum voltage regulator output,  $V_{PImin}$  (pu) — Lower limit of regulator output for PI control**

-99 (default) | real number

Minimum per-unit output voltage of the regulator PI control block.

**Maximum regulator output,  $V_{Rmax}$  (pu) — Upper limit of regulator output**

10 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output,  $V_{Rmin}$  (pu) — Lower limit of regulator output**

-10 (default) | real number

Minimum per-unit output voltage of the regulator.

**Maximum voltage error (regulator input),  $V_{Imax}$  (pu) — Voltage upper limit**

0.2 (default) | real number

Maximum per-unit voltage given as input to the regulator.

**Minimum voltage error (regulator input),  $V_{Imin}$  (pu) — Voltage lower limit**

-0.2 (default) | real number

Minimum per-unit voltage given as input to the regulator.

**Forward gain of inner loop field regulator,  $K_M$  (pu) — Inner loop forward gain**

7.93 (default) | positive number

Per-unit forward gain in the inner loop field regulator.

**Forward time constant of inner loop field regulator,  $T_M$  (s) — Inner loop forward time constant**

0.4 (default) | positive number

Forward time constant in the inner loop field regulator, in seconds.

**Maximum output of field current regulator,  $V_{Mmax}$  (pu) — Upper limit of field current regulator output**

1 (default) | real number



Maximum per-unit output voltage of the field current regulator.

**Minimum regulator output, V\_Mmin (pu) – Lower limit of field current regulator output**  
0 (default) | real number

Minimum per-unit output voltage of the field current regulator.

**Feedback gain of field current regulator, K\_G (pu) – Feedback gain of field current regulator**  
1 (default) | positive number

Per-unit feedback gain in the field current regulator.

**Maximum feedback voltage for field current regulator, V\_Gmax (pu) – Maximum feedback voltage for field current regulator**  
5.8 (default) | positive number

Maximum per-unit feedback voltage for the field current regulator.

**Alternate OEL input locations (V\_OEL) – OEL input location**  
Unused (default) | Summation point | Take-over

Overexcitation limiter input location.

**Alternate UEL input locations (V\_UEL) – UEL input location**  
Unused (default) | Summation point | Take-over

Underexcitation limiter input location.

**Alternate SCL input locations (V\_SCL) – SCL input location**  
Unused (default) | Summation point | Take-over

Stator current limiter input location:

- If you select `Summation point`, use the `V_SCLsum` input port.
- If you select `Take-over`, use the `V_SCLoel` and `V_SCLuel` input ports.

## Exciter

**Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**  
0.2 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Potential circuit (voltage) gain coefficient, K\_P (pu) – Potential circuit voltage gain coefficient**  
6.15 (default) | positive number

Per-unit voltage gain coefficient in the power source circuit.

## Dependencies

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

**Compound circuit (current) gain coefficient, K\_I (pu) – Potential circuit current gain coefficient**

0 (default) | positive number

Per-unit current gain coefficient in the power source circuit.

**Dependencies**

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

**Reactance associated with potential source, X\_L (pu) – Potential source reactance**

0.081 (default) | positive number

Per-unit reactance in the power source circuit.

**Dependencies**

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

**Potential circuit phase angle (degrees) – Potential circuit phase angle**

0 (default) | positive number

Phase angle of the power source circuit, in degrees.

**Dependencies**

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

**Maximum available exciter voltage, V\_Bmax (pu) – Upper limit of exciter voltage**

6.9 (default) | positive number

Maximum per-unit available exciter voltage value.

**Power source selector – Power source selector**

Position A: power source derived from generator terminal voltage (default) |  
Position B: power source independent of generator terminal conditions

Position of the power source selector, specified as Position A: power source derived from generator terminal voltage or Position B: power source independent of generator terminal conditions.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

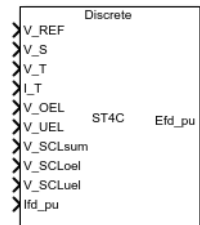
### **See Also**

SM ST1C | SM ST2C | SM ST4C

## SM ST4C

Discrete-time or continuous-time synchronous machine ST4C static excitation system with an automatic voltage regulator

**Library:** Simscape / Electrical / Control / SM Control



### Description

The SM ST4C block implements a synchronous-machine-type ST4C static excitation system model in conformance with IEEE 421.5-2016<sup>[1]</sup>.

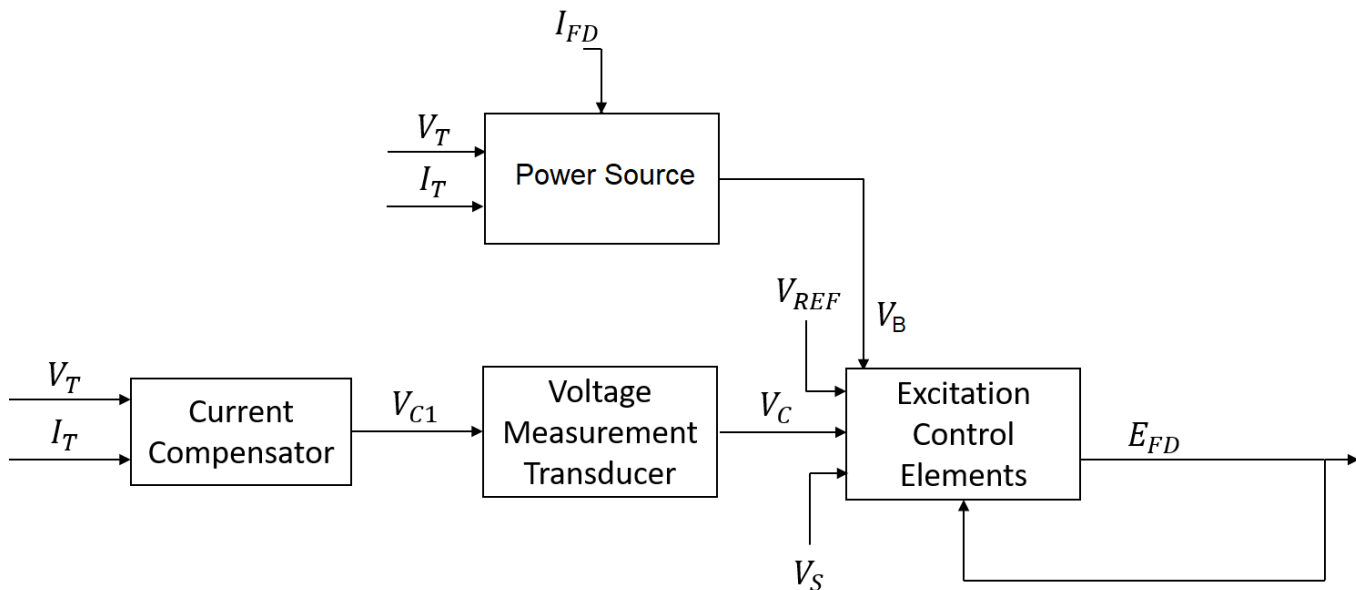
Use this block to model the control and regulation of the field voltage of a synchronous machine.

You can switch between continuous and discrete implementations of the block by using the **Sample time (-1 for inherited)** parameter. To configure the integrator for continuous time, set the **Sample time (-1 for inherited)** property to 0. To configure the integrator for discrete time, set the **Sample time (-1 for inherited)** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block.

The SM ST4C block comprises of four major components:

- The Current Compensator modifies the measured terminal voltage as a function of the terminal current.
- The Voltage Measurement Transducer simulates the dynamics of a terminal voltage transducer using a low-pass filter.
- The Excitation Control Elements component compares the voltage transducer output with a terminal voltage reference to produce a voltage error. This voltage error is then passed through a voltage regulator to produce the field voltage.
- The Power Source component models the power source for the controlled rectifier when it is independent from the terminal voltage.

This diagram shows the overall structure of the ST4C excitation system model:



In the diagram:

- $V_T$  and  $I_T$  are the measured terminal voltage and current of the synchronous machine.
- $V_{C1}$  is the current-compensated terminal voltage.
- $V_C$  is the filtered, current-compensated terminal voltage.
- $V_{REF}$  is the reference terminal voltage.
- $V_S$  is the power system stabilizer voltage.
- $V_B$  is the exciter field voltage.
- $E_{FD}$  and  $I_{FD}$  are the field voltage and current, respectively.

The following sections describe each of the major parts of the block in detail.

### Current Compensator and Voltage Measurement Transducer

The current compensator is modeled as:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

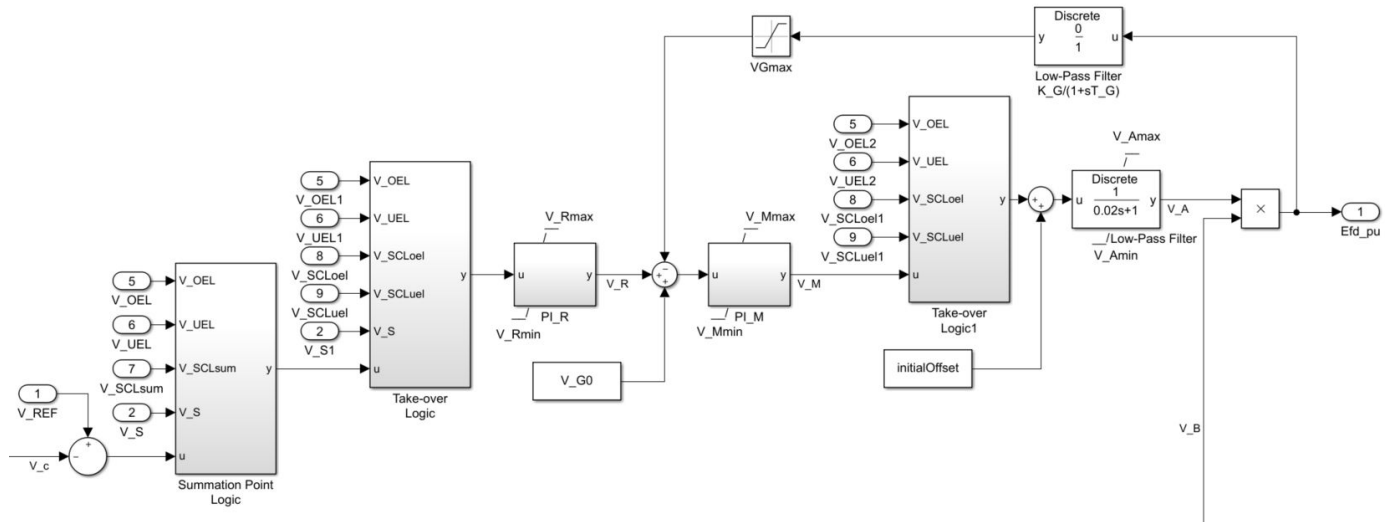
where:

- $R_C$  is the load compensation resistance.
- $X_C$  is the load compensation reactance.

The voltage measurement transducer is implemented as a Low-Pass Filter block with the time constant  $T_R$ . Refer to the documentation for the Low-Pass Filter block for the exact discrete and continuous implementations.

### Excitation Control Elements

This diagram illustrates the overall structure of the excitation control elements:



In the diagram:

- The Summation Point Logic subsystem models the summation point input location for the overexcitation limiter (OEL), underexcitation limiter (UEL), stator current limiter (SCL), and the power switch selector ( $V_S$ ) voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1976.
- There are two Take-over Logic subsystems. The subsystems model the take-over point input location for the OEL, UEL, SCL and PSS voltages. For more information about using limiters with this block, see “Field Current Limiters” on page 1-1976.
- The PI\_R subsystem models a PI controller that functions as a control structure for the automatic voltage regulator and allows the representation of an equipment retrofit with a modern digital controller. The minimum and maximum anti-windup saturation limits for the block are  $V_{PImin}$  and  $V_{PImax}$ , respectively.
- The PI\_M subsystem models a PI controller and replaces the Lead-Lag block in SM ST3C. The minimum and maximum anti-windup saturation limits for the block are  $V_{Mmin}$  and  $V_{Mmax}$ , respectively.
- An inner field voltage control loop is utilized to linearize the exciter control characteristic and it is composed of the gains  $K_M$  and  $K_G$  and the time constants  $T_M$  and  $T_G$ . The minimum and maximum anti-windup saturation limits for the Low-Pass Filter block are  $V_{Amin}$  and  $V_{Amax}$ , respectively.

### Field Current Limiters

You can use various field current limiters to modify the output of the voltage regulator under unsafe operating conditions:

- Use an overexcitation limiter to prevent overheating of the field winding due to excessive field current demand.
- Use an underexcitation limiter to boost field excitation when it is too low, which risks desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

Attach the output of any of these limiters at one of these points:

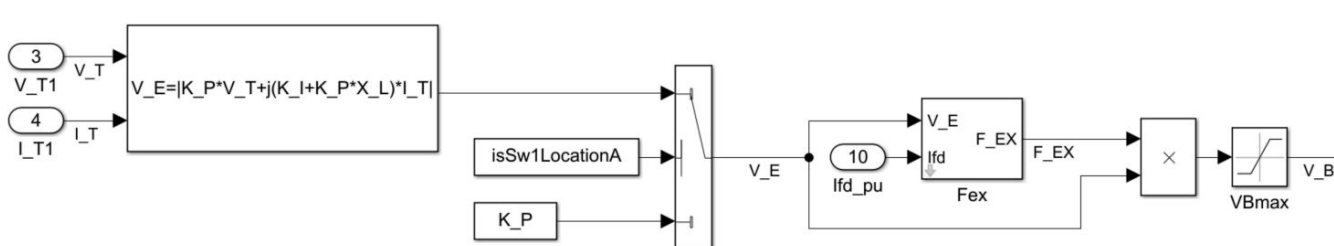
- The summation point as part of the automatic voltage regulator (AVR) feedback loop
- The take-over point to override the usual behavior of the AVR

If you are using the stator current limiter at the summation point, use the single input  $V_{SCLsum}$ . If you are using the stator current limiter at the take-over point, use both the overexcitation input,  $V_{SCLoel}$ , and the underexcitation input,  $V_{SCLuel}$ .

### Power Source

It is possible to adopt a different representation of the power source for the controlled rectifier by selecting the relevant option in the **Power source selector** parameter. The power source for the controlled rectifier can be either derived from the terminal voltage (Position A: power source derived from generator terminal voltage) or it can be independent of the terminal voltage (Position B: power source independent of generator terminal conditions).

This diagram shows a model of the exciter power source utilizing a phasor combination of the terminal voltage,  $V_T$ , and terminal current,  $I_T$ :



## Ports

### Input

#### **V\_REF — Voltage reference**

scalar

Voltage regulator reference set point, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_S — Input from stabilizer**

scalar

Input from the power system stabilizer, in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **V\_T — Terminal voltage**

scalar

Terminal voltage magnitude in per-unit representation, specified as a scalar.

Data Types: `single` | `double`

#### **I\_T — Terminal current**

scalar

Terminal current magnitude in per-unit representation, specified as a scalar.

Data Types: single | double

### **V\_OEL — Overexcitation limit signal**

scalar

Input from the overexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations (V\_OEL)** to Unused.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations (V\_OEL)** to Summation point.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations (V\_OEL)** to Take-over.

Data Types: single | double

### **V\_UEL — Underexcitation limit signal**

scalar

Input from the underexcitation limiter, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations (V\_UEL)** to Unused.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations (V\_UEL)** to Summation point.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations (V\_UEL)** to Take-over.

Data Types: single | double

### **V\_SCLsum — Summation point stator current limit signal**

scalar

Input from the stator current limiter when using the summation point, in per-unit representation, specified as a scalar.

#### **Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the summation point, set **Alternate SCL input locations (V\_SCL)** to Summation point.

Data Types: single | double

### **V\_SCLoel — Take-over stator current limit (OEL)**

scalar

Input from the stator current limiter that prevents field overexcitation when using the take-over point, in per-unit representation, specified as a scalar.



**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**V\_SCLuel – Take-over stator current limit (UEL)**

scalar

Input from the stator current limiter that prevents field underexcitation when using the take-over point, in per-unit representation, specified as a scalar.

**Dependencies**

- To ignore the input from the stator current limiter, set **Alternate SCL input locations (V\_SCL)** to Unused.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations (V\_SCL)** to Take-over.

Data Types: single | double

**Ifd\_pu – Measured field current**

scalar

Measured per-unit field current of the synchronous machine, specified as a scalar.

Data Types: single | double

**Output****Efd\_pu – Field voltage**

scalar

Per-unit field voltage to to apply to the field circuit of the synchronous machine, returned as a scalar.

Data Types: single | double

**Parameters****General****Initial field voltage, Efd0 (pu) – Initial output voltage**

1 (default) | real number

Initial per-unit voltage to apply to the field circuit of the synchronous machine.

**Initial terminal voltage, Vt0 (pu) – Initial terminal voltage**

1 (default) | real number

Initial per-unit terminal voltage.

**Initial terminal current, It0 (pu) – Initial terminal current**

1 (default) | real number

Initial per-unit terminal current.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Pre-Control****Resistive component of load compensation, R\_C (pu) — Compensation resistance**

0 (default) | positive number

Resistance used in the current compensation system. Set this parameter and **Reactance component of load compensation, X\_C (pu)** to 0 to disable current compensation.

**Reactance component of load compensation, X\_C (pu) — Compensation reactance**

0 (default) | positive number

Reactance used in the current compensation system. Set this parameter and **Resistive component of load compensation, R\_C (pu)** to 0 to disable current compensation.

**Regulator input filter time constant, T\_R (s) — Regulator time constant**

0 (default) | positive number

Equivalent time constant for the voltage transducer filtering.

**Control****Voltage regulator proportional gain, K\_PR (pu) — Regulator proportional gain**

10.75 (default) | positive number

Proportional gain associated with the voltage regulator PI control block.

**Voltage regulator integral gain, K\_IR (pu/s) — Regulator integral gain**

10.75 (default) | positive number

Integral gain associated with the voltage regulator PI control block.

**Thyristor bridge firing control equivalent time constant, T\_A (s) — Regulator lag time constant for first lead-lag block**

0.02 (default) | positive number

Equivalent time constant in the thyristor bridge firing control.

**Maximum regulator output, V\_Rmax (pu) — Upper limit of regulator output**

1 (default) | real number

Maximum per-unit output voltage of the regulator.

**Minimum regulator output, V\_Rmin (pu) – Lower limit of regulator output**

-0.87 (default) | real number

Minimum per-unit output voltage of the regulator.

**Forward proportional gain of inner loop field regulator, K\_PM (pu) – Inner loop forward proportional gain**

1 (default) | positive number

Per-unit forward proportional gain of the PI control block in the inner loop field regulator.

**Forward integral gain of inner loop field regulator, K\_IM (pu/s) – Inner loop forward integral gain**

0 (default) | positive number

Forward integral gain of the PI control block in the inner loop field regulator.

**Maximum output of field current regulator, V\_Mmax (pu) – Upper limit of field current regulator output**

99 (default) | real number

Maximum per-unit output voltage of the field current regulator.

**Minimum output of field current regulator, V\_Mmin (pu) – Lower limit of field current regulator output**

-99 (default) | real number

Minimum per-unit output voltage of the field current regulator.

**Maximum exciter output, V\_Amax (pu) – Upper limit of exciter output**

99 (default) | real number

Maximum per-unit output voltage of the exciter.

**Minimum exciter output, V\_Amin (pu) – Lower limit of exciter output**

-99 (default) | real number

Minimum per-unit output voltage of the exciter.

**Feedback gain of field current regulator, K\_G (pu) – Feedback gain of field current regulator**

0 (default) | positive number

Per-unit feedback gain of the field current regulator.

**Feedback time constant of field current regulator, T\_G (s) – Feedback time constant of field current regulator**

0 (default) | positive number

Per-unit feedback time constant of the field current regulator.

**Maximum feedback voltage for field current regulator, V\_Gmax (pu) – Maximum feedback voltage for field current regulator**

99 (default) | positive number

Maximum per-unit feedback voltage for the field current regulator.

**Alternate PSS input locations (V\_S) – PSS input location**

Summation point before take-over UEL (default) | Summation point after take-over UEL

Power system stabilizer input.

**Alternate OEL input locations (V\_OEL) – OEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage regulator input | Take-over at inner-loop output

Overexcitation limiter input location:

- If you select Summation point at voltage error, **V\_OEL** is an input of the Summation Point Logic subsystem.
- If you select Take-over at voltage regulator input, **V\_OEL** is an input of the Take-over Logic subsystem.
- If you select Take-over at inner-loop output, **V\_OEL** is an input of the Take-over Logic 1 subsystem.

**Alternate UEL input locations (V\_UEL) – UEL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage regulator input | Take-over at inner-loop output

Underexcitation limiter input location:

- If you select Summation point at voltage error, **V\_UEL** is an input of the Summation Point Logic subsystem.
- If you select Take-over at voltage regulator input, **V\_UEL** is an input of the Take-over Logic subsystem.
- If you select Take-over at inner-loop output, **V\_UEL** is an input of the Take-over Logic 1 subsystem.

**Alternate SCL input locations (V\_SCL) – SCL input location**

Unused (default) | Summation point at voltage error | Take-over at voltage regulator input | Take-over at inner-loop output

Stator current limiter input location:

- If you select Summation point, use the **V\_SCLsum** input port.
- If you select any of the Take-over options, use the **V\_SCLoel** and **V\_SCLuel** input ports.

**Exciter****Rectifier loading factor proportional to commutating reactance, K\_C (pu) – Rectifier loading factor**

0.113 (default) | positive number

Rectifier loading factor proportional to the commutating reactance.

**Potential circuit (voltage) gain coefficient, K\_P (pu) – Potential circuit voltage gain coefficient**

9.3 (default) | positive number

Per-unit voltage gain coefficient in the power source circuit.

#### Dependencies

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

#### Compound circuit (current) gain coefficient, $K_I$ (pu) – Potential circuit current gain coefficient

0 (default) | positive number

Per-unit current gain coefficient in the power source circuit.

#### Dependencies

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

#### Reactance associated with potential source, $X_L$ (pu) – Potential source reactance

0.124 (default) | positive number

Per-unit reactance in the power source circuit.

#### Dependencies

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

#### Potential circuit phase angle (degrees) – Potential circuit phase angle

0 (default) | positive number

Phase angle of the power source circuit, in degrees.

#### Dependencies

To enable this parameter, set **Power source selector** to Position A: power source derived from generator terminal voltage.

#### Maximum available exciter voltage, $V_{Bmax}$ (pu) – Exciter voltage upper limit

11.63 (default) | positive number

Maximum per-unit available exciter voltage value.

#### Power source selector – Power source selector

Position A: power source derived from generator terminal voltage (default) |  
Position B: power source independent of generator terminal conditions

Position of the power source selector, specified as Position A: power source derived from generator terminal voltage or Position B: power source independent of generator terminal conditions.

## Version History

Introduced in R2020a

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

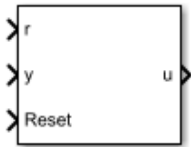
## See Also

SM ST1C | SM ST2C | SM ST3C

# Smith Predictor Controller

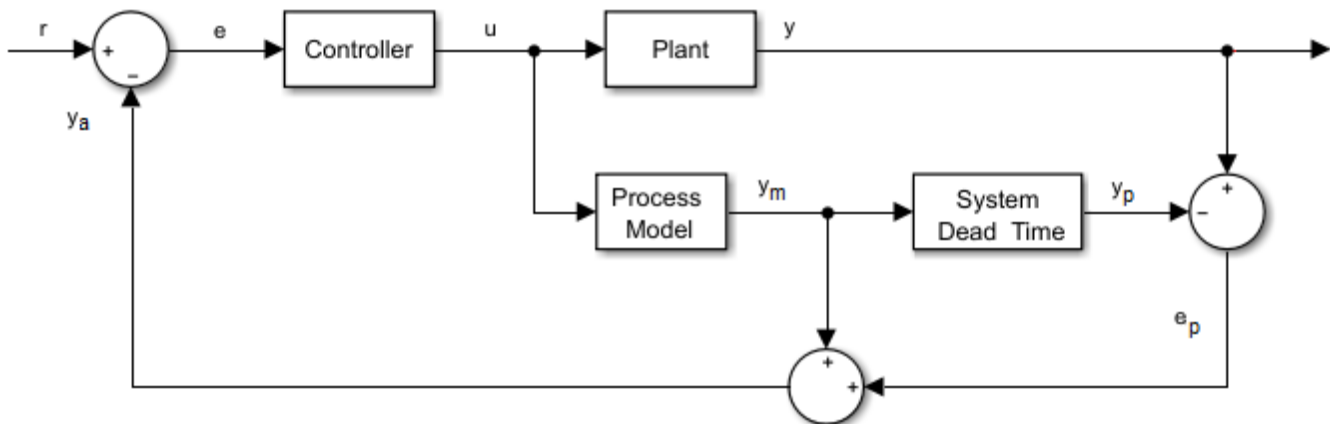
Discrete-time Smith dead-time compensator

**Library:** Simscape / Electrical / Control / General Control



## Description

The Smith Predictor Controller block compensates for dead time by implementing a Smith dead-time PI control structure in discrete time. This diagram shows the equivalent circuit for the block.



## Equations

The transfer function for a system with dead-time is

$$G_f(s) = G_p(s)e^{-\tau s},$$

where:

- $\tau$  is the system dead time.
- $G_p(s)$  is the process model.
- $G_f(s)$  is prediction error filter.

## Ports

### Input

**r – Plant reference**

scalar

Plant system reference signal.

Data Types: `single` | `double`**Reset – Integrator reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `Boolean`**y – Plant output**

scalar

Plant system output signal.

Data Types: `single` | `double`

### Output

**u – Controller output**

scalar

Control system output signal.

Data Types: `single` | `double`

## Parameters

**Proportional gain –  $K_p$** 

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the PI controller.**Integral gain –  $K_i$** 

1 (default) | positive scalar

Integral gain,  $K_i$ , of the PI controller.**Integrator initial condition – Initial integrator value**

0 (default) | scalar

Value of the integrator at simulation start time.

**Control action upper limit –  $u_{max}$** 5 (default) | scalar greater than the value of the **Control action lower limit** parameter

Upper limit for the control output signal.



**Control action lower limit —  $u_{min}$** 

0 (default) | scalar

Lower limit for the control output signal.

**Model discrete transfer function numerator — Transfer function numerator**

1 (default) | scalar or vector

Numerator of the system discretized transfer function. To determine the discrete transfer function, if you have a license for Control System Toolbox, use the `c2d` function.

**Model discrete transfer function denominator — Transfer function denominator**

[1 0.5] (default) | vector

Denominator of the system discretized transfer function. To determine the discrete transfer function, if you have a license for Control System Toolbox, use the `c2d` function.

**System dead time (samples) — Number of dead-time samples**

2 (default) | 0 or a positive integer

Number of samples of the dead time.

**Sample time (-1 for inherited) — Sampling interval**

-1 (default) | default value or a positive number

Time interval between samples. If the block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

## Version History

Introduced in R2017b

## References

- [1] Velagic, J. "Design of Smith-like Predictive Controller with Communication with Communication Delay Adaptation." *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*. Vol 2, Number 11, 2008, pp. 2447-2481.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Blocks**

State-Feedback Controller | RST Controller

# Solar Cell

Photovoltaic solar cell

**Library:** Simscape / Electrical / Sources



## Description

The Solar Cell block represents a solar cell current source.

The solar cell model includes the following components:

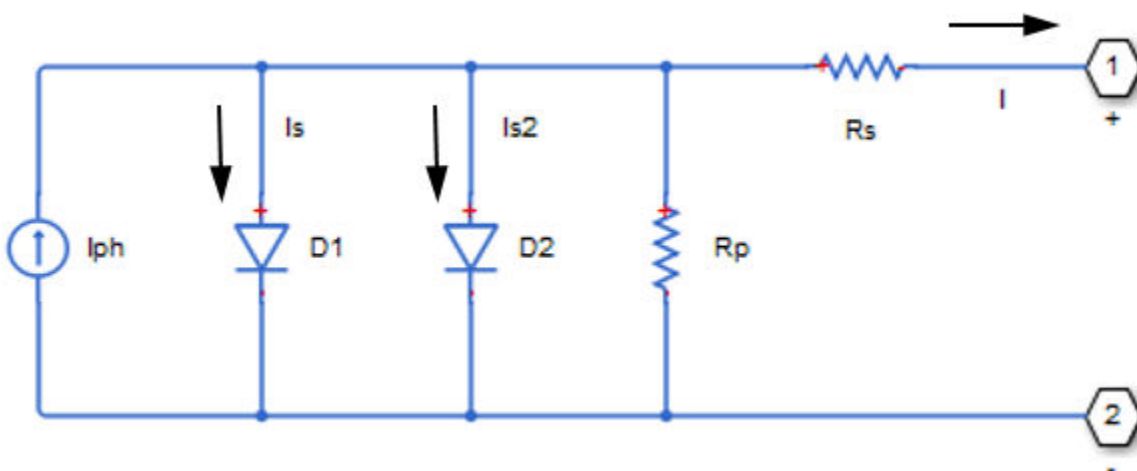
- “Solar-Induced Current” on page 1-1988
- “Temperature Dependence” on page 1-1990
- “Predefined Parameterization” on page 1-1990
- “Thermal Port” on page 1-1991

### Solar-Induced Current

The block represents a single solar cell as a resistance  $R_s$  that is connected in series with a parallel combination of the following elements:

- Current source
- Two exponential diodes
- Parallel resistor  $R_p$

The following illustration shows the equivalent circuit diagram:



The output current  $I$  is

$$I = I_{ph} - I_s * \left( e^{(V + I * R_s) / (N * V_t)} - 1 \right) - I_{s2} * \left( e^{(V + I * R_s) / (N_2 * V_t)} - 1 \right) - (V + I * R_s) / R_p$$

where:

- $I_{ph}$  is the solar-induced current:

$$I_{ph} = I_{ph0} \times \frac{I_r}{I_{r0}}$$

where:

- $I_r$  is the irradiance (light intensity), in  $W/m^2$ , falling on the cell.
- $I_{ph0}$  is the measured solar-generated current for the irradiance  $I_{r0}$ .
- $I_s$  is the saturation current of the first diode.
- $I_{s2}$  is the saturation current of the second diode.
- $V_t$  is the thermal voltage,  $kT/q$ , where:
  - $k$  is the Boltzmann constant.
  - $T$  is the **Device simulation temperature** parameter value.
  - $q$  is the elementary charge on an electron.
- $N$  is the quality factor (diode emission coefficient) of the first diode.
- $N_2$  is the quality factor (diode emission coefficient) of the second diode.
- $V$  is the voltage across the solar cell electrical ports.

The quality factor varies for amorphous cells, and is typically 2 for polycrystalline cells.

The block lets you choose between two models:

- An 8-parameter model where the preceding equation describes the output current
- A 5-parameter model that applies the following simplifying assumptions to the preceding equation:
  - The saturation current of the second diode is zero.
  - The impedance of the parallel resistor is infinite.

If you choose the 5-parameter model, you can parameterize this block in terms of the preceding equivalent circuit model parameters or in terms of the short-circuit current and open-circuit voltage the block uses to derive these parameters.

All models adjust the block resistance and current parameters as a function of temperature.

You can model any number of solar cells connected in series using a single Solar Cell block by setting the parameter **Number of series-connected cells per string** to a value larger than 1. 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.

## Temperature Dependence

Several solar cell parameters depend on temperature. The solar cell temperature is specified by the **Device simulation temperature** parameter value.

The block provides the following relationship between the solar-induced current  $I_{ph}$  and the solar cell temperature  $T$ :

$$I_{ph}(T) = I_{ph} * (1 + TIPH1 * (T - T_{meas}))$$

where:

- $TIPH1$  is the **First order temperature coefficient for Iph, TIPH1** parameter value.
- $T_{meas}$  is the **Measurement temperature** parameter value.

The block provides the following relationship between the saturation current of the first diode  $I_s$  and the solar cell temperature  $T$ :

$$I_s(T) = I_s * \left(\frac{T}{T_{meas}}\right)^{(TXIS1/N)} * e^{\left(EG * \left(\frac{T}{T_{meas}} - 1\right) / (N * V_t)\right)}$$

where  $TXIS1$  is the **Temperature exponent for Is, TXIS1** parameter value.

The block provides the following relationship between the saturation current of the second diode  $I_{s2}$  and the solar cell temperature  $T$ :

$$I_{s2}(T) = I_{s2} * \left(\frac{T}{T_{meas}}\right)^{(TXIS2/N2)} * e^{\left(EG * \left(\frac{T}{T_{meas}} - 1\right) / (N2 * V_t)\right)}$$

where  $TXIS2$  is the **Temperature exponent for Is2, TXIS2** parameter value.

The block provides the following relationship between the series resistance  $R_s$  and the solar cell temperature  $T$ :

$$R_s(T) = R_s * \left(\frac{T}{T_{meas}}\right)^{TRS1}$$

where  $TRS1$  is the **Temperature exponent for Rs, TRS1** parameter value.

The block provides the following relationship between the parallel resistance  $R_p$  and the solar cell temperature  $T$ :

$$R_p(T) = R_p * \left(\frac{T}{T_{meas}}\right)^{TRP1}$$

where  $TRP1$  is the **Temperature exponent for Rp, TRP1** parameter value.

## Predefined Parameterization

There are multiple available built-in parameterizations for the Solar Cell block.

This pre-parameterization data allows you to set up the block to represent components by specific suppliers. The parameterizations of these solar cell modules match the manufacturer data sheets. To load a predefined parameterization, double-click the Solar Cell block, click the **<click to select>**

hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

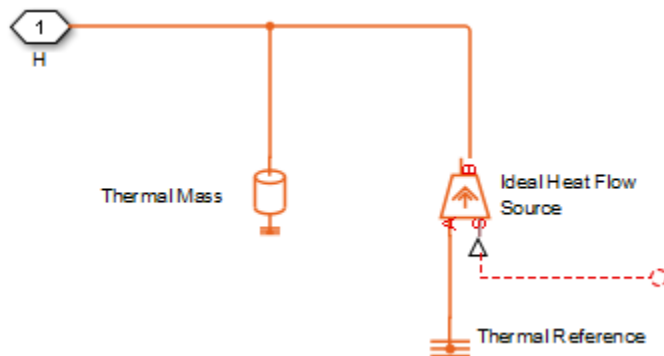
### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

The thermal port model, shown in the following illustration, represents just the thermal mass of the device. The thermal mass is directly connected to the component thermal port **H**. An internal Ideal Heat Flow Source block supplies a heat flow to the port and thermal mass. This heat flow represents the internally generated heat.



The internally generated heat in the solar cell is calculated according to the equivalent circuit diagram, shown at the beginning of the reference page, in the “Solar-Induced Current” on page 1-1988 section. It is the sum of the  $i^2 \cdot R$  losses for each of the resistors plus the losses in each of the diodes.

The internally generated heat due to electrical losses is a separate heating effect to that of the solar irradiation. To model thermal heating due to solar irradiation, you must account for it separately in your model and add the heat flow to the physical node connected to the solar cell thermal port.

## Ports

### Input

#### **Ir – Incident irradiance**

physical signal

Physical signal associated with the solar cell incident irradiance.

### Conserving

#### **+ – Positive voltage**

electrical

Electrical conserving port associated with the solar cell positive voltage

#### **- – Negative voltage**

electrical

Electrical conserving port associated with the solar cell negative voltage

#### **H – Thermal port**

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

#### **Selected part – Predefined parameterization option**

<click to select> (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see “Load Predefined Parameterizations”.

#### **Modeling option – Whether to enable thermal port**

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

### Cell Characteristics

All quantities in this setting are for a single cell.

#### **Parameterize by – Block parameterization**

By s/c current and o/c voltage, 5 parameter (default) | By equivalent circuit parameters, 5 parameter | By equivalent circuit parameters, 8 parameter

Select one of the following methods for block parameterization:

- By *s/c current and o/c voltage*, 5 parameter — Provide short-circuit current and open-circuit voltage that the block converts to an equivalent circuit model of the solar cell.
- By *equivalent circuit parameters*, 5 parameter — Provide electrical parameters for an equivalent circuit model of the solar cell using the 5-parameter solar cell model that makes the following assumptions:
  - The saturation current of the second diode is zero.
  - The parallel resistor has infinite impedance.
- By *equivalent circuit parameters*, 8 parameter — Provide electrical parameters for an equivalent circuit model of the solar cell using the 8-parameter solar cell model.

### **Short-circuit current, $I_{sc}$ — Short-circuit current**

7.34 A (default)

Current that flows when you short-circuit the solar cell.

#### **Dependencies**

This parameter is visible only when you select *By s/c current and o/c voltage*, 5 parameter for the **Parameterize by** parameter.

### **Open-circuit voltage, $V_{oc}$ — Open-circuit voltage**

0.6 V (default)

Voltage across the solar cell when it is not connected.

#### **Dependencies**

This parameter is visible only when you select *By s/c current and o/c voltage*, 5 parameter for the **Parameterize by** parameter.

### **Diode saturation current, $I_s$ — First diode saturation current**

1e-6 A (default)

Asymptotic reverse current of the first diode for increasing reverse bias in the absence of any incident light.

#### **Dependencies**

This parameter is visible only when you select *By equivalent circuit parameters*, 5 parameter or *By equivalent circuit parameters*, 8 parameter for the **Parameterize by** parameter.

### **Diode saturation current, $I_{s2}$ — Second diode saturation current**

0 A (default)

Asymptotic reverse current of the second diode for increasing reverse bias in the absence of any incident light.

#### **Dependencies**

This parameter is visible only when you select *By equivalent circuit parameters*, 8 parameter for the **Parameterize by** parameter.

**Solar-generated current for measurements,  $I_{ph0}$  — Solar-generated current for measurements**

7.34 A (default)

Solar-induced current when the irradiance is  $I_{r0}$ .**Dependencies**

This parameter is visible only when you select **By equivalent circuit parameters, 5 parameter** or **By equivalent circuit parameters, 8 parameter** for the **Parameterize by** parameter.

**Irradiance used for measurements,  $I_{r0}$  — Irradiance used for measurements**1000 W/m<sup>2</sup> (default)Irradiance that produces a current of  $I_{ph0}$  in the solar cell.**Quality factor,  $N$  — First diode emission coefficient**

1.5 (default)

Emission coefficient of the first diode.

**Quality factor,  $N2$  — Second diode emission coefficient**

2 (default)

Emission coefficient of the second diode.

**Dependencies**

This parameter is visible only when you select **By equivalent circuit parameters, 8 parameter** for the **Parameterize by** parameter.

**Series resistance,  $R_s$  — Series resistance**

0 Ohm (default)

Internal series resistance.

**Parallel resistance,  $R_p$  — Parallel resistance**

inf Ohm (default)

Internal parallel resistance.

**Dependencies**

This parameter is visible only when you select **By equivalent circuit parameters, 8 parameter** for the **Parameterize by** parameter.

**Panel Configuration****Number of series-connected cells per string — Series-connected solar cells per string**

1 (default) | positive scalar

Number of series-connected solar cells per string modeled by the block. The value must be greater than 0.



**Number of parallel-connected strings — Number of parallel-connected strings**

1 (default) | positive scalar

Number of strings connected in parallel. The value must be greater than 0.

**Temperature Dependence****First order temperature coefficient for  $I_{ph}$ ,  $TIPH1$  — First order temperature coefficient for  $I_{ph}$** 

0 1/K (default) | nonnegative scalar

Order of the linear increase in the solar-generated current as temperature increases. The value must be greater than or equal to 0.

**Energy gap,  $E_G$  — Energy gap**

1.11 eV (default)

Solar cell activation energy. The value must be greater than or equal to 0.1.

**Temperature exponent for  $I_s$ ,  $TXIS1$  — Temperature exponent for  $I_s$** 

3 (default) | positive scalar

Order of the exponential increase in the current from the first diode as temperature increases. The value must be greater than 0.

**Temperature exponent for  $I_{s2}$ ,  $TXIS2$  — Temperature exponent for  $I_{s2}$** 

3 (default) | nonnegative scalar

Order of the exponential increase in the current from the second diode as temperature increases. The value must be greater than or equal to 0.

**Dependencies**

This parameter is visible only when you select **By equivalent circuit parameters**, 8 parameter for the **Parameterize by** parameter.

**Temperature exponent for  $R_s$ ,  $TRS1$  — Temperature exponent for  $R_s$** 

0 (default)

Order of the exponential increase in the series resistance as temperature increases. The value must be greater than or equal to 0.

**Temperature exponent for  $R_p$ ,  $TRP1$  — Temperature exponent for  $R_p$** 

0 (default) | nonnegative scalar

Order of the exponential increase in the parallel resistance as temperature increases. The value must be greater than or equal to 0.

**Dependencies**

This parameter is visible only when you select **By equivalent circuit parameters**, 8 parameter for the **Parameterize by** parameter.

**Measurement temperature — Measurement temperature**

25 degC (default) | positive scalar

Temperature at which the solar cell parameters were measured. The value must be greater than 0.

**Device simulation temperature – Device simulation temperature**

25 degC (default) | positive scalar

Temperature at which the solar cell is simulated. The value must be greater than 0.

**Thermal port**

To enable these parameters, set **Modeling option** to Show thermal port.

**Thermal mass per cell – Thermal mass per cell**

100 J/K (default)

The heat energy required to raise the temperature of one solar cell by one degree. When modeling more than one cell in series, specify the thermal mass for a single cell. This value gets multiplied internally by the number of cells to determine the total thermal mass.

**Initial temperature – Initial temperature**

25 degC (default)

The temperature of the solar cell at the start of simulation.

## Version History

**Introduced in R2008a**

## References

- [1] Gow, J.A. and C.D. Manning. "Development of a Photovoltaic Array Model for Use in Power-Electronics Simulation Studies." *IEEE Proceedings of Electric Power Applications*, Vol. 146, No. 2, 1999, pp. 193-200.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

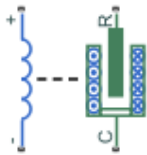
**Topics**

"List of Pre-Parameterized Components"

# Solenoid

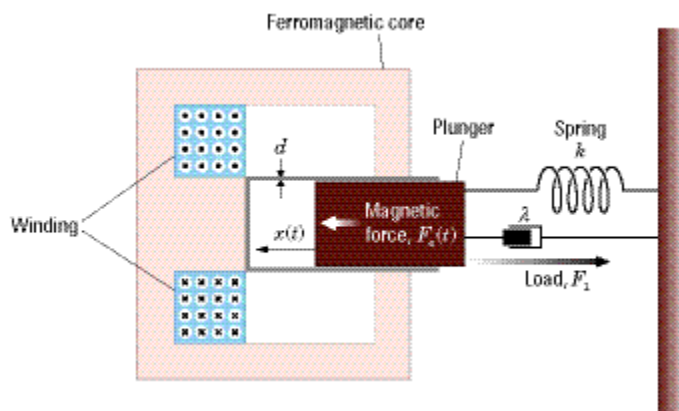
Electrical characteristics and generated force of solenoid

**Library:** Simscape / Electrical / Electromechanical / Mechatronics  
Actuators



## Description

The Solenoid block represents the electrical characteristics and generated force for the solenoid in this figure:



Both positive and negative currents generate a force that pulls the plunger towards the winding. This minimizes the reluctance and decreases the stroke. To increase the stroke, a spring or an external force must overcome the magnetic force.

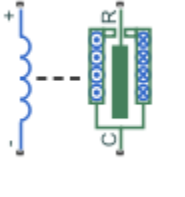
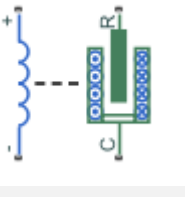
The return spring is optional. To remove the effects of this spring from the model, set the **Spring constant** parameter to 0.

To show the direction in which the solenoid acts, the block icon changes dynamically based on value of the **Motion direction under excitation** parameter.

---

**Note** If you place an external position sensor around the block, pay attention to the sensor orientation and the initial position condition. The initial position condition you specify on the sensor corresponds to the **Stroke, X** variable of the Solenoid block only if you orientate the sensor in a direction that is consistent with the value of the **Motion direction under excitation** parameter.

---

When The Motion direction under excitation Parameter Is Set To...	Block Icon
Port R moves away from port C	
Port R moves towards port C	

The equation of motion for the plunger as a function of position,  $x$ , is:

$$F_l + m\ddot{x} + \lambda\dot{x} + kx = F_e$$

where  $F_e$  is the electromagnetic force,  $F_l$  is the load force,  $\lambda$  is the viscous damping term and  $m$  is the plunger mass. The electromagnetic force is related to the solenoid current and inductance by:

$$F_e = \frac{1}{2}i^2 \frac{\partial L(x)}{\partial x}$$

The inductance, which is derived in [1], can be written as:

$$\frac{\partial L(x)}{\partial x} = \frac{-\beta}{(\alpha + \beta x)^2}$$

where  $\alpha$  and  $\beta$  are constants. Plugging the preceding equation into the equation for electromagnetic force gives the force-stroke relationship of the solenoid for a current  $i_0$ :

$$F = \frac{1}{2}i_0^2 \frac{-\beta}{(\alpha + \beta x)^2}$$

The Solenoid block solves for  $\alpha$  and  $\beta$  by taking the two specified force and stroke measurements and substituting them into the preceding equation. It solves the resulting equations for  $\alpha$  and  $\beta$ .

A positive current from the electrical + to - ports creates a negative force (i.e., a pulling force) from the mechanical C to R ports.

### Model Thermal Effects

You can expose the thermal port to model the effects of losses that convert power to heat. To expose the thermal port, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain a thermal port.
- Show thermal port — The block contains one thermal conserving port.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

Use the **Stroke, X** variable to set the target for the initial plunger position at the start of simulation.

---

**Note** The **Stroke, X** variable is the initial air gap or stroke of the solenoid, and so should be greater than or equal to  $\theta$ .

---

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the solenoid positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the solenoid negative terminal.

#### C — Case

mechanical

Mechanical translational conserving port associated with the case.

#### R — Plunger

mechanical

Mechanical translational conserving port associated with the plunger.

#### H — Thermal port

thermal

Thermal port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of losses that convert power to heat.

### Magnetic Force

#### Motion direction under excitation – Motion direction under excitation

Port R moves towards port C (default) | Port R moves away from port C

Select one of the following options for the type of motion direction under excitation:

- Port R moves away from port C
- Port R moves towards port C

#### Pull-in forces [F1 F2] – Vector of pull-in forces

[7.5, .75] N (default)

A vector of the force values at the two points on the force-stroke curve. The second measurement point must be at a stroke that is greater than that of the first measurement point. When the manufacturer does not provide a force-stroke curve, set F1 to the holding force (when X1 = 0) and F2 to the pull-in force when running the solenoid at the **Rated voltage Vdc** and **Rated current Idc** values.

#### Stroke (air gap) [X1 X2] – Vector of strokes

[1, 5] mm (default)

A vector of the stroke (plunger distance from the fully closed position) values at the two points on the force-stroke curve. The second measurement point must be at a stroke that is greater than that of the first measurement point. To ensure a finite force value, the points must meet the condition

$$\frac{X2}{X1} > \sqrt{\frac{F1}{F2}}$$

#### Rated voltage Vdc – Rated voltage

50 V (default)

The voltage at which the solenoid is rated to operate. This voltage value is used to measure the **Forces [F1 F2]** and **Stroke [X1 X2]** values.

#### Rated current Idc – Rated current

0.05 A (default)

The current that flows when the solenoid is supplied with the **Rated voltage Vdc** voltage.

### Mechanical

#### Spring constant – Spring constant

200 N/m (default)

Constant representing the stiffness of the spring that acts to retract the plunger when the solenoid is powered off. The force is zero when the plunger is displaced to the **Stroke for zero spring force** parameter value. Set the spring constant to zero if there is no spring.

**Stroke for zero spring force – Stroke for zero spring force**

5 mm (default)

The stroke at which the spring provides no force.

**Damping – Damping**

1 N/(m/s) (default)

The term  $\lambda$  in the equation of motion for the plunger as a function of position that linearly damps the plunger motion. The value can be zero.

**Plunger mass – Plunger mass**

0.05 kg (default)

The weight of the solenoid plunger. The value can be zero.

**Maximum stroke – Maximum stroke**

Inf mm (default)

The maximum amount by which the plunger can be displaced. You can use this parameter to model a hard endstop that limits the stroke. The default value is Inf mm, which means no stroke limit.

**Contact stiffness – Contact stiffness**

1e6 N/m (default)

Stiffness of the plunger contact that models the hard stop at the minimum ( $x = 0$ ) and maximum ( $x =$  **Maximum stroke**) plunger positions.

**Contact damping – Contact damping**

500 N/(m/s) (default)

Damping of the plunger contact that models the hard stop at the minimum ( $x = 0$ ) and maximum ( $x =$  **Maximum stroke**) plunger positions.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Resistance temperature coefficient.

**Measurement temperature – Measurement temperature**

25 degC (default)

Temperature for which the actuator parameters are defined.

**Thermal mass – Thermal mass**

100 J/K (default)

Thermal mass is the energy required to raise the temperature by one degree.

## **Version History**

**Introduced in R2008a**

## **References**

[1] S.E. Lyshevski. *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC, 1999.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

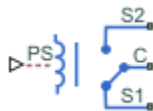
## **See Also**



# SPDT Relay

Single-pole, double-throw relay with delays and faults

**Library:** Simscape / Electrical / Switches & Breakers / Relays



## Description

The SPDT Relay block models a single-pole, double-throw relay. The block has three potential states:

- De-energized — The common contact, **C**, is connected to the normally closed contact, **S1**.
- Energized — The common contact, **C**, is connected to the normally open contact, **S2**.
- Open Circuit — The relay is open. The common contact, **C**, is not connected to the normally closed contact, **S1**, or the normally open contact, **S2**.

You can:

- Control the relay state using the physical signal input port or electrical conserving ports.
- Output the relay state.
- Delay the breaking and making of either or both of the connections.
- Introduce a behavioral or temporal fault that results in a stuck connection, an open circuit, or degraded contact resistance.

## Relay State Control

The block has two control variants:

- Physical signal (PS) control — The state of the relay depends on how the value of the input physical signal compares to the relay threshold. The relay threshold,  $th$ , is the value that you specify for the **Threshold** parameter.
- Electrical control — The state of the relay depends on how the current through the positive and negative electrical conserving ports, which represent the relay winding, compares to the relay upper and lower thresholds. The upper and lower thresholds depend on the values that you specify for the parameters in the **Winding** settings.

For the PS control variant, at the start of simulation:

- If the input signal,  $PS$ , is less than or equal to  $th$ , the relay is de-energized and the common contact, **C**, connects to the normally closed contact, **S1**.
- If the input signal,  $PS$ , greater than  $th$ , the relay is energized and **C** connects to the normally open contact, **S2**.

After the simulation starts, if  $PS$  rises above  $th$ , the block goes from a de-energized state to an energized state :

- The **C-S1** connection breaks after the delay specified in **Time-to-break C-S1 connection**.
- The **C-S2** connection closes after the delay specified in **Time-to-make C-S2 connection**.

If the block goes from an energized state to a de-energized state, that is *PS* falls to or below *th*:

- The **C-S2** connection breaks after the delay specified in **Time-to-break C-S2 connection**.
- The **C-S1** connection closes after the delay specified in **Time-to-make C-S1 connection**.

For the electrical control variant, the upper and lower thresholds depend on the specified values for the **Rated voltage**, **Percent rated voltage to energize**, **Percent rated voltage to de-energize**, and **Winding series resistance** parameters. The rated current is:

$$i_{rated} = \frac{V_{rated}}{r_{coil}}.$$

The equation for the upper current threshold,  $i_{energized}$ , is:

$$i_{energized\_current} = \frac{Rated\ Voltage * Percent\ rated\ voltage\ to\ energize}{Winding\ series\ resistance}.$$

The equation for the lower current threshold,  $i_{de-energized}$ , is:

$$i_{de-energized\_current} = \frac{Rated\ Voltage * Percent\ rated\ voltage\ to\ de - energize}{Winding\ series\ resistance}.$$

At the start of simulation:

- If the control current is less than  $i_{energized}$ , the common contact, **C**, connects to the normally closed contact, **S1**.
- If the control current is greater than  $i_{energized}$ , the relay is energized and **C** connects to the normally open contact, **S2**.

After the start of simulation, if the current rises above  $i_{energized}$ , the block goes from a de-energized state to an energized state:

- The **C-S1** connection breaks after the delay specified in **Time-to-break C-S1 connection**.
- The **C-S2** connection closes after the delay specified in **Time-to-make C-S2 connection**.

If the current falls below  $i_{de-energized}$ , the block goes from an energized state to a de-energized state:

- The **C-S2** connection breaks after the delay specified in **Time-to-break C-S2 connection**.
- The **C-S1** connection closes after the delay specified in **Time-to-make C-S1 connection**.

### Output the Relay State

To view the relay state, expose port **x**, a physical signal port that outputs the state of each connection. To expose the **x** port, in the **Main** settings, set **State port** to **Visible**.

The table shows how the state of the relay relates to the state of the connections. A closed connection has a state of 1. An open connection has a state of 0.

**Relay and Connection States**

C-S1 Connection State	C-S2 Connection State	Relay State
1	0	De-energized
0	1	Energized
0	0	Open Circuit

**Connection Delays**

You can specify delays for making and breaking connections in the **Mechanical** settings. The table shows how the make and brake time delays affect the connections between contacts **S1** and **S2** and the common contact, **C**.

Mechanical Settings		Relay State	C-S1 Connection State	C-S2 Connection State	Results, with C-S1 in Yellow and C-S2 in Blue
<b>Parameter</b>	<b>Value</b>	De-energized	1	0	
Time-to-break C-S1 connection	0	Energized	0	1	
Time-to-make C-S1 connection	0				
Time-to-break C-S2 connection	0				
Time-to-make C-S2 connection	0				
<b>Parameter</b>	<b>Value</b>	De-energized	1	0	
Time-to-break C-S1 connection	> 0				
Time-to-make C-S1 connection	0				
Time-to-break C-S2 connection	0				

Mechanical Settings		Relay State	C-S1 Connection State	C-S2 Connection State	Results, with C-S1 in Yellow and C-S2 in Blue
		Energized	1, 0	1	
<b>Parameter</b>	<b>Value</b>	De-energized	0, 1	0	
Time-to-break C-S1 connection	0	Energized	0	1	
Time-to-make C-S1 connection	> 0				
Time-to-break C-S2 connection	0				
Time-to-make C-S2 connection	0				
<b>Parameter</b>	<b>Value</b>	De-energized	1	1, 0	
Time-to-break C-S1 connection	0	Energized	0	1	
Time-to-make C-S1 connection	0				
Time-to-break C-S2 connection	> 0				
Time-to-make C-S2 connection	0				

Mechanical Settings		Relay State	C-S1 Connection State	C-S2 Connection State	Results, with C-S1 in Yellow and C-S2 in Blue
<b>Parameter</b>	<b>Value</b>	De-nergized	1	0	
<b>Time-to-break C-S1 connection</b>	0	Energized	0	0, 1	
<b>Time-to-make C-S1 connection</b>	0				
<b>Time-to-break C-S2 connection</b>	0				
<b>Time-to-make C-S2 connection</b>	> 0				

### Faults

The block provides five fault types:

- C-S1 stuck closed
- C-S2 stuck closed
- C open circuit (no path to S1 or S2)
- Degraded contact resistance
- Winding failed open circuit. This fault type is available only for the electrical control variant.

Each type of fault can take effect only when the threshold for a temporal or behavioral trigger is exceeded.

#### C-S1 Stuck Closed Fault

The **C-S1** stuck closed fault occurs if the common contact, **C**, becomes mechanically, electrically, or chemically stuck to the normally closed contact, **S1**. Causes of this type of fault include:

- 1 Contact welding, possibly due to an inrush current or breaking current that exceeds the rating or load short circuit.
- 2 Mechanical damage.
- 3 Insulation deterioration.
- 4 Inductive voltage.

The table shows the faulted state for a **C-S1** stuck closed fault.

Connection	State
C-S1	1
C-S2	0

A **C-S1** stuck closed temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time after which contactors can stick** parameter.

A **C-S1** stuck closed behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and

$$t_{i > i_{th}} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i > i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

When the temporal or behavioral fault thresholds are exceeded, if **C-S1** is:

- Closed — **C-S1** remains closed for the rest of the simulation.
- Open — **C-S1** the fault does not take effect unless the relay becomes de-energized and the **C-S1** connection closes. Once closed, the **C-S1** connection remains closed for the rest of the simulation.

#### C-S2 Stuck Closed Fault

The **C-S2** stuck closed fault occurs if the common contact, **C**, becomes mechanically, electrically, or chemically stuck to the normally closed contact, **S2**. Causes of this type of fault include:

- 1 Contact welding, possibly due to an inrush current or breaking current that exceeds the rating or load short circuit.
- 2 Mechanical damage.
- 3 Insulation deterioration.

The table shows the faulted state for a **C-S2** stuck closed fault.

Connection	State
C-S1	0

Connection	State
C-S2	1

A **C-S2** stuck closed temporal fault can occur only if the simulation time exceeds the time threshold such that:

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time after which contactors can stick** parameter.

A **C-S2** stuck closed behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and

$$t_i > i_{th} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i>i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

When the temporal or behavioral fault thresholds are exceeded, if **C-S2** is:

- Closed — **C-S2** remains closed for the rest of the simulation.
- Open — The **C-S2** the fault does not take effect unless the relay becomes energized and the **C-S2** connection closes. Once closed, the **C-S2** connection remains closed for the rest of the simulation.

#### **C Open Circuit (No Path to S1 or S2) Fault**

The **C** open circuit (no path to **S1** or **S2**) fault occurs if the common contact, **C**, becomes mechanically stuck in an open position. This type of fault can happen if:

- There is contact damage or deterioration.
- The switch connector becomes mechanically stuck in the middle of the relay.

The table shows the faulted state for a **C** open circuit (no path to **S1** or **S2**) fault.

Connection	State
C-S1	0
C-S2	0

A **C** open circuit temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time after which C can be open circuit** parameter.

A **C** open circuit behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and

$$t_{i > i_{th}} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i > i_{th}}$  is the period of time during which the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

When the temporal fault threshold is exceeded, if:

- The relay is in the open-circuit state, that is, one connection has been broken, and the other connection has not yet been made, the relay remains open for the rest of the simulation.
- Either the **C-S1** or **C-S2** connection is closed, the fault does not take effect unless the relay is energized or de-energized and the closed connection is broken. If the connection is broken, the relay becomes an open-circuit, and remains open for the rest of the simulation.

When the behavioral fault thresholds are exceeded, if:

- The relay is in the open-circuit state, that is, one connection has been broken, and the other connection has not yet been made, the relay remains open for the rest of the simulation.
- Either the **C-S1** or **C-S2** connection is closed, the relay state immediately becomes an open circuit and remains an open circuit for the rest of the simulation.

### **Degraded Contact Resistance Fault**

Causes of the degraded contact resistance fault include:

- 1** Overuse-induced overload conditions. High inrush currents and voltages can cause overload conditions, as can excessive switching of the relay. Overload conditions ultimately trigger electrical arcing, which generates heat that degrades the contact material.
- 2** Chemical contamination that interferes with the operation of the relay contacts. Contaminants, which can include oxidation films or foreign particles, tend to produce high or unstable contact resistance readings.
- 3** End of relay life.

The table shows the faulted state for a degraded contact resistance fault.



Connection	State
C-S1	0 or 1. The contact resistance is degraded.
C-S2	0 or 1. The contact resistance is degraded.

A degraded contact resistance temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.

The degraded contact resistances of **S1** and **S2** are:

$$r_{contact\_s1}(t) = r_{contact\_fault\_s1} - (r_{contact\_fault\_s1} - r_{contact})sech\left(\frac{t - t_{th\_t}}{\tau}\right),$$

$$r_{contact\_s2}(t) = r_{contact\_fault\_s2} - (r_{contact\_fault\_s2} - r_{contact})sech\left(\frac{t - t_{th\_t}}{\tau}\right),$$

where:

- $r_{contact\_fault\_s1}$  is the final value of the faulted **S1** contact resistance.
- $r_{contact\_fault\_s2}$  is the final value of the faulted **S2** contact resistance.
- $r_{contact}$  is the unfaulted **S1** and **S2** contact resistance.

A degraded contact resistance behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{S1 - C} > i_{th},$$

and

$$t_{i1 > i_{th}} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i1 > i_{th}}$  is the period of time during which the current threshold for the connection **C-S1** is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

For a behavior-triggered fault, if  $i_{S1 - C} > i_{th}$  continuously over the time interval  $t_{th\_b}$ ,

$$r_{contact\_s1}(t) = r_{contact\_fault\_s1} - (r_{contact\_fault\_s1} - r_{contact})sech\left(\frac{t_{i1 > i_{th}} - t_{th\_b}}{\tau}\right),$$

where:

- $i_{S1-C}$  is the common contact to normally closed contact, **C-S1**, current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.
- $r_{contact\_fault\_s1}$  is the final value of the faulted **S1** contact resistance.
- $r_{contact}$  is the unfaulted **S1** contact resistance.
- $\tau$  is specified value for the **Time constant for degraded contact resistance** parameter.

Likewise, for a **C-S2** connection, if

$$i_{S2-C} > i_{th}$$

and

$$t_{i2} > i_{th} > t_{th\_b},$$

then the degraded contact resistance is:

$$r_{contact\_s2}(t) = r_{contact\_fault\_s2} - (r_{contact\_fault\_s2} - r_{contact}) \operatorname{sech}\left(\frac{t_{i2} > i_{th} - t_{th\_b}}{\tau}\right),$$

where:

- $i_{S2-C}$  is the common contact to normally closed contact, **C-S2**, current.
- $r_{contact\_fault\_s2}$  is the final value of the faulted **S2** contact resistance

When the temporal fault threshold is exceeded, for both the **C-S1** and the **C-S2** connections, the contact resistance is immediately degraded and remains degraded for the rest of the simulation.

When the behavioral fault thresholds are exceeded for **C-S1**, in terms of  $i_{S1-C}$ , the resistance for the **C-S1** connection is degraded after the specified value for the **Time to fail when exceeding current** parameter and remains degraded for the rest of the simulation.

When the behavioral fault thresholds are exceeded for **C-S2**, in terms of  $i_{S2-C}$ , the resistance for the **C-S2** connection is degraded after the specified value for the **Time to fail when exceeding current** parameter and remains degraded for the rest of the simulation.

### Winding failed open circuit

The open-circuit winding fault is available only for the electrical control variant. An open circuit in the winding coil can cause this type of fault.

The table shows the faulted state for a winding failed open circuit fault.

Connection	State
<b>C-S1</b>	1
<b>C-S2</b>	0

A winding failed open circuit temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.

For time fault, the relay switches depending on the winding current, which is approximated as:

$$L \frac{di}{dt} + Ri = v_{winding} * \operatorname{sech}\left(\frac{t - t_{th\_t}}{\tau}\right),$$

where :

- $L$  is the winding inductance.
- $R$  is the winding resistance.
- $i$  is the winding current.
- $v_{winding}$  is the voltage across the winding.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.
- $\tau$  is the specified value for the **Time constant for winding open circuit transition** parameter.

A winding failed open circuit behavioral fault can occur only if one of these conditions is met:

- The winding current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold.
- The winding voltage exceeds the voltage threshold for a number of times that exceeds the threshold for the number of voltage overloads.

That is, the behavioral fault can occur only if:

$$i_{winding} > i_{th}$$

and then

$$t_i > i_{th} > t_{th\_b},$$

where:

- $i_{winding}$  is the winding current.
- $i_{th}$  is the specified value for the **Maximum permissible winding current** parameter.
- $t_{i>i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

or if:

$$v_{winding} > v_{th},$$

and then

$$N_v > v_{th} > N_{th},$$

where:

- $v_{winding}$  is the winding voltage.
- $v_{th}$  is the specified value for the **Maximum permissible winding voltage** parameter.

- $N_{v > v_{th}}$  is the number of times that the voltage threshold is exceeded.
- $N_{th}$  is the specified value for the **Number of events to fail when exceeding voltage** parameter.

If the temporal or behavioral fault thresholds are exceeded, the **C-S1** remains closed for the rest of the simulation

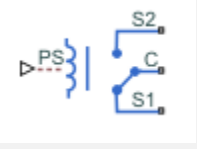
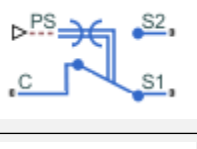
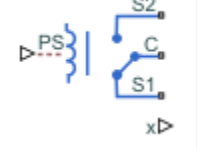
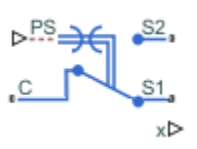
## Limitations and Assumptions

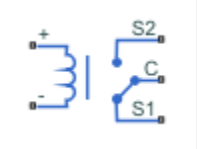
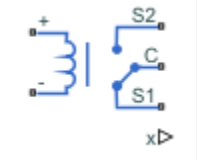
- For behavioral faults, if **time to fail when exceeding the current** threshold is greater than the time between switching events, no fault is triggered because the accumulated heat is not adequate for melting or breaking the contacts or windings.
- The energize and de-energize delays can differ, but the energize delay must be greater than or equal to the de-energize delay.

## Ports

The type, visibility, and location of the block ports depend on how you configure these parameters in the **Main** settings:

- **Control port** — Choose between a physical signal input port, **PS**, or electrical conserving ports, + and -, for relay control.
- **State port** — Set the visibility for the relay state physical signal output port, **x**.
- **Common port** — Set the location of the common port, **C**, relative to the contact ports **S1** and **S2**.

Control port	State port	Common port	Block
PS	Hidden	Adjacent to switch ports	
		Adjacent to switch ports	
	Visible	Across from switch ports	
		Across from switch ports	

Control port	State port	Common port	Block
Electrical	Hidden	Not applicable	
	Visible	Not applicable	

### Input

#### PS – Physical signal control

physical signal

Physical signal input port that energizes and de-energizes the relay.

#### Dependencies

This port is visible only when, in the **Main** settings, the **Control port** parameter is set to PS.

### Output

#### x – Relay state

physical signal

State of the relay in terms of the **C-S1** and **C-S2** connections:

- 1 – Closed connection
- 0 – Open connection

#### Dependencies

This port is visible only when, in the **Main** settings, the **State port** parameter is set to Visible.

### Conserving

#### S1 – Normally closed contact

electrical

Electrical conserving port associated with the normally closed contact.

#### C – Common contact

electrical

Electrical conserving port associated with the common contact.

#### S2 – Normally open contact

electrical

Electrical conserving port associated with the normally open contact.

**+ – Winding positive terminal**`electrical`

Electrical conserving port associated with the winding positive voltage terminal. The current through the winding controls the relay state.

**Dependencies**

This port is visible only when, in the **Main** settings, the **Control port** parameter is set to `Electrical`.

**- – Winding negative terminal**`electrical`

Electrical conserving port associated with the winding negative voltage terminal. The current through the winding controls the relay state.

**Dependencies**

This port is visible only when, in the **Main** settings, the **Control port** parameter is set to `Electrical`.

**Parameters****Main****Control port – Input port choice**`PS (default) | Electrical`

Relay state control method:

- `PS` — **PS** physical signal input port
- `Electrical` — `+` and `-` electrical conserving ports associated with the relay winding terminals

**Dependencies**

If this parameter is set to:

- `PS` — The **PS** port and related parameters in the **Main** and **Faults** settings are visible. For more information, see Control Port Parameter Dependencies.
- `Electrical` — The `+` and `-` ports, related parameters in the **Main** and **Faults** settings, and the **Windings** settings are visible. For more information, see Control Port Parameter Dependencies, Faults Parameter Dependencies, and “Winding” on page 1-2017.

The table shows how this parameter affects the visibility of other parameters in the **Main** settings. To learn how to read the table, see “Parameter Dependencies” on page A-2.

## Control Port Parameter Dependencies

Control port	
PS	Electrical
<b>Closed resistance</b>	<b>Closed resistance</b>
<b>Open conductance</b>	<b>Open conductance</b>
<b>Threshold</b>	
<b>State port</b>	<b>State port</b>
<b>Common port</b>	

### Closed resistance – Connected contact resistance

0.01 Ohm (default) | positive scalar

Resistance across closed relay contacts. The parameter value must be greater than zero.

### Open conductance – Open-circuit conductance

1e-8 1/Ohm (default) | positive scalar

Conductance across open relay contacts. The parameter value must be greater than zero.

### Threshold – Relay threshold

0 (default) | nonnegative scalar

If the physical signal input is above the threshold value, the relay is energized. Conversely, if the physical signal input falls is the threshold value, the relay is de-energized.

#### Dependencies

This parameter is visible only if the **Control port** parameter is set to PS.

### State port – Relay state output port visibility

Hidden (default) | Visible

Visibility of the physical signal port that outputs the relay state. The port outputs a vector of length two, with the first element corresponding to the **C-S1** connection and the second to the **C-S2** connection. The elements are 1 if the corresponding connection is closed, and 0 otherwise.

#### Dependencies

The **x** port is visible only if this parameter is set to Visible.

### Common port – Common port location

Adjacent to switch ports (default) | Across from switch ports

Location of the common port, **C**, relative to the **S1** and **S2** ports.

#### Dependencies

This parameter is visible only if the **Control port** parameter is set to PS.

### Winding

The **Winding** settings are visible only if, in the **Main** settings, the **Control port** parameter is set to Electrical. For more information, see “Control port” on page 1-0 .

**Rated voltage — Rated voltage**

5 V (default) | positive scalar

Standard voltage applied to the operating coil under normal operating conditions. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Percent rated voltage to energize — Percent rated voltage to energize**

75 (default) | positive scalar

Minimum percent of the rated voltage required to energize the relay. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Percent rated voltage to de-energize — Percent rated voltage to de-energize**

25 (default) | positive scalar

Percent of the rated voltage required to de-energize the relay. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding inductance — Winding inductance**

0.08 H (default) | positive scalar

Winding inductance.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding series resistance — Winding series resistance**

12.5 0hm (default) | positive scalar

Resistance between the winding terminals. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding parallel conductance — Winding parallel conductance**

1e-9 1/0hm (default) | positive scalar

Winding conductance.



## Dependencies

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

## Mechanical

### Time-to-break C-S1 connection — Delay time for breaking connection C-S1

0 s (default) | nonnegative scalar

Mechanical switching time for breaking the connection between ports **C** and **S1** when the relay is energized.

### Time-to-make C-S1 connection — Delay time for making connection C-S1

0 s (default) | nonnegative scalar

Mechanical switching time for making the connection between ports **C** and **S1** when the relay is de-energized.

### Time-to-break C-S2 connection — Delay time for breaking connection C-S2

0 s (default) | nonnegative scalar

Mechanical switching time for breaking the connection between ports **C** and **S2** when the relay is energized.

### Time-to-make C-S2 connection — Delay time for making connection C-S2

0 s (default) | nonnegative scalar

Mechanical switching time for making the connection between ports **C** and **S2** when the relay is de-energized.

## Faults

The table shows how the specified options for certain parameters in the **Faults** settings affect the visibility of other parameters in the **Faults** settings. To learn how to read the table, see “Parameter Dependencies” on page A-2.

---

**Note** The **Failure mode** parameter is visible only if, in the **Main** settings, the **Control Port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

---

**Faults Parameter Dependencies**

<b>Faults</b>																				
<b>Enable faults</b>																				
No	Yes																			
	<b>Reporting when a fault occurs</b>																			
	<b>Failure mode</b>																			
	Switch fault														Winding failed open circuit					
	<b>Switch fault</b>																			
	C-S1 stuck closed				C-S2 stuck closed				C open circuit (no path to S1 or S2)				Degraded contact resistance							
													Degraded contact resistance C-S1							
													Degraded contact resistance C-S2							
									Time constant for degraded contact resistance				Time constant for winding open circuit transition							
	<b>Enable temporal fault trigger</b>				<b>Enable temporal fault trigger</b>				<b>Enable temporal fault trigger</b>				<b>Enable temporal fault trigger</b>				<b>Enable temporal fault trigger</b>			
	No		Yes		No		Yes		No		Yes		No		Yes		No		Yes	
			Simulation time after which contactors can stick				Simulation time after which contactors can stick				Simulation time after which C can be open circuit				Simulation time for fault event				Simulation time for fault event	
	<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>	
	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes

Faults																	
		Maximum permissible load current		Maximum permissible load current		Maximum permissible load current		Maximum permissible load current		Maximum permissible load current		Maximum permissible load current		Maximum permissible load current		Maximum permissible winding voltage	Maximum permissible winding voltage
																Number of events to fail when exceeding voltage	Number of events to fail when exceeding voltage
																Maximum permissible winding current	Maximum permissible winding current

Faults																		
		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current

**Enable faults – Whether to model faults**

No (default) | Yes

Whether to model faults.

**Dependencies**

If this parameter is set to Yes, related parameters are visible. For more information, see Faults Parameter Dependencies.

**Reporting when a fault occurs – Reporting when a fault occurs**

None (default) | Warn | Error

Simulation reporting when a fault occurs:

- None — Does not generate a warning or an error.
- Warn — Generates a warning.
- Error — The simulation stops and generates an error.

**Dependencies**

This parameter is visible only if the **Enable faults** parameter is set to Yes. For more information, see Faults Parameter Dependencies.

**Failure mode – Fault model**

Winding failed open circuit (default) | Switch fault

Failure mode.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical** and, in the **Faults** settings, the **Enable faults** parameter is set to Yes. For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

The specified option for this parameter affects the visibility of other parameters in the **Faults** settings. For more information, see Faults Parameter Dependencies.

**Switch fault – Switch fault model**

C-S1 stuck closed (default) | C-S2 stuck closed | C open circuit (no path to S1 or S2) | Degraded contact resistance

Switch fault model.

#### Dependencies

This parameter is visible only if either of these conditions are met:

- In the **Main** settings, the **Control port** parameter is set to PS and, in the **Faults** settings, the **Enable faults** parameter is set to Yes.
- In the **Main** settings, the **Control port** parameter is set to Electrical, and in the **Faults** settings, the **Enable faults** parameter is set to Yes and the **Failure mode** parameter is set to Switch fault.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

#### Degraded contact resistance C-S1 — Degraded contact resistance C-S1

1e-9 Ohm (default) | positive scalar

Degraded resistance of the faulted **S1** contact.

#### Dependencies

This parameter is visible only if either one of these two sets of conditions is met:

##### Condition Set One

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.

##### Condition Set Two

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

#### Degraded contact resistance C-S2 — Degraded contact resistance C-S2

1e-9 Ohm (default) | positive scalar

Degraded resistance of the faulted **S2** contact.

#### Dependencies

This parameter is visible only if either one of these two sets of conditions is met:

##### Condition Set One

- In the **Main** settings, the **Control port** parameter is set to PS.

- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.

#### **Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

#### **Time constant for degraded contact resistance – Time constant for degraded contact resistance**

1e-3 (default) | positive scalar

Time constant for contact resistance to fall into degraded status. The degraded contact resistances for **S1** and **S2** have the same time constant.

#### **Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

#### **Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.

#### **Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

#### **Time constant for winding open circuit transition – Time constant for winding open circuit transition**

1e-3 s (default) | positive scalar

Time constant for the winding open circuit transition.

### Dependencies

This parameter is visible only if all of these conditions are met:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Enable temporal fault trigger — Whether to enable a temporal fault trigger

No (default) | Yes

Whether to enable a temporal fault trigger.

### Dependencies

This parameter is visible only if the **Enable faults** parameter is set to Yes.

If this parameter is set to Yes, related parameters become visible.

For more information, see Faults Parameter Dependencies.

### Simulation time after which contactors can stick — Simulation time after which the S1 or S2 contact can stick closed

1 s (default) | positive scalar

Simulation time after which the **S1** or **S2** contact can stick closed for a C-S1 stuck closed or C-S2 stuck closed fault.

### Dependencies

This parameter is visible only if either one of these two sets of conditions is met:

#### Condition Set One

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to either C-S1 stuck closed or C-S2 stuck closed.
  - The **Enable temporal fault trigger** parameter is set to Yes.

#### Condition Set Two

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to either C-S1 stuck closed or C-S2 stuck closed.

- The **Enable temporal fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### **Simulation time after which C can be open circuit – Simulation time after which the relay can be faulted to the open- circuit state**

1 s (default) | positive scalar

Simulation time after which, for the C open circuit (no path to S1 or S2) fault, the relay can be faulted to the open- circuit state.

#### **Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

#### **Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to C open circuit (no path to S1 or S2).
  - The **Enable temporal fault trigger** parameter is set to Yes.

#### **Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to C open circuit (no path to S1 or S2).
  - The **Enable temporal fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### **Simulation time for fault event – Simulation time for winding fault or degraded resistance fault**

1 s (default) | nonnegative scalar

Simulation time after which a fault can occur for either a Degraded contact resistance or a Winding failed open circuit fault.

#### **Dependencies**

This parameter is visible only if either one of these three sets of conditions is met:

#### **Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.



- The **Switch fault** parameter is set to Degraded contact resistance.
- The **Enable temporal fault trigger** parameter is set to Yes.

### Condition Set Two

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.
  - The **Enable temporal fault trigger** parameter is set to Yes.

### Condition Set Three

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable temporal fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Enable behavioral fault trigger — Whether to enable a behavioral fault trigger

No (default) | Yes

Whether to enable a behavioral fault trigger.

#### Dependencies

This parameter is visible only if the **Enable faults** parameter is set to Yes.

If this parameter is set to Yes, related parameters become visible.

For more information, see Faults Parameter Dependencies.

### Maximum permissible load current — Maximum permissible load current

inf A (default) | positive scalar

Upper load-current threshold for degraded contact resistance and open-circuit winding faults.

#### Dependencies

This parameter is visible only if either one of these two sets of conditions is met:

### Condition Set One

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Maximum permissible winding voltage – Maximum permissible winding voltage**

inf V (default) | positive scalar

Maximum voltage for the relay winding, above which a fault may be triggered.

**Dependencies**

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Number of events to fail when exceeding voltage – Number of events to fail when exceeding voltage**

1 (default) | positive scalar integer

Number of times the relay must exceed the winding voltage threshold before a Winding failed open circuit fault is triggered.

**Dependencies**

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Maximum permissible winding current – Maximum permissible winding current**

inf A (default) | positive scalar

Winding current threshold for triggering a behavioral Winding failed open circuit fault.

### Dependencies

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to **Electrical**.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to **Yes**.
  - The **Failure mode** parameter is set to **Winding failed open circuit**.
  - The **Enable behavioral fault trigger** parameter is set to **Yes**.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Time to fail when exceeding current – Time to fail when exceeding current

1 s (default) | positive scalar

Amount of time that the load or winding current must continuously exceed the maximum permissible current before a behavioral fault is triggered.

### Dependencies

This parameter is visible only if both of these conditions are met:

- The **Enable faults** parameter is set to **Yes**.
- The **Enable behavioral fault trigger** parameter is set to **Yes**.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

## Version History

Introduced in R2019b

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

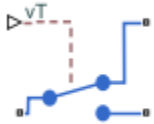
### See Also

SPST Relay | SPST Switch (Three-Phase) | SPDT Switch

## SPDT Switch

Single-pole double-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



### Description

The SPDT Switch block models a single-pole double-throw switch:

- When the switch is open, port **p** is connected to port **n1**.
- When the switch is closed, port **p** is connected to port **n2**.



Open connections are modeled by a resistor with value equal to the reciprocal of the **Open conductance** parameter value. Closed connections are modeled by a resistor with value equal to the **Closed resistance** parameter value.

If the **Threshold width** parameter is set to zero, the switch is closed if the voltage presented at the **vT** 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:

$$G = \frac{x}{R_{closed}} + (1 - x)G_{open}$$

$$\lambda = \frac{vT - \text{Threshold}}{\text{Threshold width}}$$

$$x = \begin{cases} 0 & \text{for } \lambda \leq 0 \\ 3\lambda^2 - 2\lambda^3 & \text{for } 0 < \lambda < 1 \\ 1 & \text{for } \lambda \geq 1 \end{cases}$$

The block uses the function  $3\lambda^2 - 2\lambda^3$  because its derivative is zero for  $\lambda = 0$  and  $\lambda = 1$ .

Defining a small positive **Threshold width** can help solver convergence in some models, particularly if the control port signal  $vT$  varies continuously as a function of other network variables. However, defining a nonzero threshold width precludes the solver making use of switched linear optimizations. Therefore, if the rest of your network is switched linear, set **Threshold width** to zero.

Optionally, you can add a delay between the point at which the voltage at **vT** passes the threshold and the switch opening or closing. To enable the delay, on the **Dynamics** tab, set the **Model dynamics** parameter to **Model turn-on and turn-off times**.

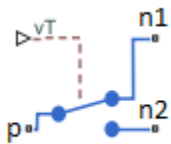
### Model Gate Control Port

You can control the switch gates by using a physical signal port or electrical conserving ports. To select the gate control port, set the **Modeling option** parameter to either:

- **PS control port** — The block contains a physical signal port that is associated with the threshold voltage.
- **Electrical control port** — The block contains electrical conserving ports that are associated with the threshold voltage.

## Ports

Refer to the figure for port locations.



### Input

#### **vT** — Switch control port

physical signal

Physical port associated with the signal that opens and closes the switch.

#### Dependencies

To enable this port, set **Modeling option** to **PS control port**.

### Conserving

#### **p** — Electrical port

electrical

Electrical conserving port.

#### **n1** — Electrical port

electrical

Electrical conserving port.

#### **n2** — Electrical port

electrical

Electrical conserving port.

**+ – Electrical port**

electrical

Electrical conserving port for the positive voltage associated with opening and closing the switch.

**Dependencies**

To enable this port, set **Modeling option** to Electrical control port.

**- – Electrical port**

electrical

Electrical conserving port for the negative voltage associated with opening and closing the switch.

**Dependencies**

To enable this port, set **Modeling option** to Electrical control port.

**Parameters****Modeling option – Whether to specify physical or electrical control port**

PS control port (default) | Electrical control port

Whether to specify physical or electrical control port for the switch gate.

**Main****Closed resistance – Closed resistance**

0.01 Ohm (default) | positive scalar

Resistance between the **p** port and ports **n1** and **n2** when the switch is closed. The value must be greater than zero.

**Open conductance – Open conductance**

1e-6 S (default) | positive scalar

Conductance between the **p** port and ports **n1** and **n2** when the switch is open. The value must be greater than zero.

**Threshold – Threshold voltage**

0.5 V (default)

The threshold voltage above which the switch will turn on.

**Threshold width – Threshold width**

0 V (default)

The minimum voltage increase above the threshold value that will move the switch from fully open to fully closed.

**Dynamics****Model dynamics – Model dynamics**

No dynamics (default) | Model turn-on and turn-off times

Select whether the block models a switching delay:

- **No dynamics** — Do not model the delay. This is the default option.
- **Model turn-on and turn-off times** — Use additional parameters to model a delay between the point at which the voltage at  $v_T$  or + and - passes the threshold and the switch opening or closing.

#### **Turn-on delay — Turn-on delay**

1e-3 (default) | positive scalar

Time between the input voltage exceeding the threshold voltage and the switch closing. The value must be greater than zero.

#### **Dependencies**

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

#### **Turn-off delay — Turn-off delay**

1e-3 (default) | positive scalar

Time between the input voltage falling below the threshold voltage and the switch opening. The value must be greater than zero.

#### **Dependencies**

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

#### **Initial input value, $v_T$ — Initial input value, $v_T$**

0 (default)

The value of the physical signal input  $v_T$  at time zero. This value is used to initialize the delayed control voltage parameter internally.

#### **Dependencies**

This parameter is visible only when you select **Model turn-on and turn-off times** for the **Model dynamics** parameter.

## **Version History**

**Introduced in R2012b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

DPDT Switch | DPST Switch | SPDT Switch (Three-Phase) | SPST Switch | SPST Switch (Three-Phase)

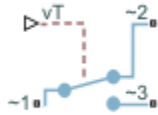
## **Topics**

“Switch Between Physical Signal and Electrical Ports”

## SPDT Switch (Three-Phase)

Three-phase single-pole double-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



### Description

The SPDT Switch (Three-Phase) block models a three-phase single-pole double-throw switch that uses an external signal to connect each phase of the port ~1 with the corresponding phase of either port ~2 or ~3.

The table shows how the external signal  $vT$  controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
$vT \leq \text{Threshold}$	Each phase of port ~1 is connected to the corresponding phase of port ~2 via internal resistance. Port ~3 is unconnected.	<b>Open conductance</b> (port ~1 to port ~3). <b>Closed resistance</b> (port ~1 to port ~2)
$vT > \text{Threshold}$	Each phase of port ~1 is connected to the corresponding phase of port ~3 via internal resistance. Port ~2 is unconnected.	<b>Open conductance</b> (port ~1 to port ~2). <b>Closed resistance</b> (port ~1 to port ~3)

### Ports

#### Input

##### $vT$ — Control port

physical signal

Physical control port.

#### Dependencies

To enable this port, set **Modeling option** to Expanded three-phase ports | PS control port or Composite three-phase ports | PS control port.

#### Conserving

##### ~1 — Expandable three-phase port

electrical

Expandable three-phase electrical conserving port.



**~2 — Expandable three-phase port**

electrical

Expandable three-phase electrical conserving port.

**~3 — Expandable three-phase port**

electrical

Expandable three-phase electrical conserving port.

**vT — Control port**

electrical

Electrical conserving control port.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports | Electrical control port or Composite three-phase ports | Electrical control port.

**Parameters****Modeling option — Connection and gate control ports**

Composite three-phase ports | PS control port (default) | Composite three-phase ports | Electrical control port | Expanded three-phase ports | PS control port | Expanded three-phase ports | Electrical control port

Whether to model composite or expanded three-phase ports and physical signal or electrical control ports.

**Closed resistance — Closed resistance**

0.001 Ohm (default)

Resistance between ports ~1 and ~3 when the switch is closed.

**Open conductance — Open conductance**

1e-6 1/Ohm (default)

Conductance between ports ~1 and ~2 when the switch is open.

**Threshold — Threshold**

0.5 V (default)

Threshold voltage for the control port vT. When the voltage is above the threshold, the switch is closed.

**Version History**

Introduced in R2013b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

[DPDT Switch](#) | [DPST Switch](#) | [SPDT Switch](#) | [SPST Switch](#) | [SPST Switch \(Three-Phase\)](#)

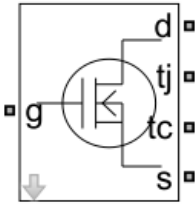
**Topics**

[“Expand and Collapse Three-Phase Ports on a Block”](#)  
[“Switch Between Physical Signal and Electrical Ports”](#)

## SPICE-Imported MOSFET

Predefined MOSFET parameterized by external SPICE subcircuit

**Library:** Simscape / Electrical / Semiconductors & Converters



### Description

The SPICE-Imported MOSFET block models a metal-oxide-semiconductor field-effect transistor (MOSFET) that has been parameterized using an external SPICE subcircuit.

You can use this block to model the MOSFET devices listed in the table. These parameterizations have already been imported to Simscape using the `subcircuit2ssc` function, and they have been numerically validated against results generated using a SPICE simulation tool.

**Available Parameterizations**

Series	Device
<b>Infineon OptiMOS6 40V</b>	IAUC120N04S6N009
	IAUC120N04S6N010
	IAUC120N04S6N013
	IAUC100N04S6N015
	IAUC100N04S6N022
	IAUC100N04S6N028
	IAUC80N04S6N036
	IAUC60N04S6N036
	IAUC120N04S6L008
	IAUC120N04S6L009
	IAUC120N04S6L012
	IAUC100N04S6L014
	IAUC100N04S6L020
	IAUC100N04S6L025
	IAUC80N04S6L032
	IAUC60N04S6L039
<b>Infineon OptiMOS5 80V</b>	IPB015N08N5
	IPB017N08N5
	IPP020N08N5
	IPT012N08N5
	IPB020N08N5
	IPP023N08N5
	IPB024N08N5
	IPP027N08N5
	IPB031N08N5
	IPP034N08N5
	IPB049N08N5
	IPP052N08N5
	IAUT300N08S5N012
	IAUT165N08S5N029
	IAUT300N08S5N014
	IAUT240N08S5N019
	IAUT200N08S5N023
	IAUS300N08S5N013T
IAUS300N08S5N015T	

Series	Device
	IAUC100N08S5N031
	IAUC100N08S5N043
	IAUC70N08S5N074
<b>Infineon OptiMOS4 60V</b>	IPB180N06S4_H1
	IPB120N06S4_H1
	IPI120N06S4_H1
	IPP120N06S4_H1
	IPB120N06S4_02
	IPI120N06S4_02
	IPP120N06S4_02
	IPB120N06S4_03
	IPI120N06S4_03
	IPP120N06S4_03
	IPB90N06S4_04
	IPD90N06S4_04
	IPD100N06S4_03
	IPI90N06S4_04
	IPP90N06S4_04
	IPB80N06S4_05
	IPD90N06S4_05
	IPI80N06S4_05
	IPP80N06S4_05
	IPB80N06S4_07
	IPD90N06S4_07
	IPI80N06S4_07
	IPP80N06S4_07
	IPB45N06S4_09
	IPD50N06S4_09
	IPI45N06S4_09
	IPP45N06S4_09
	IPG20N06S4_15
	IPB90N06S4L_04
	IPD90N06S4L_03
	IPI90N06S4L_04
	IPP90N06S4L_04
	IPB80N06S4L_05

Series	Device
	IPD90N06S4L_05
	IPI80N06S4L_05
	IPP80N06S4L_05
	IPB80N06S4L_07
	IPD90N06S4L_06
	IPI80N06S4L_07
	IPP80N06S4L_07
	IPB45N06S4L_08
	IPD50N06S4L_08
	IPI45N06S4L_08
	IPP45N06S4L_08
	IPD50N06S4L_12
	IPD30N06S4L_23
	IPD25N06S4L_30
	IPG20N06S4L_11
	IPG20N06S4L_14
	IPG20N06S4L_26
<b>Infineon OptiMOST 120V</b>	IPB100N12S3_05
	IPI100N12S3_05
	IPP100N12S3_05
	IPB70N12S3_11
	IPI70N12S3_11
	IPP70N12S3_11
	IPD70N12S3_11
	IPD70N12S3L_12
	IPB70N12S3L_12
	IPI70N12S3L_12
	IPP70N12S3L_12
	IPB50N12S3L_15
	IPD50N12S3L_15
	IPI50N12S3L_15
	IPP50N12S3L_15
	IPB35N12S3L_26
	IPD35N12S3L_24
	IPD30N12S3L_31

## Limitations

- This block works with local solver and variable-step solver. To enable the local solver, select the **Use local solver** check box in the Solver Configuration block. You must set the **Sample time** parameter to a value smaller than the fastest device dynamic. If you specify a larger sample time, the block steps over the dynamic events, which results in numerical warnings.
- Copying this block from the library into a model breaks the link to the library. If you upgrade MATLAB to later versions, you need to replace this block with a new copy of the block from the library.

## Ports

### Conserving

#### **g** – Gate terminal

electrical

Electrical conserving port associated with the gate terminal of the chosen pre-parameterized MOSFET.

#### **d** – Drain terminal

electrical

Electrical conserving port associated with the drain terminal of the chosen pre-parameterized MOSFET.

#### **tj** – Junction temperature

electrical

Electrical conserving port associated with the junction temperature of the chosen pre-parameterized MOSFET.

#### **tc** – Case temperature

electrical

Electrical conserving port associated with the case temperature of the chosen pre-parameterized MOSFET.

#### **s** – Source terminal

electrical

Electrical conserving port associated with the source of the chosen pre-parameterized MOSFET.

## Parameters

### **Series** – MOSFET series

Infineon OptiMOS6 40V (default) | Infineon OptiMOS5 80V | Infineon OptiMOS4 60V | Infineon OptiMOST 120V

Series of the pre-parameterized MOSFET, specified as either Infineon OptiMOS6 40V, Infineon OptiMOS5 80V, Infineon OptiMOS4 60V, or Infineon OptiMOST 120V.

**Device — Device name**

IPB015N08N5 | IAUC120N04S6N009 | IAUC120N04S6N010 | IAUC120N04S6N013 |  
IAUC100N04S6N015 | ...

Name of the device, specified as one of the options listed in the Available Parameterizations table.

## **Version History**

**Introduced in R2020b**

### **See Also**

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“List of Pre-Parameterized Components”



# SPICE Diode

SPICE-compatible diode

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



## Description

The SPICE Diode block represents a SPICE-compatible diode.

SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

## Equations

Variables for the SPICE Diode block equations include:

- Variables that you define by specifying parameters for the SPICE Diode block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2048.
- Geometry-adjusted variables, which depend on several values that you specify using parameters for the SPICE Diode block. For more information, see “Geometry-Adjusted Variables” on page 1-2043.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE Diode block or by specifying parameters for both the SPICE Diode block and an Environment Parameters block. For more information, see “Diode Temperature” on page 1-2044.
- Temperature-dependent variables. For more information, see “Temperature Dependence” on page 1-2046.
- Minimal conductance,  $GMIN$ , which is  $1e-12$  1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2044.
- Thermal voltage,  $V_t$ . For more information, see “Thermal Voltage” on page 1-2044.

## Geometry-Adjusted Variables

Several variables in the equations for the SPICE diode model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on variables that you define by specifying SPICE Diode block parameters. The geometry-adjusted variables depend on these variables:

- $AREA$  — Area of the device
- $SCALE$  — Number of parallel connected devices

- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$CJO_d$	Geometry-adjusted zero-bias junction capacitance	$CJO_d = CJO * AREA * SCALE$
$IBV_d$	Geometry-adjusted reverse breakdown current	$IBV_d = IBV * AREA * SCALE$
$IS_d$	Geometry-adjusted saturation current	$IS_d = IS * AREA * SCALE$
$RS_d$	Geometry-adjusted series resistance	$RS_d = \frac{RS}{AREA * SCALE}$

### Diode Temperature

You can use these options to define diode temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent from the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the Spice Diode block is set to **Fixed temperature**. For this model, the block sets  $T$  equal to *TFIXED*.
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the Spice Diode block is set to **Device temperature**. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is no Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **Spice** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- *TOFFSET* is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of  $1e-12$  1/Ohm. To specify a different value:

- 1 If there is not an Environment Parameters block in the diode circuit, add one.
- 2 In the **Spice** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Thermal Voltage

Thermal voltage,  $V_t$ , is defined by the equation

$$V_t = N \frac{k * T}{q}$$

Where:

- $N$  is the emission coefficient.
- $T$  is the diode temperature. For more information, see “Diode Temperature” on page 1-2044.
- $k$  is the Boltzmann constant.
- $q$  is the elementary charge on an electron.

### Current-Voltage Equations

These equations define the relationship between the diode current,  $I_d$ , and the diode voltage,  $V_d$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2046.

$$I_d = AREA * (I_{fwd} - I_{rev})$$

$$I_{fwd} = I_{nrm} * K_{inj} + I_{rec} * K_{gen}$$

$$I_{rev} = I_{revh} + I_{revl}$$

$$I_{nrm} = I_S e^{V_d / (N * V_t) - 1}$$

$$I_{rec} = I_{SR} e^{V_d / (NR * V_t) - 1}$$

$$K_{inj} = \left( \frac{IKF}{IKF + I_{nrm}} \right)^{0.5}$$

$$K_{gen} = \left[ \left( \frac{1 - V_d}{V_J} \right)^2 + 0.005 \right]^{\frac{M}{2}}$$

$$I_{revh} = IBV * e^{-\frac{V_d + BV}{NBV * V_t}}$$

$$I_{revl} = IBVL * e^{-\frac{V_d + BV}{NBVL * V_t}}$$

Where:

- $I_{fwd}$  is the forward current.
- $I_{rev}$  is the reverse current.
- $I_{nrm}$  is the normal current.
- $I_{rec}$  is the recombination current.
- $K_{inj}$  is the high-injection factor.
- $K_{gen}$  is the generation factor.
- $I_{revh}$  is the high-level breakdown current.
- $I_{revl}$  is the low-level breakdown current.
- $V_t$  is thermal voltage. For more information, see “Thermal Voltage” on page 1-2044.
- $I_S$  is the saturation current.
- $I_{SR}$  is the recombination current.

- $IKF$  is the forward knee current.
- $VJ$  is the junction potential.
- $N$  is the emission coefficient.
- $NR$  is the reverse emission coefficient.
- $NBV$  is the reverse breakdown emission coefficient.
- $NBVL$  is the low-level reverse breakdown ideality factor.
- $M$  is the grading coefficient.
- $BV$  is the reverse breakdown voltage.
- $IBV$  is the reverse breakdown current.
- $IBVL$  is the low-level reverse breakdown knee current.

### Junction Charge Model

The table shows the equations that define the relationship between the diode charge  $Q_d$ , and the diode voltage,  $V_d$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2046.

$V_d$ Range	$Q_d$ Equation
$V_d < FC * VJ$	$Q_d = TT * AREA * I_{fwd} + CJO_d * VJ * \frac{1 - \left(1 - \frac{V_d}{VJ}\right)^{1 - M}}{1 - M}$
$V_d \geq FC * VJ$	$Q_d = TT * AREA * I_{fwd} + CJO_d * \left( F1 + \frac{F3 * (V_d - FC * VJ) + \left(\frac{M}{2 * VJ}\right) * (V_d^2 - (FC * VJ)^2)}{F2} \right)$

Where:

- $FC$  is the forward bias depletion capacitance coefficient.
- $VJ$  is the junction potential.
- $TT$  is the transit time.
- $CJO_d$  is the geometry-adjusted zero-bias junction capacitance. For more information, see “Geometry-Adjusted Variables” on page 1-2043.
- $M$  is the grading coefficient.
- $F1 = VJ * (1 - (1 - FC)^{(1 - M)}) / (1 - M)$
- $F2 = (1 - FC)^{(1 + M)}$
- $F3 = 1 - FC * (1 + M)$

### Temperature Dependence

The relationship between the geometry-adjusted saturation current and the diode temperature is

$$IS_d(T) = IS_d * (T/TMEAS)^{\frac{XTI}{N}} * e^{\left(\frac{T}{TMEAS} - 1\right) * \frac{EG}{N * V_t}}$$

Where:

- $IS_d$  is the geometry-adjusted saturation current. For more information, see “Geometry-Adjusted Variables” on page 1-2043.
- $T$  is the diode temperature. For more information, see “Diode Temperature” on page 1-2044.
- $TMEAS$  is the parameter extraction temperature.
- $XTI$  is the saturation current temperature exponent.
- $N$  is the emission coefficient.
- $EG$  is the activation energy.
- $V_t$  is thermal voltage. For more information, see “Thermal Voltage” on page 1-2044.

The relationship between the recombination current and the diode temperature is

$$ISR(T) = ISR * \left( \frac{T}{TMEAS} \right)^{\frac{XTI}{NR}} * e^{\left( \frac{T}{TMEAS} - 1 \right) * \frac{EG}{NR * V_t}}$$

Where:

- $ISR$  is the recombination current.
- $NR$  is the reverse emission coefficient.

The relationship between the forward knee current and the diode temperature is

$$IKF(T) = IKF * [1 + TIKF * (T - TMEAS)]$$

Where:

- $IKF$  is the forward knee current.
- $TIKF$  is the linear IKF temperature coefficient.

The relationship between the breakdown voltage and the diode temperature is

$$BV(T) = BV * [1 + TBV1 * (T - TMEAS) + TBV2 * (T - TMEAS)^2]$$

Where:

- $BV$  is the breakdown voltage.
- $TBV1$  is the linear BV temperature coefficient.
- $TBV2$  is the quadratic BV temperature coefficient.

The relationship between the ohmic resistance and the diode temperature is

$$RS(T) = RS * [1 + TRS1 * (T - TMEAS) + TRS2 * (T - TMEAS)^2]$$

Where:

- $RS$  is the ohmic resistance.
- $TRS1$  is the linear RS temperature coefficient.
- $TRS2$  is the quadratic RS temperature coefficient.

The relationship between the junction potential and the diode temperature is

$$VJ(T) = VJ * \left( \frac{T}{TMEAS} \right) - 3 * V_t * \ln \left( \frac{T}{TMEAS} \right) - \left( \frac{T}{TMEAS} \right) * EG_{TMEAS} + EG_T$$

Where:

- $V_J$  is the junction potential.
- $EG_{TMEAS}$  is the activation energy for the temperature at which the diode parameters were measured. The defining equation is  $EG_{TMEAS} = 1.16eV - (7.02e - 4 * TMEAS^2) / (TMEAS + 1108)$ .
- $EG_T$  is the activation energy for the diode temperature. The defining equation is  $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$ .

The relationship between the geometry-adjusted diode zero-bias junction capacitance and the diode temperature is

$$CJO_d(T) = CJO_d * \left[ 1 + M * \left( 400e - 6 * (T - TMEAS) - \frac{V_J(T) - V_J}{V_J} \right) \right]$$

Where:

- $CJO_d$  is the geometry-adjusted zero-bias junction capacitance. For more information, see “Geometry-Adjusted Variables” on page 1-2043.
- $M$  is the grading coefficient.

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### + – Positive voltage

electrical

Electrical conserving port associated with positive voltage.

#### - – Negative voltage

electrical

Electrical conserving port associated with negative voltage.

## Parameters

### Main

#### Device area, AREA – Device area

1.0 m<sup>2</sup> (default) | positive scalar

Diode area. The value must be greater than 0.

#### Number of parallel devices, SCALE – Number of parallel devices

1 (default) | positive scalar

Number of parallel diodes that the block represents. The value must be greater than 0.

**Saturation current, IS — Saturation current**

1e-14 A/m<sup>2</sup> (default) | nonnegative scalar

Magnitude of the current that the ideal diode equation approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

**High-injection knee current, IKF — High-injection knee current**

Inf A/m<sup>2</sup> (default) | scalar

Current value at which forward-beta high-current roll-off occurs. The value must be greater than or equal to 0.

**Recombination current parameter, ISR — Recombination current**

0 A/m<sup>2</sup> (default) | scalar

Magnitude of the current generated from the process of the recombination of electrons and holes inside the junction.

**Emission coefficient, N — Emission coefficient**

1 (default) | positive scalar

Diode emission coefficient or ideality factor. The value must be greater than 0.

**Emission coefficient for ISR, NR — Emission coefficient for ISR**

2 (default) | positive scalar

Diode emission coefficient for the recombination current. The value must be greater than 0.

**Grading coefficient, M — Grading coefficient**

0.5 (default) |  $0 < M < 0.9$  | scalar

Grading coefficient,  $M$ . The value must be greater than 0 and less than 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Junction potential, VJ — Junction potential**

1 V (default) |  $VJ > 0.01$  | scalar

Junction potential,  $VJ$ . The value must be greater than 0.01 V.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Ohmic resistance, RS — Series resistance**

0.01 m<sup>2</sup>\*Ohm (default) | nonnegative scalar

Series diode connection resistance. The value must be greater than or equal to 0.

**Junction Capacitance**

**Model junction capacitance — Junction capacitance model**

No (default) | Yes

Options for modeling the junction capacitance:

- **No** — Do not include junction capacitance in the model.
- **Yes** — Specify zero-bias junction capacitance, junction potential, grading coefficient, forward-bias depletion capacitance coefficient, and transit time.

**Zero-bias junction capacitance, CJO — Zero-bias junction capacitance**

$0 \text{ F/m}^2$  (default) | nonnegative scalar

Value of the capacitance placed in parallel with the exponential diode term. The value must be greater than or equal to  $0$ .

**Dependencies**

To enable this parameter, set **Model junction capacitance** to Yes.

**Capacitance coefficient, FC — Capacitance coefficient**

$0.5$  (default) |  $0 \leq FC < 0.95$  | scalar

Fitting coefficient,  $FC$ , that quantifies the decrease of the depletion capacitance with applied voltage. The value must be greater than or equal to  $0$  and less than  $0.95$ .

**Dependencies**

To enable this parameter, set **Model junction capacitance** to Yes.

**Transit time, TT — Transit time**

$0 \text{ s}$  (default) | nonnegative scalar

Transit time,  $TT$ , of the carriers that cause diffusion capacitance. The value must be greater than or equal to  $0$ .

**Dependencies**

To enable this parameter, set **Model junction capacitance** to Yes.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- **No** — Do not specify an initial condition for the model.
- **Yes** — Specify the initial diode voltage.

---

**Note** The SPICE Diode block applies the initial diode voltage across the junction capacitors and not across the ports.

---

**Dependencies**

To enable this parameter, set **Model junction capacitance** to Yes.

**Initial condition voltage, V0 — Initial voltage**

$0 \text{ V}$  (default) | scalar

Diode voltage at the start of the simulation.



---

**Note** The block applies the initial condition across the diode junction, so the initial condition is only effective when charge storage is included, that is, when one or both of the **Zero-bias junction capacitance, CJO** and **Transit time, TT** parameters are greater than zero.

---

### Dependencies

To enable this parameter, set **Model junction capacitance** and **Specify initial condition** to Yes.

### Reverse Breakdown

#### Model reverse breakdown — Reverse breakdown model

No (default) | Yes

Options for modeling reverse breakdown:

- No — Do not model reverse breakdown.
- Yes — Introduce a second exponential term to the diode I-V relationship, thereby modeling a rapid increase in conductance as the breakdown voltage is exceeded.

#### Reverse breakdown voltage, BV — Reverse breakdown threshold voltage

Inf V (default) | nonnegative scalar

If voltage drops below this value, the block models the rapid increase in conductance that occurs at diode breakdown. The value must be greater than or equal to 0.

### Dependencies

To enable this parameter, set **Model reverse breakdown** to Yes.

#### Reverse breakdown current, IBV — Reverse breakdown current

1e-10 A/m<sup>2</sup> (default) | positive scalar

Diode current that corresponds to the voltage specified for the **Reverse breakdown voltage, BV** parameter. The value must be greater than 0.

### Dependencies

To enable this parameter, set **Model reverse breakdown** to Yes.

#### Low-level reverse breakdown knee current, IBVL — Low-level reverse breakdown knee current

0 A/m<sup>2</sup> (default) | positive scalar

Low-level reverse breakdown knee current.

### Dependencies

To enable this parameter, set **Model reverse breakdown** to Yes.

#### Reverse breakdown ideality factor, NBV — Reverse breakdown ideality factor

1 (default) | positive scalar

Ideality factor for the **Reverse breakdown voltage, BV**.

### Dependencies

To enable this parameter, set **Model reverse breakdown** to Yes.

**Low-level reverse breakdown ideality factor, NBVL — Low-level reverse breakdown ideality factor**

1 (default) | positive scalar

Ideality factor for the **Low-level reverse breakdown knee current, IBVL****Dependencies**To enable this parameter, set **Model reverse breakdown** to Yes.**Temperature****Model temperature dependence using — Temperature dependence model**

Device temperature (default) | Fixed temperature

Select one of these options for modeling the diode temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Diode Temperature” on page 1-2044.

**Saturation current temperature exponent, XTI — Saturation current temperature exponent**

3.0 (default) | positive scalar

Order of the exponential increase in the saturation current as temperature increases. The value must be greater than 0.

**Activation energy, EG — Activation energy**1.11 eV (default) |  $EG \geq 0.1$  | scalar

Diode activation energy. The value must be greater than or equal to 0.1 eV.

**IKF temperature coefficient (linear), TIKF — Temperature coefficient for IKF**

0 1/K (default) | positive scalar

Linear temperature coefficient for the **High-injection knee current, IKF**.**RS temperature coefficient (linear), TRS1 — Temperature coefficient for RS**

0 1/K (default) | positive scalar

Linear temperature coefficient for the **Ohmic resistance, RS**.**RS temperature coefficient (quadratic), TRS2 — Temperature coefficient for RS**0 1/K<sup>2</sup> (default) | positive scalarQuadratic temperature coefficient for the **Ohmic resistance, RS**.**BV temperature coefficient (linear), TBV1 — Temperature coefficient for BV**

0 1/K (default) | positive scalar

Linear temperature coefficient for the **Breakdown voltage, BV**.

**BV temperature coefficient (quadratic), TBV2 — Temperature coefficient for BV**  
 0 1/K<sup>2</sup> (default) | positive scalar

Quadratic temperature coefficient for the **Breakdown voltage, BV**.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**  
 300.15 K (default) | positive scalar

Diode simulation temperature. The value must be greater than 0 K.

#### Dependencies

To enable this parameter, set **Model temperature dependence using** to Fixed temperature.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**  
 300.15 K (default) | positive scalar

Temperature at which the diode parameters are measured. The value must be greater than 0 K.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**  
 0 K (default) | scalar

Amount by which the diode temperature differs from the circuit temperature.

#### Dependencies

To enable this parameter, set **Model temperature dependence using** to Device temperature.

## Version History

Introduced in R2008a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | Diode

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## SPICE NIGBT

SPICE-compatible N-Channel insulated gate bipolar transistor

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors

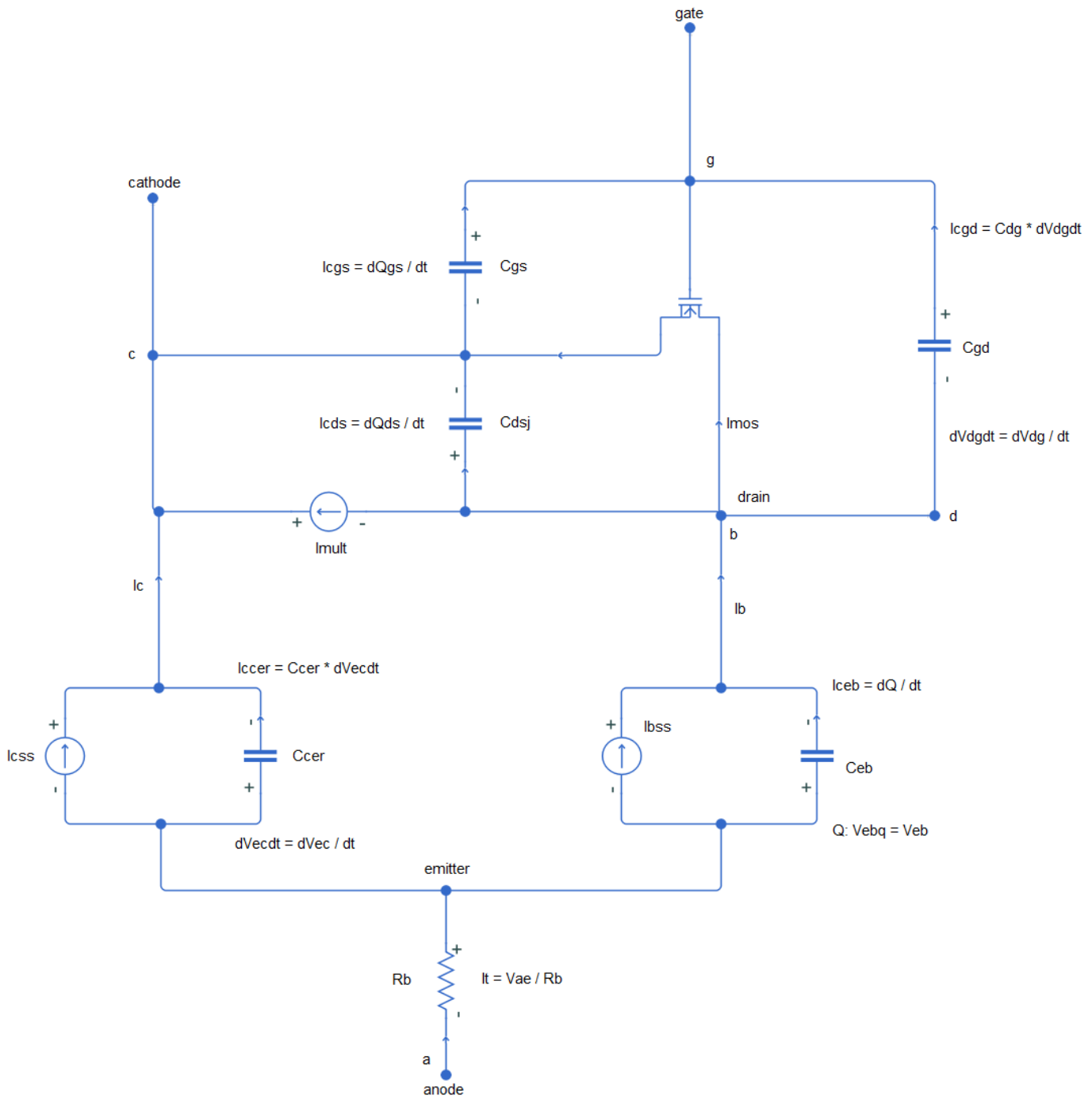


### Description

The SPICE NIGBT block models a SPICE n-type insulated gate bipolar transistor (IGBT).

SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

This figure shows the equivalent circuit for the SPICE NIGBT block:



### Equations

Variables for the SPICE NIGBT block equations include:

- Variables that you define by specifying parameters for the SPICE NIGBT block.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE NIGBT block or by specifying parameters for both the SPICE NIGBT

block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2060.

**MOSFET Channel Current**

This table shows the equations that define the relationship between the MOSFET channel current,  $I_{mos}$ , and the gate-source voltage,  $V_{gs}$ .

Applicable Range of $V_{gs}$ Values	Corresponding $I_{mos}$ Equation
$V_{gs} < VT$	$I_{mos} = GMIN * V_{ds} * SCALE$
$V_{ds} \leq \frac{V_{gs} - VT}{KF}$	$I_{mos} = SCALE * \left( \frac{KF * KP \left[ (V_{gs} - VT) V_{ds} - \frac{KF * V_{ds}^2}{2} \right]}{1 + THETA * (V_{gs} - VT)} + GMIN * V_{ds} \right)$
$V_{ds} > \frac{(V_{gs} - VT)}{KF}$	$I_{mos} = SCALE * \left( \frac{KP(V_{gs} - VT)^2}{2[1 + THETA * (V_{gs} - VT)]} + GMIN * V_{ds} \right)$

In these equations:

- $V_{ds}$  is the drain-source voltage.
- $VT$  is the threshold voltage.
- $KF$  is the triode region factor.
- $KP$  is the mosfet transconductance.
- $THETA$  is the transverse field factor.

**Bipolar Steady-State Collector Current**

This table shows the equations that define the relationship between the steady-state collector current,  $I_{css}$ , and the emitter-base capacitance,  $Q_{eb}$ .

Applicable Range of $Q_{eb}$ Values	Corresponding $I_{css}$ Equation
$Q_{eb} < 0$	$I_{css} = 0$
$Q_{eb} \geq 0$	$I_{css} = \left[ \left( \frac{1}{1 + b} \right) I_T + \left( \frac{b}{1 + b} \right) \left( \frac{4D_p}{W^2} \right) Q_{eb} \right] * SCALE$

In these equations:

- $b = \frac{MUN}{MUP}$  is the ambipolar mobility ratio.
- $D_p = \frac{K_B T}{q} * MUP$  is the diffusion coefficient for holes.
- $W = WB - W_{bcj}$  is the quasi-neutral base width, where:
  - $WB$  is the metallurgical base width.

- $W_{bcj} = \sqrt{2\epsilon_{si} \frac{V_{bc} + V_{bi}}{q * NB}}$  is the base-collector depletion width.
- $V_{bc}$  is the base-collector voltage.
- $V_{bi}$  is the build-in voltage, and it is equal to 0.6 V

**Bipolar Steady-State Base Current**

This table shows the equations that define the relationship between the steady-state base current,  $I_{bss}$ , and the emitter-base capacitance,  $Q_{eb}$ .

Applicable Range of $Q_{eb}$ Values	Corresponding $I_{bss}$ Equation
$Q_{eb} < 0$	$I_{bss} = 0$
$Q_{eb} \geq 0$	$I_{bss} = \frac{Q_{eb}}{TAU} + \left(\frac{Q_{eb}}{Q_B}\right)^2 \left(\frac{4 * NB^2}{ni^2}\right) JSNE * AREA * SCALE$

In these equations:

- $TAU$  is the ambipolar recombination lifetime.
- $JSNE$  is the emitter saturation current density.
- $ni$  is the intrinsic carrier concentration. At 300 K it is equal to  $1.45 * 10^{10}$  1/cm<sup>3</sup>.
- $Q_B = qWN_B * AREA$  is the background mobile carrier base charge, where:
  - $N_B$  is the base doping.
  - $AREA$  is the area of the device.

**Bipolar Emitter-Base Voltage**

This table shows the equations that define the relationship between the emitter-base voltage,  $V_{eb}$ , and the emitter-base capacitance,  $Q_{eb}$ .

Applicable Range of $Q_{eb}$ Values	Corresponding $V_{eb}$ Equation
$Q_{eb} < 0$	$V_{eb} = V_{ebj}$
$Q_{bi} > Q_{eb} \geq 0$	$V_{ebmin} = \min(V_{ebj}, V_{ebd})$
$Q_{eb} > Q_{bi}$	$V_{eb} = V_{ebd}$

In these equations:

- $V_{ebj} = V_{bi} - \frac{(Q_{eb} - Q_{bi})^2}{2qNB\epsilon_{si}A^2}$  is the emitter-base depletion voltage.
- $V_{bi}$  is the build-in voltage.
- $Q_{bi} = AREA * \sqrt{2\epsilon_{si}qNB * V_{bi}}$  is the emitter-base junction build-in voltage.
- $V_{ebd} = \frac{kT}{q} \ln \left[ \left( \frac{P_0}{ni^2} + \frac{1}{NB} \right) (NB + P_0) \right] - \frac{D_C}{\mu_{nc}} \ln \frac{P_0 + NB}{NB}$  is the emitter-base diffusion voltage.

**Anode Current**

The anode current is obtained from this equation:

$$I_T = \frac{V_{ae}}{R_b} * SCALE,$$

where:

- $V_{ae}$  is the applied anode-emitter voltage.
- $R_b$  is the conductivity-modulated base resistance.

This table shows the equations that define the relationship between the conductivity-modulated base resistance,  $R_b$ , and the emitter-base capacitance,  $Q_{eb}$ .

Applicable Range of $Q_{eb}$ Values	Corresponding $R_b$ Equation
$Q_{eb} < 0$	$R_b = \frac{W}{q * MUN * AREA * NB}$
$Q_{eb} \geq 0$	$R_b = \frac{W}{q * \mu_{eff} * AREA * n_{eff}}$

In these equations:

- $\mu_{eff}$  is the effective carrier mobility.
- $n_{eff}$  is the effective base doping concentration.
- $MUN$  is the electron mobility.

$\mu_{eff}$  and  $n_{eff}$  are obtained using these equations:

$$\mu_{nc} = \frac{1}{\left(\frac{1}{\mu_n} + \frac{1}{\mu_c}\right)}$$

$$\mu_{pc} = \frac{1}{\left(\frac{1}{\mu_p} + \frac{1}{\mu_c}\right)}$$

$$\mu_{eff} = \mu_{nc} + \frac{\mu_{pc} Q_{eb}}{(Q_{eb} + Q_B)}$$

$$D_c = 2 \frac{kT}{q} \frac{\mu_{nc} \mu_{pc}}{\mu_{nc} + \mu_{pc}}$$

$$L = \sqrt{D_c * TAU}$$

$$P_0 = \frac{Q}{q * AREA * L \tanh \frac{W}{2L}}$$

$$n_{eff} = \frac{\frac{W}{2L} \sqrt{N_B^2 + P_0^2 \operatorname{csch} \left( \frac{W}{L} \right)}}{\operatorname{arctanh} \left[ \frac{\sqrt{N_B^2 + P_0^2 \operatorname{csch} \left( \frac{W}{L} \right)} \tanh \left( \frac{W}{2L} \right)}{N_B + P_0 \operatorname{csch} \left( \frac{W}{L} \right) \tanh \left( \frac{W}{2L} \right)} \right]}$$

$$\bar{\delta}_p = \frac{P_0 \sinh \left( \frac{W}{2L} \right)}{\sinh \left( \frac{W}{L} \right)}$$

where:



- $\mu_{nc}$  is the electron carrier scattering mobility.
- $\mu_{pc}$  is the hole carrier scattering mobility.
- $D_c$  is the carrier-carrier scattering diffusivity.
- $L$  is the ambipolar diffusion length.
- $P_0$  is the carrier concentration at emitter's end of the base.
- $\bar{\delta}_p$  is the average carrier concentration in base.

### Avalanche Multiplication Current

The avalanche multiplication current is obtained from this equation:

$$I_{mult} = (M - 1)(I_{mos} + I_{css} + I_{ccer}) + M * I_{gen},$$

where:

- $I_{gen} = \frac{SCALE}{TAU} q n_i AREA \sqrt{2 \epsilon_{si} \frac{V_{bc}}{q N_B}}$  is the collector-base thermally generated current.
- $I_{css}$  is the steady-state collector current.
- $I_{mos}$  is the MOSFET channel current.
- $I_{ccer}$  is the collector-emitter redistribution current.

This equation defines the relationship between the base-collector voltage,  $V_{bc}$ , and the avalanche multiplication factor,  $M$ :

$$M = \frac{1}{1 - \left( \frac{V_{bc}}{BV_{cbo}} \right)^{BVN}},$$

where:

- $BV_{cbo} = BVF * 5.34e13 * NB^{-0.75}$  is the open-base collector-emitter breakdown voltage.
- $BVF$  is the avalanche uniformity factor.
- $BVN$  is the avalanche multiplication exponent.

### Capacitance Model

The gate source capacitance is obtained from this equation:

$$Q_{gs} = CGS * V_{gs} * SCALE.$$

The drain source capacitance is obtained from this equation:

$$Q_{ds} = q(AREA - AGD) * NB * W_{dsj} * SCALE,$$

where  $W_{dsj} = W_{bcj}$  is the drain-source depletion width.

This table shows the equations that define the relationship between the gate-drain capacitance,  $Q_{dg}$ , and the drain-gate voltage,  $V_{dg}$

Applicable Range of $V_{dg}$ Values	Corresponding $Q_{dg}$ Equation
$V_{dg} + VTD \leq 0$	$Q_{dg} = C_{gdo} * V_{dg} * SCALE$
$V_{dg} + VTD > 0$	$Q_{dg} = \left[ \frac{qNB\epsilon_{si}AGD}{COXD} \left( \frac{COXD * W_{dgj}}{\epsilon_{si}} \log \left( 1 + \frac{COXD * W_{dgj}}{\epsilon_{si}} \right) \right) - C_{gdo} * VTD \right] * SCALE$

In these equations:

- $C_{gdo} = COXD * AGD$  is the gate-drain overlap oxide capacitance.
- $V_{dg}$  is the drain-gate voltage.
- $\epsilon_{si}$  is the permittivity of silicon.
- $W_{dgj} = \sqrt{2\epsilon_{si} \frac{(V_{dg} + VTD)}{qNB}}$  is the drain-gate overlap depletion width.
- $VTD$  is the **Gate-drain overlap depletion threshold, VTD**.
- $COXD$  is the **Gate-drain oxide capacitance per unit area, COXD**.
- $AGD$  is the **Gate-drain overlap area, AGD**.
- $NB$  is the **Base doping, NB**.

This equation shows the relationship between the collector-emitter redistribution current,  $I_{c_{cer}}$ , and the collector-emitter redistribution capacitance,  $C_{cer}$ :

$$I_{c_{cer}} = C_{cer} * \frac{dV_{ec}}{dt} * SCALE,$$

where  $V_{ec}$  is the emitter-collector voltage.

This table shows the equations that define the relationship between the collector-emitter redistribution capacitance,  $C_{cer}$ , and the emitter-base capacitance,  $Q_{eb}$ .

Applicable Range of $Q_{eb}$ Values	Corresponding $C_{cer}$ Equation
$Q_{eb} > 0$	$C_{cer} = \frac{Q_{eb}C_{bcj}}{3Q_B} + C_{min} * AREA$
$Q_{eb} \leq 0$	$C_{cer} = C_{min} * AREA$

In these equations:

- $C_{bcj}$  is the base-collector depletion capacitance.
- $Q_B = qWN_B * AREA$  is the background mobile carrier base charge.

The implicit emitter-base capacitor current is obtained from this equation:

$$I_{q_{eb}} = \frac{dQ_{eb}}{dt} * SCALE.$$

### Transistor Temperature

You can use these options to define transistor temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent of the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NIGBT block is set to Fixed temperature. For this model, the block sets  $T$  equal to  $T_{FIXED}$ .
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NIGBT block is set to Device temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

where:

- $T_C$  is the circuit temperature.

If there is not an Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- $TOFFSET$  is the offset local circuit temperature.

## Ports

Refer to the figure for port locations.



## Conserving

### gx — Gate terminal

electrical

Electrical conserving port associated with the IGBT gate terminal.

### cx — Collector terminal

electrical

Electrical conserving port associated with the IGBT collector terminal.

### ex — Emitter terminal

electrical

Electrical conserving port associated with the IGBT emitter terminal.

## Parameters

### Dimensions

#### Gate-drain overlap area, AGD — Gate-drain overlap area

5.0e-6 m<sup>2</sup> (default) | positive scalar

Gate-drain overlap area. The value must be greater than 0.

**Area of the device, AREA — Device area**

1.0e-5 m<sup>2</sup> (default) | positive scalar

Area of the device. The value must be greater than 0.

**Number of parallel devices, SCALE — Number of parallel devices**

1 (default) | positive integer

Number of parallel transistors the block represents. The value must be an integer greater than 0.

**General****Electron mobility, MUN — Mobility of electron**

1.5e3 cm<sup>2</sup>/s/V (default) | scalar

Mobility of the electrons.

**Hole mobility, MUP — Mobility of hole**

4.5e2 cm<sup>2</sup>/s/V (default) | scalar

Mobility of the hole.

**Base doping, NB — Doping of base**

2e14 1/cm<sup>3</sup> (default) | scalar

Doping of the base.

**MOSFET****Triode region factor, KF — Factor of triode region**

1 (default)

Factor of the triode region.

**MOS transconductance, KP — MOS transconductance**

0.38 A/V<sup>2</sup> (default) | scalar

The derivative of the drain current with respect to the gate voltage. The value must be greater than or equal to 0

**Threshold voltage, VT — Threshold voltage**

4.7 V (default) | scalar

Threshold voltage.

**Transverse field factor, THETA — Transverse field factor**

0.02 1/V (default) | scalar

Transverse field factor.

**BJT****Ambipolar recombination lifetime, TAU — Ambipolar recombination lifetime**

7.1e-6 s (default) | scalar

Ambipolar recombination lifetime.

**Metallurgical base width, WB — Width of metallurgical base**

9.0e-5 m (default) | scalar

Width of the metallurgical base.

**Emitter saturation current density, JSNE — Density of emitter saturation current**

6.5e-13 A/cm<sup>2</sup> (default) | scalar

Density of the emitter saturation current.

**Avalanche uniformity factor, BVF — Avalanche uniformity factor**

1 (default) | scalar

Avalanche uniformity factor.

**Avalanche multiplication exponent, BVN — Avalanche multiplication exponent**

4 (default) | scalar

Avalanche multiplication exponent.

**Capacitance**

**Gate-source capacitance per unit area, CGS — Gate-source capacitance per unit area**

1.24e-8 F/cm<sup>2</sup> (default) | scalar

Gate-source capacitance per unit area.

**Gate-drain oxide capacitance per unit area, COXD — Gate-drain oxide capacitance per unit area**

3.5e-8 F/cm<sup>2</sup> (default) | scalar

Gate-drain oxide capacitance per unit area.

**Gate-drain overlap depletion threshold, VTD — Gate-drain overlap depletion threshold**

1e-3 V (default) | scalar

Gate-drain overlap depletion threshold.

**Specify initial condition — Whether to specify initial condition**

No (default) | Yes

Whether to specify initial condition.

**Initial condition voltage ICVGE — Initial condition voltage ICVGE**

0 V (default) | scalar

Initial condition voltage ICVGE.

**Dependencies**

To enable this parameter, set **Specify initial condition** to Yes.

**Initial condition voltage ICVCE — Initial condition voltage ICVCE**

0 V (default) | scalar

Initial condition voltage ICVCE.

**Dependencies**

To enable this parameter, set **Specify initial condition** to Yes.

**Temperature****Model temperature dependence using — Temperature dependence model**

Device temperature (default) | Fixed temperature

Options for modeling the transistor temperature dependence:

- Device temperature — Use the device temperature to model temperature dependence.
- Fixed temperature — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Temperature Dependence” on page 1-2093.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0 K.

**Dependencies**

To enable this parameter, set **Model temperature dependence using** to Fixed temperature.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

Temperature at which the transistor parameters are measured. The value must be greater than 0 K.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

To enable this parameter, set **Model temperature dependence using** to Device temperature.

## Version History

Introduced in R2020a

## References

- [1] Hefner, A.R. and Diebolt, D.M. *An experimentally verified IGBT model implemented in the Saber circuit simulator*. IEEE transactions on Power Electronics 9, no. 5 (September 1994): 532-42. <https://doi.org/10.1109/63.321038>.
- [2] Hefner, A.R., Jr. *Semiconductor measurement technology: INSTANT - IGBT Network Simulation and Transient ANALYSIS Tool*. U.S. Department of Commerce/Technology Administration, National Institute of Standards and Technology. 1992.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## SPICE NJFET

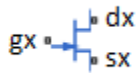
SPICE-compatible N-Channel JFET

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



### Description

The SPICE NJFET block represents a SPICE-compatible N-channel junction field-effect transistor (NJFET). If the voltage applied to the gate port, **gx**, is less than the voltage applied to the source port, **sx**, the current between the source port and drain port, **dx**, is reduced.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

### Equations

Variables for the SPICE NJFET block equations include:

- Variables that you define by specifying parameters for the SPICE NJFET block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2071.
- Geometry-adjusted variables, which depend on several values that you specify using parameters for the SPICE NJFET block. For more information, see “Geometry-Adjusted Variables” on page 1-2066.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE NJFET block or by specifying parameters for both the SPICE NJFET block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2067.
- Temperature-dependent variables. For more information, see “Temperature Dependence” on page 1-2070.
- Minimal conductance,  $GMIN$ , which is 1e-12 1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2068.

### Geometry-Adjusted Variables

Several variables in the equations for the N-channel junction field-effect transistor model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on



variables that you define by specifying SPICE NJFET block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$BETA_d$	Geometry-adjusted transconductance	$BETA_d = BETA * AREA * SCALE$
$CGD_d$	Geometry-adjusted zero-bias gate-drain capacitance	$CGD_d = CGD * AREA * SCALE$
$CGS_d$	Geometry-adjusted zero-bias gate-source capacitance	$CGS_d = CGS * AREA * SCALE$
$IS_d$	Geometry-adjusted saturation current	$IS_d = IS * AREA * SCALE$
$RS_d$	Geometry-adjusted source resistance	$RS_d = \frac{RS}{AREA * SCALE}$
$RD_d$	Geometry-adjusted drain resistance	$RD_d = \frac{RD}{AREA * SCALE}$

### Transistor Temperature

You can use these options to define transistor temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent of the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NJFET block is set to `Fixed` temperature. For this model, the block sets  $T$  equal to *TFIXED*.
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NJFET block is set to `Device` temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is not an Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- *TOFFSET* is the offset local circuit temperature.

**Minimal Conduction**

Minimal conductance, *GMIN*, has a default value of 1e-12 1/Ohm. To specify a different value:

- 1 If there is not an Environment Parameters block in the transistor circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired *GMIN* value for the **GMIN** parameter.

**Gate-Source Current-Voltage Model**

This table shows the equations that define the relationship between the gate-source current, *I<sub>gs</sub>*, and the gate-source voltage, *V<sub>gs</sub>*. As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2046.

Applicable Range of <i>V<sub>gs</sub></i> Values	Corresponding <i>I<sub>gs</sub></i> Equation
$V_{gs} > 80 * V_t$	$I_{gs} = IS_d * \left( \left( \frac{V_{gs}}{V_t} - 79 \right) e^{80} - 1 \right) + V_{gs} * Gmin$
$80 * V_t \geq V_{gs}$	$I_{gs} = IS_d * \left( e^{V_{gs}/V_t} - 1 \right) + V_{gs} * Gmin$

Where:

- *IS<sub>d</sub>* is the geometry-adjusted saturation current.
- *V<sub>t</sub>* is the thermal voltage, such that  $V_t = ND * k * T / q$ .
- *ND* is the emission coefficient.
- *q* is the elementary charge on an electron.
- *k* is the Boltzmann constant.
- *T* is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2067
- *GMIN* is the transistor minimum conductance. or more information, see “Minimal Conduction” on page 1-2068

**Gate-Drain Current-Voltage Model**

This table shows the relationship between the gate-drain current, *I<sub>gd</sub>*, and the gate-drain voltage, *V<sub>gd</sub>*. As applicable, model parameters are first adjusted for temperature.

Applicable Range of <i>V<sub>gd</sub></i> Values	Corresponding <i>I<sub>gd</sub></i> Equation
$V_{gd} > 80 * V_t$	$I_{gd} = IS_d * \left( \left( \frac{V_{gd}}{V_t} - 79 \right) e^{80} - 1 \right) + V_{gd} * Gmin$
$80 * V_t \geq V_{gd}$	$I_{gd} = IS_d * \left( e^{V_{gd}/V_t} - 1 \right) + V_{gd} * Gmin$

**Drain-Source Current-Voltage Model**

This table shows the relationship between the drain-source current, *I<sub>ds</sub>*, and the drain-source voltage, *V<sub>ds</sub>*, in normal mode (*V<sub>ds</sub>* ≥ 0). As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{gs}$ and $V_{gd}$ Values	Corresponding $I_{ds}$ Equation
$V_{gs} - V_{to} \leq 0$	$I_{ds} = 0$
$0 < V_{gs} - V_{to} \leq V_{ds}$	$I_{ds} = \beta_d(V_{gs} - V_{to})^2(1 + \lambda V_{ds})$
$0 < V_{ds} < V_{gs} - V_{to}$	$I_{ds} = \beta_d V_{ds}(2(V_{gs} - V_{to}) - V_{ds})(1 + \lambda V_{ds})$

Where:

- $V_{to}$  is the threshold voltage.
- $\beta_d$  is the geometry-adjusted transconductance.
- $\lambda$  is the channel modulation.

This table shows the relationship between the drain-source current,  $I_{ds}$ , and the drain-source voltage,  $V_{ds}$ , in inverse mode ( $V_{ds} < 0$ ). As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{gs}$ and $V_{gd}$ Values	Corresponding $I_{ds}$ Equation
$V_{gd} - V_{to} \leq 0$	$I_{ds} = 0$
$0 < V_{gd} - V_{to} \leq -V_{ds}$	$I_{ds} = -\beta_d(V_{gd} - V_{to})^2(1 - \lambda V_{ds})$
$0 < -V_{ds} < V_{gd} - V_{to}$	$I_{ds} = \beta_d V_{ds}(2(V_{gd} - V_{to}) + V_{ds})(1 - \lambda V_{ds})$

### Junction Charge Model

This table shows the relationship between the gate-source charge,  $Q_{gs}$ , and the gate-source voltage,  $V_{gs}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{gs}$ Values	Corresponding $Q_{gs}$ Equation
$V_{gs} < FC * VJ$	$Q_{gs} = \frac{CGS_d * VJ * \left(1 - \left(1 - \frac{V_{gs}}{VJ}\right)^{1 - MG}\right)}{1 - MG}$
$V_{gs} \geq FC * VJ$	$Q_{gs} = CGS_d * \left( F1 + \frac{F3 * (V_{gs} - FC * VJ) + \frac{MG * (V_{gs}^2 - (FC * VJ)^2)}{2 * VJ}}{F2} \right)$

Where:

- $FC$  is the capacitance coefficient.
- $VJ$  is the junction potential.
- $CGS_d$  is the zero-bias gate-source capacitance.
- $MG$  is the grading coefficient.
- $F1 = \frac{VJ * (1 - (1 - FC)^{1 - MG})}{1 - MG}$

- $F2 = (1 - FC)^{1 + MG}$
- $F3 = 1 - FC * (1 + MG)$

This table shows the relationship between the gate-drain charge,  $Q_{gd}$ , and the gate-drain voltage,  $V_{gd}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{gd}$ Values	Corresponding $Q_{gd}$ Equation
$V_{gd} < FC * VJ$	$Q_{gd} = \frac{CGD_d * VJ * \left(1 - \left(1 - \frac{V_{gd}}{VJ}\right)^{1 - MG}\right)}{1 - MG}$
$V_{gd} \geq FC * VJ$	$Q_{gd} = CGD_d * \left( F1 + \frac{F3 * (V_{gd} - FC * VJ) + \frac{MG * (V_{gd}^2 - (FC * VJ)^2)}{2 * VJ}}{F2} \right)$

Where  $CGD_d$  is the geometry-adjusted zero-bias gate-drain capacitance.

#### Temperature Dependence

The block provides this relationship between the saturation current  $IS$  and the transistor temperature  $T$ :

$$IS(T) = IS_d * (T/T_{meas})^{\frac{XTI}{ND}} * e^{\left(\frac{T}{T_{meas}} - 1\right) * \frac{EG}{V_t}}$$

Where:

- $IS_d$  is the geometry-adjusted saturation current.
- $T_{meas}$  is the parameter extraction temperature.
- $XTI$  is the saturation current temperature exponent.
- $EG$  is the energy gap.
- $V_t$  is the thermal voltage, such that  $V_t = ND * k * T / q$ .
- $ND$  is the emission coefficient.

The relationship between the junction potential,  $VJ$ , and the transistor temperature  $T$  is

$$VJ(T) = VJ * \left(\frac{T}{T_{meas}}\right) - \frac{3 * k * T}{q} * \log\left(\frac{T}{T_{meas}}\right) - \left(\frac{T}{T_{meas}}\right) * EG_{T_{meas}} + EG_T$$

Where:

- $VJ$  is the junction potential.
- $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
- $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The relationship between the gate-source junction capacitance,  $CGS$ , and the transistor temperature,  $T$ :

$$CGS(T) = CGS_d * \left[ 1 + MG * \left( 400e - 6 * (T - T_{meas}) - \frac{VJ(T) - VJ}{VJ} \right) \right]$$

Where  $CGS_d$  is the geometry-adjusted zero-bias gate-source capacitance.

The block uses the  $CGS(T)$  equation to calculate the gate-drain junction capacitance by substituting  $CGD_d$ , the zero-bias gate-drain capacitance, for  $CGS_d$ .

The relationship between the transconductance,  $\beta$ , and the transistor temperature  $T$  is

$$\beta(T) = \beta_d * \left( \frac{T}{T_{meas}} \right)$$

Where  $\beta_d$  is the geometry-adjusted transconductance.

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### **gx – Gate terminal**

electrical

Electrical conserving port associated with the transistor gate terminal.

#### **dx – Drain terminal**

electrical

Electrical conserving port associated with the transistor drain terminal.

#### **sx – Source terminal**

electrical

Electrical conserving port associated with the transistor source terminal.

## Parameters

### Main

#### **Device area, AREA – Device area**

1.0 m<sup>2</sup> (default) | positive scalar

Transistor area. The value must be greater than 0.

#### **Number of parallel devices, SCALE – Number of parallel devices**

1 (default) | positive integer

Number of parallel transistors the block represents. The value must be an integer greater than 0.

#### **Threshold voltage, VT0 – Threshold voltage**

-2 V (default) | scalar

Gate-source voltage above which the transistor produces a nonzero drain current.

**Transconductance, BETA — Channel modulation**

$1e-4 \text{ A/m}^2/\text{V}^2$  (default) | nonnegative scalar

Derivative of drain current with respect to gate voltage. The value must be greater than or equal to 0.

**Channel modulation, LAMBDA — Channel modulation**

0 1/V (default) | scalar

Channel modulation.

**Saturation current, IS — Saturation current**

$1e-14 \text{ A/m}^2$  (default) | nonnegative scalar

Magnitude of the current that the gate-current equation approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

**Drain resistance, RD — Series resistance**

0.01  $\text{m}^2 \cdot \Omega$  (default) | nonnegative scalar

Transistor drain resistance. The value must be greater than or equal to 0.

**Source resistance, RS — Series resistance**

0.0001  $\text{m}^2 \cdot \Omega$  (default) | nonnegative scalar

Transistor source resistance. The value must be greater than or equal to 0.

**Emission coefficient, N — Emission coefficient**

1 (default) | positive scalar

Transistor emission coefficient or ideality factor. The value must be greater than 0.

**Junction Capacitance****Model junction capacitance — Junction capacitance model**

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model.
- Yes — Specify zero-bias junction capacitance, junction potential, grading coefficient, forward-bias depletion capacitance coefficient, and transit time.

**Dependencies**

Selecting Yes exposes related parameters.

**Zero-bias GS capacitance, CGS — Zero-bias junction capacitance**

0 F/ $\text{m}^2$  (default) | nonnegative scalar

Value of the capacitance placed between the gate and the source. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Zero-bias GD capacitance, CGD — Zero-bias junction capacitance**

$0 \text{ F/m}^2$  (default) | nonnegative scalar

Value of the capacitance placed between the gate and the drain. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Junction potential, VJ — Junction potential**

1 V (default) |  $VJ > 0.01$  | scalar

Junction potential,  $VJ$ . The value must be greater than 0.01 V.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Grading coefficient, M — Grading coefficient**

0.5 (default) |  $0 < M < 0.9$  | scalar

Grading coefficient,  $M$ . The value must be greater than 0 and less than 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Capacitance coefficient, FC — Capacitance coefficient**

0.5 (default) |  $0 \leq FC < 0.95$  | scalar

Fitting coefficient,  $FC$ , that quantifies the decrease of the depletion capacitance with applied voltage. The value must be greater than or equal to 0 and less than 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- No — Do not specify an initial condition for the model.
- Yes — Specify the initial transistor voltage.

---

**Note** The SPICE NJFET block applies the initial transistor voltage across the junction capacitors and not across the ports.

---

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

Selecting Yes for this parameter exposes the **Initial condition voltage, ICVDS** and **Initial condition voltage, ICVGS** parameters.

**Initial condition voltage, ICVDS — Initial voltage**

0 V (default) | scalar

Drain-source voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**Initial condition voltage, ICVGS — Initial voltage**

0 V (default) | scalar

Gate-source voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**Temperature****Model temperature dependence using — Temperature dependence model**

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Transistor Temperature” on page 1-2067.

**Dependencies**

Selecting **Device temperature** exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting **Fixed temperature** exposes the **Fixed circuit temperature, TFIXED** parameter.

**Saturation current temperature exponent, XTI — Saturation current temperature exponent**

0 (default) | positive scalar

Order of the exponential increase in the saturation current as temperature increases. The value must be greater than 0.

**Activation energy, EG — Activation energy**

1.11 eV (default) |  $EG \geq 0.1$  | scalar

Transistor activation energy. The value must be greater than or equal to 0.1.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0.



### Dependencies

This parameter is only visible when you select `Fixed` temperature for the **Model temperature dependence using** parameter.

### Parameter extraction temperature, TMEAS — Parameter extraction temperature

300.15 K (default) | positive scalar

Temperature at which the transistor parameters are measured. The value must be greater than 0.

### Offset local circuit temperature, TOFFSET — Local circuit temperature offset

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

### Dependencies

This parameter is only visible when you select `Device` temperature for the **Model temperature dependence using** parameter.

## Version History

Introduced in R2008a

## References

[1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | N-Channel JFET | SPICE PJFET

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

## SPICE NMOS

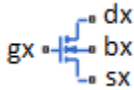
SPICE-compatible N-Channel MOSFET

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



### Description

The SPICE NMOS block represents a SPICE-compatible negative-channel (N-Channel) metal-oxide semiconductor (MOS) field-effect transistor (FET). If the gate-source voltage increases the channel conductance increases. If the gate-source voltage is decreased, the channel conductance decreases.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

### Equation Variables

Variables for the SPICE NMOS block equations include:

- Variables that you define by specifying parameters for the SPICE NMOS block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2097.
- Geometry-adjusted variables, which depend on several of the values that you specify using parameters for the SPICE NMOS block. For more information, see “Geometry-Adjusted Variables” on page 1-2076.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE NMOS block or by specifying parameters for both the SPICE NMOS block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2077.
- Minimal conductance,  $GMIN$ , which is 1e-12 1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2078.
- Thermal voltage,  $V_{tn}$ . For more information, see “Thermal Voltage” on page 1-2078.

### Geometry-Adjusted Variables

Several variables in the equations for the SPICE N-channel MOSFET model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on variables that you define by specifying SPICE NMOS block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$KP_d$	Geometry-adjusted transconductance	$KP_d = KP * AREA * SCALE$
$IS_d$	Geometry-adjusted bulk saturation current	$IS_d = IS * AREA * SCALE$
$JS_d$	Geometry-adjusted bulk junction saturation current density	$JS_d = JS * AREA * SCALE$
$CBD_d$	Geometry-adjusted zero-bias bulk-drain capacitance	$CBD_d = CBD * AREA * SCALE$
$CBS_d$	Geometry-adjusted zero-bias bulk-source capacitance	$CBS_d = CBS * AREA * SCALE$
$CGSO_d$	Geometry-adjusted gate-source overlap capacitance	$CGSO_d = CGSO * AREA * SCALE$
$CGDO_d$	Geometry-adjusted gate-drain overlap capacitance	$CGDO_d = CGDO * AREA * SCALE$
$CGBO_d$	Geometry-adjusted gate-bulk overlap capacitance	$CGBO_d = CGBO * AREA * SCALE$
$CJ$	Geometry-adjusted bottom capacitance per junction area	$CJ_d = CJ * AREA * SCALE$
$CJSW$	Geometry-adjusted sidewall capacitance per junction perimeter	$CJSW_d = CJSW * AREA * SCALE$
$RD_d$	Geometry-adjusted drain resistance	$RD_d = \frac{RD}{AREA * SCALE}$
$RS_d$	Geometry-adjusted source resistance	$RS_d = \frac{RS}{AREA * SCALE}$
$RSH_d$	Geometry-adjusted sheet resistance	$RSH_d = \frac{RSH}{AREA * SCALE}$

### Transistor Temperature

There are two different options for defining transistor temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent of the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NMOS block is set to Fixed temperature. For this model, the block sets  $T$  equal to *TFIXED*.
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NMOS block is set to Device temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is not an Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- $TOFFSET$  is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of  $1e-12$  1/0hm. To specify a different value:

- 1 If there is not already an Environment Parameters block in the circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Thermal Voltage

$V_{tn}$  is the thermal voltage, which is defined as

$$V_{tn} = N \frac{k * T}{q}$$

Where:

- $N$  is the emission coefficient.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2077.
- $k$  is the Boltzmann constant.
- $q$  is the elementary charge on an electron.

### Parameters Calculations

The tables show how the SPICE NMOS block determines some of its parameters based on values that you specify.

#### Drain Resistance

Parameter Values			Geometry-Adjusted Transistor Drain Resistance
Drain resistance, $RD$	Sheet resistance, $RSH$	Number of drain squares, $NRD$	
NaN	NaN	NaN	0
NaN	$RSH$	NaN	0
NaN	NaN	$NRD$	0
$RD$	NaN or $RSH$	NaN or $NRD$	$RD_d$
NaN	$RSH$	$NRD$	$RSH_d * NRD$

## Source Resistance

Parameter Values			Geometry-Adjusted Transistor Source Resistance
Source resistance, RS	Sheet resistance, RSH	Number of source squares, NRS	
NaN	NaN	NaN	0
NaN	RSH	NaN	0
NaN	NaN	NRS	0
RS	NaN or RSH	NaN or NRS	$RS_d$
NaN	RSH	NRS	$RSH_d * NRS$

## Transconductance and Surface Mobility

Parameter Values			Geometry-Adjusted Transconductance (level 1), in A/V <sup>2</sup>	Geometry-Adjusted Transconductance (level 3), in A/V <sup>2</sup>	Surface mobility (level 3), in cm <sup>2</sup> /s/V
Oxide thickness, TOX	Surface mobility, U0	Transconductance, KP			
NaN	NaN	NaN	2e-5 (default value)	2e-5 (default value)	600 (default value)
NaN	NaN	KP	$KP_d$	$KP_d$	600
NaN	U0	NaN	2e-5	$U0 * EPXox / 1e-7$	U0
NaN	U0	KP	$KP_d$	$KP_d$	U0
TOX	NaN	NaN	$600 * EPXox / TOX$	$600 * EPXox / TOX$	600
TOX	NaN	KP	$KP_d$	$KP_d$	600
TOX	U0	NaN	$U0 * EPXox / TOX$	$U0 * EPXox / TOX$	U0
TOX	U0	KP	$KP_d$	$KP_d$	U0

**Oxide Thickness and Threshold Voltage**

Parameter Values				Surface potential, PHI (level 1), in V	Threshold voltage, VTO (level 1), in V	Surface potential, PHI (level 3), in V	Threshold voltage, VTO (level 3), in V
Oxide thickness, TOX	Substrate doping, NSUB	Surface potential, PHI	Threshold voltage, VTO				
NaN	NaN	NaN	NaN	0.6 (default value)	0 (default value)	0.6 (default value)	0 (default value)
NaN	NaN	NaN	VTO	0.6	VTO	0.6	VTO
NaN	NaN	PHI	NaN	PHI	0	PHI	0
NaN	NaN	PHI	VTO	PHI	VTO	PHI	VTO
NaN	NSUB	NaN	NaN	0.6	0	PHI (1e-7, NSUB)	VTO (1e-7, NSUB)
NaN	NSUB	NaN	VTO	0.6	VTO	PHI (1e-7, NSUB)	VTO
NaN	NSUB	PHI	NaN	PHI	0	PHI	VTO (1e-7, NSUB)
NaN	NSUB	PHI	VTO	PHI	VTO	PHI	VTO
TOX	NaN	NaN	NaN	0.6	0	0.6	0
TOX	NaN	NaN	VTO	0.6	VTO	0.6	VTO
TOX	NaN	PHI	NaN	PHI	0	PHI	0
TOX	NaN	PHI	VTO	PHI	VTO	PHI	VTO
TOX	NSUB	NaN	NaN	PHI (NSUB, TOX)	VTO (NSUB, TOX)	PHI (NSUB, TOX)	VTO (NSUB, TOX)
TOX	NSUB	NaN	VTO	PHI (NSUB, TOX)	VTO	PHI (NSUB, TOX)	VTO
TOX	NSUB	PHI	NaN	PHI	VTO (NSUB, TOX)	PHI	VTO (NSUB, TOX)
TOX	NSUB	PHI	VTO	PHI	VTO	PHI	VTO

Where PHI (NSUB, TOX), PHI (1e-7, NSUB), VTO (NSUB, TOX), and VTO (1e-7, NSUB) are obtained using these equations:

$$PHI = 2 \frac{kT}{q} \ln \left( \frac{NSUB}{ni} \right)$$

$$GAMMA = \frac{\sqrt{2q\epsilon_{si}NSUB}}{C_{ox}}$$

$$V_{FB} = \varphi_{MS} - \frac{qNSS}{C_{ox}}$$

$$V_{T0} = V_{FB} + PHI + GAMMA * \sqrt{PHI}.$$

### Bulk-Source Diode Model

The table shows the equations that define the relationship between the bulk-source current,  $I_{bs}$ , and the bulk-source voltage,  $V_{bs}$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2093.

Applicable Range of $V_{bs}$ Values	Corresponding $I_{bs}$ Equation
$V_{bs} > 80 * V_{tn}$	$I_{bs} = IS_{bs} * \left( \left( \frac{V_{bs}}{V_{tn}} - 79 \right) e^{80} - 1 \right) + V_{bs} * G_{min}$
$80V_{tn} \geq V_{bs}$	$I_{bs} = IS_{bs} * \left( e^{V_{bs}/V_{tn}} - 1 \right) + V_{bs} * G_{min}$

Where:

- $IS_{bs}$  is the bulk saturation current, such that, if:
  - $JS_d \neq 0$  and  $AS \neq 0$ ,  $IS_{bs} = JS_d * AS$ .

Where:

- $JS_d$  is the geometry-adjusted bulk junction saturation current density.
- $AS$  is the source area.
- If  $JS_d = 0$  or  $AS = 0$ ,  $IS_{bs} = IS_d$ , where  $IS_d$  is the geometry-adjusted bulk saturation current.
- $V_{tn}$  is the thermal voltage. For more information, see “Thermal Voltage” on page 1-2078.
- $G_{min}$  is the minimal conductance. For more information, see “Minimal Conduction” on page 1-2078.

### Bulk-Drain Diode Model

The table shows the equations that define the relationship between the bulk-drain current  $I_{bd}$ , and the bulk-drain voltage,  $V_{bd}$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2093.

Applicable Range of $V_{bd}$ Values	Corresponding $I_{bd}$ Equation
$V_{bd} > 80 * V_{tn}$	$I_{bd} = IS_{bd} * \left( \left( \frac{V_{bd}}{V_{tn}} - 79 \right) e^{80} - 1 \right) + V_{bd} * G_{min}$
$80V_{tn} \geq V_{bd}$	$I_{bd} = IS_{bd} * \left( e^{V_{bd}/V_{tn}} - 1 \right) + V_{bd} * G_{min}$

Where:

- $IS_{bd}$  is the bulk drain current, such that:
  - If  $JS_d \neq 0$  and  $AD \neq 0$ ,  $IS_{bd} = JS_d * AD$ .

Where:

- $JS_d$  is the geometry-adjusted bulk junction saturation current density.
- $AD$  is the drain area.
- If  $JS_d = 0$  or  $AD = 0$ ,  $IS_{bd} = IS_d$ , where  $IS_d$  is the geometry-adjusted bulk saturation current.
- $V_{tn}$  is the thermal voltage. For more information, see “Thermal Voltage” on page 1-2078.
- $G_{min}$  is the minimal conductance. For more information, see “Minimal Conduction” on page 1-2078.

### Level 1 Drain Current Model

This table shows relationship between the drain current,  $I_d$ , and the drain-source voltage,  $V_{ds}$ , in normal mode ( $V_{ds} \geq 0$ ). As applicable, model parameters are first adjusted for temperature.

#### Normal Mode

Applicable Range of $V_{gs}$ and $V_{ds}$ Values	Corresponding $I_d$ Equation
$V_{gs} - V_{on} \leq 0$	$I_d = 0$
$0 < V_{gs} - V_{on} \leq V_{ds}$	$I_d = BETA * (V_{gs} - V_{on})^2 \frac{(1 + LAMBDA * V_{ds})}{2}$
$0 < V_{ds} < V_{gs} - V_{on}$	$I_d = BETA * V_{ds} \left( (V_{gs} - V_{on}) - \frac{V_{ds}}{2} \right) (1 + LAMBDA * V_{ds})$

Where:

- $V_{on}$  depends on  $V_{bs}$  and  $PHI$ .

Applicable Relationship of $V_{bs}$ and $PHI$ Values	Corresponding $V_{on}$ Equation
$V_{bs} \leq 0$	$V_{on} = MTYPE * VBI + GAMMA * \sqrt{PHI - V_{bs}}$
$0 < V_{bs} \leq 2 * PHI$	$V_{on} = MTYPE * VBI + GAMMA \left( \sqrt{PHI} - \frac{V_{bs}}{2\sqrt{PHI}} \right)$
$V_{bs} > 2 * PHI$	$V_{on} = MTYPE * VBI$

- $MTYPE$  is 1.
- $BETA$  is  $BETA = (KP_d * WIDTH) / (LENGTH - 2 * LD)$
- $KP$  is:
  - The **Transconductance, KP**, if this parameter has a numerical value.
  - $U0 * 3.9 * \epsilon_0 / TOX$ , if **Transconductance, KP** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $WIDTH$  is the channel width.
- $LENGTH$  is the channel length.
- $LD$  is the lateral diffusion.



- $V_{BI}$  is a built-in voltage value the block uses in calculations. The value is a function of temperature. For a detailed definition, see “Temperature Dependence” on page 1-2093.
- $PHI$  is:
  - The **Surface potential, PHI**, if this parameter has a numerical value.
  - $2 * kT_{meas} / q * \log(NSUB / n_i)$ , if **Surface potential, PHI** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $LAMBDA$  is the channel modulation.
- $GAMMA$  is:
  - The **Bulk threshold, GAMMA**, if this parameter has a numerical value.
  - $TOX * \sqrt{2 * 11.7 * \epsilon_0 * q * NSUB} / (3.9 * \epsilon_0)$ , if **Bulk threshold, GAMMA** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $\epsilon_0$  is the permittivity of free space,  $8.854214871e-12$  F/m.
- $n_i$  is the carrier concentration of intrinsic silicon,  $1.45e10$  cm<sup>-3</sup>.

This table shows relationship between the drain current  $I_d$  and the drain-source voltage  $V_{ds}$  in inverse mode ( $V_{ds} < 0$ ). As applicable, model parameters are first adjusted for temperature.

### Inverse Mode

Applicable Range of $V_{gd}$ and $V_{ds}$ Values	Corresponding $I_d$ Equation
$V_{gd} - V_{on} \leq 0$	$I_d = 0$
$0 < V_{gd} - V_{on} \leq -V_{ds}$	$I_d = -BETA(V_{gd} - V_{on})^2(1 - LAMBDA * V_{ds})/2$
$0 < -V_{ds} < V_{gd} - V_{on}$	$I_d = BETA * V_{ds}((V_{gd} - V_{on}) + V_{ds}/2)(1 - LAMBDA * V_{ds})$

$V_{on}$  depends on  $V_{bd}$  and  $PHI$ .

Applicable Relationship of $V_{bs}$ and $PHI$ Values	Corresponding $V_{on}$ Equation
$V_{bd} \leq 0$	$V_{on} = MTYPE * V_{BI} + GAMMA \sqrt{PHI - V_{bd}}$
$0 < V_{bd} \leq 2 * PHI$	$V_{on} = MTYPE * V_{BI} + GAMMA \left( \sqrt{PHI} - \frac{V_{bs}}{2\sqrt{PHI}} \right)$
$V_{bd} > 2 * PHI$	$V_{on} = MTYPE * PHI$

### Level 3 Drain Current Model

The block provides the following model for drain current  $I_{ds}$  in normal mode ( $V_{ds} \geq 0$ ) after adjusting the applicable model parameters for temperature.

$$I_{DS} = I_{DS0} * Scale_{VMAX} * Scale_{LChan} * Scale_{INV}$$

Where:

- $I_{DS0}$  is the Basic Drain Current Model on page 1-2084.
- $Scale_{VMAX}$  is the Velocity Saturation Scaling on page 1-2086.

- $Scale_{LChan}$  is the Channel Length Modulation Scaling on page 1-2086.
- $Scale_{INV}$  is the Weak Inversion Scaling on page 1-2087.

The block uses the same model for drain current in inverse mode ( $V_{ds} < 0$ ), with the following substitutions:

$$V_{bs} \equiv V_{bs} - V_{ds}$$

$$V_{gs} \equiv V_{gs} - V_{ds}$$

$$V_{ds} \equiv -V_{ds}$$

### Basic Drain Current Model

The relationship between the drain current,  $I_{ds}$ , and the drain-source voltage,  $V_{ds}$  is

$$I_{DS0} = BETA * F_{gate} * \left( V_{GSX} - V_{TH} - \frac{1 + F_B}{2} * V_{DSX} \right) * V_{DSX}$$

Where:

- $BETA$  is calculated as described in “Level 1 Drain Current Model” on page 1-2082.
- $F_{GATE}$  is calculated as

$$F_{gate} = \frac{1}{1 + THETA * (V_{gsx} - V_{TH})}$$

Where:

- $THETA$  models the dependence of the mobility on the gate-source voltage.
- $V_{gsx} = \max(V_{GS}, V_{on})$
- If you specify a nonzero value for the **Fast surface state density, NFS** parameter, the block calculates  $V_{on}$  using this equation:

$$V_{on} = V_{TH} + x_n V_T$$

Otherwise,

$$V_{on} = V_{TH}$$

- The block calculates  $x_n$  as

$$x_n = 1 + \frac{q * NFS}{COX} + \frac{\left( GAMMA * F_s * \sqrt{V_{bulk}} + \frac{F_n * V_{bulk}}{WIDTH} \right)}{2 * V_{bulk}}$$

- The block calculates  $V_{bulk}$  as follows:

- If

$$V_{BS} \leq 0,$$

$$V_{bulk} = PHI - V_{BS}.$$

- Otherwise, the block calculates  $V_{bulk}$  as

$$V_{bulk} = \frac{PHI}{\left(1 + \frac{V_{BS}}{2 * PHI}\right)^2}$$

- Thermal voltage such that

$$V_T = \frac{kT}{q}$$

- The block calculates  $V_{TH}$  using the following equation:

$$V_{TH} = V_{BI} - \frac{8.15e^{-22} * ETA}{COX * (LENGTH - 2 * LD)^3} * V_{DS} + GAMMA * F_s * \sqrt{V_{bulk}} + F_n * V_{bulk}$$

For information about how the block calculates  $V_{BI}$ , see “Temperature Dependence” on page 1-2093.

- $ETA$  is the **Vds dependence threshold volt, ETA**.

- $COX = \frac{\epsilon_{ox}}{TOX}$ ,

Where  $\epsilon_{ox}$  is the permittivity of the oxide and  $TOX$  is the **Oxide thickness, TOX**.

- If you specify a nonzero value for the **Junction depth, XJ** parameter and a value for the **Substrate doping, NSUB** parameter, the block calculates  $F_s$  using these equations:

$$\alpha = \frac{2\epsilon_{si}}{qNSUB}$$

$$XD = \sqrt{\alpha}$$

$$wc = .0631353 + .8013292 * \frac{XD * \sqrt{V_{bulk}}}{XJ} - .01110777 * \left(\frac{XD * \sqrt{V_{bulk}}}{XJ}\right)^2 + \frac{LD}{XJ}$$

$$F_s = 1 - \left( wc * \sqrt{1 - \left(\frac{XD * \sqrt{V_{bulk}}}{XJ + XD * \sqrt{V_{bulk}}}\right)^2} - \frac{LD}{XJ} \right)$$

Where  $\epsilon_{si}$  is the permittivity of silicon.

Otherwise,

$$F_s = 1$$

- The block calculates  $F_B$  as

$$F_B = \frac{GAMMA * F_s}{4 * \sqrt{V_{bulk}}} + F_n$$

- The block calculates  $F_n$  as

$$F_n = \frac{DELTA * \pi * \epsilon_{si}}{2 * COX * WIDTH}$$

- $DELTA$  is the width effect on threshold.

- $V_{DSX}$  is the lesser of  $V_{DS}$  and the saturation voltage,  $V_{dsat}$ .
- If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block calculates  $V_{dsat}$  using the following equation:

$$V_{dsat} = \frac{V_{gsx} - V_{TH}}{1 + F_B} + \frac{(LENGTH - 2 * LD) * VMAX}{UO * F_{gate}}$$

$$- \sqrt{\left(\frac{V_{gsx} - V_{TH}}{1 + F_B}\right)^2 + \left(\frac{(LENGTH - 2 * LD) * VMAX}{UO * F_{gate}}\right)^2}$$

Otherwise, the block calculates  $V_{dsat}$  as

$$V_{dsat} = \frac{V_{gsx} - V_{TH}}{1 + F_B}$$

### Velocity Saturation Scaling

If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block calculates  $Scale_{VMAX}$  as

$$Scale_{VMAX} = \frac{1}{1 + \frac{UO * F_{gate}}{(LENGTH - 2 * LD) * VMAX} * V_{DSX}}$$

Otherwise,

$$Scale_{VMAX} = 1$$

### Channel Length Modulation Scaling

The block scales the drain current to account for channel length modulation if the block meets all of the following criteria:

- $V_{DS} > V_{dsat}$
- The **Max carrier drift velocity, VMAX** is less than or equal to zero or  $\alpha$  is nonzero.

The block scales the drain current using the following equation:

$$Scale_{LChan} = \frac{1}{1 - \frac{\Delta l}{(LENGTH - 2 * LD)}}$$

To calculate  $\Delta l$  the block:

- 1 Calculates the intermediate value  $\Delta l_0$ .

- If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block computes the intermediate value  $g_{dsat}$  as the greater of 1e-12 and the result of the following equation:

$$I_{DS0} * \left(1 - \frac{1}{1 + Scale_{gdsat} * V_{DSX}}\right) * Scale_{gdsat}$$

Where:

$$Scale_{g_{dsat}} = \frac{UO * F_{gate}}{(LENGTH - 2 * LD) * VMAX}$$

Then, the block uses the following equation to calculate the intermediate value  $\Delta l_0$ :

$$\Delta l_0 = \sqrt{\left( \frac{KA * I_{DS}}{2 * (LENGTH - 2 * LD) * g_{dsat}} \right)^2 + KA * (V_{DS} - V_{dsat})} - \frac{KA * I_{DS}}{2 * (LENGTH - 2 * LD) * g_{dsat}}$$

Where

$$KA = KAPPA * \alpha .$$

- Otherwise, the block uses the following equation to calculate the intermediate value  $\Delta l_0$  as

$$\Delta l = \sqrt{KA * (V_{DS} - V_{dsat})}$$

- 2 The block checks for punch through and calculates  $\Delta l$ .

- If

$$\Delta l_0 > (LENGTH - 2 * LD) / 2,$$

the block calculates  $\Delta l$  using the following equation:

$$\Delta l = \left( 1 - \frac{(LENGTH - 2 * LD)}{4 * \Delta l_0} \right) * (LENGTH - 2 * LD)$$

- Otherwise,

$$\Delta l = \Delta l_0 .$$

### Weak Inversion Scaling

If  $V_{GS}$  is less than  $V_{on}$ , the block calculates  $Scale_{INV}$  using the following equation:

$$Scale_{INV} = e^{\frac{V_{gs} - V_{on}}{\alpha_n * V_T}}$$

Otherwise,

$$Scale_{INV} = 1$$

### Junction Charge Model

The block models “Junction Overlap Charges” on page 1-2087 and “Bulk Junction Charges” on page 1-2088.

#### Junction Overlap Charges

The block calculates the following junction overlap charges:

- $Q_{GS} = CGSO_d * WIDTH * V_{gs}$

Where:

- $Q_{GS}$  is the gate-source overlap charge.
- $CGSO_d$  is the geometry adjusted gate-source overlap capacitance.
- $WIDTH$  is the channel width.
- $Q_{GD} = CGDO_d * WIDTH * V_{gd}$

Where:

- $Q_{GD}$  is the gate-drain overlap charge.
- $CGDO_d$  is the geometry adjusted gate-drain overlap capacitance.
- $Q_{GB} = CGBO_d * (LENGTH - 2 * LD) * V_{gb}$

Where:

- $Q_{GB}$  is the gate-bulk overlap charge.
- $CGBO_d$  is the geometry adjusted gate-bulk overlap capacitance.
- $LENGTH$  is the channel length.
- $LD$  is the lateral diffusion.

### Bulk Junction Charges

This table shows relationship between the bulk-drain bottom junction charge  $Q_{bottom}$  and the junction voltage,  $V_{bd}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{bd}$ Values	Corresponding $Q_{bottom}$ Equation
$V_{bd} < FC * PB$	$Q_{bottom} = \frac{CBD_d * PB * \left(1 - \left(1 - \frac{V_{bd}}{PB}\right)^{1 - MJ}\right)}{1 - MJ} \text{ if } CBD_d > 0$ $Q_{bottom} = \frac{CJ_d * AD * PB * \left(1 - \left(1 - \frac{V_{bd}}{PB}\right)^{1 - MJ}\right)}{1 - MJ} \text{ otherwise.}$
$V_{bd} \geq FC * PB$	$Q_{bottom} = CBD_d * \left( F1 + \frac{F3 * (V_{bd} - FC * PB) + \frac{MJ * (V_{bd}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right) \text{ if } CBD_d > 0$ $Q_{bottom} = CJ_d * AD * \left( F1 + \frac{F3 * (V_{bd} - FC * PB) + \frac{MJ * (V_{bd}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right) \text{ otherwise.}$

Where:

- $PB$  is the bulk junction potential.

- $FC$  is the capacitance coefficient.
- $CBD_d$  is the geometry-adjusted zero-bias bulk-drain capacitance.
- $CJ_d$  is the geometry-adjusted bottom capacitance per junction area.
- $AD$  is the drain area.
- $MJ$  is the bottom grading coefficient.
- $$F1 = \frac{PB * (1 - (1 - FC)^{1 - MJ})}{1 - MJ}$$
- $$F2 = (1 - FC)^{1 + MJ}$$
- $$F3 = 1 - FC * (1 + MJ)$$

To calculate the bulk-source bottom junction charge, the block substitutes variables in the equations in the preceding table. The block substitutes:

- $V_{bs}$  for  $V_{bd}$
- $AS$  for  $AD$
- $CBS_d$  for  $CBD_d$

This table shows relationship between the bulk-drain sidewall junction charge  $Q_{sidewall}$  and the junction voltage  $V_{bd}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{bd}$ Values	Corresponding $Q_{sidewall}$ Equation
$V_{bd} < FC * PB$	$Q_{sidewall} = \frac{CJSW_d * PD * PB * \left(1 - \left(1 - \frac{V_{bd}}{PB}\right)^{1 - MGSW}\right)}{1 - MGSW}$
$V_{bd} \geq FC * PB$	$Q_{sidewall} = CJSW_d * PD * \left( F1 + \frac{F3 * (V_{bd} - FC * PB) + \frac{MGSW * (V_{bd}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right)$

Where:

- $CJSW_d$  is the geometry adjusted sidewall capacitance per junction perimeter.
- $PD$  is the drain perimeter.
- $MGSW$  is the side grading coefficient.
- $$F1 = \frac{PB * (1 - (1 - FC)^{1 - MJSW})}{1 - MJSW}$$
- $$F2 = (1 - FC)^{1 + MJSW}$$
- $$F3 = 1 - FC * (1 + MJSW)$$

To calculate the bulk-source sidewall junction charge and the sidewall junction voltage, the block substitutes variables in the equations in the preceding table. The block substitutes:

- $V_{bs}$  for  $V_{bd}$
- $PS$  for  $PD$

### Capacitance Model

The SPICE NMOS block allows you to model the transistor capacitance model in three different ways:

- No intrinsic capacitance model
- “Meyer Gate Capacitance Model” on page 1-2090
- “Charge Conservation Capacitance Model” on page 1-2091

### Meyer Gate Capacitance Model

This table shows the relationship between the operational regions of the transistor and the gate-bulk, gate-drain, and gate-source capacitances.

Operational region	Gate-Bulk, $C_{gb}$ , Gate-Drain, $C_{gd}$ , and Gate-Source, $C_{gs}$ , Equations	
Accumulation region, $V_{gb} < V_{FB}$	$C_{gb} = C_{oxt}$ $C_{gd} = 0$ $C_{gs} = 0$	
Depletion region, $V_{gs} < V_{TH}$	$C_{gb} = \frac{C_{oxt}}{\sqrt{1 + \frac{4}{\text{GAMMA}^2} * (V_{gb} - V_{FB})}}$ $C_{gd} = 0$ $C_{gs} = 0$	
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	if $V_{ds} < V_{satmin}$ then:	if $V_{ds} \geq V_{satmin}$ then:
	$C_{gb} = 0$ $C_{gd} = \frac{2}{3}(C_{oxt} - C_{gb}) \left( 1 - \frac{(V_{satmin})^2}{(2V_{satmin} - V_{ds})^2} \right)$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb}) \left( 1 - \frac{(V_{satmin} - V_{ds})^2}{(2V_{satmin} - V_{ds})^2} \right)$	$C_{gb} = 0$ $C_{gd} = 0$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb})$
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$C_{gb} = 0$ $C_{gd} = \frac{2}{3}(C_{oxt} - C_{gb}) * \left( 1 - \frac{(V_{gs} - V_{TH})^2}{(2 * (V_{gs} - V_{TH}) - V_{ds})^2} \right)$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb}) * \left( 1 - \frac{(V_{gs} - V_{TH} - V_{ds})^2}{(2 * (V_{gs} - V_{TH}) - V_{ds})^2} \right)$	

where:

- $C_{oxt} = \text{WIDTH} * (\text{LENGTH} - 2 * \text{LD}) * \text{COX} * \text{AREA} * \text{SCALE}$
- $V_{FB} = \text{VBI} * \text{MTYPE} - \text{PHI}$  is the flat-band voltage.
- $V_{satmin}$  is the minimum saturation voltage. It is a predefined parameter equal to 1 V.



These equations are continuous between the depletion region and the accumulation region, and discontinuous between the depletion and the inversion region. Other SPICE tools apply smoothing functions between the inversion and depletion regions.

$$C_{gb} = \frac{WIDTH * LENGTH * COX}{\sqrt{\left(1 + \frac{4}{GAMMA^2} * (V_{TH} - V_{bs} - V_{FB})\right)^m}} * smoothing$$

$$smoothing = \frac{1}{\left(1 + \frac{4}{GAMMA^2} * (V_{gs} - V_{TH})\right)^m}$$

where  $m$  is a predefined smoothing constant.

### Charge Conservation Capacitance Model

This table shows the relationship between the operational regions of the transistor and the gate, bulk, channel, drain, and source charges for a level 1 MOS.

Operational region	Level-1 Charges Equations
Accumulation region, $V_{gb} < V_{FB}$	$Q_g = Cox t * (V_{gb} - V_{FB})$ $Q_b = -Cox t * (V_{gb} - V_{FB})$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Depletion region, $V_{gs} < V_{TH}$	$Q_g = -0.5 * Cox t * GAMMA^2 * \left(1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})}\right)$ $Q_b = 0.5 * Cox t * GAMMA^2 * \left(1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})}\right)$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	$Q_g = Cox t * \left(V_{gs} - V_{FB} - PHI - \frac{V_{gs} - V_{TH}}{3}\right)$ $Q_b = -Cox t * (V_{TH} - V_{FB} - PHI)$ $Q_c = -Cox t * \left(V_{gs} - V_{TH} - \frac{V_{gs} - V_{TH}}{3}\right)$ $Q_d = 0$ $Q_s = Q_c - Q_d$

Operational region	Level-1 Charges Equations
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$Q_g = Cox_{t} * \left( V_{gs} - V_{FB} - PHI - \frac{V_{ds}}{2} + \frac{V_{ds}^2}{12(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_b = -Cox_{t} * (V_{TH} - V_{FB} - PHI)$ $Q_c = -Cox_{t} * \left( V_{gs} - V_{TH} - \frac{V_{ds}}{2} + \frac{V_{ds}^2}{12(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_d = -Cox_{t} * \left( \frac{V_{gs} - V_{TH}}{2} - \frac{3V_{ds}}{4} + \frac{V_{ds}^2}{8(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_s = Q_c - Q_d$

where:

- $Q_c = -(Q_g + Q_b) = Q_d + Q_s$  is the charge in channel.  $Q_c$  needs to be partitioned between  $Q_d$  and  $Q_s$ .

where:

- $Q_g$  is the gate charge.
- $Q_b$  is the bulk charge.
- $Q_d$  is the drain charge.
- $Q_s$  is the source charge.
- $Q_c = -(Q_g + Q_b) = Q_d + Q_s$  is the charge in channel.  $Q_c$  needs to be partitioned between  $Q_d$  and  $Q_s$ .

This table shows the relationship between the operational regions of the transistor and the gate, bulk, channel, drain, and source charges for a level-3 MOS.

Operational region	Level-3 Charges Equations
Accumulation region, $V_{gb} < V_{FB}$	$Q_g = Cox_{t} * (V_{gb} - V_{FB}) - SF_1 + SF_2$ $Q_b = -Cox_{t} * (V_{gb} - V_{FB}) - SF_1 + SF_2$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Depletion region, $V_{gs} < V_{TH}$	$Q_g = -0.5 * Cox_{t} * GAMMA^2 * \left( 1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})} \right) - SF_1 + SF_2$ $Q_b = 0.5 * Cox_{t} * GAMMA^2 * \left( 1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})} \right) + SF_1 - SF_2$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$

Operational region	Level-3 Charges Equations
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	$Q_g = Cox_t * \left( V_{gs} - V_{FB} - PHI + ETA * V_{dsat} - \frac{V_{dsat}}{2} + \frac{1 + F_B}{12 * F_i} * V_{dsat}^2 \right)$ $Q_b = -Cox_t * \left( V_{TH} - V_{FB} - PHI + ETA * V_{dsat} - \frac{F_b}{2} * V_{ds} - \frac{F_B * (1 + F_B)}{12 * F_i} * V_{dsat}^2 \right)$ $Q_c = -Cox_t * \left( V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{dsat} + \frac{(1 + F_B)^2}{12 * F_i} * V_{dsat}^2 \right)$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$Q_g = Cox_t * \left( V_{gs} - V_{FB} - PHI + ETA * V_{ds} - \frac{V_{ds}}{2} + \frac{1 + F_B}{12 * F_i} * V_{ds}^2 \right)$ $Q_b = -Cox_t * \left( V_{TH} - V_{FB} - PHI + ETA * V_{ds} + \frac{F_b}{2} * V_{ds} - \frac{F_B * (1 + F_B)}{12 * F_i} * V_{ds}^2 \right)$ $Q_c = -Cox_t * \left( V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{ds} + \frac{(1 + F_B)^2}{12 * F_i} * V_{ds}^2 \right)$ $Q_d = -Cox_t * \left( \frac{V_{gs} - V_{TH}}{2} - \frac{3(1 + F_B)}{2} * V_{ds} + \frac{(1 + F_B)^2}{8 * F_i} * V_{ds}^2 \right)$ $Q_s = Q_c - Q_d$

where:

- $V_{dsat}$  is the saturation voltage
- $F_B$  is the body effect coefficient
- $ETA$  is the drain-source voltage threshold coefficient
- $F_i = V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{ds}$
- $SF_1 = -0.5 * Cox_t * GAMMA^2 * \left( 1 - \left( 1 + \frac{4}{GAMMA^2} * 2(V_{TH} - V_{bs} - V_{FB}) \right)^{0.5} \right)$  and  
 $SF_2 = Cox_t * (V_{TH} - V_{FB} - PHI)$  are smoothing factors between depletion and accumulation regions to help with convergence.

### Temperature Dependence

The transconductance as a function of the transistor temperature is

$$KP(T) = \frac{KP_d}{\left( \frac{T}{T_{meas}} \right)^{3/2}}$$

Where:

- $KP_d$  is the geometry-adjusted transconductance.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2077.

- $T_{meas}$  is the parameter extraction temperature.

The surface potential as a function of the transistor temperature is

$$PHI(T) = \frac{T}{T_{meas}} \left( PHI + \frac{kT_{meas}}{q} \left( \log \left( \frac{T_{meas}}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_{T_{meas}}}{T_{meas}} \right) \right) \right) - \frac{kT}{q} \left( \log \left( \frac{T}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_T}{T} \right) \right)$$

Where:

- $PHI$  is the surface potential.
- $k$  is the Boltzmann constant.
- $q$  is the elementary charge on an electron, 1.6021918e-19 C.
- $EG$  is the activation energy, such that:
  - $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
  - $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The built-in voltage as a function of the transistor temperature is

$$VBI(T) = VTO + MTYPE * \left( \frac{PHI(T) - PHI}{2} - GAMMA * \sqrt{PHI} \right) + \frac{EG_{T_{meas}} - EG_T}{2}$$

Where:

- $VBI$  is the built-in voltage.
- $VTO$  is the threshold voltage.  $VTO$  depends on the value that you specify for the **Threshold voltage, VTO** parameter in the **DC currents** settings. If you specify a numerical value,  $VTO$  is evaluated as that value. If you specify a nonnumerical value (NAN) and you specify numerical values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters in the **Process** settings, then  $VTO$  is evaluated as  $\Phi - 3.25 + EG_{T_{meas}}/2 + MTYPE * PHI/2 - NSS * q * TOX / (3.9 * \epsilon_0) + MTYPE * (GAMMA * \sqrt{PHI} + PHI)$

Where:

- $\Phi$  depends on the gate type, which you specify using the **Gate type, TPG** parameter. If you specify Aluminum ( $\emptyset$ ),  $\Phi = 3.2$ . Otherwise,  $\Phi = 3.25 + EG_{T_{meas}}/2 - MTYPE * TPG * EG_{T_{meas}}/2$ , Where:
  - $MTYPE$  is the transistor type. For an N-channel MOSFET,  $MTYPE = 1$ .
  - $TPG$  represents the gate type and also depends on the option that you specify for the **Gate type, TPG** parameter in the **Process** settings. If you specify
    - Opposite of substrate (1) —  $TPG = 1$
    - Same as substrate (-1) —  $TPG = -1$

- $NSS$  is the surface state density.
- $TOX$  is the oxide thickness.
- $\epsilon_0$  is the permittivity of free space.
- $GAMMA$  is the bulk threshold.  $GAMMA$  depends on the value that you specify for the **Bulk threshold, GAMMA** parameter in the **DC currents** settings. If you specify a numerical value,  $GAMMA$  is evaluated as that value. If you specify a nonnumerical value (NAN) and you specify numerical values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters in the **Process** settings, then  $VTO$  is evaluated as  $TOX * \sqrt{2 * 11.7 * \epsilon_0 * q * NSUB} / (3.9 * \epsilon_0)$ , where  $NSUB$  is the substrate doping.

The bulk saturation current as a function of the transistor temperature is

$$IS(T) = IS_d * e^{\frac{-qEG_T}{ND * kT}} + \frac{qEG_{T_{meas}}}{ND * kT_{meas}}$$

Where:

- $IS_d$  is the geometry-adjusted bulk saturation current.
- $ND$  is the emission coefficient.

The bulk junction saturation current density as a function of the transistor temperature is

$$JS(T) = JS_d * e^{\frac{-qEG_T}{ND * kT}} + \frac{qEG_{T_{meas}}}{ND * kT_{meas}}$$

Where  $JS_d$  is the geometry-adjusted bulk junction saturation current density.

The bulk junction potential as a function of the transistor temperature is

$$PB(T) = \frac{PB + \frac{kT_{meas}}{q} \left( \log\left(\frac{T_{meas}}{300.15}\right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_{T_{meas}}}{T} \right) \right)}{T_{meas}/T} - \frac{kT}{q} \left( \log\left(\frac{T}{300.15}\right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_T}{T} \right) \right)$$

Where  $PB$  is the bulk junction potential.

The bulk-drain junction capacitance as a function of the transistor temperature is

$$CBD(T) = CBD_d \frac{pbo + MJ * (4 * 10^4 * (T - 300.15) * pbo - (PB(T) - pbo))}{pbo + MJ * (4 * 10^4 * (T_{meas} - 300.15) * pbo - (PB - pbo))}$$

Where:

- $CBD_d$  is the geometry adjusted zero-bias bulk-drain capacitance.
- $MJ$  is the bottom grading coefficient.

$$pbo = \frac{PB + \frac{kT_{meas}}{q} \left( \log\left(\frac{T_{meas}}{300.15}\right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_{T_{meas}}}{T} \right) \right)}{T_{meas}/300.15}$$

The block uses the  $CBD(T)$  equation to calculate:

- The bulk-source junction capacitance by substituting  $CBS_d$ , the geometry-adjusted zero-bias bulk-source capacitance, for  $CBD_d$ .
- The bottom junction capacitance by substituting  $CJ_d$ , the geometry-adjusted bottom capacitance per junction area for  $CBD_d$ .

The relationship between the sidewall junction capacitance  $CJSW$  and the transistor temperature,  $T$ , is

$$CJSW(T) = CJSW_d \frac{pbo + MJSW * (4 * 10^4 * (T - 300.15) * pbo - (PB(T) - pbo))}{pbo + MJSW * (4 * 10^4 * (T_{meas} - 300.15) * pbo - (PB - pbo))}$$

Where:

- $CJSW_d$  is the side geometry-adjusted sidewall capacitance per junction perimeter.
- $MJSW$  is the side grading coefficient.

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### **gx — Gate terminal**

electrical

Electrical conserving port associated with the transistor gate terminal.

#### **dx — Drain terminal**

electrical

Electrical conserving port associated with the transistor drain terminal.

#### **sx — Source terminal**

electrical

Electrical conserving port associated with the transistor source terminal.

#### **bx — Bulk terminal**

electrical

Electrical conserving port associated with the transistor bulk terminal.

## Parameters

### Model Selection

#### MOS model — Drain current model

Level 1 MOS (default) | Level 3 MOS

MOSFET drain current model options:

- Level 1 MOS — Use the “Level 1 Drain Current Model” on page 1-2082. This is the default option.
- Level 3 MOS — Use the “Level 3 Drain Current Model” on page 1-2083.

### Dependencies

The setting that you select for the **MOS model** affects the visibility of certain parameters in the **DC Currents** and **Process** settings.

### Dimensions

#### Device area factor, AREA — Device area

1.0 (default) | positive scalar

Transistor area factor for scaling. The value must be greater than 0.

#### Number of parallel devices, SCALE — Number of parallel devices

1 (default) | positive integer

The number of parallel MOS instances that the block represents. This parameter multiplies the output current and device charge. The value must be greater than 0.

#### Length of channel, LENGTH — Source-to-drain channel length

1e-4 m (default) | positive scalar

Length of the channel between the source and drain.

#### Width of channel, WIDTH — Source-to-drain channel width

1e-4 m (default) | positive scalar

Width of the channel between the source and drain.

#### Area of drain, AD — Drain area

0 m<sup>2</sup> (default) | nonnegative scalar

Area of the transistor drain diffusion. The value must be greater than or equal to 0.

#### Area of source, AS — Source area

0 m<sup>2</sup> (default) | nonnegative scalar

Area of the transistor source diffusion. The value must be greater than or equal to 0.

#### Perimeter of drain, PD — Drain perimeter

0 m (default) | nonnegative scalar

Perimeter of the transistor drain diffusion. The value must be greater than or equal to 0.

**Perimeter of source, PS — Source perimeter** $0$  m (default) | nonnegative scalar

Perimeter of the transistor source diffusion. The value must be greater than or equal to  $0$ .

**Resistors****Number of drain squares, NRD — Number of drain diffusion resistance squares** $0$  (default) | nonnegative scalar

Number of squares of resistance that make up the transistor drain diffusion. The value must be greater than or equal to  $0$ . The block only uses this parameter value if you do not specify one or both of the **Drain resistance, RD** and **Source resistance, RS** parameter values, as described in “Parameters Calculations” on page 1-2078.

**Number of source squares, NRS — Number of transistor source diffusion squares of resistance** $1$  (default) | nonnegative scalar

Number of squares of resistance that make up the transistor source diffusion. The value must be greater than or equal to  $0$ . The block only uses this parameter value if you do not specify one or both of the **Drain resistance, RD** and **Source resistance, RS** parameter values, as described in “Parameters Calculations” on page 1-2078.

**Drain resistance, RD — Series resistance** $0$  Ohm (default) | nonnegative scalar

Transistor drain resistance. The value must be greater than or equal to  $0$ .

**Source resistance, RS — Series resistance** $0$  Ohm (default) | nonnegative scalar

Transistor source resistance. The value must be greater than or equal to  $0$ .

**Sheet resistance, RSH — Per square resistance** $0$  Ohm (default) | nonnegative scalar

Resistance per square of the transistor source and drain. Check “Parameters Calculations” on page 1-2078 to see when the block uses this parameter. The value must be greater than or equal to  $0$ .

**DC Currents****Threshold voltage, VT0 — Threshold voltage** $0$  V (default) | scalar

The gate-source voltage above which the transistor produces a nonzero drain current. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Temperature Dependence” on page 1-2093.

**Transconductance, KP — Transconductance** $2e-5$  A/V<sup>2</sup> (default) | nonnegative scalar

The derivative of drain current with respect to gate voltage. The value must be greater than or equal to  $0$ . If you assign this parameter a value of NaN, the block calculates the value from the specified



values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2082 or “Level 3 Drain Current Model” on page 1-2083 as appropriate for the selected value of the **MOS model** parameter.

#### **Bulk threshold, GAMMA — Bulk threshold**

0 V (default) | nonnegative scalar

Body effect parameter, which relates the threshold voltage, VTH, to the body bias, VBS, as described in “Level 1 Drain Current Model” on page 1-2082 and “Level 3 Drain Current Model” on page 1-2083. The value must be greater than or equal to 0. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2082 or “Level 3 Drain Current Model” on page 1-2083 as appropriate for the selected value of the **MOS model** parameter.

#### **Surface potential, PHI — Surface potential**

0.6 V (default) | nonnegative scalar

Twice the voltage at which the surface electron concentration becomes equal to the intrinsic concentration and the device transitions between depletion and inversion conditions. The value must be greater than or equal to 0. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2082 or “Level 3 Drain Current Model” on page 1-2083 as appropriate for the selected value of the **MOS model** parameter.

#### **Channel modulation, LAMBDA — Channel-length modulation**

0 1/V (default) | scalar

Channel-length modulation.

#### **Dependencies**

This parameter is only visible when you select Level 1 MOS for the **MOS model** parameter in the **Model Selection** settings.

#### **Bulk saturation current, IS — Bulk saturation current magnitude**

1e-14 A (default) | nonnegative scalar

Magnitude of the current that the junction approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

#### **Emission coefficient, N — Emission coefficient**

1 (default) | nonnegative scalar

Transistor emission coefficient or ideality factor. The value must be greater than 0.

#### **Bulk jct sat current density, JS — Bulk junction saturation current density**

0 A/m<sup>2</sup> (default) | nonnegative scalar

Magnitude of the current per unit area that the junction approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

#### **Width effect on threshold, DELTA — Width factor**

0 (default) | scalar

Factor that controls the effect of transistor width on threshold voltage.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Max carrier drift velocity, VMAX — Maximum drift velocity**

0 m/s (default) | scalar

Maximum drift velocity of the carriers.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Fast surface state density, NFS — Fast surface state density**

0 1/cm<sup>2</sup> (default) | scalar

Fast surface state density adjusts the drain current for the mobility reduction caused by the gate voltage.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Vds dependence threshold volt, ETA — Drain-source voltage threshold**

0 (default) | scalar

The coefficient that controls how the drain voltage affects the mobility in the drain current calculation.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Vgs dependence on mobility, THETA — Mobility dependence coefficient**

0 1/V (default) | scalar

The coefficient that controls how the gate voltage affects the mobility in the drain current calculation.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Mobility modulation, KAPPA — Mobility modulation coefficient**

0.2 (default) | scalar

Coefficient of channel-length modulation for the level 3 MOS model.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**C-V****Model gate capacitance (CGS, CGD, CGB) — Gate capacitance model**

No intrinsic capacitance (default) | Meyer gate capacitances | Charge conservation capacitances

Options for modeling the gate capacitance:

- No intrinsic capacitance — Do not include gate capacitance in the model.
- Meyer gate capacitances —
- Charge conservation capacitances —

**Model gate overlap capacitance (CGSO, CGDO, CGBO) — Gate overlap model**

No (default) | Yes

Options for modeling the gate overlap capacitance:

- No — Do not include gate overlap capacitance in the model.
- Yes — Specify the gate-source, gate-drain, and gate-bulk capacitances.

**Dependencies**

Selecting Yes exposes related parameters.

**G-S overlap capacitance, CGSO — Gate-source overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-source capacitance due to lateral diffusion of the source. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGBO)** parameter.

**G-D overlap capacitance, CGDO — Gate-drain overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-drain capacitance due to lateral diffusion of the drain. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGBO)** parameter.

**G-B overlap capacitance, CGBO — Gate-bulk overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-bulk capacitance due to gate extending beyond the channel width. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGBO)** parameter.

**Model junction capacitance (CBD, CBS) – Junction capacitance model**

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model.
- Yes — Specify zero-bias junction capacitance, junction potential, grading coefficient, forward-bias depletion and capacitance coefficient.

**Dependencies**

Selecting Yes exposes related parameters.

**Zero-bias BD capacitance, CBD – Zero-bias junction capacitance** $0\text{ F}$  (default) |  $0$  or  $\geq 1\text{e-}18$ Capacitance between the bulk and the drain. The value must be equal to  $0$  or greater than or equal to  $C_{min}$ .  $C_{min}$  is a built-in model constant whose value is  $1\text{e-}18$ .**Dependencies**This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.**Zero-bias BS capacitance, CBS – Zero-bias bulk-source capacitance** $0\text{ F}$  (default) | scalar |  $0$  or  $\geq 1\text{e-}18$ Capacitance between the bulk and the source. The value must be equal to  $0$  or greater than or equal to  $C_{min}$ .  $C_{min}$  is a built-in model constant whose value is  $1\text{e-}18$ .**Dependencies**This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.**Bulk junction potential, PB – Bulk junction potential** $0.8\text{ V}$  (default) | scalar |  $0$  or  $\geq 0.01$ Potential across the bulk junction. This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter. The value must be equal to  $0$  or greater than or equal to  $VJ_{min}$ .  $VJ_{min}$  is a built-in model constant whose value is  $0.01$ .**Dependencies**This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.**Bottom junction cap per area, CJ – Zero-bias bulk junction bottom capacitance per junction area** $0\text{ F/m}^2$  (default) | scalar |  $0$  or  $\geq 1\text{e-}18$ Zero-bias bulk junction bottom capacitance per junction area. The value must be equal to  $0$  or greater than or equal to  $C_{min}$ .  $C_{min}$  is a built-in model constant whose value is  $1\text{e-}18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Bottom grading coefficient, MJ — Bottom grading coefficient**

0.5 (default) | scalar | 0 or < 0.9

Transistor bottom grading coefficient. The value must be equal to 0 or less than  $MGmax$ .  $MGmax$  is a built-in model constant whose value is 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Side jct cap/area of jct perimeter, CJSW — Zero-bias bulk junction sidewall capacitance per junction perimeter**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Zero-bias bulk junction sidewall capacitance per junction perimeter. The value must be equal to 0 or greater than or equal to  $Cmin$ .  $Cmin$  is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Side grading coefficient, MJSW — Sidewall grading coefficient**

0.5 (default) | scalar |  $0 < MJSW < 0.9$

Transistor sidewall grading coefficient. The value must be equal to 0 or less than  $MGmax$ .  $MGmax$  is a built-in model constant whose value is 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Capacitance coefficient, FC — Capacitance coefficient**

0.5 (default) | 0 or  $\leq 0.95$  | scalar

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage. The value must be equal to 0 or less than or equal to  $FCmax$ .  $FCmax$  is a built-in model constant whose value is 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- No — Do not specify an initial condition for the model.

- Yes — Specify the initial transistor voltage.

---

**Note** The block applies the initial transistor voltage across the junction capacitors and not across the ports.

---

**Dependencies**

To enable this parameter, set either:

- **Model gate capacitance (CGS, CGD, CGB)** to No intrinsic capacitance and either or both **Model gate overlap capacitance (CGSO, CGDO, CGBO)** and **Model junction capacitance (CBD, CBS)** to Yes.
- **Model gate capacitance (CGS, CGD, CGB)** to Meyer gate capacitances or Charge conservation capacitances.

**Initial condition voltage, ICVDS — Initial voltage**

0 V (default) | scalar

Drain-source voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

**Initial condition voltage, ICVGS — Initial voltage**

0 V (default) | scalar

Gate-source voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

**Initial condition voltage, ICVBS — Initial voltage**

0 V (default) | scalar

Bulk-source voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

**Process****Oxide thickness, TOX — Gate oxide thickness**

1e-7 m (default) | nonnegative scalar

Thickness of the gate oxide. The value must be greater than or equal to 0.

**Lateral diffusion, LD — Length of lateral diffusion**

0 m (default) | scalar

Length of lateral diffusion.

**Surface mobility, U0 — Zero-bias surface mobility coefficient**

600 cm<sup>2</sup>/s/V (default)

Zero-bias surface mobility coefficient.

### Substrate doping, NSUB — Substrate doping

NaN 1/cm<sup>3</sup> (default) | scalar  $\geq 1.45e10$

Substrate doping. The value must be greater than or equal to 1.45e10 (the carrier concentration of intrinsic silicon).

### Gate type, TPG — Gate type

Opposite of substrate (1) (default) | Same as substrate (-1) | Aluminum (0)

MOSFET gate materials (as compared to the substrate):

- **Opposite of substrate** — The gate material is the opposite of the substrate. This means that TPG = 1 in the device equations. This is the default option.
- **Same as substrate** — The gate material is the same as the substrate. This means that TPG = -1 in the device equations.
- **Aluminum** — The gate material is aluminum. This means that TPG = 0 in the device equations.

### Surface state density, NSS — Surface state density

NaN 1/cm<sup>2</sup> (default) | scalar

Surface state density.

### Junction depth, XJ — Junction depth

0m (default)

Junction depth.

### Dependencies

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

### Temperature

#### Model temperature dependence using — Temperature dependence model

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Temperature Dependence” on page 1-2093.

### Dependencies

Selecting Device temperature exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting Fixed temperature exposes the **Fixed circuit temperature, TFIXED** parameter.

#### Fixed circuit temperature, TFIXED — Fixed circuit temperature

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0 K.

**Dependencies**

This parameter is only visible when you select Fixed temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

The temperature at which the transistor parameters are measured. The value must be greater than 0 K.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

The amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is only visible when you select Device temperature for the **Model temperature dependence using** parameter.

## Version History

Introduced in R2009a

## References

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.
- [2] Ping Yang, et al. 'An Investigation of the Charge Conservation Problem for MOSFET Circuit Simulation'. *IEEE Journal of Solid-State Circuits*, vol. 18, no. 1, Feb. 1983, pp. 128-38. DOI.org (Crossref), doi:10.1109/JSSC.1983.1051909.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | SPICE PMOS

**Functions**

subcircuit2ssc

**Topics**

"Additional Parameterization Workflows"  
"Converting a SPICE Netlist to Simscape Blocks"



“Parameterize an Exponential Diode from SPICE Netlist”

## SPICE NPN

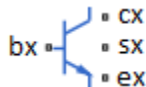
SPICE-compatible Gummel-Poon NPN Transistor

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



### Description

The SPICE NPN block represents a SPICE-compatible four-terminal Gummel-Poon NPN bipolar junction transistor. A capacitor connects the substrate port, **sx**, to the transistor base, **bx**. Therefore, the device is equivalent to a three-terminal transistor when you use the default value of 0 for the **C-S junction capacitance, CJS** parameter and connect the substrate port to any other port, including the emitter port, **ex**, or the collector port, **cx**.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

### Equations

Variables for the SPICE NPN block equations include:

- Variables that you define by specifying parameters for the SPICE NPN block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2116.
- Geometry-adjusted variables, which depend on several values that you specify using parameters for the SPICE NPN block. For more information, see “Geometry-Adjusted Variables” on page 1-2108.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE NPN block or by specifying parameters for both the SPICE NPN block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2109.
- Temperature-dependent variables. For more information, see “Temperature Dependence” on page 1-2114.
- Minimal conductance,  $GMIN$ , which is  $1e-12$  1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2110.

### Geometry-Adjusted Variables

Several variables in the equations for the SPICE NPN bipolar junction transistor model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on

variables that you define by specifying SPICE NPN block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$IS_d$	Geometry-adjusted transport saturation current	$IS_d = IS * AREA * SCALE$
$IKF_d$	Geometry-adjusted forward knee current	$IKF_d = IKF * AREA * SCALE$
$IS_d$	Geometry-adjusted base-emitter leakage current	$ISE_d = ISE * AREA * SCALE$
$IKR_d$	Geometry-adjusted reverse knee current	$IKR_d = IKR * AREA * SCALE$
$ISC_d$	Geometry-adjusted base-collector leakage current	$ISC_d = ISC * AREA * SCALE$
$IRB_d$	Geometry-adjusted half base resistance current	$IRB_d = IRB * AREA * SCALE$
$CJE_d$	Geometry-adjusted base-emitter depletion capacitance	$CJE_d = CJE * AREA * SCALE$
$ITF_d$	Geometry-adjusted forward transit time coefficient	$ITF_d = ITF * AREA * SCALE$
$CJC_d$	Geometry-adjusted base-collector depletion capacitance	$CJC_d = CJC * AREA * SCALE$
$CJS_d$	Geometry-adjusted collector-substrate junction capacitance	$CJS_d = CJS * AREA * SCALE$
$RB_d$	Geometry-adjusted zero-bias base resistance	$RB_d = \frac{RB}{AREA * SCALE}$
$RBM_d$	Geometry-adjusted minimum base resistance	$RBM_d = \frac{RBM}{AREA * SCALE}$
$RE_d$	Geometry-adjusted emitter resistance	$RE_d = \frac{RE}{AREA * SCALE}$
$RC_d$	Geometry-adjusted collector resistance	$RC_d = \frac{RC}{AREA * SCALE}$

### Transistor Temperature

You can use these options to define transistor temperature, *T*:

- Fixed temperature — The block uses a temperature that is independent from the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NPN block is set to Fixed temperature. For this model, the block sets *T* equal to *TFIXED*.

- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE NPN block is set to Device temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is no Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- $TOFFSET$  is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of 1e-12 1/Ohm. To specify a different value:

- 1 If there is not an Environment Parameters block in the circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Current-Voltage and Base Charge Model

The current-voltage relationships and base charge relationships for the transistor are described in terms of “Base-Emitter and Base-Collector Junction Currents” on page 1-2110, “Terminal Currents” on page 1-2111, and “Base Charge Model” on page 1-2112. As applicable, the model parameters are first adjusted for temperature.

#### Base-Emitter and Base-Collector Junction Currents

The base-emitter junction current depends on the base-emitter voltage,  $V_{BE}$  such that:

- When  $V_{BE} > 80 * V_{TF}$ :

$$I_{bef} = IS_d * \left( \left( \frac{V_{BE}}{V_{TF}} - 79 \right) * e^{80} - 1 \right) + G_{min} * V_{BE}$$

$$I_{bee} = ISE_d * \left( (V_{BE} - 80 * V_{TF} + V_{TE}) * \frac{e^{(80 * V_{TF}/V_{TE})}}{V_{TE}} - 1 \right)$$

- When  $V_{BE} \leq 80 * V_{TF}$ :

$$I_{bef} = IS_d * \left( e^{(V_{BE}/V_{TF})} - 1 \right) + G_{min} * V_{BE}$$

$$I_{bee} = ISE_d * \left( e^{(V_{BE}/V_{TE})} - 1 \right)$$

The base-collector junction current depends on the base collector voltage,  $V_{BC}$ , such that:

- When  $V_{BC} > 80 * V_{TR}$ :

$$I_{bcr} = IS_d * \left( \left( \frac{V_{BC}}{V_{TR}} - 79 \right) * e^{80} - 1 \right) + G_{min} * V_{BC}$$

$$I_{bcc} = ISC_d * \left( (V_{BC} - 80 * V_{TR} + V_{TC}) * \frac{e^{(80 * V_{TR}/V_{TC})}}{V_{TC}} - 1 \right)$$

- When  $V_{BC} \leq 80 * V_{TR}$ :

$$I_{bcr} = ISC_d * (e^{(V_{BC}/V_{TR})} - 1) + G_{min} * V_{BC}$$

$$I_{bcc} = ISC_d * (e^{(V_{BC}/V_{TC})} - 1)$$

Where:

- $V_{BE}$  is the base-emitter voltage.
- $V_{BC}$  is the base-collector voltage.
- $V_{TE}$  is the emitter thermal voltage, such that  $V_{TE} = NE * k * T / q$ .
- $V_{TC}$  is the collector thermal voltage, such that  $V_{TC} = NC * k * T / q$ .
- $V_{TF}$  is the forward thermal voltage, such that  $V_{TF} = NF * k * T / q$ .
- $V_{TR}$  is the reverse thermal voltage, such that  $V_{TR} = NR * k * T / q$ .
- $ISC_d$  is the geometry-adjusted base-collector leakage current.
- $ISE_d$  is the geometry-adjusted base-emitter leakage current.
- $NE$  is the base-emitter emission coefficient.
- $NC$  is the base-collector emission coefficient.
- $NF$  is the forward emission coefficient.
- $NR$  is the reverse emission coefficient.
- $q$  is the elementary charge on an electron.
- $k$  is the Boltzmann constant.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2109.
- $G_{min}$  is the minimum conductance. For more information, see “Minimal Conduction” on page 1-2044.

### Terminal Currents

The terminal currents are calculated as:

$$I_B = \left( \frac{I_{bef}}{BF} + I_{bee} + \frac{I_{bcr}}{BR} + I_{bcc} \right)$$

$$I_C = \left( \frac{I_{bef} - I_{bcr}}{q_b} - \frac{I_{bcr}}{BR} - I_{bcc} \right)$$

Where:

- $I_B$  is the base terminal current.
- $I_C$  is the collector terminal current.

- $BF$  is the forward beta.
- $BR$  is the reverse beta.

### Base Charge Model

The base charge,  $q_b$ , is calculated using these equations:

$$q_b = \frac{q_1}{2} \left( 1 + \sqrt{0.5 \left( \sqrt{(1 + 4q_2 - eps)^2 + eps^2} + 1 + 4q_2 - eps \right) + eps} \right)$$

$$q_1 = \left( 1 - \frac{V_{BC}}{V_{AF}} - \frac{V_{BE}}{V_{AR}} \right)^{-1}$$

$$q_2 = \frac{I_{bef}}{IKF_d} + \frac{I_{bcr}}{IKR_d}$$

Where:

- $q_b$  is the base charge.
- $V_{AF}$  is the forward Early voltage.
- $V_{AR}$  is the reverse Early voltage.
- $IKF_d$  is the geometry-adjusted forward knee current.
- $IKR_d$  is the geometry-adjusted reverse knee current.
- $eps$  is 1e-4.

### Base Resistance Model

You can use these options to model base resistance,  $r_{bb}$ :

- If you use the default value of infinity for the **Half base resistance cur, IRB** parameter, the block calculates the base resistance as

$$r_{bb} = RBM_d + \frac{RB_d - RBM_d}{q_b}$$

Where:

- $r_{bb}$  is base resistance.
- $RBM_d$  is the geometry-adjusted minimum base resistance.
- $RB_d$  is the geometry-adjusted zero-bias base resistance.
- If you specify a finite value for the **Half base resistance cur, IRB** parameter, the block calculates the base resistance as

$$r_{bb} = RBM_d + 3 * (RB_d - RBM_d) * \left( \frac{\tan z - z}{z * \tan^2 z} \right)$$

Where

$$z = \frac{\sqrt{1 + 144I_B / (\pi^2 IRB_d)} - 1}{(24/\pi^2) \sqrt{I_B / IRB_d}}$$

### Transit Charge Modulation Model

If you specify nonzero values for the **Coefficient of TF, XTF** parameter, the block models transit charge modulation by scaling the forward transit time as

$$TF_{\text{mod}} = \frac{TF * \left[ 1 + XTF * e^{V_{BC}/(1.44V_{TF})} \left( \frac{I_{BE}}{I_{BE} + ITF_d} \right)^2 \right]}{q_b}$$

Where  $ITF_d$  is the geometry-adjusted coefficient of the forward transit time.

### Junction Charge Model

The block lets you model junction charge. The base-collector charge,  $Q_{bc}$ , and the base-emitter charge,  $Q_{be}$ , depend on an intermediate value,  $Q_{dep}$ . As applicable, the model parameters are first adjusted for temperature.

- For the internal base-emitter junctions

$$Q_{be} = TF_{\text{mod}} * I_{be} + Q_{dep}$$

- For the internal base-collector junctions

$$Q_{bc} = TR * I_{bc} + XCJC * Q_{dep}$$

- For the external base-collector junctions

$$Q_{b_{\text{ext}c}} = (1 - XCJC) * Q_{dep}$$

$Q_{dep}$  depends on the junction voltage,  $V_{jct}$  ( $V_{BE}$  for the base-emitter junction and  $V_{BC}$  for the base-collector junction), as follows.

Applicable Range of $V_{jct}$ Values	Corresponding $Q_{dep}$ Equation
$V_{jct} < FC * VJ$	$Q_{dep} = C_{jct} * VJ * \frac{1 - (1 - V_{jct}/VJ)^{(1 - MJ)}}{1 - MJ}$
$V_{jct} \geq FC * VJ$	$Q_{dep} = C_{jct} * \left[ F1 + \frac{F3 * (V_{jct} - FC * VJ) + \frac{MJ * [V_{jct}^2 - (FC * VJ)^2]}{2 * VJ}}{F2} \right]$

Where:

- $FC$  is the capacitance coefficient.
- $VJ$  is:
  - The base-emitter built-in potential,  $V_{JE}$ , for the base-emitter junction.
  - The base-collector built-in potential,  $V_{JC}$ , for the base-collector junction.
- $MJ$  is:
  - The base-emitter exponential factor,  $M_{JE}$ , for the base-emitter junction.
  - The base-collector exponential factor,  $M_{JC}$ , for the base-collector junction.

- $C_{jct}$  is:
  - The geometry-adjusted base-emitter depletion capacitance,  $CJE_d$ , for the base-emitter junction.
  - The geometry-adjusted base-collector depletion capacitance,  $CJC_d$ , for the base-collector junction.
- $F1 = VJ * (1 - (1 - FC)^{(1 - MJ)}) / (1 - MJ)$
- $F2 = (1 - FC)^{(1 + MJ)}$
- $F3 = 1 - FC * (1 + MJ)$

The collector-substrate charge,  $Q_{cs}$ , depends on the collector-substrate voltage,  $V_{cs}$ . As applicable, the model parameters are first adjusted for temperature.

Applicable Range of $V_{cs}$ Values	Corresponding $Q_{cs}$ Equation
$V_{cs} < 0$	$Q_{cs} = CJS_d * VJS * \left( \frac{1 - (1 - V_{cs}/VJS)^{(1 - MJS)}}{1 - MJS} \right)$
$V_{cs} \geq 0$	$Q_{cs} = CJS_d * (1 + MJS * V_{cs} / (2 * VJS)) * V_{cs}$

Where:

- $CJS_d$  is the geometry-adjusted collector-substrate junction capacitance.
- $VJS$  is the substrate built-in potential.
- $MJS$  is the substrate exponential factor.

#### Temperature Dependence

The relationship between the saturation current,  $IS_d$ , and the transistor temperature,  $T$ , is

$$IS(T) = IS_d * (T/T_{meas})^{XTI} * e^{\left(\frac{T}{T_{meas}} - 1\right) * \frac{EG}{V_t}}$$

Where:

- $IS_d$  is the geometry-adjusted transport saturation current.
- $T_{meas}$  is the parameter extraction temperature.
- $XTI$  is the transport saturation current temperature exponent.
- $EG$  is the energy gap.
- $V_t = kT/q$ .

The relationship between the base-emitter junction potential,  $VJE$ , and the transistor temperature,  $T$ , is

$$VJE(T) = VJE * \left(\frac{T}{T_{meas}}\right) - \frac{3 * k * T}{q} * \log\left(\frac{T}{T_{meas}}\right) - \left(\frac{T}{T_{meas}}\right) * EG_{T_{meas}} + EG_T$$

Where:

- $VJE$  is the base-emitter built-in potential.



- $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
- $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The block uses the  $VJE(T)$  equation to calculate the base-collector junction potential by substituting  $VJC$ , the base-collector built-in potential, for  $VJE$ .

The relationship between the base-emitter junction capacitance,  $CJE$ , and the transistor temperature,  $T$ , is

$$CJE(T) = CJE_d * \left[ 1 + MJE * \left( 400e - 6 * (T - T_{meas}) - \frac{VJE(T) - VJE}{VJE} \right) \right]$$

Where:

- $CJE_d$  is the geometry-adjusted base-emitter depletion capacitance.
- $MJE$  is the base-emitter exponential factor.

The block uses the  $CJE(T)$  equation to calculate the base-collector junction capacitance by substituting  $CJC_d$ , geometry-adjusted base-collector depletion capacitance, for  $CJE_d$  and  $MJC$ , base-collector exponential factor, for  $MJE$ .

The relationship between the forward and reverse beta and the transistor temperature,  $T$ , is

$$\beta(T) = \beta * \left( \frac{T}{T_{meas}} \right)^{XTB}$$

Where:

- $\beta$  is the forward beta or reverse beta.
- $XTB$  is the beta temperature exponent.

The relationship between the base-emitter leakage current,  $ISE$ , and the transistor temperature,  $T$ , is

$$ISE(T) = ISE_d * \left( \frac{T}{T_{meas}} \right)^{-XTB} * \left( \frac{IS(T)}{IS_d} \right)^{1/NE}$$

Where:

- $ISE_d$  is the geometry-adjusted base-emitter leakage current.
- $NE$  is the base-emitter emission coefficient.

The block uses this equation to calculate the base-collector leakage current by substituting,  $ISC_d$ , the geometry-adjusted base-collector leakage current for  $ISE_d$  and  $NC$ , the base-collector emission coefficient, for  $NE$ .

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### **bx – Base terminal**

electrical

Electrical conserving port associated with the transistor base terminal.

#### **cx – Collector terminal**

electrical

Electrical conserving port associated with the transistor collector terminal.

#### **ex – Emitter terminal**

electrical

Electrical conserving port associated with the transistor emitter terminal.

#### **sx – Substrate terminal**

electrical

Electrical conserving port associated with the transistor substrate terminal.

## Parameters

### Main

#### **Device area, AREA – Device area**

1.0 m<sup>2</sup> (default) | positive scalar

Device area. The value must be greater than 0.

#### **Number of parallel devices, SCALE – Number of parallel devices**

1 (default) | positive scalar

Number of parallel transistors that the block represents. The value must be greater than 0.

### Forward Gain

#### **Transport saturation current, IS – Saturation current magnitude**

1e-16 A/m<sup>2</sup> (default) | positive scalar

Magnitude of the current at which the transistor saturates. The value must be greater than 0.

#### **Forward beta, BF – Ideal maximum forward beta**

100 (default) | positive scalar

Ideal maximum forward beta. The value must be greater than 0.

#### **Forward emission coefficient, NF – Forward emission coefficient**

1 (default) | positive scalar

Forward emission coefficient or ideality factor. The value must be greater than 0.

**Forward Early voltage, VAF — Forward Early voltage**

Inf V (default) | nonnegative scalar

Forward Early voltage. The value must be greater than or equal to 0.

**Forward knee current, IKF — Forward-beta high-current roll-off current**Inf A/m<sup>2</sup> (default) | nonnegative scalar

Current value at which forward-beta high-current roll-off occurs. The value must be greater than or equal to 0.

**B-E leakage current, ISE — Base-emitter leakage current**0 A/m<sup>2</sup> (default) | nonnegative scalar

Base-emitter leakage current. The value must be greater than or equal to 0.

**B-E emission coefficient, NE — Base-emitter emission coefficient**

1.5 (default) | positive scalar

Base-emitter emission coefficient or ideality factor. The value must be greater than 0.

**Reverse Gain****Reverse beta, BR — Ideal maximum reverse beta**

1 (default) | positive scalar

Ideal maximum reverse beta. The value must be greater than 0.

**Reverse emission coefficient, NR — Reverse emission coefficient**

1 (default) | positive scalar

Reverse emission coefficient or ideality factor. The value must be greater than 0.

**Reverse Early voltage, VAR — Reverse Early voltage**

Inf V (default) | nonnegative scalar

Reverse Early voltage. The value must be greater than or equal to 0.

**Reverse knee current, IKR — Reverse-beta high-current roll-off current**Inf A/m<sup>2</sup> (default) | nonnegative scalar

Current value at which reverse-beta high-current roll-off occurs. The value must be greater than or equal to 0.

**B-C leakage current, ISC — Base-collector leakage current**0 A/m<sup>2</sup> (default) | nonnegative scalar

Base-collector leakage current. The value must be greater than or equal to 0.

**B-C emission coefficient, NC — Base-collector emission coefficient**

2 (default) | positive scalar

Base-collector emission coefficient or ideality factor. The value must be greater than 0.

## Resistors

### Zero-bias base resistance, **RB** — Base resistance

1 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Maximum resistance of the base. The value must be greater than or equal to 0.

### Half base resistance cur, **IRB** — Base resistance half zero-bias drop current

Inf A/m<sup>2</sup> (default) | nonnegative scalar

Base current at which the base resistance has dropped to half of its zero-bias value. The value must be greater than or equal to 0. If you do not want to model the change in base resistance as a function of base current, use the default value of Inf.

### Minimum base resistance, **RBM** — Minimum base resistance

0 m<sup>2</sup>\*0hm (default) | scalar < *RB*

Minimum resistance of the base. The value must be less than or equal to the **Zero-bias base resistance, RB** parameter value.

### Emitter resistance, **RE** — Emitter resistance

1e-4 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Resistance of the emitter. The value must be greater than or equal to 0.

### Collector resistance, **RC** — Collector resistance

0.01 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Resistance of the collector. The value must be greater than or equal to 0.

## Capacitance

### Model junction capacitance — Junction capacitance model

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model. This is the default option.
- Yes — Include junction capacitance in the model.

### Dependencies

Selecting Yes for the **Model junction capacitance** parameter exposes other **Capacitance** parameters and these capacitance junction settings:

- **B-E Capacitance** — Base-emitter parameters
- **B-C Capacitance** — Base-collector parameters
- **C-S Capacitance** — Collector-substrate parameters

### Capacitance coefficient, **FC** — Capacitance coefficient

0.5 (default) |  $0 \leq FC < 0.95$  | scalar

Fitting coefficient, *FC*, that quantifies the decrease of the depletion capacitance with applied voltage. The value must be greater than or equal to 0 and less than 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- No — Do not specify an initial condition for the model. This is the default option.
- Yes — Specify the initial transistor conditions.

---

**Note** The block applies the initial transistor voltages across the junction capacitors and not across the ports.

---

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

Selecting Yes for the **Specify initial condition** parameter exposes related parameters.

**Initial condition voltage ICVBE — Initial base-emitter voltage**

0 V (default) | scalar

Base-emitter voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**Initial condition voltage ICVCE — Initial base-collector voltage**

0 V (default) | scalar

Base-collector voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**B-E Capacitance**

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

**B-E depletion capacitance, CJE — Base-emitter junction depletion capacitance**

0 F/m<sup>2</sup> (default) | nonnegative scalar

Depletion capacitance across the base-emitter junction. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**B-E built-in potential, VJE — Base-emitter junction potential**

0.75 V (default) | scalar |  $0.01 \leq VJE$

Base-emitter junction potential. The value must be greater than or equal to  $0.01$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**B-E exponential factor, MJE — Base-emitter junction grading coefficient**

$0.33$  (default) | scalar |  $0 \leq MJC \leq 0.9$

Grading coefficient for the base-emitter junction. The value must be greater than or equal to  $0$  and less than or equal to  $0.9$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Forward transit time, TF — Minority carrier transit time**

$0$  s (default) | nonnegative scalar

Transit time of the minority carriers that cause diffusion capacitance when the base-emitter junction is forward-biased. The value must be greater than or equal to  $0$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Coefficient of TF, XTF — Base-emitter transit time bias dependence coefficient**

$0$  (default) | nonnegative scalar

Coefficient for the base-emitter bias dependence of the transit time, which produces a charge across the base-emitter junction. The value must be greater than or equal to  $0$ . If you do not want to model the effect of base-emitter bias on transit time, use the default value of  $0$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**VBC dependence of TF, VTF — Base-collector transit time bias dependence coefficient**

$\text{Inf } V$  (default) | nonnegative scalar

Voefficient for the base-collector bias dependence of the transit time. The value must be greater than or equal to  $0$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Coefficient of TF, ITF — Collector current transit time dependence coefficient**

$0$  A/m<sup>2</sup> (default) | nonnegative scalar

Coefficient for the dependence of the transit time on collector current. The value must be greater than or equal to  $0$ . If you do not want to model the effect of collector current on transit time, use the default value of  $0$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## B-C Capacitance

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

### B-C depletion capacitance, CJC — base-collector junction depletion capacitance

$0 \text{ F/m}^2$  (default) | positive scalar

Depletion capacitance across the base-collector junction. The value must be greater than 0.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### B-C built-in potential, VJC — Base-collector junction potential

$0.75 \text{ V}$  (default) | scalar |  $0.01 \leq VJC$

Base-collector junction potential. The value must be greater than or equal to  $0.01 \text{ V}$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### B-C exponential factor, MJC — Base-collector junction grading coefficient

$0.33$  (default) |  $0 \leq MJC \leq 0.9$

Grading coefficient for the base-collector junction. The value must be greater than or equal to 0 and less than or equal to 0.9.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### B-C capacitance fraction, XCJC — Base-collector depletion capacitance fraction

$1$  (default) | scalar |  $0 \leq XCJC \leq 1$

Fraction of the base-collector depletion capacitance that is connected between the internal base and the internal collector. The rest of the base-collector depletion capacitance is connected between the external base and the internal collector. The value must be greater than or equal to 0 and less than or equal to 1.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### Reverse transit time, TR — Minority carrier transit time

$0 \text{ s}$  (default) | nonnegative scalar

Transit time of the minority carriers that cause diffusion capacitance when the base-collector junction is forward-biased. The value must be greater than or equal to 0.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## C-S Capacitance

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

### C-S junction capacitance, CJS — Collector-substrate junction capacitance

$0\text{F/m}^2$  (default) | nonnegative scalar

Collector-substrate junction capacitance. The value must be greater than or equal to 0.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### Substrate built-in potential, VJS — Substrate potential

0.75 V (default) | scalar |  $0.01 \leq V_{JC}$

Potential of the substrate. The value must be greater than or equal to 0.01 V.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### Substrate exponential factor, MJS — Collector-substrate junction grading coefficient

0 (default) | scalar |  $0 \leq M_{JS} \leq 0.9$

Grading coefficient for the collector-substrate junction. The value must be greater than or equal to 0 and less than or equal to 0.9.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## Temperature

### Model temperature dependence using — Temperature dependence model

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Transistor Temperature” on page 1-2109.

#### Dependencies

Selecting **Device temperature** exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting **Fixed temperature** exposes the **Fixed circuit temperature, TFIXED** parameter.

### Beta temperature exponent, XTB — Forward and reverse beta temperature exponent

0 (default) | nonnegative scalar

Forward and reverse beta temperature exponent that models base current temperature dependence. The value must be greater than or equal to 0.



**Energy gap, EG — Energy gap**1.11 eV (default) |  $EG \geq 0.1$  | scalar

Energy gap that affects the increase in the saturation current as temperature increases. The value must be greater than or equal to 0.1.

**Temperature exponent for IS, XTI — Saturation current exponential increase order**

3 (default) | nonnegative scalar

Order of the exponential increase in the saturation current as temperature increases. The value must be greater than or equal to 0.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is only visible when you select Device temperature for the **Model temperature dependence using** parameter.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0 K.

**Dependencies**

This parameter is only visible when you select Fixed temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

Temperature at which the transistor parameters are measured. The value must be greater than 0 K.

## Version History

Introduced in R2008a

## References

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

Environment Parameters | NPN Bipolar Transistor

### **Functions**

subcircuit2ssc

### **Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# SPICE PJFET

SPICE-compatible P-Channel JFET

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



## Description

The SPICE PJFET block represents a SPICE-compatible P-channel junction field-effect transistor (PJFET). If the voltage applied to the gate port, **gx**, is greater than the voltage applied to the source port, **sx**, the current between the source port and drain port, **dx**, is reduced.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

## Equations

Variables for the SPICE PJFET block equations include:

- Variables that you define by specifying parameters for the SPICE PJFET block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2130.
- Geometry-adjusted variables, which depend on several of the values that you specify using parameters for the SPICE PJFET block. For more information, see “Geometry-Adjusted Variables” on page 1-2125.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE PJFET block or by specifying parameters for both the SPICE PJFET block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2067.
- Temperature-dependent variables. For more information, see “Temperature Dependence” on page 1-2129.
- Minimal conductance,  $GMIN$ , which is  $1e-12$  1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2127.

## Geometry-Adjusted Variables

Several variables in the equations for the P-channel junction field-effect transistor model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on

variables that you define by specifying SPICE PJFET block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$BETA_d$	Geometry-adjusted transconductance	$BETA_d = BETA * AREA * SCALE$
$CGD_d$	Geometry-adjusted zero-bias gate-drain capacitance	$CGD_d = CGD * AREA * SCALE$
$CGS_d$	Geometry-adjusted zero-bias gate-source capacitance	$CGS_d = CGS * AREA * SCALE$
$IS_d$	Geometry-adjusted saturation current	$IS_d = IS * AREA * SCALE$
$RS_d$	Geometry-adjusted source resistance	$RS_d = \frac{RS}{AREA * SCALE}$
$RD_d$	Geometry-adjusted drain resistance	$RD_d = \frac{RD}{AREA * SCALE}$

### Transistor Temperature

You can use these options to define transistor temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent of the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PJFET block is set to `Fixed` temperature. For this model, the block sets  $T$  equal to *TFIXED*.
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PJFET block is set to `Device` temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is not an Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- *TOFFSET* is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of  $1e-12$  1/0hm. To specify a different value:

- 1 If there is not an Environment Parameters block in the transistor circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Source-Gate Current-Voltage Model

This table shows the equations that define the relationship between the source-gate current,  $I_{sg}$ , and the source-gate voltage,  $V_{sg}$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2046.

Applicable Range of $V_{sg}$ Values	Corresponding $I_{sg}$ Equation
$V_{sg} > 80 * V_t$	$I_{sg} = IS_d * \left( \left( \frac{V_{sg}}{V_t} - 79 \right) e^{80} - 1 \right) + V_{sg} * Gmin$
$80 * V_t \geq V_{sg}$	$I_{sg} = IS_d * \left( e^{V_{sg}/V_t} - 1 \right) + V_{sg} * Gmin$

Where:

- $IS_d$  is the geometry-adjusted saturation current.
- $V_t$  is the thermal voltage, such that  $V_t = ND * k * T / q$ .
- $ND$  is the emission coefficient.
- $q$  is the elementary charge on an electron.
- $k$  is the Boltzmann constant.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2126
- $GMIN$  is the transistor minimum conductance. or more information, see “Minimal Conduction” on page 1-2127

### Drain-Gate Current-Voltage Model

This table shows the relationship between the drain-gate current,  $I_{dg}$ , and the drain-gate voltage,  $V_{dg}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{dg}$ Values	Corresponding $I_{dg}$ Equation
$V_{dg} > 80 * V_t$	$I_{dg} = IS_d * \left( \left( \frac{V_{dg}}{V_t} - 79 \right) e^{80} - 1 \right) + V_{dg} * Gmin$
$80 * V_t \geq V_{dg}$	$I_{dg} = IS_d * \left( e^{V_{dg}/V_t} - 1 \right) + V_{dg} * Gmin$

### Source-Drain Current-Voltage Model

This table shows the relationship between the source-drain current,  $I_{sd}$ , and the source-drain voltage,  $V_{sd}$ , in normal mode ( $V_{sd} \geq 0$ ). As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{sg}$ and $V_{dg}$ Values	Corresponding $I_{sd}$ Equation
$V_{sg} - V_{to} \leq 0$	$I_{sd} = 0$
$0 < V_{sg} - V_{to} \leq V_{sd}$	$I_{sd} = \beta_d (V_{sg} - V_{to})^2 (1 + \lambda V_{sd})$
$0 < V_{sd} < V_{sg} - V_{to}$	$I_{sd} = \beta_d V_{sd} (2(V_{sg} - V_{to}) - V_{sd}) (1 + \lambda V_{sd})$

Where:

- $V_{to}$  is the threshold voltage.
- $\beta_d$  is the geometry-adjusted transconductance.
- $\lambda$  is the channel modulation.

This table shows the relationship between the source-drain current,  $I_{sd}$ , and the source-drain voltage,  $V_{sd}$ , in inverse mode ( $V_{sd} < 0$ ). As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{sggs}$ and $V_{dg}$ Values	Corresponding $I_{sd}$ Equation
$V_{dg} - V_{to} \leq 0$	$I_{sd} = 0$
$0 < V_{dg} - V_{to} \leq -V_{sd}$	$I_{sd} = -\beta_d (V_{dg} - V_{to})^2 (1 - \lambda V_{sd})$
$0 < -V_{sd} < V_{dg} - V_{to}$	$I_{sd} = \beta_d V_{sd} (2(V_{dg} - V_{to}) + V_{sd}) (1 - \lambda V_{sd})$

### Junction Charge Model

This table shows the relationship between the source-gate charge,  $Q_{sg}$ , and the source-gate voltage,  $V_{sg}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{sg}$ Values	Corresponding $Q_{sg}$ Equation
$V_{sg} < FC * VJ$	$Q_{sg} = \frac{CGS_d * VJ * \left(1 - \left(1 - \frac{V_{sg}}{VJ}\right)^{1 - MG}\right)}{1 - MG}$
$V_{sg} \geq FC * VJ$	$Q_{sg} = CGS_d * \left( F1 + \frac{F3 * (V_{sg} - FC * VJ) + \frac{MG * (V_{sg}^2 - (FC * VJ)^2)}{2 * VJ}}{F2} \right)$

Where:

- $FC$  is the capacitance coefficient.
- $VJ$  is the junction potential.
- $CGS_d$  is the zero-bias gate-source capacitance.
- $MG$  is the grading coefficient.
- $F1 = \frac{VJ * (1 - (1 - FC)^{1 - MG})}{1 - MG}$

- $F2 = (1 - FC)^{1 + MG}$
- $F3 = 1 - FC * (1 + MG)$

This table shows the relationship between the drain-gate charge,  $Q_{dg}$ , and the drain-gate voltage,  $V_{dg}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{dg}$ Values	Corresponding $Q_{dg}$ Equation
$V_{dg} < FC * VJ$	$Q_{dg} = \frac{CGD_d * VJ * \left(1 - \left(1 - \frac{V_{dg}}{VJ}\right)^{1 - MG}\right)}{1 - MG}$
$V_{dg} \geq FC * VJ$	$Q_{dg} = CGD_d * \left( F1 + \frac{F3 * (V_{dg} - FC * VJ) + \frac{MG * (V_{dg}^2 - (FC * VJ)^2)}{2 * VJ}}{F2} \right)$

Where:

- $CGD_d$  is the geometry-adjusted zero-bias gate-drain capacitance.

#### Temperature Dependence

The block provides this relationship between the saturation current  $IS$  and the transistor temperature  $T$ :

$$IS(T) = IS_d * (T/T_{meas})^{\frac{XTI}{ND}} * e^{\left(\frac{T}{T_{meas}} - 1\right) * \frac{EG}{V_t}}$$

Where:

- $IS_d$  is the geometry-adjusted saturation current.
- $T_{meas}$  is the parameter extraction temperature.
- $XTI$  is the saturation current temperature exponent.
- $EG$  is the energy gap.
- $V_t$  is the thermal voltage, such that  $V_t = ND * k * T / q$ .
- $ND$  is the emission coefficient.

The relationship between the junction potential,  $VJ$ , and the transistor temperature  $T$  is

$$VJ(T) = VJ * \left(\frac{T}{T_{meas}}\right) - \frac{3 * k * T}{q} * \log\left(\frac{T}{T_{meas}}\right) - \left(\frac{T}{T_{meas}}\right) * EG_{T_{meas}} + EG_T$$

Where:

- $VJ$  is the junction potential.
- $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
- $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The relationship between the gate-source junction capacitance,  $CGS$ , and the transistor temperature,  $T$ :

$$CGS(T) = CGS_d * \left[ 1 + MG * \left( 400e - 6 * (T - T_{meas}) - \frac{VJ(T) - VJ}{VJ} \right) \right]$$

Where:

- $CGS_d$  is the geometry-adjusted zero-bias gate-source capacitance.

The block uses the  $CGS(T)$  equation to calculate the gate-drain junction capacitance by substituting  $CGD_d$ , the zero-bias gate-drain capacitance, for  $CGS_d$ .

The relationship between the transconductance,  $\beta$ , and the transistor temperature  $T$  is

$$\beta(T) = \beta_d * \left( \frac{T}{T_{meas}} \right)$$

Where  $\beta_d$  is the geometry-adjusted transconductance.

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### gx – Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal.

#### dx – Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal.

#### sx – Source terminal

electrical

Electrical conserving port associated with the transistor source terminal.

## Parameters

### Main

#### Device area, AREA – Device area

1.0 m<sup>2</sup> (default) | positive scalar

Transistor area. The value must be greater than 0.

#### Number of parallel devices, SCALE – Number of parallel devices

1 (default) | positive integer

Number of parallel transistors the block represents. The value must be an integer greater than 0.



**Threshold voltage, VT0 — Threshold voltage**

-2 V (default) | scalar

Source-gate voltage above which the transistor produces a nonzero drain current.

**Transconductance, BETA — Channel modulation**1e-4 A/m<sup>2</sup>/V<sup>2</sup> (default) | nonnegative scalar

Derivative of drain current with respect to gate voltage. The value must be greater than or equal to 0.

**Channel modulation, LAMBDA — Channel modulation**

0 1/V (default) | scalar

Channel modulation.

**Saturation current, IS — Saturation current**1e-14 A/m<sup>2</sup> (default) | nonnegative scalar

Magnitude of the current that the gate-current equation approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

**Drain resistance, RD — Series resistance**0.01 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Transistor drain resistance. The value must be greater than or equal to 0.

**Source resistance, RS — Series resistance**0.0001 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Transistor source resistance. The value must be greater than or equal to 0.

**Emission coefficient, N — Emission coefficient**

1 (default) | positive scalar

Transistor emission coefficient or ideality factor. The value must be greater than 0.

**Junction Capacitance****Model junction capacitance — Junction capacitance model**

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model.
- Yes — Specify zero-bias junction capacitance, junction potential, grading coefficient, forward-bias depletion capacitance coefficient, and transit time.

**Dependencies**

Selecting Yes exposes related parameters.

**Zero-bias GS capacitance, CGS — Zero-bias junction capacitance**0 F/m<sup>2</sup> (default) | nonnegative scalar

Value of the capacitance placed between the gate and the source. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Zero-bias GD capacitance, CGD — Zero-bias junction capacitance**

0 F/m<sup>2</sup> (default) | nonnegative scalar

Value of the capacitance placed between the gate and the drain. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Junction potential, VJ — Junction potential**

1 V (default) |  $VJ > 0.01$  | scalar

Junction potential, *VJ*. The value must be greater than 0.01 V.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Grading coefficient, M — Grading coefficient**

0.5 (default) |  $0 < M < 0.9$  | scalar

Grading coefficient, *M*. The value must be greater than 0 and less than 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Capacitance coefficient, FC — Capacitance coefficient**

0.5 (default) |  $0 \leq FC < 0.95$  | scalar

Fitting coefficient, *FC*, that quantifies the decrease of the depletion capacitance with applied voltage. The value must be greater than or equal to 0 and less than 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- No — Do not specify an initial condition for the model.
- Yes — Specify the initial transistor voltage.

---

**Note** The SPICE PJFET block applies the initial transistor voltage across the junction capacitors and not across the ports.

---

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

Selecting Yes for this parameter exposes the **Initial condition voltage, ICVDS** and **Initial condition voltage, ICVGS** parameters.

#### **Initial condition voltage, ICVDS — Initial voltage**

0 V (default) | scalar

Drain-source voltage at the start of the simulation.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

#### **Initial condition voltage, ICVGS — Initial voltage**

0 V (default) | scalar

Gate-source voltage at the start of the simulation.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

### **Temperature**

#### **Model temperature dependence using — Temperature dependence model**

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Transistor Temperature” on page 1-2126.

#### **Dependencies**

Selecting **Device temperature** exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting **Fixed temperature** exposes the **Fixed circuit temperature, TFIXED** parameter.

#### **Saturation current temperature exponent, XTI — Saturation current temperature exponent**

0 (default) | positive scalar

Order of the exponential increase in the saturation current as temperature increases. The value must be greater than 0.

#### **Activation energy, EG — Activation energy**

1.11 eV (default) |  $EG \geq 0.1$  | scalar

Transistor activation energy. The value must be greater than or equal to 0.1.

#### **Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0.

**Dependencies**

This parameter is only visible when you select Fixed temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

Temperature at which the transistor parameters are measured. The value must be greater than 0.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is only visible when you select Device temperature for the **Model temperature dependence using** parameter.

## Version History

Introduced in R2008a

## References

[1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | P-Channel JFET | SPICE NJFET

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# SPICE PMOS

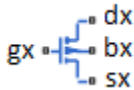
SPICE-compatible P-Channel MOSFET

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



## Description

The SPICE PMOS block represents a SPICE-compatible positive-channel (P-Channel) metal-oxide semiconductor (MOS) field-effect transistor (FET). If the gate-source voltage decreases, the channel conductance increases. If the gate-source voltage is increased, the channel conductance decreases.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

## Equation Variables

Variables for the SPICE PMOS block equations include:

- Variables that you define by specifying parameters for the SPICE PMOS block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2156.
- Geometry-adjusted variables, which depend on several of the values that you specify using parameters for the SPICE PMOS block. For more information, see “Geometry-Adjusted Variables” on page 1-2135.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE PMOS block or by specifying parameters for both the SPICE PMOS block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2136.
- Minimal conductance,  $GMIN$ , which is  $1e-12$  1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2137.
- Thermal voltage,  $V_{tn}$ . For more information, see “Thermal Voltage” on page 1-2137.

## Geometry-Adjusted Variables

Several variables in the equations for the SPICE P-channel MOSFET model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on variables that you define by specifying SPICE PMOS block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$KP_d$	Geometry-adjusted transconductance	$KP_d = KP * AREA * SCALE$
$IS_d$	Geometry-adjusted bulk saturation current	$IS_d = IS * AREA * SCALE$
$JS_d$	Geometry-adjusted bulk junction saturation current density	$JS_d = JS * AREA * SCALE$
$CBD_d$	Geometry-adjusted zero-bias bulk-drain capacitance	$CBD_d = CBD * AREA * SCALE$
$CBS_d$	Geometry-adjusted zero-bias bulk-source capacitance	$CBS_d = CBS * AREA * SCALE$
$CGSO_d$	Geometry-adjusted gate-source overlap capacitance	$CGSO_d = CGSO * AREA * SCALE$
$CGDO_d$	Geometry-adjusted gate-drain overlap capacitance	$CGDO_d = CGDO * AREA * SCALE$
$CGBO_d$	Geometry-adjusted gate-bulk overlap capacitance	$CGBO_d = CGBO * AREA * SCALE$
$CJ$	Geometry-adjusted bottom capacitance per junction area	$CJ_d = CJ * AREA * SCALE$
$CJSW$	Geometry-adjusted sidewall capacitance per junction perimeter	$CJSW_d = CJSW * AREA * SCALE$
$RD_d$	Geometry-adjusted drain resistance	$RD_d = \frac{RD}{AREA * SCALE}$
$RS_d$	Geometry-adjusted source resistance	$RS_d = \frac{RS}{AREA * SCALE}$
$RSH_d$	Geometry-adjusted sheet resistance	$RSH_d = \frac{RSH}{AREA * SCALE}$

### Transistor Temperature

There are two different options for defining transistor temperature,  $T$ :

- Fixed temperature — The block uses a temperature that is independent of the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PMOS block is set to Fixed temperature. For this model, the block sets  $T$  equal to *TFIXED*.
- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PMOS block is set to Device temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is not an Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- $TOFFSET$  is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of  $1e-12$  1/0hm. To specify a different value:

- 1 If there is not already an Environment Parameters block in the circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Thermal Voltage

$V_{tn}$  is the thermal voltage, which is defined as

$$V_{tn} = N \frac{k * T}{q}$$

Where:

- $N$  is the emission coefficient.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2136.
- $k$  is the Boltzmann constant.
- $q$  is the elementary charge on an electron.

### Parameters Calculations

The tables show how the SPICE PMOS block determines some of its parameters based on values that you specify.

#### Drain Resistance

Parameter Values			Geometry-Adjusted Transistor Drain Resistance
Drain resistance, $RD$	Sheet resistance, $RSH$	Number of drain squares, $NRD$	
NaN	NaN	NaN	0
NaN	$RSH$	NaN	0
NaN	NaN	$NRD$	0
$RD$	NaN or $RSH$	NaN or $NRD$	$RD_d$
NaN	$RSH$	$NRD$	$RSH_d * NRD$

**Source Resistance**

Parameter Values			Geometry-Adjusted Transistor Source Resistance
Source resistance, RS	Sheet resistance, RSH	Number of source squares, NRS	
NaN	NaN	NaN	0
NaN	RSH	NaN	0
NaN	NaN	NRS	0
RS	NaN or RSH	NaN or NRS	$RS_d$
NaN	RSH	NRS	$RSH_d * NRS$

**Transconductance and Surface Mobility**

Parameter Values			Geometry-Adjusted Transconductance (level 1), in A/V <sup>2</sup>	Geometry-Adjusted Transconductance (level 3), in A/V <sup>2</sup>	Surface mobility (level 3), in cm <sup>2</sup> /s/V
Oxide thickness, TOX	Surface mobility, U0	Transconductance, KP			
NaN	NaN	NaN	2e-5 (default value)	2e-5 (default value)	600 (default value)
NaN	NaN	KP	$KP_d$	$KP_d$	600
NaN	U0	NaN	2e-5	$U0 * EPXox / 1e-7$	U0
NaN	U0	KP	$KP_d$	$KP_d$	U0
TOX	NaN	NaN	$600 * EPXox / TOX$	$600 * EPXox / TOX$	600
TOX	NaN	KP	$KP_d$	$KP_d$	600
TOX	U0	NaN	$U0 * EPXox / TOX$	$U0 * EPXox / TOX$	U0
TOX	U0	KP	$KP_d$	$KP_d$	U0



## Oxide Thickness and Threshold Voltage

Parameter Values				Surface potential, PHI (level 1), in V	Threshold voltage, VTO (level 1), in V	Surface potential, PHI (level 3), in V	Threshold voltage, VTO (level 3), in V
Oxide thickness, TOX	Substrate doping, NSUB	Surface potential, PHI	Threshold voltage, VTO				
NaN	NaN	NaN	NaN	0.6 (default value)	0 (default value)	0.6 (default value)	0 (default value)
NaN	NaN	NaN	VTO	0.6	VTO	0.6	VTO
NaN	NaN	PHI	NaN	PHI	0	PHI	0
NaN	NaN	PHI	VTO	PHI	VTO	PHI	VTO
NaN	NSUB	NaN	NaN	0.6	0	PHI (1e-7, NSUB)	VTO (1e-7, NSUB)
NaN	NSUB	NaN	VTO	0.6	VTO	PHI (1e-7, NSUB)	VTO
NaN	NSUB	PHI	NaN	PHI	0	PHI	VTO (1e-7, NSUB)
NaN	NSUB	PHI	VTO	PHI	VTO	PHI	VTO
TOX	NaN	NaN	NaN	0.6	0	0.6	0
TOX	NaN	NaN	VTO	0.6	VTO	0.6	VTO
TOX	NaN	PHI	NaN	PHI	0	PHI	0
TOX	NaN	PHI	VTO	PHI	VTO	PHI	VTO
TOX	NSUB	NaN	NaN	PHI (NSUB, TOX)	VTO (NSUB, TOX)	PHI (NSUB, TOX)	VTO (NSUB, TOX)
TOX	NSUB	NaN	VTO	PHI (NSUB, TOX)	VTO	PHI (NSUB, TOX)	VTO
TOX	NSUB	PHI	NaN	PHI	VTO (NSUB, TOX)	PHI	VTO (NSUB, TOX)
TOX	NSUB	PHI	VTO	PHI	VTO	PHI	VTO

Where PHI (NSUB, TOX), PHI (1e-7, NSUB), VTO (NSUB, TOX), and VTO (1e-7, NSUB) are obtained using these equations:

$$PHI = 2 \frac{kT}{q} \ln \left( \frac{NSUB}{ni} \right)$$

$$GAMMA = \frac{\sqrt{2q\epsilon_{si}NSUB}}{C_{ox}}$$

$$V_{FB} = \varphi_{MS} - \frac{qNSS}{C_{ox}}$$

$$V_{T0} = V_{FB} - PHI - GAMMA * \sqrt{PHI}.$$

### Bulk-Source Diode Model

The table shows the equations that define the relationship between the source-bulk current,  $I_{sb}$ , and voltage,  $V_{sb}$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2152.

Applicable Range of $V_{sb}$ Values	Corresponding $I_{gs}$ Equation
$V_{sb} > 80 * V_{tn}$	$I_{sb} = IS_{sb} * \left( \left( \frac{V_{sb}}{V_{tn}} - 79 \right) e^{80} - 1 \right) + V_{sb} * G_{min}$
$80V_{tn} \geq V_{sb}$	$I_{sb} = IS_{sb} * \left( e^{V_{sb}/V_{tn}} - 1 \right) + V_{sb} * G_{min}$

Where:

- $IS_{sb}$  is the bulk saturation current, such that, if:
  - $JS_d \neq 0$  and  $AS \neq 0$ ,  $IS_{sb} = JS_d * AS$ .

Where:

- $JS_d$  is the geometry-adjusted bulk junction saturation current density.
- $AS$  is the source area.
- If  $JS_d = 0$  or  $AS = 0$ ,  $IS_{sb} = IS_d$ , where  $IS_d$  is the geometry-adjusted bulk saturation current.
- $V_{tn}$  is the thermal voltage. For more information, see “Thermal Voltage” on page 1-2137.
- $G_{min}$  is the minimal conductance. For more information, see “Minimal Conduction” on page 1-2137.

### Bulk-Drain Diode Model

The table shows the equations that define the relationship between the drain-bulk current,  $I_{db}$ , and voltage,  $V_{db}$ . As applicable, the model parameters are first adjusted for temperature. For more information, see “Temperature Dependence” on page 1-2152.

Applicable Range of $V_{db}$ Values	Corresponding $I_{db}$ Equation
$V_{db} > 80 * V_{tn}$	$I_{db} = IS_{db} * \left( \left( \frac{V_{db}}{V_{tn}} - 79 \right) e^{80} - 1 \right) + V_{db} * G_{min}$
$80V_{tn} \geq V_{db}$	$I_{db} = IS_{db} * \left( e^{V_{db}/V_{tn}} - 1 \right) + V_{db} * G_{min}$

Where:

- $IS_{db}$  is the bulk drain current, such that:
  - If  $JS_d \neq 0$  and  $AD \neq 0$ ,  $IS_{db} = JS_d * AD$ .

Where:

- $JS_d$  is the geometry-adjusted bulk junction saturation current density.
- $AD$  is the drain area.
- If  $JS_d = 0$  or  $AD = 0$ ,  $IS_{db} = IS_d$ , where  $IS_d$  is the geometry-adjusted bulk saturation current.
- $V_{tn}$  is the thermal voltage. For more information, see “Thermal Voltage” on page 1-2137.
- $G_{min}$  is the minimal conductance. For more information, see “Minimal Conduction” on page 1-2137.

### Level 1 Drain Current Model

This table shows relationship between the drain current  $I_{sd}$  and the source-drain voltage  $V_{sd}$  in normal mode ( $V_{sd} \geq 0$ ). As applicable, model parameters are first adjusted for temperature.

#### Normal Mode

Applicable Range of $V_{sg}$ and $V_{sd}$ Values	Corresponding $I_{sd}$ Equation
$V_{sg} - V_{on} \leq 0$	$I_{sd} = 0$
$0 < V_{sg} - V_{on} \leq V_{sd}$	$I_{sd} = BETA * (V_{sg} - V_{on})^2 \frac{(1 + LAMBDA * V_{sd})}{2}$
$0 < V_{sd} < V_{sg} - V_{on}$	$I_{sd} = BETA * V_{sd} \left( (V_{sg} - V_{on}) - \frac{V_{sd}}{2} \right) (1 + LAMBDA * V_{sd})$

Where:

- $V_{on}$  depends on  $V_{sb}$  and  $PHI$ .

Applicable Relationship of $V_{sb}$ and $PHI$ Values	Corresponding $V_{on}$ Equation
$V_{sb} \leq 0$	$V_{on} = MTYPE * VBI + GAMMA \sqrt{PHI - V_{sb}}$
$0 < V_{sb} \leq 2 * PHI$	$V_{on} = MTYPE * VBI + GAMMA \left( \sqrt{PHI} - \frac{V_{sb}}{2\sqrt{PHI}} \right)$
$V_{sb} > 2 * PHI$	$V_{on} = MTYPE * VBI$

- $MTYPE$  is -1.
- $BETA$  is  $BETA = (KP_d * WIDTH) / (LENGTH - 2 * LD)$
- $KP$  is:
  - The **Transconductance, KP**, if this parameter has a numerical value.
  - $U0 * 3.9 * \epsilon_0 / TOX$ , if **Transconductance, KP** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $WIDTH$  is the channel width.
- $LENGTH$  is the channel length.

- $LD$  is the lateral diffusion.
- $VBI$  is a built-in voltage value the block uses in calculations. The value is a function of temperature. For a detailed definition, see “Temperature Dependence” on page 1-2152.
- $PHI$  is:
  - The **Surface potential, PHI**, if this parameter has a numerical value.
  - $2 * kT_{meas} / q * \log(NSUB / n_i)$ , if **Surface potential, PHI** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $LAMBDA$  is the channel modulation.
- $GAMMA$  is:
  - The **Bulk threshold, GAMMA**, if this parameter has a numerical value.
  - $TOX * \sqrt{2 * 11.7 * \epsilon_0 * q * NSUB} / (3.9 * \epsilon_0)$ , if **Bulk threshold, GAMMA** is NaN and you specify values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters.
- $\epsilon_0$  is the permittivity of free space, 8.854214871e-12 F/m.
- $n_i$  is the carrier concentration of intrinsic silicon, 1.45e10 cm<sup>-3</sup>.

This table shows relationship between the drain current  $I_{sd}$  and the source-drain voltage  $V_{sd}$  in inverse mode ( $V_{sd} < 0$ ). As applicable, model parameters are first adjusted for temperature.

**Inverse Mode**

Applicable Range of $V_{dg}$ and $V_{sd}$ Values	Corresponding $I_{sd}$ Equation
$V_{dg} - V_{on} \leq 0$	$I_{sd} = 0$
$0 < V_{dg} - V_{on} \leq -V_{sd}$	$I_{sd} = -BETA(V_{dg} - V_{on})^2(1 - LAMBDA * V_{sd})/2$
$0 < -V_{sd} < V_{dg} - V_{on}$	$I_{sd} = BETA * V_{sd}((V_{dg} - V_{on}) + V_{sd}/2)(1 - LAMBDA * V_{sd})$

$V_{on}$  depends on  $V_{db}$  and  $PHI$ .

Applicable Relationship of $V_{db}$ and $PHI$ Values	Corresponding $V_{on}$ Equation
$V_{db} \leq 0$	$V_{on} = MTYPE * VBI + GAMMA \sqrt{PHI - V_{db}}$
$0 < V_{db} \leq 2 * PHI$	$V_{on} = MTYPE * VBI + GAMMA \left( \sqrt{PHI} - \frac{V_{db}}{2\sqrt{PHI}} \right)$
$V_{db} > 2 * PHI$	$V_{on} = MTYPE * PHI$

**Level 3 Drain Current Model**

The block provides the following model for drain current  $I_{sd}$  in normal mode ( $V_{sd} \geq 0$ ) after adjusting the applicable model parameters for temperature.

$$I_{SD} = I_{SD0} * Scale_{VMAX} * Scale_{LChan} * Scale_{INV}$$

Where:

- $I_{DS0}$  is the Basic Drain Current Model on page 1-2143.
- $Scale_{VMAX}$  is the Velocity Saturation Scaling on page 1-2145.
- $Scale_{LChan}$  is the Channel Length Modulation Scaling on page 1-2145.
- $Scale_{INV}$  is the Weak Inversion Scaling on page 1-2146.

The block uses the same model for drain current in inverse mode ( $V_{sd} < 0$ ), with the following substitutions:

- $V_{sb} \equiv V_{sb} - V_{sd}$
- $V_{sg} \equiv V_{sg} - V_{sd}$
- $V_{sd} \equiv -V_{sd}$

### Basic Drain Current Model

The relationship between the drain current  $I_{sd}$  and the source-drain voltage  $V_{sd}$  is

$$I_{SD0} = BETA * F_{gate} * \left( V_{SGX} - V_{TH} - \frac{1 + F_B}{2} * V_{SDX} \right) * V_{SDX}$$

Where:

- $BETA$  is calculated as described in “Level 1 Drain Current Model” on page 1-2141.
- $F_{GATE}$  is calculated as

$$F_{gate} = \frac{1}{1 + THETA * (V_{sgx} - V_{TH})}$$

Where:

- $THETA$  models the dependence of the mobility on the gate-source voltage.
- $V_{sgx} = \max(V_{SG}, V_{on})$
- If you specify a nonzero value for the **Fast surface state density, NFS** parameter, the block calculates  $V_{on}$  using this equation:

$$V_{on} = V_{TH} + x_n V_T$$

Otherwise,

$$V_{on} = V_{TH}$$

- The block calculates  $x_n$  as

$$x_n = 1 + \frac{q * NFS}{COX} + \frac{\left( GAMMA * F_s * \sqrt{V_{bulk}} + \frac{F_n * V_{bulk}}{WIDTH} \right)}{2 * V_{bulk}}$$

- The block calculates  $V_{bulk}$  as follows:

- If

$$V_{SB} \leq 0,$$

$$V_{bulk} = PHI - V_{SB}.$$

- Otherwise, the block calculates  $V_{bulk}$  as

$$V_{bulk} = \frac{PHI}{\left(1 + \frac{V_{SB}}{2 * PHI}\right)^2}$$

- Thermal voltage such that

$$V_T = \frac{kT}{q}$$

- The block calculates  $V_{TH}$  using the following equation:

$$V_{TH} = V_{BI} - \frac{8.15e^{-22} * ETA}{COX * (LENGTH - 2 * LD)^3} * V_{SD} + GAMMA * F_s * \sqrt{V_{bulk}} + F_n * V_{bulk}$$

- For information about how the block calculates  $V_{BI}$ , see “Temperature Dependence” on page 1-2152.
- **ETA** is the **Vds dependence threshold volt, ETA**.
- $COX = \frac{\epsilon_{ox}}{TOX}$ ,

Where  $\epsilon_{ox}$  is the permittivity of the oxide and **TOX** is the **Oxide thickness, TOX**.

- If you specify a nonzero value for the **Junction depth, XJ** parameter and a value for the **Substrate doping, NSUB** parameter, the block calculates  $F_s$  using these equations:

$$\alpha = \frac{2\epsilon_{si}}{qNSUB}$$

$$XD = \sqrt{\alpha}$$

$$wc = .0631353 + .8013292 * \frac{XD * \sqrt{V_{bulk}}}{XJ} - .01110777 * \left(\frac{XD * \sqrt{V_{bulk}}}{XJ}\right)^2 + \frac{LD}{XJ}$$

$$F_s = 1 - \left( wc * \sqrt{1 - \left(\frac{XD * \sqrt{V_{bulk}}}{XJ + XD * \sqrt{V_{bulk}}}\right)^2} - \frac{LD}{XJ} \right)$$

Where  $\epsilon_{si}$  is the permittivity of silicon.

Otherwise,

$$F_s = 1$$

- The block calculates  $F_B$  as

$$F_B = \frac{GAMMA * F_s}{4 * \sqrt{V_{bulk}}} + F_n$$

- The block calculates  $F_n$  as

$$F_n = \frac{DELTA * \pi * \epsilon_{si}}{2 * COX * WIDTH}$$

- *DELTA* is the width effect on threshold.
- $V_{SDX}$  is the lesser of  $V_{SD}$  and the saturation voltage,  $V_{dsat}$ .
- If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block calculates  $V_{dsat}$  using the following equation:

$$V_{dsat} = \frac{V_{sgx} - V_{TH}}{1 + F_B} + \frac{(LENGTH - 2 * LD) * VMAX}{UO * F_{gate}}$$

$$- \sqrt{\left(\frac{V_{sgx} - V_{TH}}{1 + F_B}\right)^2 + \left(\frac{(LENGTH - 2 * LD) * VMAX}{UO * F_{gate}}\right)^2}$$

Otherwise, the block calculates  $V_{dsat}$  as

$$V_{dsat} = \frac{V_{sgx} - V_{TH}}{1 + F_B}$$

### Velocity Saturation Scaling

If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block calculates  $Scale_{VMAX}$  as

$$Scale_{VMAX} = \frac{1}{1 + \frac{UO * F_{gate}}{(LENGTH - 2 * LD) * VMAX} * V_{SDX}}$$

Otherwise,

$$Scale_{VMAX} = 1$$

### Channel Length Modulation Scaling

The block scales the drain current to account for channel length modulation if  $V_{SD} > V_{dsat}$  and the **Max carrier drift velocity, VMAX** is less than or equal to zero or  $\alpha$  is nonzero.

The block scales the drain current using the following equation:

$$Scale_{LChan} = \frac{1}{1 - \frac{\Delta l}{(LENGTH - 2 * LD)}}$$

To calculate  $\Delta l$  the block:

- 1 Calculates the intermediate value  $\Delta l_0$ .

- If you specify a positive value for the **Max carrier drift velocity, VMAX** parameter, the block computes the intermediate value  $g_{dsat}$  as the greater of 1e-12 and the result of the following equation:

$$I_{SD0} * \left(1 - \frac{1}{1 + Scale_{g_{dsat}} * V_{SDX}}\right) * Scale_{g_{dsat}}$$

Where:

$$Scale_{g_{dsat}} = \frac{UO * F_{gate}}{(LENGTH - 2 * LD) * VMAX}$$

Then, the block uses the following equation to calculate the intermediate value  $\Delta l_0$ :

$$\Delta l_0 = \sqrt{\left(\frac{KA * I_{SD}}{2 * (LENGTH - 2 * LD) * g_{dsat}}\right)^2 + KA * (V_{SD} - V_{dsat})} - \frac{KA * I_{SD}}{2 * (LENGTH - 2 * LD) * g_{dsat}}$$

Where

$$KA = KAPPA * \alpha .$$

- Otherwise, the block uses the following equation to calculate the intermediate value  $\Delta l_0$  as

$$\Delta l = \sqrt{KA * (V_{SD} - V_{dsat})}$$

- 2 The block checks for punch through and calculates  $\Delta l$ .

- If

$$\Delta l_0 > (LENGTH - 2 * LD) / 2,$$

the block calculates  $\Delta l$  using the following equation:

$$\Delta l = \left(1 - \frac{(LENGTH - 2 * LD)}{4 * \Delta l_0}\right) * (LENGTH - 2 * LD)$$

- Otherwise,

$$\Delta l = \Delta l_0 .$$

### Weak Inversion Scaling

If  $V_{SG}$  is less than  $V_{on}$ , the block calculates  $Scale_{INV}$  using the following equation:

$$Scale_{INV} = e^{\frac{V_{sg} - V_{on}}{\alpha_n * V_T}}$$

Otherwise,

$$Scale_{INV} = 1$$

### Junction Charge Model

The block models the “Junction Overlap Charges” on page 1-2146 and the “Bulk Junction Charges” on page 1-2147.

#### Junction Overlap Charges

The block calculates the following junction overlap charges:

- $Q_{SG} = CGSO_d * WIDTH * V_{sg}$

Where:

- $Q_{SG}$  is the source-gate overlap charge.
- $CGSO_d$  is the geometry adjusted gate-source overlap capacitance.



- *WIDTH* is the channel width.
- $Q_{DG} = CGDO_d * WIDTH * V_{dg}$

Where:

- $Q_{DG}$  is the drain-gate overlap charge.
- $CGDO_d$  is the geometry adjusted gate-drain overlap capacitance.
- $Q_{BG} = CGBO_d * (LENGTH - 2 * LD) * V_{bg}$

Where:

- $Q_{BG}$  is the bulk-gate overlap charge.
- $CGBO_d$  is the geometry adjusted gate-bulk overlap capacitance.
- *LENGTH* is the channel length.
- *LD* is the lateral diffusion.

### Bulk Junction Charges

This table shows relationship between the bulk-drain bottom junction charge  $Q_{bottom}$  and the junction voltage,  $V_{db}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{db}$ Values	Corresponding $Q_{bottom}$ Equation
$V_{db} < FC * PB$	$Q_{bottom} = \frac{CBD_d * PB * \left(1 - \left(1 - \frac{V_{db}}{PB}\right)^{1 - MJ}\right)}{1 - MJ} \text{ if } CBD > 0.$ $Q_{bottom} = \frac{CJ_d * AD * PB * \left(1 - \left(1 - \frac{V_{db}}{PB}\right)^{1 - MJ}\right)}{1 - MJ} \text{ otherwise.}$
$V_{db} \geq FC * PB$	$Q_{bottom} = CBD_d * \left( F1 + \frac{F3 * (V_{db} - FC * PB) + \frac{MJ * (V_{db}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right) \text{ if } CBD_d > 0.$ $Q_{bottom} = CJ_d * AD * \left( F1 + \frac{F3 * (V_{db} - FC * PB) + \frac{MJ * (V_{db}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right)$ <p>otherwise.</p>

Where:

- *PB* is the bulk junction potential.
- *FC* is the capacitance coefficient.

- $CBD_d$  is the geometry-adjusted zero-bias bulk-drain capacitance.
- $CJ_d$  is the geometry-adjusted bottom capacitance per junction area.
- $AD$  is the drain area.
- $MJ$  is the bottom grading coefficient.
- $$F1 = \frac{PB * (1 - (1 - FC)^{1 - MJ})}{1 - MJ}$$
- $F2 = (1 - FC)^{1 + MJ}$
- $F3 = 1 - FC * (1 + MJ)$

To calculate the bulk-source bottom junction charge, the block substitutes variables in the equations in the preceding table. The block substitutes:

- $V_{sb}$  replaces  $V_{db}$ .
- $AS$  for  $AD$
- $CBS_d$  for  $CBD_d$

This table shows relationship between the bulk-drain sidewall junction charge  $Q_{sidewall}$  and the junction voltage  $V_{db}$ . As applicable, model parameters are first adjusted for temperature.

Applicable Range of $V_{db}$ Values	Corresponding $Q_{sidewall}$ Equation
$V_{db} < FC * PB$	$Q_{sidewall} = \frac{CJ_{SW_d} * PD * PB * \left(1 - \left(1 - \frac{V_{db}}{PB}\right)^{1 - MJ_{SW}}\right)}{1 - MJ_{SW}}$
$V_{db} \geq FC * PB$	$Q_{sidewall} = CJ_{SW_d} * PD * \left( F1 + \frac{F3 * (V_{db} - FC * PB) + \frac{MJ_{SW} * (V_{db}^2 - (FC * PB)^2)}{2 * PB}}{F2} \right)$

Where:

- $CJ_{SW_d}$  is the geometry adjusted sidewall capacitance per junction perimeter.
- $PD$  is the drain perimeter.
- $MJ_{SW}$  is the side grading coefficient.
- $$F1 = \frac{PB * (1 - (1 - FC)^{1 - MJ_{SW}})}{1 - MJ_{SW}}$$
- $F2 = (1 - FC)^{1 + MJ_{SW}}$
- $F3 = 1 - FC * (1 + MJ_{SW})$

To calculate the bulk-source sidewall junction charge and the sidewall junction voltage, the block substitutes variables in the equations in the preceding table. The block substitutes:

- $V_{sb}$  replaces  $V_{db}$ .

- *PS* for *PD*

### Capacitance Model

The SPICE PMOS block allows you to model the transistor capacitance model in three different ways:

- No intrinsic capacitance model
- “Meyer Gate Capacitance Model” on page 1-2149
- “Charge Conservation Capacitance Model” on page 1-2150

### Meyer Gate Capacitance Model

This table shows the relationship between the operational regions of the transistor and the gate-bulk, gate-drain, and gate-source capacitances.

Operational region	Gate-Bulk, $C_{gb}$ , Gate-Drain, $C_{gd}$ , and Gate-Source, $C_{gs}$ , Equations	
Accumulation region, $V_{gb} < V_{FB}$	$C_{gb} = C_{oxt}$ $C_{gd} = 0$ $C_{gs} = 0$	
Depletion region, $V_{gs} < V_{TH}$	$C_{gb} = \frac{C_{oxt}}{\sqrt{1 + \frac{4}{\text{GAMMA}^2} * (V_{gb} - V_{FB})}}$ $C_{gd} = 0$ $C_{gs} = 0$	
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	if $V_{ds} < V_{satmin}$ then:	if $V_{ds} \geq V_{satmin}$ then:
	$C_{gb} = 0$ $C_{gd} = \frac{2}{3}(C_{oxt} - C_{gb}) \left( 1 - \frac{(V_{satmin})^2}{(2V_{satmin} - V_{ds})^2} \right)$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb}) \left( 1 - \frac{(V_{satmin} - V_{ds})^2}{(2V_{satmin} - V_{ds})^2} \right)$	$C_{gb} = 0$ $C_{gd} = 0$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb})$
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$C_{gb} = 0$ $C_{gd} = \frac{2}{3}(C_{oxt} - C_{gb}) * \left( 1 - \frac{(V_{gs} - V_{TH})^2}{(2 * (V_{gs} - V_{TH}) - V_{ds})^2} \right)$ $C_{gs} = \frac{2}{3}(C_{oxt} - C_{gb}) * \left( 1 - \frac{(V_{gs} - V_{TH} - V_{ds})^2}{(2 * (V_{gs} - V_{TH}) - V_{ds})^2} \right)$	

where:

- $C_{oxt} = \text{WIDTH} * (\text{LENGTH} - 2 * \text{LD}) * \text{COX} * \text{AREA} * \text{SCALE}$
- $V_{FB} = \text{VBI} * \text{MTYPE} - \text{PHI}$  is the flat-band voltage.
- $V_{satmin}$  is the minimum saturation voltage. It is a predefined parameter equal to 1 V.

These equations are continuous between the depletion region and the accumulation region, and discontinuous between the depletion and the inversion region. Other SPICE tools apply smoothing functions between the inversion and depletion regions.

$$C_{gb} = \frac{WIDTH * LENGTH * COX}{\sqrt{\left(1 + \frac{4}{GAMMA^2} * (V_{TH} - V_{bs} - V_{FB})\right)^m}} * smoothing$$

$$smoothing = \frac{1}{\left(1 + \frac{4}{GAMMA^2} * (V_{gs} - V_{TH})\right)^m}$$

where  $m$  is a predefined smoothing constant.

### Charge Conservation Capacitance Model

This table shows the relationship between the operational regions of the transistor and the gate, bulk, channel, drain, and source charges for a level 1 MOS.

Operational region	Level-1 Charges Equations
Accumulation region, $V_{gb} < V_{FB}$	$Q_g = Cox t * (V_{gb} - V_{FB})$ $Q_b = -Cox t * (V_{gb} - V_{FB})$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Depletion region, $V_{gs} < V_{TH}$	$Q_g = -0.5 * Cox t * GAMMA^2 * \left(1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})}\right)$ $Q_b = 0.5 * Cox t * GAMMA^2 * \left(1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})}\right)$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	$Q_g = Cox t * \left(V_{gs} - V_{FB} - PHI - \frac{V_{gs} - V_{TH}}{3}\right)$ $Q_b = -Cox t * (V_{TH} - V_{FB} - PHI)$ $Q_c = -Cox t * \left(V_{gs} - V_{TH} - \frac{V_{gs} - V_{TH}}{3}\right)$ $Q_d = 0$ $Q_s = Q_c - Q_d$

Operational region	Level-1 Charges Equations
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$Q_g = Cox_{t} * \left( V_{gs} - V_{FB} - PHI - \frac{V_{ds}}{2} + \frac{V_{ds}^2}{12(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_b = -Cox_{t} * (V_{TH} - V_{FB} - PHI)$ $Q_c = -Cox_{t} * \left( V_{gs} - V_{TH} - \frac{V_{ds}}{2} + \frac{V_{ds}^2}{12(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_d = -Cox_{t} * \left( \frac{V_{gs} - V_{TH}}{2} - \frac{3V_{ds}}{4} + \frac{V_{ds}^2}{8(V_{gs} - V_{TH} - 0.5V_{ds})} \right)$ $Q_s = Q_c - Q_d$

where:

- $Q_g$  is the gate charge.
- $Q_b$  is the bulk charge.
- $Q_d$  is the drain charge.
- $Q_s$  is the source charge.
- $Q_c = -(Q_g + Q_b) = Q_d + Q_s$  is the charge in channel.  $Q_c$  needs to be partitioned between  $Q_d$  and  $Q_s$ .

This table shows the relationship between the operational regions of the transistor and the gate, bulk, channel, drain, and source charges for a level-3 MOS.

Operational region	Level-3 Charges Equations
Accumulation region, $V_{gb} < V_{FB}$	$Q_g = Cox_{t} * (V_{gb} - V_{FB}) - SF_1 + SF_2$ $Q_b = -Cox_{t} * (V_{gb} - V_{FB}) - SF_1 + SF_2$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Depletion region, $V_{gs} < V_{TH}$	$Q_g = -0.5 * Cox_{t} * GAMMA^2 * \left( 1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})} \right) - SF_1 + SF_2$ $Q_b = 0.5 * Cox_{t} * GAMMA^2 * \left( 1 - \sqrt{1 + \frac{4}{GAMMA^2} * (V_{gb} - V_{FB})} \right) + SF_1 - SF_2$ $Q_c = 0$ $Q_d = 0$ $Q_s = Q_c - Q_d$

Operational region	Level-3 Charges Equations
Saturation region, $V_{gs} - V_{TH} < V_{ds}$	$Q_g = Cox_t * \left( V_{gs} - V_{FB} - PHI + ETA * V_{dsat} - \frac{V_{dsat}}{2} + \frac{1 + F_B}{12 * F_i} * V_{dsat}^2 \right)$ $Q_b = - Cox_t * \left( V_{TH} - V_{FB} - PHI + ETA * V_{dsat} - \frac{F_b}{2} * V_{ds} - \frac{F_B * (1 + F_B)}{12 * F_i} * V_{dsat}^2 \right)$ $Q_c = - Cox_t * \left( V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{dsat} + \frac{(1 + F_B)^2}{12 * F_i} * V_{dsat}^2 \right)$ $Q_d = 0$ $Q_s = Q_c - Q_d$
Linear region, $V_{gs} - V_{TH} > V_{ds}$	$Q_g = Cox_t * \left( V_{gs} - V_{FB} - PHI + ETA * V_{ds} - \frac{V_{ds}}{2} + \frac{1 + F_B}{12 * F_i} * V_{ds}^2 \right)$ $Q_b = - Cox_t * \left( V_{TH} - V_{FB} - PHI + ETA * V_{ds} + \frac{F_B}{2} * V_{ds} - \frac{F_B * (1 + F_B)}{12 * F_i} * V_{ds}^2 \right)$ $Q_c = - Cox_t * \left( V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{ds} + \frac{(1 + F_B)^2}{12 * F_i} * V_{ds}^2 \right)$ $Q_d = - Cox_t * \left( \frac{V_{gs} - V_{TH}}{2} - \frac{3(1 + F_B)}{2} * V_{ds} + \frac{(1 + F_B)^2}{8 * F_i} * V_{ds}^2 \right)$ $Q_s = Q_c - Q_d$

where:

- $V_{dsat}$  is the saturation voltage
- $F_B$  is the body effect coefficient
- $ETA$  is the drain-source voltage threshold coefficient
- $F_i = V_{gs} - V_{TH} - \frac{1 + F_B}{2} * V_{ds}$
- $SF_1 = -0.5 * Cox_t * GAMMA^2 * \left( 1 - \left( 1 + \frac{4}{GAMMA^2} * 2(V_{TH} - V_{bs} - V_{FB}) \right)^{0.5} \right)$  and  
 $SF_2 = Cox_t * (V_{TH} - V_{FB} - PHI)$  are smoothing factors between depletion and accumulation regions to help with convergence.

### Temperature Dependence

The transconductance as a function of the transistor temperature is

$$KP(T) = \frac{KP_d}{\left( \frac{T}{T_{meas}} \right)^{3/2}}$$

Where:

- $KP_d$  is the geometry-adjusted transconductance.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2136.

- $T_{meas}$  is the parameter extraction temperature.

The surface potential as a function of the transistor temperature is

$$PHI(T) = \frac{T}{T_{meas}} \left( PHI + \frac{kT_{meas}}{q} \left( \log \left( \frac{T_{meas}}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_{T_{meas}}}{T_{meas}} \right) \right) \right) - \frac{kT}{q} \left( \log \left( \frac{T}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_T}{T} \right) \right)$$

Where:

- $PHI$  is the surface potential.
- $k$  is the Boltzmann constant.
- $q$  is the elementary charge on an electron, 1.6021918e-19 C.
- $EG$  is the activation energy, such that:
  - $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
  - $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The built-in voltage as a function of the transistor temperature is

$$VBI(T) = VTO + MTYPE * \left( \frac{PHI(T) - PHI}{2} - GAMMA \sqrt{PHI} \right) + \frac{EG_{T_{meas}} - EG_T}{2}$$

Where:

- $VBI$  is the built-in voltage.
- $VTO$  is the threshold voltage.  $VTO$  depends on the value that you specify for the **Threshold voltage, VTO** parameter in the **DC currents** settings. If you specify a numerical value,  $VTO$  is evaluated as that value. If you specify a nonnumerical value (NAN) and you specify numerical values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters in the **Process** settings, then  $VTO$  is evaluated as  $\Phi - 3.25 + EG_{T_{meas}}/2 + MTYPE * PHI/2 - NSS * q * TOX / (3.9 * \epsilon_0) + MTYPE * (GAMMA * \sqrt{PHI} + PHI)$

Where:

- $\Phi$  depends on the gate type, which you specify using the **Gate type, TPG** parameter. If you specify Aluminum ( $\emptyset$ ),  $\Phi = 3.2$ . Otherwise,  $\Phi = 3.25 + EG_{T_{meas}}/2 - MTYPE * TPG * EG_{T_{meas}}/2$ , Where:
  - $MTYPE$  is the transistor type. For an P-channel MOSFET,  $MTYPE = -1$ .
  - $TPG$  represents the gate type and also depends on the option that you specify for the **Gate type, TPG** parameter in the **Process** settings. If you specify
    - Opposite of substrate (1) —  $TPG = 1$
    - Same as substrate (-1) —  $TPG = -1$

- $NSS$  is the surface state density.
- $TOX$  is the oxide thickness.
- $\epsilon_0$  is the permittivity of free space.
- $GAMMA$  is the bulk threshold.  $GAMMA$  depends on the value that you specify for the **Bulk threshold, GAMMA** parameter in the **DC currents** settings. If you specify a numerical value,  $GAMMA$  is evaluated as that value. If you specify a nonnumerical value (NAN) and you specify numerical values for both the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters in the **Process** settings, then  $VTO$  is evaluated as  $TOX * \sqrt{2 * 11.7 * \epsilon_0 * q * NSUB} / (3.9 * \epsilon_0)$ , where  $NSUB$  is the substrate doping.

The bulk saturation current as a function of the transistor temperature is

$$IS(T) = IS_d * e^{\frac{-qEG_T}{ND * kT}} + \frac{qEG_{T_{meas}}}{ND * kT_{meas}}$$

Where:

- $IS_d$  is the geometry-adjusted bulk saturation current.
- $ND$  is the emission coefficient.

The bulk junction saturation current density as a function of the transistor temperature is

$$JS(T) = JS_d * e^{\frac{-qEG_T}{ND * kT}} + \frac{qEG_{T_{meas}}}{ND * kT_{meas}}$$

Where:

- $JS_d$  is the geometry-adjusted bulk junction saturation current density.

The bulk junction potential as a function of the transistor temperature is

$$PB(T) = \frac{PB + \frac{kT_{meas}}{q} \left( \log \left( \frac{T_{meas}}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_{T_{meas}}}{T} \right) \right)}{T_{meas}/T} - \frac{kT}{q} \left( \log \left( \frac{T}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EG_T}{T} \right) \right)$$

Where:

- $PB$  is the bulk junction potential.

The bulk-drain junction capacitance as a function of the transistor temperature is

$$CBD(T) = CBD_d \frac{pbo + MJ * (4 * 10^4 * (T - 300.15) * pbo - (PB(T) - pbo))}{pbo + MJ * (4 * 10^4 * (T_{meas} - 300.15) * pbo - (PB - pbo))}$$

Where:

- $CBD_d$  is the geometry adjusted zero-bias bulk-drain capacitance.
- $MJ$  is the bottom grading coefficient.



$$pbo = \frac{PB + \frac{kT_{meas}}{q} \left( \log \left( \frac{T_{meas}}{300.15} \right)^3 + \frac{q}{k} \left( \frac{1.115}{300.15} - \frac{EGT_{meas}}{T} \right) \right)}{T_{meas}/300.15}$$

The block uses the  $CBD(T)$  equation to calculate:

- The bulk-source junction capacitance by substituting  $CBS_d$ , the geometry-adjusted zero-bias bulk-source capacitance, for  $CBD_d$ .
- The bottom junction capacitance by substituting  $CJ_d$ , the geometry-adjusted bottom capacitance per junction area for  $CBD_d$ .

The relationship between the sidewall junction capacitance  $CJSW$  and the transistor temperature,  $T$ , is

$$CJSW(T) = CJSW_d \frac{pbo + MJSW * (4 * 10^4 * (T - 300.15) * pbo - (PB(T) - pbo))}{pbo + MJSW * (4 * 10^4 * (T_{meas} - 300.15) * pbo - (PB - pbo))}$$

Where:

- $CJSW_d$  is the side geometry-adjusted sidewall capacitance per junction perimeter.
- $MJSW$  is the side grading coefficient.

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### gx – Gate terminal

electrical

Electrical conserving port associated with the transistor gate terminal.

#### dx – Drain terminal

electrical

Electrical conserving port associated with the transistor drain terminal.

#### sx – Source terminal

electrical

Electrical conserving port associated with the transistor source terminal.

#### bx – Bulk terminal

electrical

Electrical conserving port associated with the transistor bulk terminal.

## Parameters

### Model Selection

#### **MOS model – Drain current model**

Level 1 MOS (default) | Level 3 MOS

MOSFET drain current model options:

- Level 1 MOS — Use the “Level 1 Drain Current Model” on page 1-2141. This is the default option.
- Level 3 MOS — Use the “Level 3 Drain Current Model” on page 1-2142.

### Dependencies

The setting that you select for the **MOS model** affects the visibility of certain parameters in the **DC Currents** and **Process** settings.

### Dimensions

#### **Device area factor, AREA – Device area**

1.0 (default) | positive scalar

Transistor area factor for scaling. The value must be greater than 0.

#### **Number of parallel devices, SCALE – Number of parallel devices**

1 (default) | positive integer

Number of parallel MOS instances that the block represents. This parameter multiplies the output current and device charge. The value must be greater than 0.

#### **Length of channel, LENGTH – Source-to-drain channel length**

1e-4 m (default) | positive scalar

Length of the channel between the source and drain.

#### **Width of channel, WIDTH – Source-to-drain channel width**

1e-4 m (default) | positive scalar

Width of the channel between the source and drain.

#### **Area of drain, AD – Drain area**

0 m<sup>2</sup> (default) | nonnegative scalar

Area of the transistor drain diffusion. The value must be greater than or equal to 0.

#### **Area of source, AS – Source area**

0 m<sup>2</sup> (default) | nonnegative scalar

Area of the transistor source diffusion. The value must be greater than or equal to 0.

#### **Perimeter of drain, PD – Drain perimeter**

0 m (default) | nonnegative scalar

Perimeter of the transistor drain diffusion. The value must be greater than or equal to 0.

**Perimeter of source, PS — Source perimeter**

0 m (default) | nonnegative scalar

Perimeter of the transistor source diffusion. The value must be greater than or equal to 0.

**Resistors****Number of drain squares, NRD — Number of drain diffusion resistance squares**

0 (default) | nonnegative scalar

Number of squares of resistance that make up the transistor drain diffusion. The value must be greater than or equal to 0. The block only uses this parameter value if you do not specify one or both of the **Drain resistance, RD** and **Source resistance, RS** parameter values, as described in “Parameters Calculations” on page 1-2137.

**Number of source squares, NRS — Number of transistor source diffusion squares of resistance**

0 (default) | nonnegative scalar

Number of squares of resistance that make up the transistor source diffusion. The value must be greater than or equal to 0. The block only uses this parameter value if you do not specify one or both of the **Drain resistance, RD** and **Source resistance, RS** parameter values, as described in “Parameters Calculations” on page 1-2137.

**Drain resistance, RD — Series resistance**

0.01 Ohm (default) | nonnegative scalar

Transistor drain resistance. The value must be greater than or equal to 0.

**Source resistance, RS — Series resistance**

0.0001 Ohm (default) | nonnegative scalar

Transistor source resistance. The value must be greater than or equal to 0.

**Sheet resistance, RSH — Per square resistance**

0 Ohm (default) | nonnegative scalar

Resistance per square of the transistor source and drain. Check “Parameters Calculations” on page 1-2137 to see when the block uses this parameter. The value must be greater than or equal to 0.

**DC Currents****Threshold voltage, VT0 — Threshold voltage**

0 V (default) | scalar

Source-gate voltage above which the transistor produces a nonzero drain current. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Temperature Dependence” on page 1-2152 .

**Transconductance, KP — Transconductance**

2e-5 A/V<sup>2</sup> (default) | nonnegative scalar

Derivative of drain current with respect to gate voltage. The value must be greater than or equal to 0. If you assign this parameter a value of NaN, the block calculates the value from the specified values of

the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2141 or “Level 3 Drain Current Model” on page 1-2142 as appropriate for the selected value of the **MOS model** parameter.

#### **Bulk threshold, GAMMA — Bulk threshold**

0 V<sup>0.50000</sup> (default) | nonnegative scalar

Body effect parameter, which relates the threshold voltage, VTH, to the body bias, VBS, as described in “Level 1 Drain Current Model” on page 1-2141 and “Level 3 Drain Current Model” on page 1-2142. The value must be greater than or equal to 0. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2141 or “Level 3 Drain Current Model” on page 1-2142 as appropriate for the selected value of the **MOS model** parameter.

#### **Surface potential, PHI — Surface potential**

0.6 V (default) | nonnegative scalar

Twice the voltage at which the surface electron concentration becomes equal to the intrinsic concentration and the device transitions between depletion and inversion conditions. The value must be greater than or equal to 0. If you assign this parameter a value of NaN, the block calculates the value from the specified values of the **Oxide thickness, TOX** and **Substrate doping, NSUB** parameters. For more information about this calculation, see “Level 1 Drain Current Model” on page 1-2141 or “Level 3 Drain Current Model” on page 1-2142 as appropriate for the selected value of the **MOS model** parameter.

#### **Channel modulation, LAMBDA — Channel-length modulation**

0 1/V (default) | scalar

Channel-length modulation.

#### **Dependencies**

This parameter is only visible when you select Level 1 MOS for the **MOS model** parameter in the **Model Selection** settings.

#### **Bulk saturation current, IS — Bulk saturation current magnitude**

1e-14 A (default) | nonnegative scalar

Magnitude of the current that the junction approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

#### **Emission coefficient, N — Emission coefficient**

1 (default) | nonnegative scalar

Transistor emission coefficient or ideality factor. The value must be greater than 0.

#### **Bulk jct sat current density, JS — Bulk junction saturation current density**

0 A/m<sup>2</sup> (default) | nonnegative scalar

Magnitude of the current per unit area that the junction approaches asymptotically for very large reverse bias levels. The value must be greater than or equal to 0.

#### **Width effect on threshold, DELTA — Width factor**

0 (default) | scalar

Factor that controls the effect of transistor width on threshold voltage.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Max carrier drift velocity, VMAX — Maximum drift velocity**

0 m/s (default) | scalar

Maximum drift velocity of the carriers.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Fast surface state density, NFS — Fast surface state density**

0 1/cm<sup>2</sup> (default) | scalar

Fast surface state density adjusts the drain current for the mobility reduction caused by the gate voltage.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Vds dependence threshold volt, ETA — Drain-source voltage threshold**

0 (default) | scalar

Coefficient that controls how the drain voltage affects the mobility in the drain current calculation.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Vgs dependence on mobility, THETA — Mobility dependence coefficient**

0 1/V (default) | scalar

Coefficient that controls how the gate voltage affects the mobility in the drain current calculation.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Mobility modulation, KAPPA — Mobility modulation coefficient**

0.2 (default) | scalar

Coefficient of channel-length modulation for the level 3 MOS model.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**C-V****Model gate capacitance (CGS, CGD, CGB) — Gate capacitance model**

No intrinsic capacitance (default) | Meyer gate capacitances | Charge conservation capacitances

Options for modeling the gate capacitance:

- No intrinsic capacitance — Do not include gate capacitance in the model.
- Meyer gate capacitances — Model capacitances using Meyer gate capacitances.
- Charge conservation capacitances — Model capacitances using charge conservation capacitances.

**Model gate overlap capacitance (CGSO, CGDO, CGB0) — Gate overlap model**

No (default) | Yes

Options for modeling the gate overlap capacitance:

- No — Do not include gate overlap capacitance in the model.
- Yes — Specify the gate-source, gate-drain, and gate-bulk capacitances.

**Dependencies**

Selecting Yes exposes related parameters.

**G-S overlap capacitance, CGSO — Gate-source overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-source capacitance due to lateral diffusion of the source. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGB0)** parameter.

**G-D overlap capacitance, CGDO — Gate-drain overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-drain capacitance due to lateral diffusion of the drain. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGB0)** parameter.

**G-B overlap capacitance, CGB0 — Gate-bulk overlap capacitance**

0 F/m (default) | scalar | 0 or  $\geq 1e-18$

Gate-bulk capacitance due to gate extending beyond the channel width. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model gate overlap capacitance (CGSO, CGDO, CGBO)** parameter.

**Model junction capacitance (CBD, CBS) — Junction capacitance model**

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model.
- Yes — Specify zero-bias junction capacitance, junction potential, grading coefficient, forward-bias depletion and capacitance coefficient.

**Dependencies**

Selecting Yes exposes related parameters.

**Zero-bias BD capacitance, CBD — Zero-bias junction capacitance**

0 F (default) | 0 or  $\geq 1e-18$

Capacitance between the bulk and the drain. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Zero-bias BS capacitance, CBS — Zero-bias bulk-source capacitance**

0 F (default) | scalar | 0 or  $\geq 1e-18$

Capacitance between the bulk and the source. The value must be equal to 0 or greater than or equal to *Cmin*. *Cmin* is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Bulk junction potential, PB — Bulk junction potential**

0.8 V (default) | scalar | 0 or  $\geq 0.01$

Potential across the bulk junction. This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter. The value must be equal to 0 or greater than or equal to *VJmin*. *VJmin* is a built-in model constant whose value is 0.01.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Bottom junction cap per area, CJ — Zero-bias bulk junction bottom capacitance per junction area**

0 F/m<sup>2</sup> (default) | scalar | 0 or  $\geq 1e-18$

Zero-bias bulk junction bottom capacitance per junction area. The value must be equal to 0 or greater than or equal to  $C_{min}$ .  $C_{min}$  is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Bottom grading coefficient, MJ – Bottom grading coefficient**

0.5 (default) | scalar |  $0 < 0.9$

Transistor bottom grading coefficient. The value must be equal to 0 or less than  $MG_{max}$ .  $MG_{max}$  is a built-in model constant whose value is 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Side jct cap/area of jct perimeter, CJSW – Zero-bias bulk junction sidewall capacitance per junction perimeter**

0 F/m (default) | scalar |  $0 \text{ or } \geq 1e-18$

Zero-bias bulk junction sidewall capacitance per junction perimeter. The value must be equal to 0 or greater than or equal to  $C_{min}$ .  $C_{min}$  is a built-in model constant whose value is  $1e-18$ .

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Side grading coefficient, MJSW – Sidewall grading coefficient**

0.5 (default) | scalar |  $0 < MJSW < 0.9$

Transistor sidewall grading coefficient. The value must be equal to 0 or less than  $MG_{max}$ .  $MG_{max}$  is a built-in model constant whose value is 0.9.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Capacitance coefficient, FC – Capacitance coefficient**

0.5 (default) |  $0 \text{ or } \leq 0.95$  | scalar

Fitting coefficient that quantifies the decrease of the depletion capacitance with applied voltage. The value must be equal to 0 or less than or equal to  $FC_{max}$ .  $FC_{max}$  is a built-in model constant whose value is 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance (CBD, CBS)** parameter.

**Specify initial condition – Initial condition model**

No (default) | Yes

Options for specifying initial conditions:



- No — Do not specify an initial condition for the model.
- Yes — Specify the initial transistor voltage.

---

**Note** The block applies the initial transistor voltage across the junction capacitors and not across the ports.

---

### Dependencies

To enable this parameter, set either:

- **Model gate capacitance (CGS, CGD, CGB)** to No intrinsic capacitance and either or both **Model gate overlap capacitance (CGSO, CGDO, CGBO)** and **Model junction capacitance (CBD, CBS)** to Yes.
- **Model gate capacitance (CGS, CGD, CGB)** to Meyer gate capacitances or Charge conservation capacitances.

### Initial condition voltage, ICVDS — Initial voltage

0 V (default) | scalar

Drain-source voltage at the start of the simulation.

### Dependencies

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

### Initial condition voltage, ICVGS — Initial voltage

0 V (default) | scalar

Gate-source voltage at the start of the simulation.

### Dependencies

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

### Initial condition voltage, ICVBS — Initial voltage

0 V (default) | scalar

Bulk-source voltage at the start of the simulation.

### Dependencies

This parameter is only visible when you select Yes for the **Specify initial condition** parameter.

### Process

#### Oxide thickness, TOX — Gate oxide thickness

1e-7 m (default) | nonnegative scalar

Thickness of the gate oxide. The value must be greater than or equal to 0.

#### Lateral diffusion, LD — Length of lateral diffusion

0 m (default) | scalar

Length of lateral diffusion.

**Surface mobility, U0 — Zero-bias surface mobility coefficient**600 cm<sup>2</sup>/s/V (default)

Zero-bias surface mobility coefficient.

**Substrate doping, NSUB — Substrate doping**NaN 1/cm<sup>3</sup> (default) | scalar  $\geq 1.45e10$ 

Substrate doping. The value must be greater than or equal to 1.45e10 (the carrier concentration of intrinsic silicon).

**Gate type, TPG — Gate type**

Opposite of substrate (1) (default) | Same as substrate (-1) | Aluminum (0)

MOSFET gate materials (as compared to the substrate):

- **Opposite of substrate** — The gate material is the opposite of the substrate. This means that TPG = 1 in the device equations. This is the default option.
- **Same as substrate** — The gate material is the same as the substrate. This means that TPG = -1 in the device equations.
- **Aluminum** — The gate material is aluminum. This means that TPG = 0 in the device equations.

**Surface state density, NSS — Surface state density**NaN 1/cm<sup>2</sup> (default) | scalar

Surface state density.

**Junction depth, XJ — Junction depth**

0m (default)

Junction depth.

**Dependencies**

This parameter is only visible when you select Level 3 MOS for the **MOS model** parameter in the **Model Selection** settings.

**Temperature****Model temperature dependence using — Temperature dependence model**

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Temperature Dependence” on page 1-2152.

**Dependencies**

Selecting Device temperature exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting Fixed temperature exposes the **Fixed circuit temperature, TFIXED** parameter.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0 K.

**Dependencies**

This parameter is only visible when you select Fixed temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

The temperature at which the transistor parameters are measured. The value must be greater than 0 K.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is only visible when you select Device temperature for the **Model temperature dependence using** parameter.

## Version History

Introduced in R2009a

## References

- [1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.
- [2] Ping Yang, et al. 'An Investigation of the Charge Conservation Problem for MOSFET Circuit Simulation'. *IEEE Journal of Solid-State Circuits*, vol. 18, no. 1, Feb. 1983, pp. 128-38. DOI.org (Crossref), doi:10.1109/JSSC.1983.1051909.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | SPICE NMOS

**Functions**

subcircuit2ssc

**Topics**

"Additional Parameterization Workflows"

“Converting a SPICE Netlist to Simscape Blocks”  
“Parameterize an Exponential Diode from SPICE Netlist”

## SPICE PNP

SPICE-compatible Gummel-Poon PNP Transistor

**Library:** Simscape / Electrical / Additional Components / SPICE Semiconductors



### Description

The SPICE PNP block represents a SPICE-compatible four-terminal Gummel-Poon PNP bipolar junction transistor. A capacitor connects the substrate port, **sx**, to the transistor base, **bx**. Therefore, the device is equivalent to a three-terminal transistor when you use the default value of 0 for the **C-S junction capacitance, CJS** parameter and connect the substrate port to any other port, including the emitter port, **ex**, or the collector port, **cx**.



SPICE, or Simulation Program with Integrated Circuit Emphasis, is a simulation tool for electronic circuits. You can convert some SPICE subcircuits into equivalent Simscape Electrical models using the Environment Parameters block and SPICE-compatible blocks from the “Additional Components” library. For more information, see `subcircuit2ssc`.

### Equations

Variables for the SPICE PNP block equations include:

- Variables that you define by specifying parameters for the SPICE PNP block. The visibility of some of the parameters depends on the value that you set for other parameters. For more information, see “Parameters” on page 1-2175.
- Geometry-adjusted variables, which depend on several values that you specify using parameters for the SPICE PNP block. For more information, see “Geometry-Adjusted Variables” on page 1-2167.
- Temperature,  $T$ , which is 300.15 K by default. You can use a different value by specifying parameters for the SPICE PNP block or by specifying parameters for both the SPICE PNP block and an Environment Parameters block. For more information, see “Transistor Temperature” on page 1-2168.
- Temperature-dependent variables. For more information, see “Temperature Dependence” on page 1-2173.
- Minimal conductance,  $GMIN$ , which is  $1e-12$  1/Ohm by default. You can use a different value by specifying a parameter for an Environment Parameters block. For more information, see “Minimal Conduction” on page 1-2169.

### Geometry-Adjusted Variables

Several variables in the equations for the SPICE PNP bipolar junction transistor model consider the geometry of the device that the block represents. These geometry-adjusted variables depend on

variables that you define by specifying SPICE PNP block parameters. The geometry-adjusted variables depend on these variables:

- *AREA* — Area of the device
- *SCALE* — Number of parallel connected devices
- The associated unadjusted variable

The table includes the geometry-adjusted variables and the defining equations.

Variable	Description	Equation
$IS_d$	Geometry-adjusted transport saturation current	$IS_d = IS * AREA * SCALE$
$IKF_d$	Geometry-adjusted forward knee current	$IKF_d = IKF * AREA * SCALE$
$IS_d$	Geometry-adjusted base-emitter leakage current	$ISE_d = ISE * AREA * SCALE$
$IKR_d$	Geometry-adjusted reverse knee current	$IKR_d = IKR * AREA * SCALE$
$ISC_d$	Geometry-adjusted base-collector leakage current	$ISC_d = ISC * AREA * SCALE$
$IRB_d$	Geometry-adjusted half base resistance current	$IRB_d = IRB * AREA * SCALE$
$CJE_d$	Geometry-adjusted base-emitter depletion capacitance	$CJE_d = CJE * AREA * SCALE$
$ITF_d$	Geometry-adjusted forward transit time coefficient	$ITF_d = ITF * AREA * SCALE$
$CJC_d$	Geometry-adjusted base-collector depletion capacitance	$CJC_d = CJC * AREA * SCALE$
$CJS_d$	Geometry-adjusted collector-substrate junction capacitance	$CJS_d = CJS * AREA * SCALE$
$RB_d$	Geometry-adjusted zero-bias base resistance	$RB_d = \frac{RB}{AREA * SCALE}$
$RBM_d$	Geometry-adjusted minimum base resistance	$RBM_d = \frac{RBM}{AREA * SCALE}$
$RE_d$	Geometry-adjusted emitter resistance	$RE_d = \frac{RE}{AREA * SCALE}$
$RC_d$	Geometry-adjusted collector resistance	$RC_d = \frac{RC}{AREA * SCALE}$

### Transistor Temperature

You can use these options to define transistor temperature, *T*:

- Fixed temperature — The block uses a temperature that is independent from the circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PNP block is set to Fixed temperature. For this model, the block sets *T* equal to *TFIXED*.

- Device temperature — The block uses a temperature that depends on circuit temperature when the **Model temperature dependence using** parameter in the **Temperature** settings of the SPICE PNP block is set to Device temperature. For this model, the block defines temperature as

$$T = T_C + TOFFSET$$

Where:

- $T_C$  is the circuit temperature.

If there is no Environment Parameters block in the circuit,  $T_C$  is equal to 300.15 K.

If there is an Environment Parameters block in the circuit,  $T_C$  is equal to the value that you specify for the **Temperature** parameter in the **SPICE** settings of the Environment Parameters block. The default value for the **Temperature** parameter is 300.15 K.

- $TOFFSET$  is the offset local circuit temperature.

### Minimal Conduction

Minimal conductance,  $GMIN$ , has a default value of 1e-12 1/Ohm. To specify a different value:

- 1 If there is not an Environment Parameters block in the circuit, add one.
- 2 In the **SPICE** settings of the Environment Parameters block, specify the desired  $GMIN$  value for the **GMIN** parameter.

### Current-Voltage and Base Charge Model

The current-voltage relationships and base charge relationships for the transistor are described in terms of “Base-Emitter and Base-Collector Junction Currents” on page 1-2169, “Terminal Currents” on page 1-2170, and “Base Charge Model” on page 1-2171. As applicable, the model parameters are first adjusted for temperature.

#### Base-Emitter and Base-Collector Junction Currents

The base-emitter junction current depends on the emitter-base voltage,  $V_{EB}$  such that:

- When  $V_{EB} > 80 * V_{TF}$ :

$$I_{bef} = IS_d * \left( \left( \frac{V_{EB}}{V_{TF}} - 79 \right) * e^{80} - 1 \right) + G_{min} * V_{EB}$$

$$I_{bee} = ISE_d * \left( (V_{EB} - 80 * V_{TF} + V_{TE}) * \frac{e^{(80 * V_{TF}/V_{TE})}}{V_{TE}} - 1 \right)$$

- When  $V_{EB} \leq 80 * V_{TF}$ :

$$I_{bef} = IS_d * \left( e^{(V_{EB}/V_{TF})} - 1 \right) + G_{min} * V_{EB}$$

$$I_{bee} = ISE_d * \left( e^{(V_{EB}/V_{TE})} - 1 \right)$$

The base-collector junction current depends on the collector-base voltage,  $V_{CB}$ , such that:

- When  $V_{CB} > 80 * V_{TR}$ :

$$I_{bcr} = IS_d * \left( \left( \frac{V_{CB}}{V_{TR}} - 79 \right) * e^{80} - 1 \right) + G_{min} * V_{CB}$$

$$I_{bcc} = ISC_d * \left( (V_{CB} - 80 * V_{TR} + V_{TC}) * \frac{e^{(80 * V_{TR}/V_{TC})}}{V_{TC}} - 1 \right)$$

- When  $V_{CB} \leq 80 * V_{TR}$ :

$$I_{bcr} = ISC_d * (e^{(V_{CB}/V_{TR})} - 1) + G_{min} * V_{CB}$$

$$I_{bcc} = ISC_d * (e^{(V_{CB}/V_{TC})} - 1)$$

Where:

- $V_{EB}$  is the emitter-base voltage.
- $V_{CB}$  is the collector-base voltage.
- $V_{TE}$  is the emitter thermal voltage, such that  $V_{TE} = NE * k * T / q$ .
- $V_{TC}$  is the collector thermal voltage, such that  $V_{TC} = NC * k * T / q$ .
- $V_{TF}$  is the forward thermal voltage, such that  $V_{TF} = NF * k * T / q$ .
- $V_{TR}$  is the reverse thermal voltage, such that  $V_{TR} = NR * k * T / q$ .
- $ISC_d$  is the geometry-adjusted base-collector leakage current.
- $ISE_d$  is the geometry-adjusted base-emitter leakage current.
- $NE$  is the base-emitter emission coefficient.
- $NC$  is the base-collector emission coefficient.
- $NF$  is the forward emission coefficient.
- $NR$  is the reverse emission coefficient.
- $q$  is the elementary charge on an electron.
- $k$  is the Boltzmann constant.
- $T$  is the transistor temperature. For more information, see “Transistor Temperature” on page 1-2168.
- $G_{min}$  is the minimum conductance. For more information, see “Minimal Conduction” on page 1-2169.

### Terminal Currents

The terminal currents are calculated as:

$$I_B = - \left( \frac{I_{bef}}{BF} + I_{bee} + \frac{I_{bcr}}{BR} + I_{bcc} \right)$$

$$I_C = - \left( \frac{I_{bef} - I_{bcr}}{q_b} - \frac{I_{bcr}}{BR} - I_{bcc} \right)$$

Where:

- $I_B$  is the base terminal current.
- $I_C$  is the collector terminal current.



- $BF$  is the forward beta.
- $BR$  is the reverse beta.

### Base Charge Model

The base charge,  $q_b$ , is calculated using these equations:

$$q_b = \frac{q_1}{2} \left( 1 + \sqrt{0.5 \left( \sqrt{(1 + 4q_2 - eps)^2 + eps^2} + 1 + 4q_2 - eps \right) + eps} \right)$$

$$q_1 = \left( 1 - \frac{V_{CB}}{V_{AF}} - \frac{V_{EB}}{V_{AR}} \right)^{-1}$$

$$q_2 = \frac{I_{bef}}{IKF_d} + \frac{I_{bcr}}{IKR_d}$$

Where:

- $q_b$  is the base charge.
- $V_{AF}$  is the forward Early voltage.
- $V_{AR}$  is the reverse Early voltage.
- $IKF_d$  is the geometry-adjusted forward knee current.
- $IKR_d$  is the geometry-adjusted reverse knee current.
- $eps$  is 1e-4.

### Base Resistance Model

You can use these options to model base resistance,  $r_{bb}$ :

- If you use the default value of infinity for the **Half base resistance cur, IRB** parameter, the block calculates the base resistance as

$$r_{bb} = RBM_d + \frac{RB_d - RBM_d}{q_b}$$

Where:

- $r_{bb}$  is base resistance.
- $RBM_d$  is the geometry-adjusted minimum base resistance.
- $RB_d$  is the geometry-adjusted zero-bias base resistance.
- If you specify a finite value for the **Half base resistance cur, IRB** parameter, the block calculates the base resistance as

$$r_{bb} = RBM_d + 3 * (RB_d - RBM_d) * \left( \frac{\tan z - z}{z * \tan^2 z} \right)$$

Where

$$z = \frac{\sqrt{1 + 144I_B / (\pi^2 IRB_d)} - 1}{(24/\pi^2) \sqrt{I_B / IRB_d}}$$

**Transit Charge Modulation Model**

If you specify nonzero values for the **Coefficient of TF, XTF** parameter, the block models transit charge modulation by scaling the forward transit time as

$$TF_{\text{mod}} = \frac{TF * \left[ 1 + XTF * e^{V_{CB}/(1.44V_{TF})} \left( \frac{I_{EB}}{I_{EB} + ITF_d} \right)^2 \right]}{q_b}$$

Where  $ITF_d$  is the geometry-adjusted coefficient of the forward transit time.

**Junction Charge Model**

The block lets you model junction charge. The base-collector charge,  $Q_{bc}$ , and the base-emitter charge,  $Q_{be}$ , depend on an intermediate value,  $Q_{dep}$ . As applicable, the model parameters are first adjusted for temperature.

- For the internal base-emitter junctions

$$Q_{be} = TF_{\text{mod}} * I_{be} + Q_{dep}$$

- For the internal base-collector junctions

$$Q_{bc} = TR * I_{bc} + XCJC * Q_{dep}$$

- For the external base-collector junctions

$$Q_{b_{\text{ext}C}} = (1 - XCJC) * Q_{dep}$$

$Q_{dep}$  depends on the junction voltage,  $V_{jct}$  ( $V_{BE}$  for the base-emitter junction and  $V_{BC}$  for the base-collector junction), as follows.

Applicable Range of $V_{jct}$ Values	Corresponding $Q_{dep}$ Equation
$V_{jct} < FC * VJ$	$Q_{dep} = C_{jct} * VJ * \frac{1 - (1 - V_{jct}/VJ)^{(1 - MJ)}}{1 - MJ}$
$V_{jct} \geq FC * VJ$	$Q_{dep} = C_{jct} * \left[ F1 + \frac{F3 * (V_{jct} - FC * VJ) + \frac{MJ * [V_{jct}^2 - (FC * VJ)^2]}{2 * VJ}}{F2} \right]$

Where:

- $FC$  is the capacitance coefficient.
- $VJ$  is:
  - The base-emitter built-in potential,  $VJE$ , for the base-emitter junction.
  - The base-collector built-in potential,  $VJC$ , for the base-collector junction.
- $MJ$  is:
  - The base-emitter exponential factor,  $MJE$ , for the base-emitter junction.
  - The base-collector exponential factor,  $MJC$ , for the base-collector junction.

- $C_{jct}$  is:
  - The geometry-adjusted base-emitter depletion capacitance,  $CJE_d$ , for the base-emitter junction.
  - The geometry-adjusted base-collector depletion capacitance,  $CJC_d$ , for the base-collector junction.
- $F1 = VJ * (1 - (1 - FC)^{(1 - MJ)}) / (1 - MJ)$
- $F2 = (1 - FC)^{(1 + MJ)}$
- $F3 = 1 - FC * (1 + MJ)$

The collector-substrate charge,  $Q_{cs}$ , depends on the substrate-collector voltage,  $V_{sc}$ . As applicable, the model parameters are first adjusted for temperature.

Applicable Range of $V_{sc}$ Values	Corresponding $Q_{cs}$ Equation
$V_{sc} < 0$	$Q_{cs} = CJS_d * VJS * \left( \frac{1 - (1 - V_{sc}/VJS)^{(1 - MJS)}}{1 - MJS} \right)$
$V_{sc} \geq 0$	$Q_{cs} = CJS_d * (1 + MJS * V_{sc} / (2 * VJS)) * V_{sc}$

Where:

- $CJS_d$  is the geometry-adjusted collector-substrate junction capacitance.
- $VJS$  is the substrate built-in potential.
- $MJS$  is the substrate exponential factor.

#### Temperature Dependence

The relationship between the saturation current,  $IS_d$ , and the transistor temperature,  $T$ , is

$$IS(T) = IS_d * (T/T_{meas})^{XTI} * e^{\left(\frac{T}{T_{meas}} - 1\right) * \frac{EG}{V_t}}$$

Where:

- $IS_d$  is the geometry-adjusted transport saturation current.
- $T_{meas}$  is the parameter extraction temperature.
- $XTI$  is the transport saturation current temperature exponent.
- $EG$  is the energy gap.
- $V_t = kT/q$ .

The relationship between the base-emitter junction potential,  $VJE$ , and the transistor temperature,  $T$ , is

$$VJE(T) = VJE * \left(\frac{T}{T_{meas}}\right) - \frac{3 * k * T}{q} * \log\left(\frac{T}{T_{meas}}\right) - \left(\frac{T}{T_{meas}}\right) * EG_{T_{meas}} + EG_T$$

Where:

- $VJE$  is the base-emitter built-in potential.

- $EG_{T_{meas}} = 1.16eV - (7.02e - 4 * T_{meas}^2) / (T_{meas} + 1108)$
- $EG_T = 1.16eV - (7.02e - 4 * T^2) / (T + 1108)$

The block uses the  $VJE(T)$  equation to calculate the base-collector junction potential by substituting  $VJC$ , the base-collector built-in potential, for  $VJE$ .

The relationship between the base-emitter junction capacitance,  $CJE$ , and the transistor temperature,  $T$ , is

$$CJE(T) = CJE_d * \left[ 1 + MJE * \left( 400e - 6 * (T - T_{meas}) - \frac{VJE(T) - VJE}{VJE} \right) \right]$$

Where:

- $CJE_d$  is the geometry-adjusted base-emitter depletion capacitance.
- $MJE$  is the base-emitter exponential factor.

The block uses the  $CJE(T)$  equation to calculate the base-collector junction capacitance by substituting  $CJC_d$ , geometry-adjusted base-collector depletion capacitance, for  $CJE_d$  and  $MJC$ , base-collector exponential factor, for  $MJE$ .

The relationship between the forward and reverse beta and the transistor temperature,  $T$ , is

$$\beta(T) = \beta * \left( \frac{T}{T_{meas}} \right)^{XTB}$$

Where:

- $\beta$  is the forward beta or reverse beta.
- $XTB$  is the beta temperature exponent.

The relationship between the base-emitter leakage current,  $ISE$ , and the transistor temperature,  $T$ , is

$$ISE(T) = ISE_d * \left( \frac{T}{T_{meas}} \right)^{-XTB} * \left( \frac{IS(T)}{IS_d} \right)^{1/NE}$$

Where:

- $ISE_d$  is the geometry-adjusted base-emitter leakage current.
- $NE$  is the base-emitter emission coefficient.

The block uses this equation to calculate the base-collector leakage current by substituting,  $ISC_d$ , the geometry-adjusted base-collector leakage current for  $ISE_d$  and  $NC$ , the base-collector emission coefficient, for  $NE$ .

## Assumptions and Limitations

- The block does not support noise analysis.
- The block applies initial conditions across junction capacitors and not across the block ports.

## Ports

### Conserving

#### **bx – Base terminal**

electrical

Electrical conserving port associated with the transistor base terminal.

#### **cx – Collector terminal**

electrical

Electrical conserving port associated with the transistor collector terminal.

#### **ex – Emitter terminal**

electrical

Electrical conserving port associated with the transistor emitter terminal.

#### **sx – Substrate terminal**

electrical

Electrical conserving port associated with the transistor substrate terminal.

## Parameters

### Main

#### **Device area, AREA – Device area**

1.0 m<sup>2</sup> (default) | positive scalar

Device area. The value must be greater than 0.

#### **Number of parallel devices, SCALE – Number of parallel devices**

1 (default) | positive scalar

Number of parallel transistors that the block represents. The value must be greater than 0.

### Forward Gain

#### **Transport saturation current, IS – Saturation current magnitude**

1e-16 A/m<sup>2</sup> (default) | positive scalar

Magnitude of the current at which the transistor saturates. The value must be greater than 0.

#### **Forward beta, BF – Ideal maximum forward beta**

100 (default) | positive scalar

Ideal maximum forward beta. The value must be greater than 0.

#### **Forward emission coefficient, NF – Forward emission coefficient**

1 (default) | positive scalar

Forward emission coefficient or ideality factor. The value must be greater than 0.

**Forward Early voltage, VAF — Forward Early voltage**

Inf V (default) | nonnegative scalar

Forward Early voltage. The value must be greater than or equal to  $\theta$ .

**Forward knee current, IKF — Forward-beta high-current roll-off current**Inf A/m<sup>2</sup> (default) | nonnegative scalar

Current value at which forward-beta high-current roll-off occurs. The value must be greater than or equal to  $\theta$ .

**B-E leakage current, ISE — Base-emitter leakage current** $\theta$  A/m<sup>2</sup> (default) | nonnegative scalar

Base-emitter leakage current. The value must be greater than or equal to  $\theta$ .

**B-E emission coefficient, NE — Base-emitter emission coefficient**

1.5 (default) | positive scalar

Base-emitter emission coefficient or ideality factor. The value must be greater than  $\theta$ .

**Reverse Gain****Reverse beta, BR — Ideal maximum reverse beta**

1 (default) | positive scalar

Ideal maximum reverse beta. The value must be greater than  $\theta$ .

**Reverse emission coefficient, NR — Reverse emission coefficient**

1 (default) | positive scalar

Reverse emission coefficient or ideality factor. The value must be greater than  $\theta$ .

**Reverse Early voltage, VAR — Reverse Early voltage**

Inf V (default) | nonnegative scalar

Reverse Early voltage. The value must be greater than or equal to  $\theta$ .

**Reverse knee current, IKR — Reverse-beta high-current roll-off current**Inf A/m<sup>2</sup> (default) | nonnegative scalar

Current value at which reverse-beta high-current roll-off occurs. The value must be greater than or equal to  $\theta$ .

**B-C leakage current, ISC — Base-collector leakage current** $\theta$  A/m<sup>2</sup> (default) | nonnegative scalar

Base-collector leakage current. The value must be greater than or equal to  $\theta$ .

**B-C emission coefficient, NC — Base-collector emission coefficient**

2 (default) | positive scalar

Base-collector emission coefficient or ideality factor. The value must be greater than  $\theta$ .

## Resistors

### Zero-bias base resistance, **RB** — Base resistance

1 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Maximum resistance of the base. The value must be greater than or equal to 0.

### Half base resistance cur, **IRB** — Base resistance half zero-bias drop current

Inf A/m<sup>2</sup> (default) | nonnegative scalar

Base current at which the base resistance has dropped to half of its zero-bias value. The value must be greater than or equal to 0. If you do not want to model the change in base resistance as a function of base current, use the default value of Inf.

### Minimum base resistance, **RBM** — Minimum base resistance

0 m<sup>2</sup>\*0hm (default) | scalar < *RB*

Minimum resistance of the base. The value must be less than or equal to the **Zero-bias base resistance, RB** parameter value.

### Emitter resistance, **RE** — Emitter resistance

1e-4 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Resistance of the emitter. The value must be greater than or equal to 0.

### Collector resistance, **RC** — Collector resistance

0.01 m<sup>2</sup>\*0hm (default) | nonnegative scalar

Resistance of the collector. The value must be greater than or equal to 0.

## Capacitance

### Model junction capacitance — Junction capacitance model

No (default) | Yes

Options for modeling the junction capacitance:

- No — Do not include junction capacitance in the model. This is the default option.
- Yes — Include junction capacitance in the model.

## Dependencies

Selecting Yes for the **Model junction capacitance** parameter exposes other **Capacitance** parameters and these capacitance junction settings:

- **B-E Capacitance** — Base-emitter parameters
- **B-C Capacitance** — Base-collector parameters
- **C-S Capacitance** — Collector-substrate parameters

### Capacitance coefficient, **FC** — Capacitance coefficient

0.5 (default) |  $0 \leq FC < 0.95$  | scalar

Fitting coefficient, *FC*, that quantifies the decrease of the depletion capacitance with applied voltage. The value must be greater than or equal to 0 and less than 0.95.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**Specify initial condition — Initial condition model**

No (default) | Yes

Options for specifying initial conditions:

- No — Do not specify an initial condition for the model. This is the default option.
- Yes — Specify the initial transistor conditions.

---

**Note** The block applies the initial transistor voltages across the junction capacitors and not across the ports.

---

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

Selecting Yes for the **Specify initial condition** parameter exposes related parameters.

**Initial condition voltage ICVBE — Initial base-emitter voltage**

0 V (default) | scalar

Base-emitter voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**Initial condition voltage ICVCE — Initial base-collector voltage**

0 V (default) | scalar

Base-collector voltage at the start of the simulation.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** and Yes for the **Specify initial condition** parameter.

**B-E Capacitance**

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

**B-E depletion capacitance, CJE — Base-emitter junction depletion capacitance**

0 F/m<sup>2</sup> (default) | nonnegative scalar

Depletion capacitance across the base-emitter junction. The value must be greater than or equal to 0.

**Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

**B-E built-in potential, VJE — Base-emitter junction potential**

0.75 V (default) | scalar |  $0.01 \leq VJE$



Base-emitter junction potential. The value must be greater than or equal to  $0.01$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

#### **B-E exponential factor, MJE — Base-emitter junction grading coefficient**

$0.33$  (default) | scalar |  $0 \leq MJC \leq 0.9$

Grading coefficient for the base-emitter junction. The value must be greater than or equal to  $0$  and less than or equal to  $0.9$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

#### **Forward transit time, TF — Minority carrier transit time**

$0$  s (default) | nonnegative scalar

Transit time of the minority carriers that cause diffusion capacitance when the base-emitter junction is forward-biased. The value must be greater than or equal to  $0$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

#### **Coefficient of TF, XTF — Base-emitter transit time bias dependence coefficient**

$0$  (default) | nonnegative scalar

Coefficient for the base-emitter bias dependence of the transit time, which produces a charge across the base-emitter junction. The value must be greater than or equal to  $0$ . If you do not want to model the effect of base-emitter bias on transit time, use the default value of  $0$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

#### **VBC dependence of TF, VTF — Base-collector transit time bias dependence coefficient**

$\ln f V$  (default) | nonnegative scalar

Coefficient for the base-collector bias dependence of the transit time. The value must be greater than or equal to  $0$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

#### **Coefficient of TF, ITF — Collector current transit time dependence coefficient**

$0$  A/m<sup>2</sup> (default) | nonnegative scalar

Coefficient for the dependence of the transit time on collector current. The value must be greater than or equal to  $0$ . If you do not want to model the effect of collector current on transit time, use the default value of  $0$ .

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## **B-C Capacitance**

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

### **B-C depletion capacitance, CJC — base-collector junction depletion capacitance**

$0 \text{ F/m}^2$  (default) | positive scalar

Depletion capacitance across the base-collector junction. The value must be greater than 0.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### **B-C built-in potential, VJC — Base-collector junction potential**

$0.75 \text{ V}$  (default) | scalar |  $0.01 \leq VJC$

Base-collector junction potential. The value must be greater than or equal to  $0.01 \text{ V}$ .

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### **B-C exponential factor, MJC — Base-collector junction grading coefficient**

$0.33$  (default) |  $0 \leq MJC \leq 0.9$

Grading coefficient for the base-collector junction. The value must be greater than or equal to 0 and less than or equal to 0.9.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### **B-C capacitance fraction, XCJC — Base-collector depletion capacitance fraction**

$1$  (default) | scalar |  $0 \leq XCJC \leq 1$

Fraction of the base-collector depletion capacitance that is connected between the internal base and the internal collector. The rest of the base-collector depletion capacitance is connected between the external base and the internal collector. The value must be greater than or equal to 0 and less than or equal to 1.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### **Reverse transit time, TR — Minority carrier transit time**

$0 \text{ s}$  (default) | nonnegative scalar

Transit time of the minority carriers that cause diffusion capacitance when the base-collector junction is forward-biased. The value must be greater than or equal to 0.

#### **Dependencies**

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## C-S Capacitance

These settings are exposed if you select Yes for the **Model junction capacitance** parameter in the **Capacitance** settings.

### C-S junction capacitance, CJS — Collector-substrate junction capacitance

0F/m<sup>2</sup> (default) | nonnegative scalar

Collector-substrate junction capacitance. The value must be greater than or equal to 0.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### Substrate built-in potential, VJS — Substrate potential

0.75 V (default) | scalar |  $0.01 \leq VJS$

Potential of the substrate. The value must be greater than or equal to 0.01 V.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

### Substrate exponential factor, MJS — Collector-substrate junction grading coefficient

0 (default) | scalar |  $0 \leq MJS \leq 0.9$

Grading coefficient for the collector-substrate junction. The value must be greater than or equal to 0 and less than or equal to 0.9.

#### Dependencies

This parameter is only visible when you select Yes for the **Model junction capacitance** parameter.

## Temperature

### Model temperature dependence using — Temperature dependence model

Device temperature (default) | Fixed temperature

Select one of these options for modeling the transistor temperature dependence:

- **Device temperature** — Use the device temperature to model temperature dependence.
- **Fixed temperature** — Use a temperature that is independent of the circuit temperature to model temperature dependence.

For more information, see “Transistor Temperature” on page 1-2168.

#### Dependencies

Selecting **Device temperature** exposes the **Offset local circuit temperature, TOFFSET** parameter. Selecting **Fixed temperature** exposes the **Fixed circuit temperature, TFIXED** parameter.

### Beta temperature exponent, XTB — Forward and reverse beta temperature exponent

0 (default) | nonnegative scalar

Forward and reverse beta temperature exponent that models base current temperature dependence. The value must be greater than or equal to 0.

**Energy gap, EG — Energy gap**1.11 eV (default) |  $EG \geq 0.1$  | scalar

Energy gap that affects the increase in the saturation current as temperature increases. The value must be greater than or equal to 0.1.

**Temperature exponent for IS, XTI — Saturation current exponential increase order**

3 (default) | nonnegative scalar

Order of the exponential increase in the saturation current as temperature increases. The value must be greater than or equal to 0.

**Offset local circuit temperature, TOFFSET — Local circuit temperature offset**

0 K (default) | scalar

Amount by which the transistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is only visible when you select Device temperature for the **Model temperature dependence using** parameter.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default) | positive scalar

Transistor simulation temperature. The value must be greater than 0 K.

**Dependencies**

This parameter is only visible when you select Fixed temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default) | positive scalar

Temperature at which the transistor parameters are measured. The value must be greater than 0 K.

## Version History

Introduced in R2008a

## References

[1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition. New York: McGraw-Hill, 1993.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

### Simscape Blocks

Environment Parameters | Generic Linear Actuator

### Functions

subcircuit2ssc

### Topics

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# SPICE Resistor

Model SPICE-compatible resistor

**Library:** Simscape / Electrical / Additional Components / SPICE  
Passives



## Description

The SPICE Resistor block represents a SPICE-compatible resistor. You can specify the resistance in one of the following ways:

- As a resistance value
- As process information that the block uses to calculate a resistance value

The block models temperature dependence. There are two ways to specify the resistor temperature:

- When you select `Device` temperature for the **Model temperature dependence using** parameter, the resistor temperature is

$$T = T_C + T_O$$

where:

- $T_C$  is the **Temperature** parameter value from the Environment Parameters block. If this block doesn't exist in the circuit,  $T_C$  is the default value of this parameter.
- $T_O$  is the **Offset local circuit temperature, TOFFSET** parameter value.
- When you select `Fixed` temperature for the **Model temperature dependence using** parameter, the resistor temperature is the **Fixed circuit temperature, TFIXED** parameter value.

The block adjusts the specified or calculated resistance value for temperature using the following equation:

$$R = R_0(1 + TC1(T - T_{nom}) + TC2(T - T_{nom})^2)$$

Where

- $R_0$  is the specified or calculated resistance value.
- $TC1$  is the **First order temperature coefficient, TC1** parameter value.
- $TC2$  is the **Second order temperature coefficient, TC2** parameter value.
- $T_{nom}$  is the **Parameter extraction temperature, TMEAS** parameter value.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the SPICE resistor positive voltage terminal

**- – Negative terminal**

electrical

Electrical conserving port associated with the SPICE resistor negative voltage terminal

**Parameters****Resistance****Device scale factor, SCALE – Device scale factor**

1.0 (default)

Number of parallel resistors that the block represents. This value multiplies the output current.

**Resistor parameterization – Parameterization options**

Use specified resistance (default) | Calculate from process information

Select one of the following options for specifying the resistor value:

- Use specified resistance — Provide the resistance value directly.
- Calculate from process information — Provide process parameters that the block uses to calculate the resistance value.

When you select this option, the block calculates the resistance using the following equation:

$$R = RSH * \frac{(LENGTH - NARROW)}{(WIDTH - NARROW)}$$

where:

- *RSH* is the **Sheet resistance**, **RSH** parameter value.
- *LENGTH* is the **Resistor length**, **LENGTH** parameter value.
- *WIDTH* is the **Resistor width**, **WIDTH** parameter value.
- *NARROW* is the **Etch narrowing**, **NARROW** parameter value.

**Resistance, R – Resistance**

0 Ohm (default)

Resistance value.

**Dependencies**

This parameter is visible only when you select Use specified resistance for the **Resistor parameterization** parameter.

**Sheet resistance, RSH – Resistance per square**

0 Ohm (default)

Resistance per square of the resistor.

**Dependencies**

This parameter is visible only when you select Calculate from process information for the **Resistor parameterization** parameter.

**Resistor length, LENGTH — Resistor length**

1e-6 m (default)

Length dimension of the resistor.

**Dependencies**

This parameter is visible only when you select `Calculate` from process information for the **Resistor parameterization** parameter.

**Resistor width, WIDTH — Resistor width**

1e-6 m (default)

Width dimension of the resistor.

**Dependencies**

This parameter is visible only when you select `Calculate` from process information for the **Resistor parameterization** parameter.

**Etch narrowing, NARROW — Etch narrowing**

0 m (default)

Amount by which the resistor length and width are reduced due to side etching.

**Dependencies**

This parameter is visible only when you select `Calculate` from process information for the **Resistor parameterization** parameter.

**Temperature****Model temperature dependence using — Temperature dependence parameterization**

Device temperature (default) | Fixed temperature

Select one of the following options for modeling the resistor temperature dependence:

- `Device temperature` — Use the device temperature, which is the **Temperature** parameter value (from the Environment Parameters block, if one exists in the circuit, or the default value for this block otherwise) plus the **Offset local circuit temperature, TOFFSET** parameter value.
- `Fixed temperature` — Use a temperature that is independent of the circuit temperature to model temperature dependence.

**First order temperature coefficient, TC1 — First order temperature coefficient**

0 1/K (default)

Coefficient for the linear term in the equation that the block uses to adjust the specified or calculated resistance value for temperature.

**Second order temperature coefficient, TC2 — Second order temperature coefficient**0 1/K<sup>2</sup> (default)

Coefficient for the quadratic term in the equation the block uses to adjust the specified or calculated resistance value for temperature.

**Offset local circuit temperature, TOFFSET — Offset local circuit temperature**

0 K (default)



The amount by which the resistor temperature differs from the circuit temperature.

**Dependencies**

This parameter is visible only when you select `Device` temperature for the **Model temperature dependence using** parameter.

**Fixed circuit temperature, TFIXED — Fixed circuit temperature**

300.15 K (default)

The temperature at which to simulate the resistor.

**Dependencies**

This parameter is visible only when you select `Fixed` temperature for the **Model temperature dependence using** parameter.

**Parameter extraction temperature, TMEAS — Parameter extraction temperature**

300.15 K (default)

The temperature at which the resistor parameters were measured.

## Version History

Introduced in R2012b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Diode | Environment Parameters

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

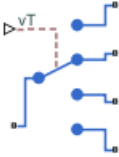
“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# SPMT Switch

Single-pole multiple-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



## Description

The SPMT Switch block models a single-pole multiple-throw switch. You can specify three to eight throw connections by setting the **Number of throw ports** parameter to the desired amount. If you only need one or two throw ports, see the SPST Switch and SPDT Switch blocks.

The switch control port physical signal, **vT**, specifies which throw port connects to the pole:

- When **vT** is in the  $[-\infty, 1.5]$  range, the pole always connects to throw port 1, **n1**.
- When **vT** is between one and the number of throw ports, the signal value is rounded to the nearest integer. The pole then connects to the throw port that corresponds to the integer value.
- When **vT** is higher than the number of throw ports, the pole always connects to the last throw port.

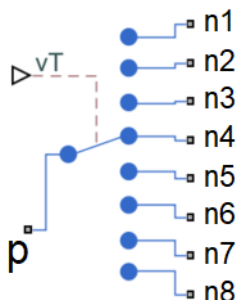
Open connections are modeled by a resistor with a value equal to the reciprocal of the **Open conductance** parameter value. Closed connections are modeled by a resistor with a value equal to the **Closed resistance** parameter value.

## Assumptions and Limitations

- The values of the open conductance and closed resistance are the same for all throw ports.
- Switch dynamics and custom thresholds are not modeled.

## Ports

Refer to the figure for port locations.



## Input

### **vT — Switch control port**

physical signal

Signal that opens and closes the switch, specified as a physical signal.

## Conserving

### **p — Common port**

electrical

Electrical conserving port associated with the common pole.

### **n1 — Throw port 1**

electrical

Electrical conserving port associated with the first throw.

### **n2 — Throw port 2**

electrical

Electrical conserving port associated with the second throw.

### **n3 — Throw port 3**

electrical

Electrical conserving port associated with the third throw.

### **n4 — Throw port 4**

electrical

Electrical conserving port associated with the fourth throw.

## Dependencies

To enable this port, set **Number of throw ports** to 4, 5, 6, 7, or 8.

### **n5 — Throw port 5**

electrical

Electrical conserving port associated with the fifth throw.

## Dependencies

To enable this port, set **Number of throw ports** to 5, 6, 7, or 8.

### **n6 — Throw port 6**

electrical

Electrical conserving port associated with the sixth throw.

## Dependencies

To enable this port, set **Number of throw ports** to 6, 7, or 8.

### **n7 — Throw port 7**

electrical

Electrical conserving port associated with the seventh throw.

**Dependencies**

To enable this port, set **Number of throw ports** to 7 or 8.

**n8 — Throw port 8**

electrical

Electrical conserving port associated with the eighth throw.

**Dependencies**

To enable this port, set **Number of throw ports** to 8.

**Parameters****Main****Number of throw ports — Number of throw ports**

4 (default) | 3 | 5 | 6 | 7 | 8

Number of throw connections.

**Closed resistance — Closed resistance**

0.01 Ohm (default) | positive scalar

Resistance between the **p** port and the throw ports when the switch is closed. The value must be greater than zero.

**Open conductance — Open conductance**

1e-6 S (default) | positive scalar

Conductance between the **p** port and the throw ports when the switch is open. The value must be greater than zero.

**Faults****Enable throw port position stuck fault — Whether to enable throw port stuck fault**

Off (default) | On

Whether throw port stuck faults can occur when the specified conditions are met.

**Reporting when a fault occurs — Type of fault reporting**

None (default) | Warn | Error

Type of reporting when a fault occurs:

- None — Does not generate a warning or an error.
- Warn — Generates a warning.
- Error — The simulation stops and generates an error.

**Dependencies**

To enable this parameter, set **Enable throw port position stuck fault** to On.

**Fault trigger – Type of fault trigger**

Temporal (default) | Behavioral

Type of fault trigger.

**Dependencies**

To enable this parameter, set **Enable throw port position stuck fault** to On.

**Simulation time for fault event – Simulation time for fault event**

1 s (default) | positive scalar

Amount of time before a temporal fault is triggered.

**Dependencies**

To enable this parameter, set:

- **Enable throw port position stuck fault** to On.
- **Fault trigger** to Temporal.

**Maximum permissible common port current – Maximum permissible common port current**

inf A (default) | positive scalar

Maximum current that is permitted to flow through the common port, above which a fault may be triggered.

**Dependencies**

To enable this parameter, set:

- **Enable throw port position stuck fault** to On.
- **Fault trigger** to Behavioral.

**Time to fail when exceeding current – Time to fail when exceeding current**

1 s (default) | positive scalar

Amount of time that the common port current must continuously exceed the maximum permissible current before a behavioral fault is triggered.

**Dependencies**

To enable this parameter, set:

- **Enable throw port position stuck fault** to On.
- **Fault trigger** to Behavioral.

**Version History**

Introduced in R2020a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

[DPDT Switch](#) | [DPST Switch](#) | [SPDT Switch](#) | [SPDT Switch \(Three-Phase\)](#) | [SPST Switch](#) | [SPST Switch \(Three-Phase\)](#)

# SPST Relay

Single-pole single-throw relay with delays and faults

**Library:** Simscape / Electrical / Switches & Breakers / Relays



## Description

The SPST Relay block models a single-pole single-throw relay. The block has three potential states:

- De-energized — The common contact, **C**, changes from being connected to disconnected from the normally open contact, **S**.
- Energized — The common contact, **C**, changes from being disconnected to connected to the normally open contact, **S**.
- Open Circuit — The relay is open. The common contact, **C**, is not connected to the normally open contact, **S**.

You can:

- Control the relay state using a physical signal input port or electrical conserving ports.
- Output the relay state.
- Delay the breaking and making of the connection.
- Introduce a behavioral or temporal fault that results in a stuck connection, an open circuit, or degraded contact resistance.

## Relay State Control

The block has two control variants:

- Physical signal (PS) control — The state of the relay depends on how the value of the input physical signal compares to the relay threshold. The relay threshold, *th*, is the value that you specify for the **Threshold** parameter.
- Electrical control — The state of the relay depends on how the current through the positive and negative electrical conserving ports, which represent the relay winding, compares to the relay upper and lower thresholds. The upper and lower thresholds depend on the values that you specify for the parameters in the **Winding** settings.

For the PS control variant, at the start of simulation:

- If the input signal, *PS*, is less than or equal to *th*, the relay is de-energized and the common contact, **C**, disconnects from the normally open contact, **S**.
- If the input signal, *PS*, greater than *th*, the relay is energized and **C** connects to the normally open contact, **S**.

After the simulation starts, if  $PS$  rises above  $th$ , the block goes from a de-energized state to an energized state and the **C-S** connection closes after the delay specified in **Time-to-make C-S connection**.

If the block goes from an energized state to a de-energized state, that is  $PS$  falls to or below  $th$ , the **C-S** connection breaks after the delay specified in **Time-to-break C-S connection**.

For the electrical control variant, the upper and lower thresholds depend on the specified values for the **Rated voltage**, **Percent rated voltage to energize**, **Percent rated voltage to de-energize**, and **Winding series resistance** parameters. The rated current is:

$$i_{rated} = \frac{V_{rated}}{r_{coil}}.$$

The equation for the upper current threshold,  $i_{energized}$ , is:

$$i_{energized\_current} = \frac{Rated\ Voltage * Percent\ rated\ voltage\ to\ energize}{Winding\ series\ resistance}.$$

The equation for the lower current threshold,  $i_{de-energized}$ , is:

$$i_{de-energized\_current} = \frac{Rated\ Voltage * Percent\ rated\ voltage\ to\ de - energize}{Winding\ series\ resistance}.$$

At the start of simulation, if the control current is greater than  $i_{energized}$ , the relay is energized and **C** connects to the normally open contact, **S**.

After the start of simulation, if the current rises above  $i_{energized}$ , the block goes from a de-energized state to an energized state. The **C-S** connection closes after the delay specified in **Time-to-make C-S connection**.

If the current falls below  $i_{de-energized}$ , the block goes from an energized state to a de-energized state. The **C-S** connection breaks after the delay specified in **Time-to-break C-S connection**.

### Output the Relay State

To view the relay state, expose port **x**, a physical signal port that outputs the state of each connection. To expose the **x** port, in the **Main** settings, set **State port** to **Visible**.

The table shows how the state of the relay relates to the state of the connection. A closed connection has a state of 1. An open connection has a state of 0.

### Relay and Connection States

C-S Connection State	Relay State
0	De-energized or open circuit
1	Energized

### Connection Delays

You can specify delays for making and breaking connections in the **Mechanical** settings. The table shows how the make and break time delays affect the connections between contacts **S1** and **S2** and the common contact, **C**.



Mechanical Settings		Relay State	C-S Connection State	Results, with C-S in Blue and PS in Yellow
<b>Parameter</b>	<b>Value</b>	De-energized	0	
<b>Time-to-break C-S connection</b>	0	Energized	1	
<b>Time-to-make C-S connection</b>	0			
<b>Parameter</b>	<b>Value</b>	De-energized	1, 0	
<b>Time-to-break C-S connection</b>	> 0	Energized	1	
<b>Time-to-make C-S connection</b>	0			

Mechanical Settings		Relay State	C-S Connection State	Results, with C-S in Blue and PS in Yellow
<b>Parameter</b>	<b>Value</b>	De-energized	0	
<b>Time-to-break C-S connection</b>	0	Energized	0, 1	
<b>Time-to-make C-S connection</b>	> 0			

### Faults

The block provides four fault types:

- C-S stuck closed
- C open circuit (no path to S)
- Degraded contact resistance
- Winding failed open circuit. This fault type is available only for the electrical control variant.

Each type of fault can take effect only when the threshold for a temporal or behavioral trigger is exceeded.

#### C-S Stuck Closed Fault

The **C-S** stuck closed fault occurs if the common contact, **C**, becomes mechanically, electrically, or chemically stuck to the normally closed contact, **S**. Causes of this type of fault include:

- 1 Contact welding, possibly due to an inrush current or breaking current that exceeds the rating or load short circuit.
- 2 Mechanical damage.
- 3 Insulation deterioration.
- 4 Inductive voltage.

The connection state for a **C-S** stuck closed fault is equal to 1.

A **C-S** stuck closed temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t}$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time after which contactors can stick** parameter.

A **C-S** stuck closed behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and

$$t_{i > i_{th}} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i > i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

When the temporal or behavioral fault thresholds are exceeded, if **C-S** is:

- Closed — **C-S** remains closed for the rest of the simulation.
- Open — The fault does not take effect unless the relay becomes energized and the **C-S** connection closes. Once closed, the **C-S** connection remains closed for the rest of the simulation.

### **C Open Circuit (No Path to S) Fault**

The **C** open circuit (no path to **S**) fault occurs if the common contact, **C**, becomes mechanically stuck in an open position. This type of fault can happen if:

- There is contact damage or deterioration.
- The switch connector becomes mechanically stuck in the middle of the relay.

The faulted state for a **C** open circuit (no path to **S**) fault is equal to  $\emptyset$ .

A **C** open circuit temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time after which C can be open circuit** parameter.

A **C** open circuit behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and then

$$t_i > i_{th} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i>i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

When the temporal fault threshold is exceeded, if:

- The relay is in the open-circuit state, that is, the connection has been broken and not yet been made, the relay remains open for the rest of the simulation.
- The **C-S** connection is closed, the fault does not take effect unless the relay is de-energized and the closed connection is broken. If the connection is broken, the relay becomes an open-circuit, and remains open for the rest of the simulation.

When the behavioral fault thresholds are exceeded, if:

- The relay is in the open-circuit state, that is, the connection has been broken and not yet been made, the relay remains open for the rest of the simulation.
- The **C-S** connection is closed, the relay state immediately becomes an open circuit and remains an open circuit for the rest of the simulation.

### **Degraded Contact Resistance Fault**

Causes of the degraded contact resistance fault include:

- 1** Overuse-induced overload conditions. High inrush currents and voltages can cause overload conditions, as can excessive switching of the relay. Overload conditions ultimately trigger electrical arcing, which generates heat that degrades the contact material.
- 2** Chemical contamination that interferes with the operation of the relay contacts. Contaminants, which can include oxidation films or foreign particles, tend to produce high or unstable contact resistance readings.
- 3** End of relay life.

The faulted state for a degraded contact resistance fault for the **C-S** connection is equal to either 0 or 1.

A degraded contact resistance temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.

The current contact resistance of **S** is:

$$r_{contact\_s}(t) = r_{contact\_fault\_s} - (r_{contact\_fault\_s} - r_{contact})sech\left(\frac{t - t_{th\_t}}{\tau}\right),$$

where:

- $r_{contact\_fault\_s}$  is the final value of the faulted **S** contact resistance.
- $r_{contact}$  is the unfaulted **S** contact resistance.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.

A degraded contact resistance behavioral fault can occur only if the load current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold. That is, the behavioral fault can occur only if:

$$i_{load} > i_{th},$$

and then

$$t_i > i_{th} > t_{th\_b},$$

where:

- $i_{load}$  is the load current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{i>i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

For a behavior-triggered fault, if  $i_{S-C} > i_{th}$  continuously over the time interval  $t_{th\_b}$ ,

$$r_{contact\_s}(t) = r_{contact\_fault\_s} - (r_{contact\_fault\_s} - r_{contact})sech\left(\frac{t_{i>i_{th}} - t_{th\_b}}{\tau}\right),$$

where:

- $i_{S-C}$  is the common contact to normally closed contact, **C-S**, current.
- $i_{th}$  is the specified value for the **Maximum permissible load current** parameter.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.
- $r_{contact\_fault\_s}$  is the final value of the faulted **S** contact resistance.
- $r_{contact}$  is the unfaulted **S** contact resistance.
- $\tau$  is specified value for the **Time constant for degraded contact resistance** parameter.

When the temporal fault threshold is exceeded for the **C-S** connection, the contact resistance is immediately degraded and remains degraded for the rest of the simulation.

When the behavioral fault thresholds are exceeded for **C-S**, in terms of  $i_{S-C}$ , the resistance for the **C-S** connection is immediately degraded and remains degraded for the rest of the simulation.

### Winding failed open circuit

The open-circuit winding fault is available only for the electrical control variant. An open circuit in the winding coil can cause this type of fault.

A winding failed open circuit temporal fault can occur only if the simulation time exceeds the time threshold such that

$$t > t_{th\_t},$$

where:

- $t$  is the simulation time.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.

For time fault, the relay switches depend on the winding current, which is approximated as:

$$L \frac{di}{dt} + Ri = v_{winding} * sech\left(\frac{t - t_{th\_t}}{\tau}\right),$$

where :

- $L$  is the winding inductance.
- $R$  is the winding resistance.
- $i$  is the winding current.
- $v_{winding}$  is the voltage across the winding.
- $t_{th\_t}$  is the specified value for the **Simulation time for fault event** parameter.
- $\tau$  is the specified value for the **Time constant for winding open circuit transition** parameter.

A winding failed open circuit behavioral fault can occur only if one of these conditions is met:

- The winding current exceeds the current threshold for a period of time that exceeds the behavioral trigger time threshold.
- The winding voltage exceeds the voltage threshold for a number of times that exceeds the threshold for the number of voltage overloads.

That is, the behavioral fault can occur only if:

$$i_{winding} > i_{th}$$

and then

$$t_i > i_{th} > t_{th\_b},$$

where:

- $i_{winding}$  is the winding current.
- $i_{th}$  is the specified value for the **Maximum permissible winding current** parameter.
- $t_{i>i_{th}}$  is the time that the current threshold is exceeded.
- $t_{th\_b}$  is the specified value for the **Time to fail when exceeding current** parameter.

or if:

$$v_{winding} > v_{th},$$

and then

$$N_v > v_{th} > N_{th},$$

where:

- $v_{winding}$  is the winding voltage.
- $v_{th}$  is the specified value for the **Maximum permissible winding voltage** parameter.
- $N_{v>v_{th}}$  is the number of times that the voltage threshold is exceeded.
- $N_{th}$  is the specified value for the **Number of events to fail when exceeding voltage** parameter.

If the temporal or behavioral fault thresholds are exceeded, **C-S** remains closed for the rest of the simulation.

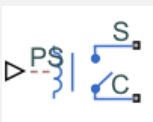
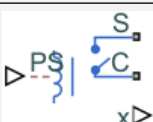
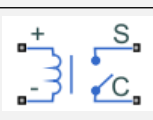
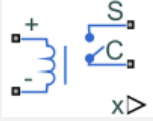
## Limitations and Assumptions

- For behavioral faults, if **Time to fail when exceeding current** is greater than the time between switches, no fault is triggered because the accumulated heat is too low to melt or break the contacts or windings.
- The energize and de-energize delays can differ, but the energize delay must be greater than or equal to the de-energize delay.

## Ports

The type, visibility, and location of the block ports depend on how you configure these parameters in the **Main** settings:

- **Control port** — Choose between a physical signal input port, **PS**, or electrical conserving ports, **+** and **-**, for relay control.
- **State port** — Set the visibility for the relay state physical signal output port, **x**.

Control port	State port	Block
PS	Hidden	
	Visible	
Electrical	Hidden	
	Visible	

## Input

### PS — Physical signal control

physical signal

Physical signal input port that energizes and de-energizes the relay.

**Dependencies**

This port is visible only when, in the **Main** settings, the **Control port** parameter is set to PS.

**Output****x – Relay state**

physical signal

State of the relay in terms of the **C-S** connection:

- 1 – Closed connection
- 0 – Open connection

**Dependencies**

This port is visible only when, in the **Main** settings, the **State port** parameter is set to Visible.

**Conserving****S – Normally-closed contact**

electrical

Electrical conserving port associated with the SPST relay normally-closed contact.

**C – Common contact**

electrical

Electrical conserving port associated with the SPST relay common contact.

**+ – Winding positive terminal**

electrical

Electrical conserving port associated with the SPST relay winding positive voltage terminal. The current through the winding controls the relay state.

**Dependencies**

This port is visible only if, in the **Main** settings, the **Control port** parameter is set to Electrical.

**- – Winding negative terminal**

electrical

Electrical conserving port associated with the SPST relay winding negative voltage terminal. The current through the winding controls the relay state.

**Dependencies**

This port is visible only if, in the **Main** settings, the **Control port** parameter is set to Electrical.

**Parameters****Main****Control port – Input port choice**

PS (default) | Electrical



Relay state control method:

- **PS** — **PS** physical signal input port
- **Electrical** — + and - electrical conserving ports associated with the relay winding terminals

### Dependencies

If this parameter is set to:

- **PS** — The **PS** port and related parameters in the **Main** and **Faults** settings are visible. For more information, see Control Port Parameter Dependencies.
- **Electrical** — The + and - ports, related parameters in the **Main** and **Faults** settings, and the **Windings** settings are visible. For more information, see Control Port Parameter Dependencies, Faults Parameter Dependencies, and “Winding” on page 1-2204.

The table shows how this parameter affects the visibility of other parameters in the **Main** settings. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Control Port Parameter Dependencies

Control port	
PS	Electrical
<b>Closed resistance</b>	<b>Closed resistance</b>
<b>Open conductance</b>	<b>Open conductance</b>
<b>Threshold</b>	
<b>State port</b>	<b>State port</b>

#### Closed resistance — Connected contact resistance

0.01 Ohm (default) | positive scalar

Resistance across closed relay contacts. The parameter value must be greater than zero.

#### Open conductance — Open-circuit conductance

1e-8 1/Ohm (default) | positive scalar

Conductance across open relay contacts. The parameter value must be greater than zero.

#### Threshold — Relay threshold

0 (default) | nonnegative scalar

Threshold of the relay. If the physical signal input rises above the threshold value, the relay is energized. Conversely, if the physical signal input falls below the threshold value, the relay is de-energized.

### Dependencies

This parameter is visible only if the **Control port** parameter is set to **PS**.

#### State port — Relay state output port visibility

Hidden (default) | Visible

Visibility of the physical signal port that outputs the relay state. This parameter value is 1 if the connection is closed, and 0 otherwise.

**Dependencies**

The **x** port is visible only if this parameter is set to **Visible**.

**Winding**

The **Winding** settings are visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Rated voltage — Rated voltage**

5 V (default) | positive scalar

Standard voltage applied to the operating coil under normal operating conditions. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Percent rated voltage to energize — Percent rated voltage to energize**

75 (default) | positive scalar

Minimum percent of the rated voltage required to energize the relay. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Percent rated voltage to de-energize — Percent rated voltage to de-energize**

25 (default) | positive scalar

Percent of the rated voltage required to de-energize the relay. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding inductance — Winding inductance**

0.08 H (default) | positive scalar

Winding inductance.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding series resistance — Winding series resistance**

12.5 Ohm (default) | positive scalar

Resistance between the winding terminals. The current threshold depends on the value of this parameter.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Winding parallel conductance — Winding parallel conductance**

1e-9 1/0hm (default) | positive scalar

Winding conductance.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

**Mechanical****Time-to-break C-S connection — Delay time for breaking connection C-S**

0 s (default) | nonnegative scalar

Mechanical switching time for breaking the connection between ports **C** and **S** when the relay is energized.

**Time-to-make C-S connection — Delay time for making connection C-S**

0 s (default) | nonnegative scalar

Mechanical switching time for making the connection between ports **C** and **S** when the relay is de-energized.

**Faults**

The table shows how the specified options for certain parameters in the **Faults** settings affect the visibility of other parameters in the **Faults** settings. To learn how to read the table, see “Parameter Dependencies” on page A-2.

---

**Note** The **Failure mode** parameter is visible only if, in the **Main** settings, the **Control Port** parameter is set to **Electrical**. For more information, see “Control port” on page 1-0 .

---

**Faults Parameter Dependencies**

<b>Faults</b>														
<b>Enable faults</b>														
No	Yes													
	<b>Reporting when a fault occurs</b>													
	<b>Failure mode</b>													
	Switch fault											Winding failed open circuit		
	<b>Switch fault</b>													
	C-S stuck closed				C open circuit (no path to S)			Degraded contact resistance		Time constant for winding open circuit transition				
								<b>Degraded contact resistance</b>						
								<b>Time constant for degraded contact resistance</b>						
	<b>Enable temporal fault trigger</b>				<b>Enable temporal fault trigger</b>			<b>Enable temporal fault trigger</b>		<b>Enable temporal fault trigger</b>				
	No		Yes			No		Yes	No		Yes	No		Yes
			<b>Simulation time after which contactors can stick</b>					<b>Simulation time after which C can be open circuit</b>			<b>Simulation time for fault event</b>			<b>Simulation time for fault event</b>
	<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>			<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>	<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>		<b>Enable behavioral fault trigger</b>	
	No	Yes	No	Yes			No	Yes	No	Yes	No	Yes	No	Yes
		<b>Maximum permissible load current</b>						<b>Maximum permissible load current</b>		<b>Maximum permissible load current</b>		<b>Maximum permissible load current</b>		<b>Maximum permissible winding voltage</b>

Faults													
				Maximum permissible load current								Number of events to fail when exceeding voltage	Number of events to fail when exceeding voltage
												Maximum permissible winding current	Maximum permissible winding current
		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current		Time to fail when exceeding current	Time to fail when exceeding current

**Enable faults – Whether to model faults**

No (default) | Yes

Whether to model faults.

**Dependencies**

If this parameter is set to Yes, related parameters are visible. For more information, see Faults Parameter Dependencies.

**Reporting when a fault occurs — Reporting when a fault occurs**

None (default) | Warn | Error

Simulation reporting when a fault occurs:

- **None** — Does not generate a warning or an error.
- **Warn** — Generates a warning.
- **Error** — The simulation stops and generates an error.

**Dependencies**

This parameter is visible only if the **Enable faults** parameter is set to Yes. For more information, see Faults Parameter Dependencies.

**Failure mode — Fault model**

Winding failed open circuit (default) | Switch fault

Failure mode.

**Dependencies**

This parameter is visible only if, in the **Main** settings, the **Control port** parameter is set to **Electrical** and, in the **Faults** settings, the **Enable faults** parameter is set to Yes. For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

The specified option for this parameter affects the visibility of other parameters in the **Faults** settings. For more information, see Faults Parameter Dependencies.

**Switch fault — Switch fault model**

C-S stuck closed (default) | C open circuit (no path to S) | Degraded contact resistance

Switch fault model.

**Dependencies**

This parameter is visible only if either of these conditions are met:

- In the **Main** settings, the **Control port** parameter is set to **PS** and, in the **Faults** settings, the **Enable faults** parameter is set to Yes.
- In the **Main** settings, the **Control port** parameter is set to **Electrical**, and in the **Faults** settings, the **Enable faults** parameter is set to Yes and the **Failure mode** parameter is set to **Switch fault**.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Degraded contact resistance C-S — Degraded contact resistance C-S**

1e-9 Ohm (default) | positive scalar

Degraded resistance of the faulted **S** contact.**Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Time constant for degraded contact resistance — Time constant for degraded contact resistance**

1e-3 (default) | positive scalar

Time constant for contact resistance to fall into degraded status.

**Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Time constant for winding open circuit transition — Time constant for winding open circuit transition**

1e-3 s (default) | positive scalar

Time constant for the winding open circuit transition.

**Dependencies**

This parameter is visible only if all of these conditions are met:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Enable temporal fault trigger — Whether to enable a temporal fault trigger**

No (default) | Yes

Whether to enable a temporal fault trigger.

**Dependencies**

This parameter is visible only if the **Enable faults** parameter is set to Yes.

If this parameter is set to Yes, related parameters become visible.

For more information, see Faults Parameter Dependencies.

**Simulation time after which contactors can stick — Simulation time after which the S contact can stick closed**

1 s (default) | positive scalar

Simulation time after which the **S** contact can stick closed for a C-S stuck closed fault.

**Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to C-S stuck closed.
  - The **Enable temporal fault trigger** parameter is set to Yes.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to C-S stuck closed.
  - The **Enable temporal fault trigger** parameter is set to Yes.



For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Simulation time after which C can be open circuit – Simulation time after which the relay can be faulted to the open-circuit state**

1 s (default) | positive scalar

Simulation time after which, for the C open circuit (no path to S) fault, the relay can be faulted to the open-circuit state.

**Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to C open circuit (no path to S).
  - The **Enable temporal fault trigger** parameter is set to Yes.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to C open circuit (no path to S).
  - The **Enable temporal fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Simulation time for fault event – Simulation time for winding fault or degraded resistance fault**

1 s (default) | nonnegative scalar

Simulation time after which a fault can occur for either a Degraded contact resistance or a Winding failed open circuit fault.

**Dependencies**

This parameter is visible only if either one of these three sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Switch fault** parameter is set to Degraded contact resistance.
  - The **Enable temporal fault trigger** parameter is set to Yes.

**Condition Set Two**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Switch fault** parameter is set to Degraded contact resistance.
  - The **Enable temporal fault trigger** parameter is set to Yes.

**Condition Set Three**

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable temporal fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Enable behavioral fault trigger — Whether to enable a behavioral fault trigger**

No (default) | Yes

Whether to enable a behavioral fault trigger.

**Dependencies**

This parameter is visible only if the **Enable faults** parameter is set to Yes.

If this parameter is set to Yes, related parameters become visible.

For more information, see Faults Parameter Dependencies.

**Maximum permissible load current — Maximum permissible load current**

inf A (default) | positive scalar

Upper load-current threshold for degraded contact resistance and open-circuit winding faults.

**Dependencies**

This parameter is visible only if either one of these two sets of conditions is met:

**Condition Set One**

- In the **Main** settings, the **Control port** parameter is set to PS.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

### Condition Set Two

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Switch fault.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Maximum permissible winding voltage – Maximum permissible winding voltage

inf V (default) | positive scalar

Maximum voltage for the relay winding, above which a fault may be triggered.

#### Dependencies

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Number of events to fail when exceeding voltage – Number of events to fail when exceeding voltage

1 (default) | positive scalar integer

Number of times the relay must exceed the winding voltage threshold before a Winding failed open circuit fault is triggered.

#### Dependencies

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to Electrical.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to Yes.
  - The **Failure mode** parameter is set to Winding failed open circuit.
  - The **Enable behavioral fault trigger** parameter is set to Yes.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

### Maximum permissible winding current – Maximum permissible winding current

inf A (default) | positive scalar

Winding current threshold for triggering a behavioral Winding failed open circuit fault.

**Dependencies**

This parameter is visible only if:

- In the **Main** settings, the **Control port** parameter is set to **Electrical**.
- In the **Faults** settings:
  - The **Enable faults** parameter is set to **Yes**.
  - The **Failure mode** parameter is set to **Winding failed open circuit**.
  - The **Enable behavioral fault trigger** parameter is set to **Yes**.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

**Time to fail when exceeding current – Time to fail when exceeding current**

1 s (default) | positive scalar

Amount of time that the load or winding current must continuously exceed the maximum permissible current before a behavioral fault is triggered.

**Dependencies**

This parameter is visible only if both of these conditions are met:

- The **Enable faults** parameter is set to **Yes**.
- The **Enable behavioral fault trigger** parameter is set to **Yes**.

For more information, see “Control port” on page 1-0 and Faults Parameter Dependencies.

## Version History

Introduced in R2019b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

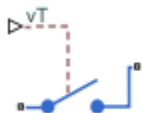
**See Also**

Switch | SPST Switch (Three-Phase) | SPDT Relay

# SPST Switch

Single-pole single-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



## Description

The SPST Switch block models a single-pole single-throw switch:

- When the switch is open, port **p** is connected to port **n** through a resistance with value equal to the reciprocal of the **Open conductance** parameter value.
- When the switch is closed, port **p** is connected to port **n** through a resistance with value equal to the **Closed resistance** parameter value.



If the **Threshold width** parameter is set to zero, the switch is closed if the voltage presented at the **vT** 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:

$$G = \frac{x}{R_{closed}} + (1 - x)G_{open}$$

$$\lambda = \frac{vT - \text{Threshold}}{\text{Threshold width}}$$

$$x = \begin{cases} 0 & \text{for } \lambda \leq 0 \\ 3\lambda^2 - 2\lambda^3 & \text{for } 0 < \lambda < 1 \\ 1 & \text{for } \lambda \geq 1 \end{cases}$$

The block uses the function  $3\lambda^2 - 2\lambda^3$  because its derivative is zero for  $\lambda = 0$  and  $\lambda = 1$ .

Defining a small positive **Threshold width** can help solver convergence in some models, particularly if the control port signal  $vT$  varies continuously as a function of other network variables. However, defining a nonzero threshold width precludes the solver making use of switched linear optimizations. Therefore, if the rest of your network is switched linear, set **Threshold width** to zero.

Optionally, you can add a delay between the point at which the voltage at **vT** passes the threshold and the switch opening or closing. To enable the delay, on the **Dynamics** tab, set the **Model dynamics** parameter to Model turn-on and turn-off times.

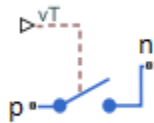
## Model Gate Control Port

You can control the switch gates by using a physical signal port or electrical conserving ports. To select the gate control port, set the **Modeling option** parameter to either:

- `PS control port` — The block contains a physical signal port that is associated with the threshold voltage.
- `Electrical control port` — The block contains electrical conserving ports that are associated with the threshold voltage.

## Ports

Refer to the figure for port locations.



### Input

#### **vT — Switch control port**

physical signal

Physical port associated with the signal that opens and closes the switch.

#### **Dependencies**

To enable this port, set **Modeling option** to `PS control port`.

### Conserving

#### **p — Electrical port**

electrical

Electrical conserving port.

#### **n — Electrical port**

electrical

Electrical conserving port.

#### **+ — Electrical port**

electrical

Electrical conserving port for the positive voltage associated with opening and closing the switch.

#### **Dependencies**

To enable this port, set **Modeling option** to `Electrical control port`.

#### **- — Electrical port**

electrical

Electrical conserving port for the negative voltage associated with opening and closing the switch.

### Dependencies

To enable this port, set **Modeling option** to `Electrical control port`.

## Parameters

### Modeling option — Whether to specify physical or electrical control port

`PS control port` (default) | `Electrical control port`

Whether to specify physical or electrical control port for the switch gate.

### Main

#### Closed resistance — Closed resistance

`0.01 Ohm` (default) | positive scalar

Resistance between the **p** port and **n** when the switch is closed. The value must be greater than zero.

#### Open conductance — Open conductance

`1e-6 S` (default) | positive scalar

Conductance between the **p** port and **n** when the switch is open. The value must be greater than zero.

#### Threshold — Threshold voltage

`0.5 V` (default)

The threshold voltage above which the switch will turn on.

#### Threshold width — Threshold width

`0 V` (default)

The minimum voltage increase above the threshold value that will move the switch from fully open to fully closed.

### Dynamics

#### Model dynamics — Model dynamics

`No dynamics` (default) | `Model turn-on and turn-off times`

Select whether the block models a switching delay:

- `No dynamics` — Do not model the delay. This is the default option.
- `Model turn-on and turn-off times` — Use additional parameters to model a delay between the point at which the voltage at **vT** or **+** and **-** passes the threshold and the switch opening or closing.

#### Turn-on delay — Turn-on delay

`1e-3` (default) | positive scalar

Time between the input voltage exceeding the threshold voltage and the switch closing. The value must be greater than zero.

**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

**Turn-off delay — Turn-off delay**

1e-3 (default) | positive scalar

Time between the input voltage falling below the threshold voltage and the switch opening. The value must be greater than zero.

**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

**Initial input value,  $vT$  — Initial input value,  $vT$** 

0 (default)

The value of the physical signal input  $vT$  at time zero. This value is used to initialize the delayed control voltage parameter internally.

**Dependencies**

This parameter is visible only when you select `Model` `turn-on` and `turn-off` times for the **Model dynamics** parameter.

## Version History

Introduced in R2012b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

[DPDT Switch](#) | [DPST Switch](#) | [SPDT Switch](#) | [SPDT Switch \(Three-Phase\)](#) | [SPST Switch \(Three-Phase\)](#)

**Topics**

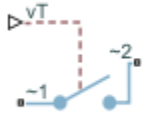
“Switch Between Physical Signal and Electrical Ports”



# SPST Switch (Three-Phase)

Three-phase single-pole single-throw switch

**Library:** Simscape / Electrical / Switches & Breakers



## Description

The SPST Switch (Three-Phase) block models a three-phase single-throw switch that uses an external signal to connect each phase of port ~1 with the corresponding phase of port ~2 via internal resistance.

The table shows how the external signal  $vT$  controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
$vT \leq \text{Threshold}$	The switch is open. Each phase in the composite three-phase port ~1 connects to the corresponding phase in the port ~2 via large internal resistance.	Open conductance
$vT > \text{Threshold}$	The switch is closed. Each phase in the composite three-phase port ~1 connects to the corresponding phase in the port ~2 via small internal resistance.	Closed resistance

## Ports

### Input

#### $vT$ — Control port

physical signal

Physical control port.

### Dependencies

To enable this port, set **Modeling option** to Expanded three-phase ports | PS control port or Composite three-phase ports | PS control port.

### Conserving

#### ~1 — Expandable three-phase port

electrical

Expandable three-phase electrical conserving port.

**~2 — Expandable three-phase port**

electrical

Expandable three-phase electrical conserving port.

**vT — Control port**

electrical

Electrical conserving control port.

**Dependencies**

To enable this port, set **Modeling option** to Expanded three-phase ports | Electrical control port or Composite three-phase ports | Electrical control port.

**Parameters****Modeling option — Connection and gate control ports**

Composite three-phase ports | PS control port (default) | Composite three-phase ports | Electrical control port | Expanded three-phase ports | PS control port | Expanded three-phase ports | Electrical control port

Whether to model composite or expanded three-phase ports and physical signal or electrical control ports.

**Closed resistance — Closed resistance**

0.001 Ohm (default)

Resistance between ports ~1 and ~2 when the switch is closed.

**Open conductance — Open conductance**

1e-6 1/Ohm (default)

Conductance between ports ~1 and ~2 when the switch is open.

**Threshold — Threshold**

0.5 V (default)

Threshold voltage for the control port vT. When the voltage is above the threshold, the switch is closed.

**Version History**

Introduced in R2013b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

DPDT Switch | DPST Switch | SPDT Switch | SPDT Switch (Three-Phase) | SPST Switch

**Topics**

“Expand and Collapse Three-Phase Ports on a Block”  
“Switch Between Physical Signal and Electrical Ports”

# SRM Commutation Logic

Commutation logic for switched reluctance machines

**Library:** Simscape / Electrical / Control / SRM Control



## Description

The SRM Commutation Logic block provides a logic signal for the Switched Reluctance Machine (SRM) block. The signal indicates when to switch the supply for each phase on and off.

The commutation signal for each phase is:

- 1 if  $\theta_{on} \leq \theta_{ph} \leq \theta_{off}$ .
- 0 if  $\theta_{ph} < \theta_{on}$ .
- 0 if  $\theta_{ph} > \theta_{off}$ .

where:

- $\theta_{ph}$  is the phase angle in the interval  $[0, \beta]$ .
- $\beta$  is the torque capability angle.
- $\theta_{on}$  is the switch-on angle.
- $\theta_{off}$  is the switch-off angle.

## Ports

### Input

#### **theta — Phase angle**

scalar

Phase angle,  $\theta_{ph}$ , in the interval  $[0, \beta]$ .

Data Types: single | double

#### **thetaON — Switch-on angle**

scalar

Lower threshold for turning on the switch.

Data Types: single | double

#### **thetaOFF — Switch-off angle**

scalar

Upper threshold for turning off the switch.

Data Types: `single` | `double`

## Output

### **abcOnOff** – Commutation signal

vector

Commutation signal for each phase: 1 for the switch-on condition and 0 for switch-off condition

Data Types: `single` | `double`

## Parameters

### **Torque capability angle (rad) – Torque angle**

$\pi/2$  (default)

The torque production capability of one rotor pole, in radians.

### **Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

Introduced in R2018a

## Extended Capabilities

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

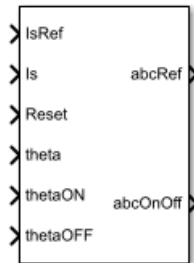
## See Also

SRM Current Controller | SRM Hysteresis Current Controller | SRM Current Controller with PWM Generation | Switched Reluctance Machine

# SRM Current Controller

Current control for switched reluctance machines

**Library:** Simscape / Electrical / Control / SRM Control



## Description

The SRM Current Controller block performs discrete-time proportional-integral (PI) current control for the Switched Reluctance Machine (SRM) block.

## Equations

To determine the duty cycle, the block implements discrete-time proportional-integral (PI) current control in accordance with this equation.

$$D = \left( K_p + K_i \frac{T_s z}{z-1} \right) (I_{s\_ref} - I_s)$$

Where:

- $D$  is the duty cycle.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the sample time.
- $I_{s\_ref}$  is the reference current.
- $I_s$  is the measured current.

To obtain control signals for the three-phases, the block then multiplies the duty cycle with the commutation signals. The resulting three control signals are normalized over the interval [0, 1].

## Ports

### Input

#### IsRef — Reference current

scalar

Reference current for control.

Data Types: single | double

**Is — Measured current**

scalar

Measured current.

Data Types: `single` | `double`**Reset — Integrator reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `Boolean`**theta — Theta-rotor angle**

scalar

Rotor angle in the interval  $[0, \beta]$ .Data Types: `single` | `double`**theta0N — Switch-on angle**

scalar

Angle for switching on the phase supply.

Data Types: `single` | `double`**theta0FF — Switch-off angle**

scalar

Angle for switching on the phase supply.

Data Types: `single` | `double`**Output****abcRef — Reference voltage**

vector

Control signal normalized in the interval  $[0, 1]$ .Data Types: `single` | `double`**abc0n0ff — Switch signal**

vector

Switch signal for the  $a$ -,  $b$ -, and  $c$ -phases. 1 for the switch-on condition and 0 for switch-off conditionData Types: `single` | `double`**Parameters****Proportional gain — Controller proportional gain,  $K_p$** 

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

**Integral gain – Integral gain,  $K_i$** 

5 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.**Anti-windup gain – Anti-windup gain,  $K_{aw}$** 

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.**Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

Introduced in R2018a

## References

- [1] Saha, N. and S. Panda. "Speed control with torque ripple reduction of switched reluctance motor by Hybrid Many Optimizing Liaison Gravitational Search technique." *Engineering Science and Technology*. Vol 20 (2017): 909–921.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

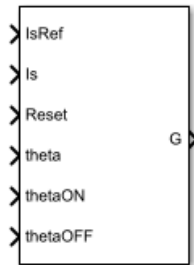
SRM Commutation Logic | SRM Hysteresis Current Controller | SRM Current Controller with PWM Generation | Switched Reluctance Machine



# SRM Current Controller with PWM Generation

Current controller with internal pulse width modulation for switched reluctance machines

**Library:** Simscape / Electrical / Control / SRM Control

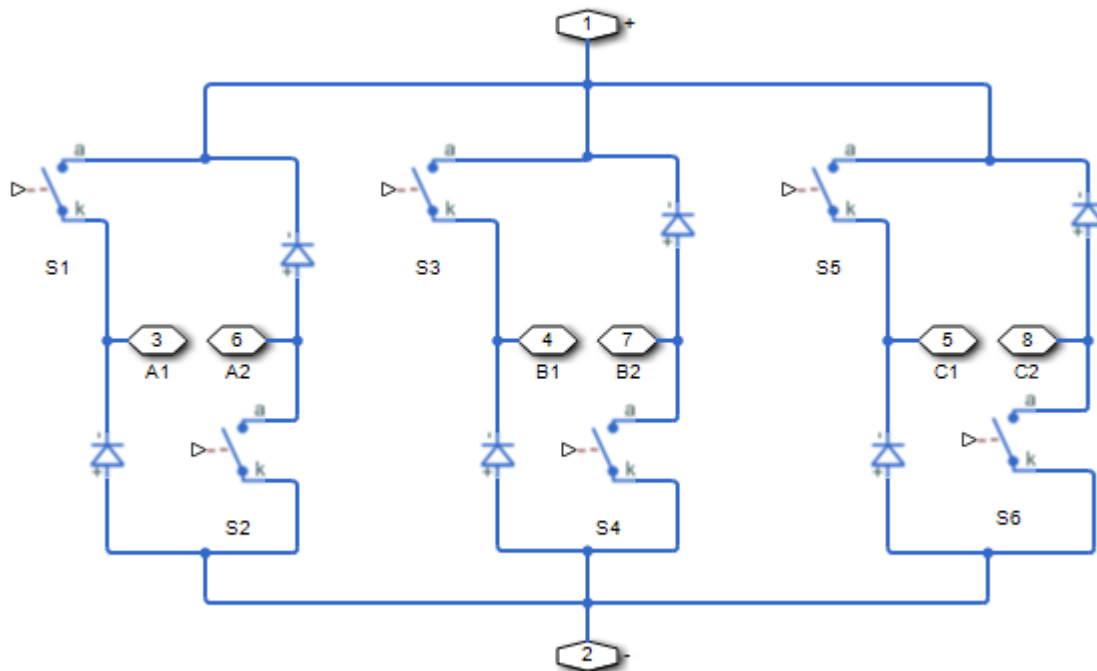


## Description

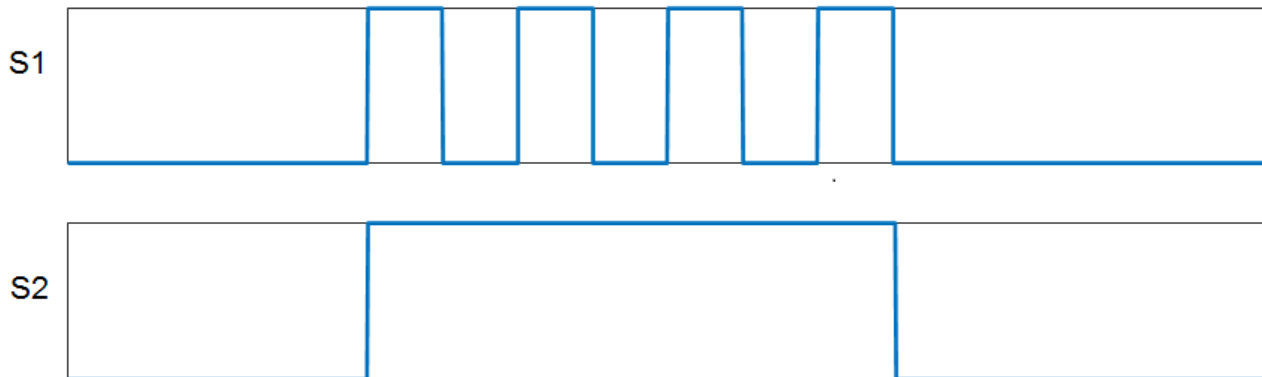
The SRM Current Controller with PWM Generation block performs discrete-time proportional-integral (PI) current control for the Switched Reluctance Machine (SRM) block. The block includes pulse width modulation (PWM).

### PWM Generation Model

The figure shows the converter structure for an SRM.



As the figure shows, the PWM generation signal is for high side switching devices.



When the control signal is greater than the carrier counter value, the PWM generator outputs 1. Otherwise, it outputs 0.

### Equations

To determine the duty cycle, the block implements PI current control in the rotor reference frame in accordance with this equation.

$$D = \left( K_p + K_i \frac{T_s z}{z-1} \right) (I_{s\_ref} - I_s)$$

Where:

- $D$  is the duty cycle.
- $K_p$  is the proportional gain.
- $K_i$  is the integral gain.
- $T_s$  is the sample time.
- $I_{s\_ref}$  is the reference current.
- $I_s$  is the measured current.

To obtain control signals for the three-phases, the block then multiplies the duty cycle with the commutation signals. The resulting three control signals are normalized over the interval [0, 1].

### Ports

#### Input

##### IsRef – Reference current

scalar

Reference current for control.

Data Types: `single` | `double`

##### Is – Measured current

scalar

Actual current.

Data Types: single | double

### **Reset — Integrator reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: Boolean

### **theta — Theta-rotor angle**

scalar

Rotor angle in the interval  $[0, \beta]$ .

Data Types: single | double

### **theta0N — Switch-on angle**

scalar

Angle for switching on the phase supply.

Data Types: single | double

### **theta0FF — Switch-off angle**

scalar

Angle for switching on the phase supply.

Data Types: single | double

## **Output**

### **G — Gate control**

vector

Pulse waveforms that determine switching behavior.

Data Types: single | double

## **Parameters**

### **Control Parameters**

#### **Proportional gain — Controller proportional gain, $K_p$**

1 (default) | positive scalar

Proportional gain,  $K_p$ , of the controller.

#### **Integral gain — Integral gain, $K_i$**

5 (default) | positive scalar

Integral gain,  $K_i$ , of the controller.

#### **Anti-windup gain — Anti-windup gain, $K_{aw}$**

1 (default) | positive scalar

Anti-windup gain,  $K_{aw}$ , of the controller.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**PWM Generator****Carrier counter – Carrier counter model**

Up (default) | Down | Up-Down

Use the carrier counter strategy to change the initial behavior of the PWM output:

- Up counter — PWM output begins at the start of the on state.
- Down counter — PWM output begins at the start of the off state.
- Up-down counter — PWM output begins in the middle of the on state.

**Timer period (s) – PWM timer period**

0.001 (default) | positive scalar

Pulse width modulation timer period,  $T_{per}$ , in seconds.

**Fundamental sample time (s) – Sample time for PWM generation**

0.0001 (default) | positive scalar

Time, in s, between consecutive PWM generator executions. During execution, the block produces PWM output and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

To ensure adequate resolution in the generated PWM signal, set the fundamental sample time so that  $0 < T_{s\_pwm} \leq 10T_{per}$ , where:

- $T_{s\_pwm}$  is the **Fundamental sample time (s)**.
- $T_{per}$  is the **Timer period (s)**.

## Version History

Introduced in R2018a

## References

- [1] Saha, N. and S. Panda. "Speed control with torque ripple reduction of switched reluctance motor by Hybrid Many Optimizing Liaison Gravitational Search technique." *Engineering Science and Technology*. Vol 20 (2017): 909-921.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

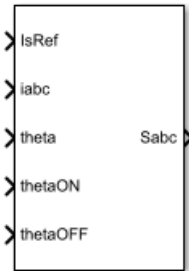
### **See Also**

SRM Commutation Logic | SRM Current Controller | SRM Hysteresis Current Controller | Switched Reluctance Machine

# SRM Hysteresis Current Controller

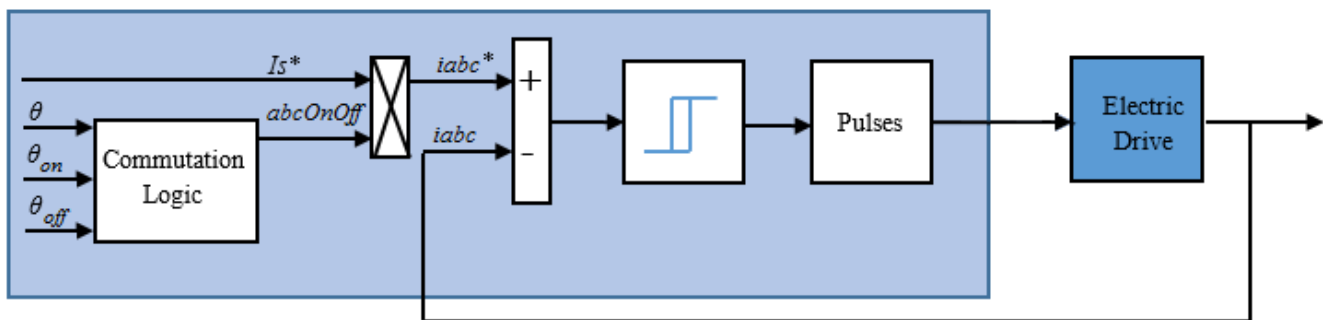
Hysteresis current control for switched reluctance machines

**Library:** Simscape / Electrical / Control / SRM Control



## Description

The SRM Hysteresis Current Controller block implements hysteresis current control for the Switched Reluctance Machine (SRM) block.



## Ports

### Input

#### **IsRef** — Reference current

scalar

Reference current for control.

Data Types: single | double

#### **iabc** — Measured current

vector

Measured three-phase current.

Data Types: single | double

#### **Reset** — Integrator reset

scalar

External reset signal (rising edge) for the integrator.

Data Types: Boolean

### **theta — Theta-rotor angle**

scalar

Rotor angle in the interval  $[0, \beta]$ .

Data Types: single | double

### **theta0N — Switch-on angle**

scalar

Angle for switching on the phase supply.

Data Types: single | double

### **theta0FF — Switch-off angle**

scalar

Angle for switching on the phase supply.

Data Types: single | double

## **Output**

### **Sabc — Voltage switch signal**

vector | 0 or 1

Signal for switching on and off the three-phase voltage.

Data Types: single | double

## **Parameters**

### **Torque capability angle (rad) — Torque capability angle**

$\pi/2$  (default) |  $[0, 2*\pi/\text{number of rotor poles}]$

The torque production capability of one rotor pole, in radians.

### **Hysteresis band (A) — Hysteresis bandwidth**

5 (default)

Hysteresis band,  $h$ , for the current controller. The switch-on point is  $h/2$  and the switch-off point is  $-h/2$ .

### **Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## **Version History**

**Introduced in R2018a**

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

#### **See Also**

SRM Commutation Logic | SRM Current Controller | SRM Current Controller with PWM Generation | Switched Reluctance Machine



# Stair Generator

Discrete-time or continuous-time stair signal generator

**Library:** Simscape / Electrical / Control / General Control



## Description

The Stair Generator block generates a stair signal based on the values specified in the **Time (s)** and **Amplitude** parameters.

The block can output a repeating stair pattern or execute the pattern only once by keeping the last amplitude active for all subsequent times.

## Ports

### Output

#### Port\_1 — Stair signal

Simulink signal

Stair signal, returned as a Simulink output signal.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

## Parameters

### Main

#### Time (s) — Transition times

[0.5 1 3 3.5] (default) | row vector of nonnegative values

Transition times at which the output changes to the value specified in the **Amplitude** parameter, in seconds. The value of this parameter must be strictly ascending.

#### Amplitude — Amplitudes of stair signal values

[1 -1 2.5 0] (default) | row vector

Amplitudes of the stair signal at the time specified by the **Time (s)** parameter. This vector must have the same number of elements as the **Time (s)** parameter.

#### Sample time (-1 for inherited) — Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

### Repeat stair pattern indefinitely – Stair pattern repetition

off (default) | on

Whether to indefinitely repeat the stair pattern. If you select this parameter, the output repeats the stair pattern defined by the **Time (s)** and **Amplitude** parameters indefinitely.

If you clear this parameter, the output executes the stair pattern only once and then keeps the last amplitude active for all subsequent times.

### Signal Attributes

#### Output data type – Specify the output data type

double (default) | Inherit: auto | Inherit: Inherit via internal rule | Inherit: Inherit via back propagation | single | int8 | uint8 | int16 | uint16 | int32 | uint32 | int64 | uint64 | <data type expression>

Data type for the output. The type can be inherited, specified directly, or expressed as a data type object such as `Simulink.NumericType`.

When you select an inherited option, the block exhibits these behaviors:

- **Inherit: Inherit via internal rule** — Simulink chooses a data type to balance numerical accuracy, performance, and generated code size, while taking into account the properties of the embedded target hardware. If you change the embedded target settings, the data type selected by the internal rule might change. It is not always possible for the software to optimize efficiency and numerical accuracy at the same time.
- **Inherit: Inherit via back propagation** — Simulink automatically determines the output data type of the block during data type propagation (see “Data Type Propagation”). In this case, the block uses the data type of a downstream block or signal object.
- **Inherit: auto** — Use a rule that inherits a data type.

For more information, see “Control Data Types of Signals”.

### Programmatic Use

**Block Parameter:** `OutDataTypeStr`

**Type:** character vector

**Values:** 'Inherit: auto' | 'Inherit: Inherit via internal rule' | 'Inherit: Inherit via back propagation' | 'single' | 'double' | 'int8' | 'uint8' | 'int16' | 'uint16' | 'int32' | 'uint32' | 'int64' | 'uint64' | '<data type expression>'

**Default:** 'Inherit: Inherit via internal rule'

## Version History

Introduced in R2020a

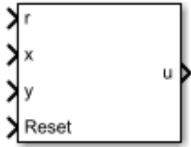
**See Also**

Sine Wave Generator (Three-Phase)

# State-Feedback Controller

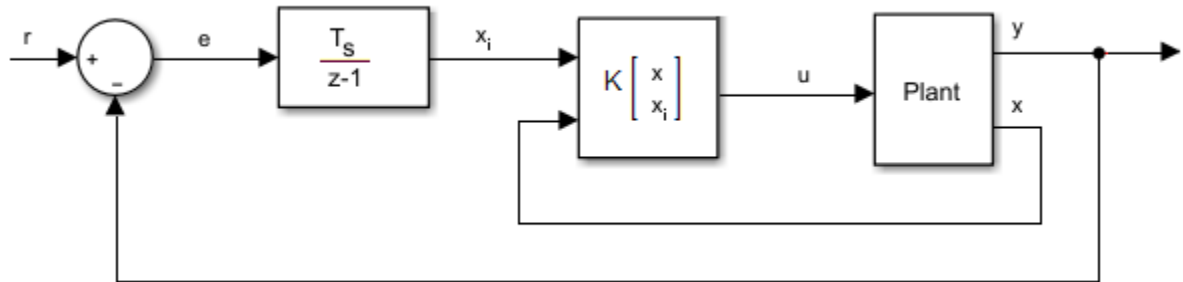
Discrete-time state-feedback controller with integral action

**Library:** Simscape / Electrical / Control / General Control



## Description

The State-Feedback Controller block implements a discrete-time state-feedback controller with integral action. Use this block to control linear systems with single or multiple inputs and single or multiple outputs. The integral action serves to eliminate steady-state error in the controlled outputs. You can define the controller using a precomputed optimal gain or use the state-space model of your system to generate this gain using pole placement.



## Equations

The integral of the tracking error,  $x_i$ , is an additional state that ensures zero steady-state error for the closed-loop system. The extended state vector is

$$x_e = \begin{bmatrix} x \\ x_i \end{bmatrix},$$

Where:

- $x$  is the state vector.
- $x_i$  is the integral of the tracking error.
- $x_e$  is the extended state vector.

Therefore, the control action is

$$u = Kx_e,$$

Where:

- $K$  is the feedback matrix, that is, the pole placement.
- $u$  is the controller output.

### Assumptions

System state measurement and estimation occur outside the controller.

## Ports

### Input

#### **r — Plant reference**

scalar

Plant system reference signal.

Data Types: `single` | `double`

#### **x — State vector**

vector

Measured or estimated system state vector.

Data Types: `single` | `double`

#### **Reset — Integrator reset**

scalar

External reset signal (rising edge) for the integrator.

Data Types: `Boolean`

#### **y — Plant output**

scalar

Plant system output signal.

Data Types: `single` | `double`

### Output

#### **u — Controller output**

scalar

Control system output signal.

Data Types: `single` | `double`

## Parameters

### **State-feedback design — Controller generation**

State-feedback gain (default) | Desired eigenvalues

Select the strategy for parameterizing controller gain:

- `State-feedback gain` — Specify the controller gain directly

- **Desired eigenvalues** — Specify the plant model and desired eigenvalues from which to generate the controller gain

### **State-feedback parameterization — State-feedback parameterization**

Discrete-time (default) | Continuous-time

Select the strategy for parameterizing the state-space matrices and desired poles for the controller. The block implementation is discrete regardless of this parameterization.

#### **Dependencies**

To enable this parameter, set **State-feedback design** to `Desired eigenvalues`.

### **Controller matrix — Controller matrix**

[1 1] (default) | matrix

Controller feedback matrix. To determine the controller matrix, if you have a license for Control System Toolbox, use the `lqr` or `lqi` function.

#### **Dependencies**

To enable this parameter, set **State-feedback design** to `State-feedback gain`.

### **Discrete-time A matrix — A matrix in discrete time**

1 (default) | real scalar or matrix

State matrix of the discrete-time state-space model. The A matrix must be square, with the number of rows and columns equal to the order of the system.

#### **Dependencies**

To enable this parameter, set **State-feedback parameterization** to `Discrete-time`.

### **Discrete-time B matrix — B matrix in discrete time**

1 (default) | real scalar or matrix

Input matrix of the discrete-time state-space model. The B matrix must have the number of rows equal to the order of the system, and the number of columns equal to the number of system inputs.

#### **Dependencies**

To enable this parameter, set **State-feedback parameterization** to `Discrete-time`.

### **Discrete-time C matrix — C matrix in discrete time**

1 (default) | real scalar or matrix

Output matrix of the discrete-time state-space model. The C matrix must have the number of rows equal to the number of outputs of the system, and the number of columns equal to the order of the system.

#### **Dependencies**

To enable this parameter, set **State-feedback parameterization** to `Discrete-time`.

### **Discrete-time D matrix — D matrix in discrete time**

1 (default) | real scalar or matrix

Feedthrough matrix of the discrete-time state-space model. The D matrix must have the number of rows equal to the number of system outputs, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-feedback parameterization** to Discrete-time.

**Continuous-time A matrix — A matrix in continuous time**

1 (default) | real scalar or matrix

State matrix of the continuous-time state-space model. The A matrix must be square, with the number of rows and columns equal to the order of the system.

**Dependencies**

To enable this parameter, set **State-feedback parameterization** to Continuous-time.

**Continuous-time B matrix — B matrix in continuous time**

1 (default) | real scalar or matrix

Input matrix of the continuous-time state-space model. The B matrix must have the number of rows equal to the order of the system, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-feedback parameterization** to Continuous-time.

**Continuous-time C matrix — C matrix in continuous time**

1 (default) | real scalar or matrix

Output matrix of the continuous-time state-space model. The C matrix must have the number of rows equal to the number of outputs of the system, and the number of columns equal to the order of the system.

**Dependencies**

To enable this parameter, set **State-feedback parameterization** to Continuous-time.

**Continuous-time D matrix — D matrix in continuous time**

1 (default) | real scalar or matrix

Feedthrough matrix of the continuous-time state-space model. The D matrix must have the number of rows equal to the number of system outputs, and the number of columns equal to the number of system inputs.

**Dependencies**

To enable this parameter, set **State-feedback parameterization** to Continuous-time.

**Discretization sample time — Discretization sample time**

0.1 (default) | positive real number

Value used to discretize the state space matrices and also approximate the discrete-time eigenvalues.

### Dependencies

To enable this parameter, set **State-feedback parameterization** to Continuous-time and **Sample time (-1 for inherited)** to -1.

### Desired eigenvalues (discrete) – Observer eigenvalues

0 (default) | real vector

Specify the location of the eigenvalues to lie within the unit circle. The controller gain is then calculated based on these eigenvalues. The size of the vector must be equal to the system order plus the number of outputs.

### Control action upper limit – $u_{max}$

5 (default) | scalar greater than the value of the **Control action lower limit** parameter

Upper limit for the control output signal.

### Control action lower limit – $u_{min}$

0 (default) | scalar

Lower limit for the control output signal.

### Sample time (-1 for inherited) – Sampling interval

-1 (default) | default value or a positive number

Time interval between samples. If the block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

## Version History

Introduced in R2017b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

RST Controller | Smith Predictor Controller



# Stepper Motor

Stepper motor suitable for whole-, half- and micro-stepping representation

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Stepper Motor block represents a stepper motor. It uses the input pulse trains, A and B, to control the mechanical output according to the following equations:

$$e_A = -K_m \omega \sin(N_r \theta)$$

$$e_B = K_m \omega \cos(N_r \theta)$$

$$\frac{di_A}{dt} = (v_A - Ri_A - e_A)/L$$

$$\frac{di_B}{dt} = (v_B - Ri_B - e_B)/L$$

$$J \frac{d\omega}{dt} + B\omega = T_e$$

$$T_e = -K_m \left( i_A - \frac{e_A}{R_m} \right) \sin(N_r \theta) + K_m \left( i_B - \frac{e_B}{R_m} \right) \cos(N_r \theta) - T_d \sin(4N_r \theta)$$

$$\frac{d\theta}{dt} = \omega$$

where:

- $e_A$  and  $e_B$  are the back electromotive forces (emfs) induced in the A and B phase windings, respectively.
- $i_A$  and  $i_B$  are the A and B phase winding currents.
- $v_A$  and  $v_B$  are the A and B phase winding voltages.
- $K_m$  is the motor torque constant.
- $N_r$  is the number of teeth on each of the two rotor poles. The **Full step size** parameter is  $(\pi/2)/N_r$ .
- $R$  is the winding resistance.
- $L$  is the winding inductance.
- $R_m$  is the magnetizing resistance.
- $B$  is the rotational damping.
- $J$  is the inertia.

- $\omega$  is the rotor speed.
- $\theta$  is the rotor angle.
- $T_d$  is the detent torque amplitude.
- $T_e$  is the electrical torque.

If the initial rotor is zero or some multiple of  $(\pi/2)/N_r$ , the rotor is aligned with the phase winding of pulse A. This happens when there is a positive current flowing from the **A+** to the **A-** ports and there is no current flowing from the **B+** to the **B-** ports.

Use the Stepper Motor Driver block to create the pulse trains for the Stepper Motor block.

The Stepper Motor block produces a positive torque acting from the mechanical **C** to **R** ports when the phase of pulse A leads the phase of pulse B.

### Averaged Mode

If you set the **Simulation mode** parameter to **Averaged**, both for a Stepper Motor block and for the Stepper Motor Driver block that controls it, then the individual steps are not simulated. This can be a good way to speed up simulation. In Averaged mode, under nonslipping conditions, the motor and driver are represented by a second-order linear system that tracks the specified step rate. The demanded step rate is determined directly from voltage across **A+** and **A-**. So, for example, a voltage of +10 V across the **A+** and **A-** terminals is interpreted as a step rate demand of 10 steps per second. See the Stepper Motor Driver block reference page for more information on how to connect the driver block to your step angle controller.

Averaged mode includes a slip estimator to predict whether the stepper motor would have slipped if running in Stepping simulation mode. Slip is predicted if the motor torque exceeds the **Vector of maximum torque values** parameter value for longer than one step period, the step period being determined from the current step rate demand. Upon detecting slip, the simulation will proceed or stop with an error, according to the **Action on slipping** parameter value. If you choose the action that lets the simulation continue, note that simulation results may be incorrect. When slipping occurs, the torque generated by the motor is not generally the maximum available torque; the maximum torque is only achieved if the stepper controller detects slip and adjusts the step rate command accordingly.

The dynamics of the equivalent second-order system are determined from the values that you specify for the **Approximate total load inertia** and **Maximum step rate command** parameters. It is important that you set as accurate values as possible for these parameters, so that the step rate command is tracked, and the block does not generate false slipping warnings or errors.

If you run the motor in Averaged mode with the optional thermal ports exposed (see “Model Thermal Effects” on page 1-2245), then heat is added to the thermal ports, assuming that the windings are always powered even when the step rate command is zero. The block makes adjustments for half stepping and for reduced torque (and winding currents) at higher speeds. For these adjustments to be correct, the **Vector of maximum torque** parameter values must be correct. For half stepping, at zero speed the heat generated by the block is the average of that generated when stopped at a half step and at a full step.

To validate Averaged mode model configurations where you predict slip to occur, compare results with the same simulation performed in stepping mode.

## Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

If you expose these ports, winding resistance is assumed linearly dependent on temperature, and is given by:

$$R = R_0 (1 + \alpha (T - T_0)) \quad (1-53)$$

where:

- $R$  is the resistance at temperature  $T$ .
- $R_0$  is the resistance at the measurement (or reference) temperature  $T_0$ . Specify the reference temperature using the **Measurement temperature** parameter.
- $\alpha$  is the resistance temperature coefficient, which you specify with the **Resistance temperature coefficients, [alpha\_A alpha\_B]** parameter. A typical value for copper is 0.00393/K.

The block calculates temperature of each of the windings and the rotor by

$$M \frac{dT}{dt} = Q$$

where

- $M$  is the thermal mass. Specify this value for the windings using the **Winding thermal masses, [M\_A M\_B]** parameter, and for the rotor using the **Rotor thermal mass** parameter.
- $T$  is the temperature. Specify the initial values for the windings using the **Winding initial temperatures, [T\_A T\_B]** parameter, and for the rotor using the **Rotor initial temperature** parameter.
- $Q$  is the heat flow, which is calculated from the iron losses of the windings:

$$Q_A = i_a^2 R_A (1 - \rho_m / 100)$$

$$Q_B = i_b^2 R_B (1 - \rho_m / 100)$$

$$Q_R = Q_A (\rho_m / 100) + Q_B (\rho_m / 100)$$

where  $\rho_m$  is the percentage of magnetizing resistance associated with the rotor. Specify this percentage using the **Percentage of magnetizing resistance associated with the rotor** parameter.

## Predefined Parameterization

There are multiple available built-in parameterizations for the Stepper Motor block.

This pre-parameterization data allows you to set up the block to represent a specific supplier component. To load a predefined parameterization, double-click the Stepper Motor block, click the

<click to select> hyperlink of the **Selected part** parameter and, in the Block Parameterization Manager window, select the part you want to use from the list of available components.

---

**Note** The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

---

For more information about pre-parameterization and for a list of the available components, see “List of Pre-Parameterized Components”.

## Assumptions and Limitations

The model is based on the following assumptions:

- This model neglects magnetic saturation effects and any magnetic coupling between phases.
- When you select the **Start simulation from steady state** check box in the Simscape Solver Configuration block, this block will not initialize an **Initial rotor angle** value between  $-\pi$  and  $\pi$ .
- To use Averaged mode, the Stepper Motor block must be directly connected to a Stepper Motor Driver block also running in Averaged mode.
- The Averaged mode is an approximation, and exact step tracking compared to the Stepping mode should not be expected.
- Slip detection in Averaged mode is approximate, and depends on a good estimate for load inertia and maximum step rate. Incorrect values may result in false slip detection.
- When simulating slip in Averaged mode, it is assumed that the stepper motor controller adjusts the step rate command so as to achieve maximum possible torque.

## Ports

### Conserving

#### **A+** — **A-phase positive terminal**

electrical

Electrical conserving port associated with the A-phase positive terminal.

#### **A-** — **A-phase negative terminal**

electrical

Electrical conserving port associated with the A-phase negative terminal.

#### **B+** — **B-phase positive terminal**

electrical

Electrical conserving port associated with the B-phase positive terminal.

#### **B-** — **B-phase negative terminal**

electrical

Electrical conserving port associated with the B-phase negative terminal.

### **C – Machine case**

mechanical rotational

Mechanical rotational conserving port connected to the motor case.

### **R – Machine rotor**

mechanical rotational

Mechanical rotational conserving port connected to the rotor.

### **HA – Winding A thermal mass**

thermal

Thermal conserving port associated with the thermal mass of winding A.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

### **HB – Winding B thermal mass**

thermal

Thermal conserving port associated with the thermal mass of winding B.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

### **HR – Rotor thermal mass**

thermal

Thermal conserving port associated with the thermal mass of the rotor.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

## **Parameters**

### **Selected part – Predefined parameterization option**

<click to select> (default)

Whether to parameterize the block to represent components by specific suppliers. Click the **<click to select>** hyperlink to open the Block Parameterization Manager window. For more information on the Block Parameterization Manager, see “Load Predefined Parameterizations”.

### **Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

## Electrical Torque

### Simulation mode — Simulation mode

Stepping (default) | Averaged

Use Averaged only if the block is connected directly to a Stepper Motor Driver block also running in Averaged mode.

### Vector of rotational speeds — Rotor speeds

[0, 1000, 3000] rpm (default) | positive vector

Vector of rotational speeds at which to define maximum torque values, for slip prediction.

#### Dependencies

To enable this parameter, set **Simulation mode** to Averaged.

### Vector of maximum torque values — Maximum torques

[2, 2, 1] N\*m (default) | positive vector

Vector of maximum torque values, to be used for slip prediction with the **Vector of rotational speeds** parameter. These values are often given on a datasheet, and correspond to the supply voltage and stepping type (half step or full step) specified in the driver.

#### Dependencies

To enable this parameter, set **Simulation mode** to Averaged.

### Action on slipping — Slip response

none (default) | warn | error

Select the action for the block to perform during simulation upon detecting slip:

- none — Continue simulation, limiting the load torque according to the **Vector of maximum torque values**.
- warn — Continue simulation, limiting the load torque according to the **Vector of maximum torque values**, and generate a warning that the rotor is slipping.
- error — Stop the simulation and generate an error message that the rotor is slipping.

If you choose an action that lets the simulation continue, simulation results might be incorrect. When slipping occurs, the motor does not always generate maximum torque. The maximum torque is only achieved if the stepper controller detects slip and adjusts the step rate command accordingly.

#### Dependencies

To enable this parameter, set **Simulation mode** to Averaged.

### Approximate total load inertia — Total load inertia

1e-4 kg\*m<sup>2</sup> (default) | positive number

Approximate total load inertia, including the rotor inertia. This value is used to help predict slip due to rapid acceleration demands.

#### Dependencies

To enable this parameter, set **Simulation mode** to Averaged.

**Maximum step rate command — Maximum command frequency**

10 Hz (default) | positive number

Maximum commanded step rate of the system. It is used to determine a suitable bandwidth for the second order system approximation to the stepper motor and driver.

**Dependencies**

To enable this parameter, set **Simulation mode** to Averaged.

**Phase winding resistance — Phase resistance**

0.55 Ohm (default) | positive number

Resistance of the A and B phase windings.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Phase winding inductance — Phase inductance**

1.5e-3 Ohm (default) | positive number

Inductance of the A and B phase windings.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Motor torque constant — Torque constant**

0.19 N\*m/A (default) | positive number

Motor torque constant,  $K_m$ .

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Detent torque — Torque variation amplitude**

0 N\*m (default) | positive number

Amplitude of the sinusoidal torque variation observed when rotating the shaft of the unpowered motor.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Magnetizing resistance — Phase magnetizing resistance**

inf (default) | strictly positive number

Total magnetizing resistance seen from each of the phase windings. The value must be greater than zero. The default value is Inf, which implies that there are no iron losses.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Full step size — Step size**

1.8 deg (default) | positive number

Step size when changing the polarity of either the A or B phase current.

### **Mechanical**

#### **Rotor inertia – Rotational inertia**

4.5e-5 kg\*m<sup>2</sup> (default) | zero or positive number

Conservative force resisting rotor acceleration.

#### **Rotor damping – Rotational damping**

8.0e-4 N\*m/(rad/s) (default) | zero or positive number

Dissipative force resisting rotor speed.

#### **Initial rotor speed – Initial speed**

0 rpm (default) | real number

Speed of the rotor at the start of the simulation.

#### **Initial rotor angle – Initial angle**

0 rad (default) | real number

Angle of the rotor at the start of the simulation.

### **Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Resistance temperature coefficients, [alpha\_A alpha\_B] – Temperature coefficients**

[.00393, .00393] 1/K (default) | positive vector

Two-element row vector defining the coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Model Thermal Effects” on page 1-2245. The first element corresponds to winding A, and the second to winding B. The default value is for copper.

#### **Measurement temperature – Reference temperature**

25 degC (default) | real number

Temperature for which motor parameters are defined.

### **Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Winding thermal masses, [M\_A M\_B] – Winding thermal masses**

[100, 100] J/K (default) | positive vector

Two-element row vector defining the thermal mass for the A and B windings. The thermal mass is the energy required to raise the temperature by one degree.

#### **Winding initial temperatures, [T\_A T\_B] – Initial winding temperatures**

[25, 25] degC (default) | real number

Two-element row vector defining the temperature of the A and B thermal ports at the start of simulation.



**Rotor thermal mass — Rotor thermal mass**

50 J/K (default) | positive number

Thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Rotor initial temperature — Initial rotor temperature**

25 degC (default) | real number

Temperature of the rotor at the start of simulation.

**Percentage of magnetizing resistance associated with the rotor — Magnetizing resistance rotor percentage**

90 (default) | positive number between 0 and 100

Percentage of the magnetizing resistance associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port, **HR**, and winding thermal ports, **HA** and **HB**.

## Version History

Introduced in R2008a

## References

- [1] M. Bodson, J. N. Chiasson, R. T. Novotnak and R. B. Rekowski. "High-Performance Nonlinear Feedback Control of a Permanent Magnet Stepper Motor." IEEE Transactions on Control Systems Technology, Vol. 1, No. 1, March 1993.
- [2] P. P. Acarnley. *Stepping Motors: A Guide to Modern Theory and Practice*. New York: Peregrinus, 1982.
- [3] S.E. Lyshevski. *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC, 1999.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Stepper Motor Driver | Unipolar Stepper Motor

**Topics**

"List of Pre-Parameterized Components"

# Stepper Motor Driver

Driver for stepper motor

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Stepper Motor Driver block represents a driver for a stepper motor. It creates the pulse trains, *A* and *B*, required to control the motor. This block initiates a step each time the voltage at the **ENA** port rises above the **Enable threshold voltage** parameter value.

If the voltage at the **REV** port is less than or equal to the **Reverse threshold voltage** parameter value, pulse *A* leads pulse *B* by 90 degrees. If the voltage at the **REV** port is greater than the **Reverse threshold voltage** value, pulse *B* leads pulse *A* by 90 degrees and the motor direction is reversed.

At time zero, pulse *A* is positive and pulse *B* is negative.

If you set the **Stepping mode** parameter to **Half stepping**, the Stepper Motor Driver block can produce the output waveforms required for half stepping. In this mode, there is 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. At half steps, windings that are not powered are short-circuited. This approximates the effect of a freewheeling diode connected across the windings.

## Microstepping

To advance the rotor of a fraction of a step instead of a full or half step, set the **Stepping mode** parameter to **Microstepping**. This mode enables circular current-controlled microstepping and increases the resolution in the position of the rotor. To select the fractional stepping resolution, specify the value of the **Number of microsteps per step** parameter.

The block initiates a micro-step each time the voltage at the **ENA** port rises above the value of the **Enable threshold voltage** parameter.

In microstepping mode, the current in each motor winding follows a discretized sinusoidal waveform. The amplitude of the current waveform is equal to the value of the **Maximum continuous current per phase** parameter.

To increase or decrease the current in each microstep, the driver applies a positive or negative over-voltage until the current matches the target value.

If you set the **Power supply** parameter to **Internal**, the applied over-voltage is equal to the value of the **Supply voltage** parameter. If you set the **Power supply** parameter to **External** the applied over-voltage is equal to the voltage between the **VM** and **GND** ports.

## Averaged Mode

If you set the **Simulation mode** parameter to *Averaged*, both for a Stepper Motor Driver block and for the Stepper Motor block connected to it, then the individual steps are not simulated. This can be a good way to speed up simulation. The *Averaged* mode assumes that the external controller provides a step rate demand. This step rate demand is determined from the voltage applied between the **ENA** and **REF** ports on the Stepper Motor Driver block, by multiplying this voltage by the value of the **Step rate sensitivity** parameter. The rotation direction is set by the **REF** port in the same way as for the *Stepping* mode.

*Averaged* mode needs to communicate the step rate demand and also output voltage amplitude information to the Stepper Motor block. To do this, the step rate demand is applied as an equivalent voltage across the **A+** and **A-** ports. Similarly the output voltage amplitude information is conveyed by applying a steady-state voltage across the **B+** and **B-** ports with value equal to the **Output voltage amplitude** parameter.

## Assumptions and Limitations

- To use *Averaged* mode, the Stepper Motor Driver block must be directly connected to a Stepper Motor block also running in *Averaged* mode.
- When changing from *Stepping* to *Averaged* mode and back, you must modify your upstream blocks that provide the input voltages to the Stepper Motor Driver block. One way to achieve this easily is to use Simulink variant subsystems.

## Ports

### Conserving

#### **A+ — A-phase positive terminal**

electrical

Electrical conserving port associated with the *A*-phase positive terminal.

#### **A- — A-phase negative terminal**

electrical

Electrical conserving port associated with the *A*-phase negative terminal.

#### **B+ — B-phase positive terminal**

electrical

Electrical conserving port associated with the *B*-phase positive terminal.

#### **B- — B-phase negative terminal**

electrical

Electrical conserving port associated with the *B*-phase negative terminal.

#### **ENA — Triggering input step voltage**

electrical

Electrical conserving port associated with the step trigger input.

**REF — Input floating reference voltage**

electrical

Electrical conserving port associated with the floating reference voltage.

**REV — Input voltage that controls motor direction**

electrical

Electrical conserving port associated with the motor direction input.

**VM — Supply voltage**

electrical

Electrical conserving port associated with the supply voltage.

**Dependencies**

To enable this port, set **Power supply** to External.

**GND — Supply ground reference**

electrical

Electrical conserving port associated with the supply ground reference.

In external power supply mode, the **A-** and **B-** ports are internally connected to **GND**. You must provide positive supply voltages and the supply must always be connected (for example, you cannot put a series switch between supply and driver).

**Dependencies**

To enable this port, set **Power supply** to External.

**Parameters****Simulation mode — Simulation mode**

Stepping (default) | Averaged

Use Averaged only if the block is connected directly to a Stepper Motor block also running in Averaged mode.

**Step rate sensitivity — Step rate demand sensitivity**

10 Hz/V (default) | positive number

This parameter converts the voltage presented across the **ENA** and **REF** ports into a step rate demand.

**Dependencies**

To enable this parameter, set **Simulation mode** to Averaged.

**Enable threshold voltage — Step voltage threshold**

2.5 V (default) | positive number

When the voltage at the **ENA** port rises above this threshold, the Stepper Motor Driver block initiates a step.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Reverse threshold voltage — Reversal voltage threshold**

2.5 V (default) | positive number

When the voltage at the **REV** port rises above this threshold, pulse *B* leads pulse *A* by 90 degrees, and the motor direction is reversed.

**Output voltage amplitude — Pulse output amplitude**

10 V (default) | positive number

Amplitude of the output pulse trains.

**Dependencies**

To enable this parameter, either:

- Set **Simulation mode** to Averaged.
- Set **Simulation mode** to Stepping and **Power supply** to Internal.

**Supply voltage — Supply voltage**

10 V (default) | positive number

Voltage from external power supply.

**Dependencies**

To enable this parameter, set **Stepping mode** to Microstepping and **Power supply** to Internal.

**Stepping mode (averaged) — Step size for averaged simulation**

Full stepping (default) | Half stepping

Whether to enable Full stepping or Half stepping mode.

**Dependencies**

To enable this parameter, set **Simulation mode** to Averaged.

**Stepping mode — Step size for stepping simulation**

Full stepping (default) | Half stepping | Microstepping

Whether to enable Full stepping, Half stepping, or Microstepping mode.

**Dependencies**

To enable this parameter, set **Simulation mode** to Stepping.

**Number of microsteps per step — Number of microsteps per step**

4 (default) | positive number

Number of microsteps per step. This value must be a positive power of two.

**Dependencies**

To enable this parameter, set **Stepping mode** to Microstepping.

**Maximum continuous current per phase — Maximum continuous current per phase**

1.2 A (default)

Maximum continuous current per phase

**Dependencies**

To enable this parameter, set **Stepping mode** to Microstepping.

## **Version History**

**Introduced in R2008a**

## **Extended Capabilities**

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Controlled PWM Voltage | Stepper Motor

# Strain Gauge

Deformation sensor

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Strain Gauge block represents a sensor that generates a change in resistance as a function of strain using the following equation:

$$\frac{\Delta R}{R} = K\varepsilon$$

where:

- $\Delta R/R$  is the fractional change in resistance.
- $\varepsilon$  is the strain at port E.
- $K$  is the **Gauge factor** parameter value.

## Ports

### Input

#### E — Strain input

physical signal

Physical signal port associated with the strain input.

### Conserving

#### + — Positive port

electrical

Electrical conserving port associated with the strain gauge positive port.

#### - — Negative port

electrical

Electrical conserving port associated with the strain gauge negative port.

## Parameters

#### Gauge resistance — Unstressed gauge resistance

100 Ohm (default)

Unstressed gauge resistance.

**Gauge factor – Gauge factor**

2 (default)

The ratio  $K$  of the fractional change in resistance to the fractional change in length.

## **Version History**

**Introduced in R2008a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**



# Supercapacitor

Electrochemical double-layer capacitor

**Library:** Simscape / Electrical / Passive



## Description

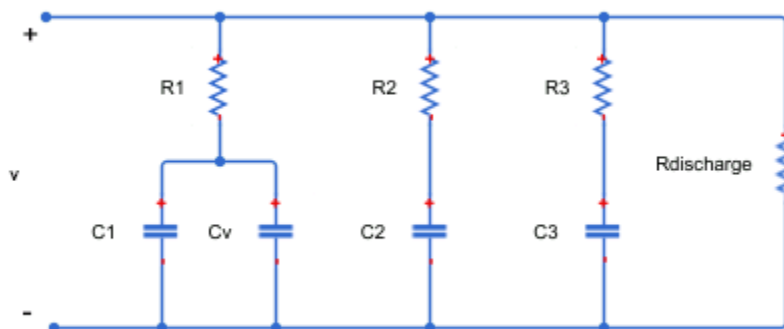
The Supercapacitor block represents an electrochemical double-layer capacitor (ELDC), which is commonly referred to as a supercapacitor or an ultracapacitor. The capacitance values for supercapacitors are orders of magnitude larger than the values for regular capacitors. Supercapacitors can provide bursts of energy because they can charge and discharge rapidly.

You can model any number of supercapacitor cells connected in series or in parallel using a single Supercapacitor block. To do so, set the relevant parameter, that is **Number of series cells** or **Number of parallel cells**, to a value larger than 1. Internally, the block simulates only the equations for a single supercapacitor cell, but it calculates:

- The output voltage according to the number of series-connected cells
- The current according to the number of parallel-connected cells

Calculating the output of a multiple-cell supercapacitor based on the output for a single cell is more efficient than simulating the equations for each cell individually.

The figure shows the equivalent circuit for a single cell in the Supercapacitor block. The circuit is a network of resistors and capacitors that is commonly used to model supercapacitor behavior.



Capacitors  $C_1$ ,  $C_2$ , and  $C_3$  have fixed capacitances. The capacitance of capacitor  $C_v$  depends on the voltage across it. Resistors  $R_1$ ,  $R_2$ , and  $R_3$  have fixed resistances. The voltage across each individual fixed capacitor in the Supercapacitor block is calculated as

$$V_{cn} = \frac{v}{N_{series}} - i_n R_n,$$

where:

- $v$  is the voltage across the block.
- $N_{series}$  is the number of cells in series.
- $n$  is the branch number.  $n = [1, 2, 3]$ .
- $i_n$  is the current through the  $n$ th branch.
- $R_n$  is the resistance in the  $n$ th branch.
- $V_{cn}$  is voltage across the capacitor in the  $n$ th branch.

The equation for the current through the first branch of the supercapacitor depends on the voltage across the capacitors in the branch. If the capacitors experience a positive voltage, that is

$$V_{c1} > 0,$$

then

$$i_1 = (C_1 + K_v V_{c1}) \frac{dV_{c1}}{dt},$$

else

$$i_1 = C_1 \frac{dV_{c1}}{dt},$$

where:

- $V_{c1}$  is voltage across the capacitors in the first branch.
- $C_1$  is the capacitance of the fixed capacitor in the first branch.
- $K_v$  is the voltage-dependent capacitance gain.
- $i_1$  is the current through the first branch.

For the remaining branches, the current is defined as

$$i_n = C_n \frac{dV_{cn}}{dt},$$

where:

- $n$  is the branch number.  $n = [2, 3]$ .
- $C_n$  is the capacitance of the  $n$ th branch.

The total current through the Supercapacitor block is

$$i = N_{parallel} \left( i_1 + i_2 + i_3 + \frac{v}{R_{discharge}} \right),$$

where:

- $N_{parallel}$  is the number of cells in parallel.
- $R_{discharge}$  is the self-discharge resistance of the supercapacitor.
- $i$  is the current through the supercapacitor.

## Ports

### Conserving

#### + – Positive electrical terminal

electrical

Electrical conserving port associated with the positive terminal.

#### - – Negative electrical terminal

electrical

Electrical conserving port associated with the negative terminal.

## Parameters

### Cell Characteristics

#### Fixed resistances, [R1 R2 R3] – Fixed resistance values for each branch

[0.2, 90.0, 1000.0] Ohm (default)

Specify the resistances for the fixed resistors in the individual branches of the supercapacitor as an array.

#### Fixed capacitances, [C1 C2 C3] – Fixed capacitance values for each branch

[2.5, 1.5, 4.0] F (default)

Specify the individual capacitances for the fixed capacitors in the supercapacitor as an array.

#### Voltage-dependent capacitor gain – Variable capacitance coefficient for the first branch

0.95 F/V (default)

Specify the variable capacitance coefficient,  $K_v$ , for the voltage-dependent capacitor in the first branch of the supercapacitor. For information on determining the variable capacitance coefficient, see [1] on page 1-2262.

#### Self-discharge resistance – Resistance to self-discharge

inf (default)

Specify the self-discharge resistance of the supercapacitor that is connected between the two terminals.

### Configuration

#### Number of series cells – Number of supercapacitor cells in series

1 (default)

Specify the number of cells in the supercapacitor that are in a series configuration.

#### Number of parallel cells – Number of supercapacitor cells in parallel

1 (default)

Specify the number of cells in the supercapacitor that are in a parallel configuration.

**Variables****Beginning Value — Initial target value**

0 (default)

Use the **Variables** tab to set the priority and initial target values for the block variables before simulation. For more information, see “Set Priority and Initial Target for Block Variables”.

**Version History**

Introduced in R2016b

**References**

- [1] Zubieta, L. and R. Bonert. “Characterization of Double-Layer Capacitors for Power Electronics Applications.” *IEEE Transactions on Industry Applications*, Vol. 36, No. 1, 2000, pp. 199-205.
- [2] Weddell, A. S., G. V. Merrett, T. J. Kazmierski, and B. M. Al-Hashimi. “Accurate Supercapacitor Modeling for Energy-Harvesting Wireless Sensor Nodes.” *IEEE Transactions on Circuits And Systems-II: Express Briefs*, Vol. 58, No. 12, 2011, pp. 911-915.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

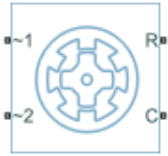
**See Also****Simscape Blocks**

Battery | Controlled Current Source (Three-Phase) | Current Source (Three-Phase) | Voltage Source (Three-Phase)

# Switched Reluctance Machine

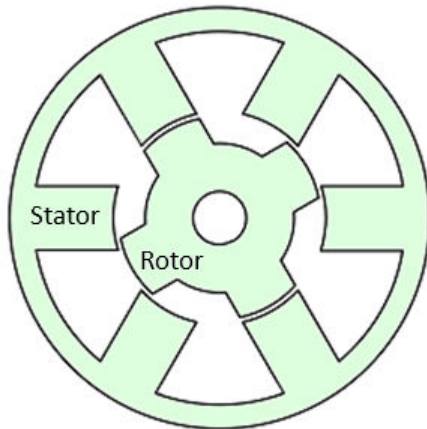
Three-phase switched reluctance machine

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Switched Reluctance Machine block represents a three-phase switched reluctance machine (SRM). The stator has three pole pairs, carrying the three motor windings, and the rotor has several nonmagnetic poles. The motor produces torque by energizing a stator pole pair, inducing a force on the closest rotor poles and pulling them toward alignment. The diagram shows the motor construction.



Choose this machine in your application to take advantage of these properties:

- Low cost
- Relatively safe failing currents
- Robustness to high temperature operation
- High torque-to-inertia ratio

Use this block to model an SRM using easily measurable or estimable parameters. To model an SRM using FEM data, see "Switched Reluctance Motor Parameterized with FEM Data".

## Equations

### Switched Reluctance Machine Block

The rotor stroke angle for a three-phase machine is

$$\theta_{st} = \frac{2\pi}{3N_r},$$

where:

- $\theta_{st}$  is the stoke angle.
- $N_r$  is the number of rotor poles.

The torque production capability,  $\beta$ , of one rotor pole is

$$\beta = \frac{2\pi}{N_r}.$$

The mathematical model for a switched reluctance machine (SRM) is highly nonlinear due to influence of the magnetic saturation on the flux linkage-to-angle,  $\lambda(\theta_{ph})$  curve. The phase voltage equation for an SRM is

$$v_{ph} = R_s i_{ph} + \frac{d\lambda_{ph}(i_{ph}, \theta_{ph})}{dt}$$

where:

- $v_{ph}$  is the voltage per phase.
- $R_s$  is the stator resistance per phase.
- $i_{ph}$  is the current per phase.
- $\lambda_{ph}$  is the flux linkage per phase.
- $\theta_{ph}$  is the angle per phase.

Rewriting the phase voltage equation in terms of partial derivatives yields this equation:

$$v_{ph} = R_s i_{ph} + \frac{\partial \lambda_{ph}}{\partial i_{ph}} \frac{di_{ph}}{dt} + \frac{\partial \lambda_{ph}}{\partial \theta_{ph}} \frac{d\theta_{ph}}{dt}.$$

Transient inductance is defined as

$$L_t(i_{ph}, \theta_{ph}) = \frac{\partial \lambda_{ph}(i_{ph}, \theta_{ph})}{\partial i_{ph}},$$

or more simply as

$$\frac{\partial \lambda_{ph}}{\partial i_{ph}}.$$

Back electromotive force is defined as

$$E_{ph} = \frac{\partial \lambda_{ph}}{\partial \theta_{ph}} \omega_r.$$

Substituting these terms into the rewritten voltage equation yields this voltage equation:

$$v_{ph} = R_s i_{ph} + L_t(i_{ph}, \theta_{ph}) \frac{di_{ph}}{dt} + E_{ph}.$$

Applying the co-energy formula to equations for torque,

$$T_{ph} = \frac{\partial W(\theta_{ph})}{\partial \theta_r},$$

and energy,

$$W(i_{ph}, \theta_{ph}) = \int_0^{i_{ph}} \lambda_{ph}(i_{ph}, \theta_{ph}) di_{ph}$$

yields an integral equation that defines the instantaneous torque per phase, that is,

$$T_{ph}(i_{ph}, \theta_{ph}) = \int_0^{i_{ph}} \frac{\partial \lambda_{ph}(i_{ph}, \theta_{ph})}{\partial \theta_{ph}} di_{ph}.$$

Integrating over the phases give this equation, which defines the total instantaneous torque for a three-phase SRM:

$$T = \sum_{j=1}^3 T_{ph}(j).$$

The equation for motion is

$$J \frac{d\omega}{dt} = T - T_L - B_m \omega$$

where:

- $J$  is the rotor inertia.
- $\omega$  is the mechanical rotational speed.
- $T$  is the rotor torque. For the Switched Reluctance Machine block, torque flows from the machine case (block conserving port **C**) to the machine rotor (block conserving port **R**).
- $T_L$  is the load torque.
- $J$  is the rotor inertia.
- $B_m$  is the rotor damping.

For high-fidelity modeling and control development, use empirical data and finite element calculation to determine the flux linkage curve in terms of current and angle, that is,

$$\lambda_{ph}(i_{ph}, \theta_{ph}).$$

For low-fidelity modeling, you can also approximate the curve using analytical techniques. One such technique [2] uses this exponential function:

$$\lambda_{ph}(i_{ph}, \theta_{ph}) = \lambda_{sat} \left( 1 - e^{-i_{ph} f(\theta_{ph})} \right),$$

where:

- $\lambda_{sat}$  is the saturated flux linkage.
- $f(\theta_r)$  is obtained by Fourier expansion.

For the Fourier expansion, use the first two even terms of this equation:

$$f(\theta_{ph}) = a + b\cos(N_r\theta_{ph})$$

where  $a > b$ ,

$$a = \frac{L_{min} + L_{max}}{2\lambda_{sat}},$$

and

$$b = \frac{L_{max} - L_{min}}{2\lambda_{sat}}.$$

### Switched Reluctance Motor Block

The flux linkage curve is approximated based on parametric and geometric data:

$$\lambda_{ph}(i_{ph}, \theta_{ph}) = \lambda_{sat} \left( 1 - e^{-L_0(\theta) i_{ph} / \lambda_{sat}} \right),$$

where  $L_0$  is the unsaturated inductance.

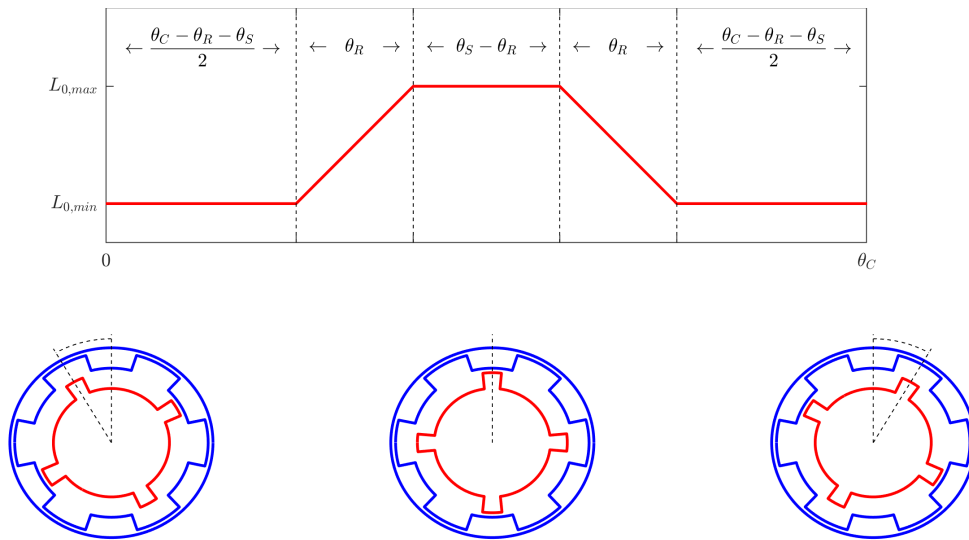
The effects of saturation are more prominent as the product of current and unsaturated inductance approach the saturated flux linkage value. Specify this value using the **Saturated flux linkage** parameter.

Differentiating the flux equation then gives the winding inductance:

$$L(\theta_{ph}) = L_0(\theta_{ph}) e^{(-L_0(\theta_{ph}) i_{ph} / \lambda_{sat})}$$

The unsaturated inductance varies between a minimum and maximum value. The minimum value occurs when a rotor pole is directly between two stator poles. The maximum occurs when the rotor pole is aligned with a stator pole. In between these two points, the block approximates the unsaturated inductance linearly as a function of rotor angle. This figure shows the unsaturated inductance as a rotor pole passes over a stator pole.





In the figure:

- $\theta_R$  corresponds to the angle subtended by the rotor pole. Set it using the **Angle subtended by each rotor pole** parameter.
- $\theta_S$  corresponds to the angle subtended by the stator pole. Set it using the **Angle subtended by each stator pole** parameter.
- $\theta_C$  corresponds to the angle subtended by this full cycle, determined by  $2\pi/2n$  where  $n$  is the number of stator pole pairs.

### Calculating Iron Losses

The Switched Reluctance Machine (Multi-Phase) block implements the iron losses as a reduction in the electrical torque.

If you set the **Stator parameterization** parameter to either Specify parametric data or Specify parametric and geometric data, specify the **Magnetizing resistance** parameter to model iron losses.

If you set the **Stator parameterization** parameter to Specify tabulated flux data, the block tabulates the losses as the power lost in each stator phase. The losses for each stator only depends on the stator current in that phase, the rotor angle (geometrically shifted to that phase), and the rotor speed. The iron loss power of the rotor is computed as a percentage of the total iron loss power, as specified in the **Percentage of total iron losses associated with the rotor** parameter.

If you tabulate the losses with current, speed, and angle, the iron loss torque reduces the electrical torque of a generic "A" phase:

$$\tau_{elecA} = \tau(i_A, \theta_A) - \tau_{loss}(i_A, w, \theta_A)$$

where  $\tau_{loss}(i_A, w, \theta_A) = \text{sign}(w) \frac{P_{loss}(\text{abs}(i_A), \text{abs}(w), \theta_A)}{\text{abs}(w) + w_{min}}$  is the torque loss computed as the mechanical power loss divided by the rotor speed.  $w_{min}$  is the minimum rotor speed and it is equal to 1 rad/s. This value prevents a division by zero when the rotor speed is equal to 0.

If you tabulate the losses in function of only current and speed, the iron loss torque reduces the electrical torque of a generic "A" phase:

$$\tau_{loss}(i_A, w) = \text{sign}(w) \frac{P_{loss}(\text{abs}(i_A), \text{abs}(w))}{\text{abs}(w) + w_{min}}.$$

The iron losses are applied to the mechanical side as a reduction in torque. If you create the iron loss table,  $P_{loss}$ , from FEM results, the FEM tool might express the power lost as electrical power instead of mechanical power.

The total electrical torque is equal to the sum of the electrical torque of each phase:

$$\tau_{elec} = \tau_{elecA} + \tau_{elecB} + \tau_{elecC}.$$

The total power dissipated is equal to the sum of the iron losses and the copper losses:

$$P_{dissipated} = (\tau_{lossA} + \tau_{lossB} + \tau_{lossC})w + R_a i_a^2 + R_b i_b^2 + R_c i_c^2.$$

### Iron losses with thermal ports

If you expose the thermal ports of the block, the power dissipated in each phase act as heat sources in the motor and in the rotor:

$$P_{dissA} = R_a i_a^2 + P_{ironA}$$

$$P_{dissB} = R_b i_b^2 + P_{ironB}$$

$$P_{dissC} = R_c i_c^2 + P_{ironC}$$

$$P_{dissR} = P_{ironR}$$

The iron loss in the rotor is equal to a percentage of the total iron losses. This loss remains constant and does not depend on other states:

$$P_{ironR} = \frac{\text{percent}_{rotor}}{100} (\tau_{lossA} + \tau_{lossB} + \tau_{lossC})w.$$

Then the remaining iron loss is distributed among the different stator phases:

$$P_{ironA} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossA} w$$

$$P_{ironB} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossB} w$$

$$P_{ironC} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossC} w$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain thermal ports.
- Show thermal port — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Numerical Smoothing

In practice, magnetic edge effects prevent the inductance from taking a trapezoidal shape as a rotor pole passes over a stator pole. To model these effects, and to avoid gradient discontinuities that hinder solver convergence, the block smooths the gradient  $\partial L_{\phi}/\partial\theta$  at inflection points. To change the angle over which this smoothing is applied, use the **Angle over which flux gradient changes are smoothed** parameter.

### Assumptions

The block assumes that a zero rotor angle corresponds to a rotor pole that is aligned perfectly with the *a*-phase.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### ~1 – Positive three-phase composite terminal

electrical

Electrical conserving three-phase port associated with the positive terminals of the stator windings.

#### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### ~2 – Negative three-phase composite terminal

electrical

Electrical conserving three-phase port associated with the negative terminals of the stator windings.

#### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### a1 – Positive *a*-phase terminal

electrical

Electrical conserving port associated with the positive terminal of stator winding *a*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**a2 — Negative a-phase terminal**

electrical

Electrical conserving port associated with the negative terminal of stator winding *a*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b1 — Positive b-phase terminal**

electrical

Electrical conserving port associated with the positive terminal of stator winding *b*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b2 — Negative b-phase terminal**

electrical

Electrical conserving port associated with the negative terminal of stator winding *b*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c1 — Positive c-phase terminal**

electrical

Electrical conserving port associated with the positive terminal of stator winding *c*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c2 — Negative c-phase terminal**

electrical

Electrical conserving port associated with the negative terminal of stator winding *c*.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**R — Rotor**

mechanical

Mechanical rotational conserving port associated with the rotor.

**C — Casing**

mechanical

Mechanical rotational conserving port associated with the stator or casing.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Number of rotor poles — Rotor poles**

4 (default) | positive integer

Number of rotor poles.

**Stator resistance per phase – Resistance**

3 Ohm (default) | positive scalar

Per-phase resistance of each of the stator windings.

**Stator parameterization – Parameterization method**

Specify parametric data (default) | Specify parametric and geometric data | Specify tabulated flux data

Method for parameterizing the stator.

**Dependencies**

Selecting Specify parametric data enables these parameters:

- **Magnetizing resistance**
- **Saturated flux linkage**
- **Aligned inductance**
- **Unaligned inductance**

Selecting Specify parametric and geometric data enables these parameters:

- **Magnetizing resistance**
- **Saturated flux linkage**
- **Aligned inductance**
- **Unaligned inductance**
- **Angle subtended by each stator pole**
- **Angle subtended by each rotor pole**
- **Angle over which flux gradient changes are smoothed**

Selecting Specify tabulated flux data enables these parameters:

- **Current vector,  $i$**
- **Angle vector,  $\theta$**
- **Flux linkage matrix,  $\Phi(i,\theta)$**

**Magnetizing resistance – Magnetic losses**

inf Ohm (default) | real scalar

The total magnetizing resistance for each of the phase windings. The default value `inf` indicates that there are no iron losses.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric data or Specify parametric and geometric data.

**Saturated flux linkage – Flux linkage**

0.43 Wb (default) | positive scalar

Saturated flux linkage per phase.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric data or Specify parametric and geometric data.

**Aligned inductance – Inductance**

0.0046 H (default) | positive scalar

The value of this parameter must be greater than the value of the **Unaligned inductance** parameter.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric data or Specify parametric and geometric data.

**Unaligned inductance – Inductance**

6.7e-4 H (default) | positive scalar

The value of this parameter must be less than the value of the **Aligned inductance** parameter.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric data or Specify parametric and geometric data.

**Angle subtended by each stator pole – Stator tooth angle**

45 deg (default) | positive scalar

Angle spanned by each stator tooth. This value must be greater than or equal to the value of **Angle subtended by each rotor pole**.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric and geometric data.

**Angle subtended by each rotor pole – Rotor tooth angle**

42 deg (default) | positive scalar

Angle spanned by each rotor tooth.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric and geometric data.

**Angle over which flux gradient changes are smoothed – Inflection point smoothing**

0.5 deg (default) | positive scalar

Angle over which sharp edges in trapezoidal inductance curve are smoothed. This value must be smaller than the value of **Angle subtended by each rotor pole**.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify parametric and geometric data.

**Current vector,  $i$  – Current**

[0, 50, 100] A (default) | vector

Current vector used to identify the flux linkage curve family.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify tabulated flux data.

**Angle vector,  $\theta$  – Angle**

[0, 45, 90] deg (default) | vector

Angle vector used to identify the flux linkage curve family.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify tabulated flux data.

**Flux linkage matrix,  $\Phi(i, \theta)$  – Flux**

[0, 0, 0; .37, .06, .37; .43, .1, .43] Wb (default) | matrix

Flux linkage matrix that defines the flux linkage curve family.

**Dependencies**

This parameter is exposed when you set **Stator parameterization** to Specify tabulated flux data.

**Iron Losses**

For more information on calculating iron losses, see “Calculating Iron Losses” on page 1-2267.

**Model – Iron losses modeling option**

None (default) | Tabulate losses from current and speed | Tabulate losses from current, speed, and angle

Whether to enable iron losses modeling.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify tabulated flux data.

**Magnetizing resistance – Magnetizing resistance for iron losses**

inf Ohm (default)

Magnetizing resistance. The value must be greater than zero. The default value is inf, which implies that there are no iron losses.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify parametric data or Specify parametric and geometric data.

**Rotor speed vector,  $w$  – Rotor speed**

[500, 1000, 1500] rpm (default)



Vector of rotor speed at which the iron loss is tabulated.

#### Dependencies

To enable this parameter, set **Model** to either Tabulate losses from current and speed or Tabulate losses from current, speed, and angle.

#### Iron loss table, $P_{loss}(i,w)$ — Iron loss table tabulated on current and speed

$\text{zeros}(3, 3)$  W (default)

Mechanical power lost in the shaft due to iron losses in a particular phase, tabulated as a function of the instantaneous phase current and the rotor speed.

#### Dependencies

To enable this parameter, set **Model** to Tabulate losses from current and speed.

#### Iron loss table, $P_{loss}(i,w,\theta)$ — Iron loss table tabulated on current, speed, and angle

$\text{zeros}(3, 3, 2)$  W (default)

Mechanical power lost in the shaft due to iron losses in a particular phase, tabulated as a function of the instantaneous phase current, the rotor speed, and the rotor angle.

#### Dependencies

To enable this parameter, set **Model** to Tabulate losses from current, speed, and angle.

### Mechanical

#### Rotor inertia — Inertia

$0.01 \text{ kg}\cdot\text{m}^2$  (default) | zero or positive scalar

Inertia of the rotor attached to mechanical translational port **R**.

#### Rotor Damping — Damping

$0 \text{ N}\cdot\text{m}/(\text{rad}/\text{s})$  (default) | positive scalar

Rotary damping.

### Thermal

To enable these parameters, set **Modeling option** to Show thermal port.

#### Resistance temperature coefficient — Temperature coefficient

$3.93\text{e-}3 \text{ 1}/\text{K}$  (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in "Thermal Model for Actuator Blocks". The default value,  $3.93\text{e-}3 \text{ 1}/\text{K}$ , is for copper.

#### Measurement temperature — Rated temperature

$25 \text{ degC}$  (default) | real scalar

The temperature for which motor parameters are quoted.

#### Winding thermal mass — Winding thermal mass

$100 \text{ J}/\text{K}$  (default) | positive scalar

The thermal mass value for the  $a$ -,  $b$ -, and  $c$ -windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass — Rotor thermal mass**

50 J/K (default) | positive scalar

The thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

**Percentage of total iron losses associated with the rotor — Percentage of total iron losses**

50 (default)

Percentage of total iron losses associated with the rotor.

## Version History

**Introduced in R2017b****Electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

[1] Boldea, I. and S. A. Nasar. *Electric Drives, Second Edition*. New York: CRC, 2005.

[2] Ilic'-Spong, M., R. Marino, S. Peresada, and D. Taylor. "Feedback linearizing control of switched reluctance motors." *IEEE Transactions on Automatic Control*. Vol. 32, No. 5, 1987, pp. 371-379.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

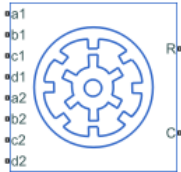
## See Also

BLDC | Synchronous Reluctance Machine | Synchronous Machine Field Circuit | PMSM | Synchronous Machine Measurement

# Switched Reluctance Machine (Multi-Phase)

Four, five, or six-phase switched reluctance machine (SRM)

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Switched Reluctance Machine (Multi-Phase) block represents a four-phase, five-phase, or six-phase switched reluctance machine (SRM).

The diagram shows the motor construction for the four-phase machine.



The diagram shows the motor construction for the five-phase machine.



## Equations

The rotor stroke angle for a multiphase machine is

$$\theta_{st} = \frac{2\pi}{N_s N_r}$$

where:

- $\theta_{st}$  is the stoke angle.

- $N_s$  is the number of phases.
- $N_r$  is the number of rotor poles.

The torque production capability,  $\beta$ , of one rotor pole is

$$\beta = \frac{2\pi}{N_r}.$$

The mathematical model for a switched reluctance machine (SRM) is highly nonlinear due to influence of the magnetic saturation on the flux linkage-to-angle curve,  $\lambda(\theta_{ph})$ . The phase voltage equation for an SRM is

$$v_{ph} = R_s i_{ph} + \frac{d\lambda_{ph}(i_{ph}, \theta_{ph})}{dt}$$

where:

- $v_{ph}$  is the voltage per phase.
- $R_s$  is the stator resistance per phase.
- $i_{ph}$  is the current per phase.
- $\lambda_{ph}$  is the flux linkage per phase.
- $\theta_{ph}$  is the angle per phase.

Rewriting the phase voltage equation in terms of partial derivatives yields this equation:

$$v_{ph} = R_s i_{ph} + \frac{\partial \lambda_{ph}}{\partial i_{ph}} \frac{di_{ph}}{dt} + \frac{\partial \lambda_{ph}}{\partial \theta_{ph}} \frac{d\theta_{ph}}{dt}.$$

Transient inductance is defined as

$$L_t(i_{ph}, \theta_{ph}) = \frac{\partial \lambda_{ph}(i_{ph}, \theta_{ph})}{\partial i_{ph}},$$

or more simply as

$$\frac{\partial \lambda_{ph}}{\partial i_{ph}}.$$

Back electromotive force is defined as

$$E_{ph} = \frac{\partial \lambda_{ph}}{\partial \theta_{ph}} \omega_r.$$

Substituting these terms into the rewritten voltage equation yields this voltage equation:

$$v_{ph} = R_s i_{ph} + L_t(i_{ph}, \theta_{ph}) \frac{di_{ph}}{dt} + E_{ph}.$$

Applying the co-energy formula to equations for torque,

$$T_{ph} = \frac{\partial W(\theta_{ph})}{\partial \theta_r},$$

and energy,

$$W(i_{ph}, \theta_{ph}) = \int_0^{i_{ph}} \lambda_{ph}(i_{ph}, \theta_{ph}) di_{ph}$$

yields an integral equation that defines the instantaneous torque per phase, that is,

$$T_{ph}(i_{ph}, \theta_{ph}) = \int_0^{i_{ph}} \frac{\partial \lambda_{ph}(i_{ph}, \theta_{ph})}{\partial \theta_{ph}} di_{ph}.$$

Integrating over the phases gives this equation, which defines the total instantaneous torque as

$$T = \sum_{j=1}^{N_s} T_{ph}(j).$$

The equation for motion is

$$J \frac{d\omega}{dt} = T - T_L - B_m \omega$$

where:

- $J$  is the rotor inertia.
- $\omega$  is the mechanical rotational speed.
- $T$  is the rotor torque. For the Switched Reluctance Machine block, torque flows from the machine case (block conserving port **C**) to the machine rotor (block conserving port **R**).
- $T_L$  is the load torque.
- $J$  is the rotor inertia.
- $B_m$  is the rotor damping.

For high-fidelity modeling and control development, use empirical data and finite element calculation to determine the flux linkage curve in terms of current and angle, that is,

$$\lambda_{ph}(i_{ph}, \theta_{ph}).$$

For low-fidelity modeling, you can also approximate the curve using analytical techniques. One such technique [2] uses this exponential function:

$$\lambda_{ph}(i_{ph}, \theta_{ph}) = \lambda_{sat} \left( 1 - e^{-i_{ph} f(\theta_{ph})} \right),$$

where:

- $\lambda_{sat}$  is the saturated flux linkage.
- $f(\theta_r)$  is obtained by Fourier expansion.

For the Fourier expansion, use the first two even terms of this equation:

$$f(\theta_{ph}) = a + b \cos(N_r \theta_{ph})$$

where  $a > b$ ,

$$a = \frac{L_{\min} + L_{\max}}{2\lambda_{\text{sat}}},$$

and

$$b = \frac{L_{\max} - L_{\min}}{2\lambda_{\text{sat}}}.$$

### Calculating Iron Losses

The Switched Reluctance Machine (Multi-Phase) block implements the iron losses as a reduction in the electrical torque.

If you set the **Stator parameterization** parameter to **Specify parametric data**, specify the **Magnetizing resistance** parameter to model iron losses.

If you set the **Stator parameterization** parameter to **Specify tabulated flux data**, the block tabulates the losses as the power lost in each stator phase. The losses for each stator only depends on the stator current in that phase, the rotor angle (geometrically shifted to that phase), and the rotor speed. The iron loss power of the rotor is computed as a percentage of the total iron loss power, as specified in the **Percentage of total iron losses associated with the rotor** parameter.

If you tabulate the losses with current, speed, and angle, the iron loss torque reduces the electrical torque of a generic "A" phase:

$$\tau_{elecA} = \tau(i_A, \theta_A) - \tau_{loss}(i_A, w, \theta_A)$$

where  $\tau_{loss}(i_A, w, \theta_A) = \text{sign}(w) \frac{P_{loss}(\text{abs}(i_A), \text{abs}(w), \theta_A)}{\text{abs}(w) + w_{\min}}$  is the torque loss computed as the mechanical power loss divided by the rotor speed.  $w_{\min}$  is the minimum rotor speed and it is equal to 1 rad/s. This value prevents a division by zero when the rotor speed is equal to 0.

If you tabulate the losses in function of only current and speed, the iron loss torque reduces the electrical torque of a generic "A" phase:

$$\tau_{loss}(i_A, w) = \text{sign}(w) \frac{P_{loss}(\text{abs}(i_A), \text{abs}(w))}{\text{abs}(w) + w_{\min}}.$$

The iron losses are applied to the mechanical side as a reduction in torque. If you create the iron loss table,  $P_{loss}$ , from FEM results, the FEM tool might express the power lost as electrical power instead of mechanical power.

The total electrical torque is equal to the sum of the electrical torque of each phase:

$$\tau_{elec} = \tau_{elecA} + \tau_{elecB} + \tau_{elecC}.$$

The total power dissipated is equal to the sum of the iron losses and the copper losses:

$$P_{dissipated} = (\tau_{lossA} + \tau_{lossB} + \tau_{lossC})w + R_a i_a^2 + R_b i_b^2 + R_c i_c^2.$$

### Iron losses with thermal ports

If you expose the thermal ports of the block, the power dissipated in each phase act as heat sources in the motor and in the rotor:

$$P_{dissA} = R_a i_a^2 + P_{ironA}$$

$$P_{dissB} = R_b i_b^2 + P_{ironB}$$

$$P_{dissC} = R_c i_c^2 + P_{ironC}$$

$$P_{dissR} = P_{ironR}$$

The iron loss in the rotor is equal to a percentage of the total iron losses. This loss remains constant and does not depend on other states:

$$P_{ironR} = \frac{\text{percent}_{rotor}}{100} (\tau_{lossA} + \tau_{lossB} + \tau_{lossC}) W.$$

Then the remaining iron loss is distributed among the different stator phases:

$$P_{ironA} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossA} W$$

$$P_{ironB} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossB} W$$

$$P_{ironC} = \left(1 - \frac{\text{percent}_{rotor}}{100}\right) \tau_{lossC} W$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Assumptions

A zero rotor angle corresponds to a rotor pole that is aligned perfectly with the *a*-phase, that is, peak flux.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

Data Types: double

#### **C — Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

Data Types: double

#### **a1 — a-phase positive supply**

electrical

Electrical positive supply for phase-*a*.

Data Types: double

#### **b1 — b-phase positive supply**

electrical

Electrical positive supply for phase-*b*.

Data Types: double

#### **c1 — c-phase positive supply**

electrical

Electrical positive supply for phase-*c*.

Data Types: double

#### **d1 — d-phase positive supply**

electrical

Electrical positive supply for phase-*d*.

Data Types: double

#### **e1 — e-phase positive supply**

electrical

Electrical positive supply for phase-*e*.

This port is visible if you select Five-phase for the **Number of stator phases** parameter.

Data Types: double

#### **f1 — f-phase positive supply**

electrical

Electrical positive supply for phase-*f*.



This port is visible if you select Six-phase for the **Number of stator phases** parameter.

Data Types: double

**a2 — a-phase negative supply**

electrical

Electrical negative supply for phase-*a*.

Data Types: double

**b2 — b-phase negative supply**

electrical

Electrical negative supply for phase-*b*.

Data Types: double

**c2 — c-phase negative supply**

electrical

Electrical negative supply for phase-*c*.

Data Types: double

**d2 — d-phase negative supply**

electrical

Electrical negative supply for phase-*d*.

Data Types: double

**e2 — e-phase negative supply**

electrical

Electrical negative supply for phase-*e*.

This port is visible if you select Five-phase for the **Number of stator phases** parameter.

Data Types: double

**f2 — f-phase negative supply**

electrical

Electrical negative supply for phase-*f*.

This port is visible if you select Six-phase for the **Number of stator phases** parameter.

Data Types: double

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — *b*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — *c*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HD — *d*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *d*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HE — *e*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *e*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port and **Number of stator phases** to Five-phase.

**HF — *f*-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *f*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port and **Number of stator phases** to Six-phase.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Whether to enable thermal ports

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

### Main

### Number of stator phases — Four- or five-phase SRM

Four-phase (default) | Five-phase | Six-phase

Type of multiphase SRM in terms of the number of stator phases.

### Number of rotor poles — Rotor pole

8 (default) | positive integer

Number of pole pairs on the rotor.

### Stator resistance per phase — Resistance

3 0hm (default) | positive scalar

Per-phase resistance of each of the stator windings.

### Stator parameterization — Parameterization method

Specify saturated flux linkage (default) | Specify flux characteristic

Method for parameterizing the stator.

### Dependencies

Selecting Specify saturated flux linkage enables these parameters:

- **Saturated flux linkage**
- **Aligned inductance**
- **Unaligned inductance**

Selecting Specify flux characteristic enables these parameters:

- **Current vector,  $i$**
- **Angle vector,  $\theta$**
- **Flux linkage matrix,  $\Phi(i,\theta)$**

### Saturated flux linkage — Flux linkage

0.43 Wb (default) | positive scalar

Saturated flux linkage per phase.

### Dependencies

To enable this parameter, set **Stator parameterization** to Specify saturated flux linkage.

### Aligned inductance — Inductance

0.0046 H (default) | positive scalar

Inductance when the axis of the rotor pole is identical to the axis of the excited stator pole. The value of this parameter must be greater than the value of the **Unaligned inductance** parameter.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify saturated flux linkage.

**Unaligned inductance – Inductance**

6.7e-4 H (default) | positive scalar

Inductance when the axis between two rotor poles is identical to the axis of the excited stator pole. The value of this parameter must be less than the value of the **Aligned inductance** parameter.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify saturated flux linkage.

**Current vector, *i* – Current**

[0, 50, 100] A (default) | positive vector

Current vector used to identify the flux linkage curve family.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify flux characteristic.

**Angle vector, *theta* – Angle**

[0, 22.5, 45] deg (default) | positive vector

Angle vector used to identify the flux linkage curve family.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify flux characteristic.

**Flux linkage matrix, *Phi(i,theta)* – Flux**

[0, 0, 0; .37, .06, .37; .43, .1, .43] Wb (default) | scalar

Flux linkage matrix that defines the flux linkage curve family.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify flux characteristic.

**Iron Losses**

For more information on calculating iron losses, see “Calculating Iron Losses” on page 1-2280.

**Model – Iron losses modeling option**

None (default) | Tabulate losses from current and speed | Tabulate losses from current, speed, and angle

Whether to enable iron losses modeling.

**Dependencies**

To enable this parameter, set **Stator parameterization** to Specify tabulated flux data.

**Magnetizing resistance — Magnetizing resistance for iron losses**`inf Ohm (default)`

Magnetizing resistance. The value must be greater than zero. The default value is `inf`, which implies that there are no iron losses.

**Dependencies**

To enable this parameter, set **Stator parameterization** to `Specify parametric data`.

**Rotor speed vector,  $w$  — Rotor speed**`[500, 1000, 1500] rpm (default)`

Vector of rotor speed at which the iron loss is tabulated.

**Dependencies**

To enable this parameter, set **Model** to either `Tabulate losses from current and speed` or `Tabulate losses from current, speed, and angle`.

**Iron loss table,  $P_{loss}(i,w)$  — Iron loss table tabulated on current and speed**`zeros(3, 3) W (default)`

Mechanical power lost in the shaft due to iron losses in a particular phase, tabulated as a function of the instantaneous phase current and the rotor speed.

**Dependencies**

To enable this parameter, set **Model** to `Tabulate losses from current and speed`.

**Iron loss table,  $P_{loss}(i,w,\theta)$  — Iron loss table tabulated on current, speed, and angle**`zeros(3, 3, 2) W (default)`

Mechanical power lost in the shaft due to iron losses in a particular phase, tabulated as a function of the instantaneous phase current, the rotor speed, and the rotor angle.

**Dependencies**

To enable this parameter, set **Model** to `Tabulate losses from current, speed, and angle`.

**Mechanical****Rotor inertia — Inertia**`0.01 kg*m^2 (default) | positive scalar`

Inertia of the rotor attached to mechanical translational port **R**.

**Rotor Damping — Damping**`0 N*m/(rad/s) (default) | scalar`

Rotary damping.

**Thermal**

To enable these parameters, set **Modeling option** to `Show thermal port`.

**Measurement temperature — Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient — Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Thermal mass for each stator winding — Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass — Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

**Percentage of total iron losses associated with the rotor — Percentage of total iron losses**

50 (default)

Percentage of total iron losses associated with the rotor.

**Dependencies**

To enable this parameter, set **Model** to either `Tabulate losses from current and speed` or `Tabulate losses from current, speed, and angle`.

## Version History

**Introduced in R2018b**

## References

- [1] Boldea, I. and S. A. Nasar. *Electric Drives*. 2nd Ed. New York: CRC Press, 2005.
- [2] Ilić-Spong, M., R. Marino, S. Peresada, and D. Taylor. “Feedback linearizing control of switched reluctance motors.” *IEEE Transactions on Automatic Control*. Vol. 32, Number 5, 1987, pp. 371-379.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

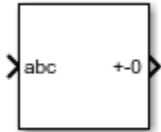
**See Also**

Synchronous Reluctance Machine | Switched Reluctance Machine

# Symmetrical-Components Transform

Implement  $abc$  to  $+0$  transform

**Library:** Simscape / Electrical / Control / Mathematical Transforms



## Description

The Symmetrical-Components Transform block implements a symmetrical transform of a set of phasors. The transform splits an unbalanced set of three phasors into three balanced sets of phasors.

In an unbalanced system with balanced impedances, use this block to decouple the system into three independent networks. In a balanced system, use this block to simplify the set of three-phasors to an equivalent one-line network. In this case, the positive set represents the one-line network.

Use the Power invariant property to choose between the Fortescue transform, and the alternative, power-invariant version.

## Equations

The symmetrical-components transform separates an unbalanced three-phase signal given in phasor quantities into three balanced sets of phasors:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} v_{a+} \\ v_{b+} \\ v_{c+} \end{bmatrix} + \begin{bmatrix} v_{a-} \\ v_{b-} \\ v_{c-} \end{bmatrix} + \begin{bmatrix} v_{a0} \\ v_{b0} \\ v_{c0} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  make up the original, unbalanced set of phasors.
- $v_{a+}$ ,  $v_{b+}$ , and  $v_{c+}$  make up the balanced, positive set of phasors.
- $v_{a-}$ ,  $v_{b-}$ , and  $v_{c-}$  make up the balanced, negative set of phasors.
- $v_{a0}$ ,  $v_{b0}$ , and  $v_{c0}$  make up the balanced, zero set of phasors.

The block calculates the symmetric  $a$ -phase using the transformation:

$$\begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix} = \frac{K}{3} \begin{bmatrix} 1 & a & a^2 \\ 1 & a^2 & a \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}.$$

where  $a$  is the complex rotation operator

$$a = e^{2\pi i/3},$$

and  $K$  is the constant that determines the type of transform:



$$\begin{cases} K = 1 & \text{Fortescue transform} \\ K = \sqrt{3} & \text{Power-invariant transform} \end{cases}$$

To select the power-invariant transform and simplify the power calculation in the  $+0$  domain, enable the **Power invariant** property.

Because the remaining two sets of symmetrical phasors are not often used in calculation, the block does not calculate them. However, they are given in terms of simple rotations of the first set:

$$\begin{bmatrix} V_{b+} \\ V_{b-} \\ V_{b0} \end{bmatrix} = \begin{bmatrix} a^2 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix},$$

and

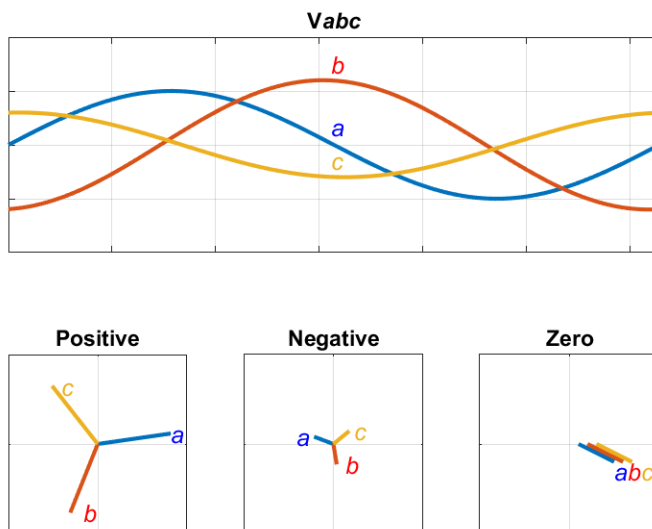
$$\begin{bmatrix} V_{c+} \\ V_{c-} \\ V_{c0} \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}.$$

### Operating Principle

The three sets of balanced phasors generated by the transform have the following properties:

- The positive set has the same order as the unbalanced set of phasors  $a$ - $b$ - $c$ .
- The negative set has the opposite order as the unbalanced set of phasors  $a$ - $c$ - $b$ .
- The zero set has no order because all three phasor angles are equal.

This diagram visualizes the separation performed by the transform.



In the diagram, the top axis shows an unbalanced three-phase signal with components  $a$ ,  $b$ , and  $c$ . The bottom set of axes separates the three-phase signal into symmetrical positive, negative, and zero phasors.

Observe that in each case, the  $a$ ,  $b$ , and  $c$  components are symmetrical and are separated by:

- +120 degrees for the positive set.
- -120 degrees for the negative set.
- 0 degrees for the zero set.

### Inverse Transform

The symmetrical-components transform is unique and invertible:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{K} \begin{bmatrix} 1 & 1 & 1 \\ a^2 & a & 1 \\ a & a^2 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}.$$

Use the Inverse Symmetrical-Components Transform block to perform this inverse transform.

## Ports

### Input

#### abc — $a$ , $b$ , and $c$ phasors

vector

Three-phase set of unbalanced phasors to be separated, given as a complex signal.

Data Types: single | double

### Output

#### + - 0 — Balanced $a$ phasor components

vector

Positive, negative and zero  $a$  phasors, output as a complex signal. Use the rotations given in the equations section to compute the  $b$  and  $c$  phasor sets.

Data Types: single | double

## Parameters

#### Power invariant — Transform type

off (default) | on

Power invariant toggle. Select this parameter to use the power-invariant alternative of the original Fortescue transform.

## Version History

Introduced in R2017b

## References

[1] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

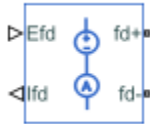
### Blocks

Inverse Symmetrical-Components Transform | Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform

## Synchronous Machine Field Circuit

Synchronous machine field circuit per-unit interface with voltage input and current output

**Library:** Simscape / Electrical / Electromechanical / Synchronous



### Description

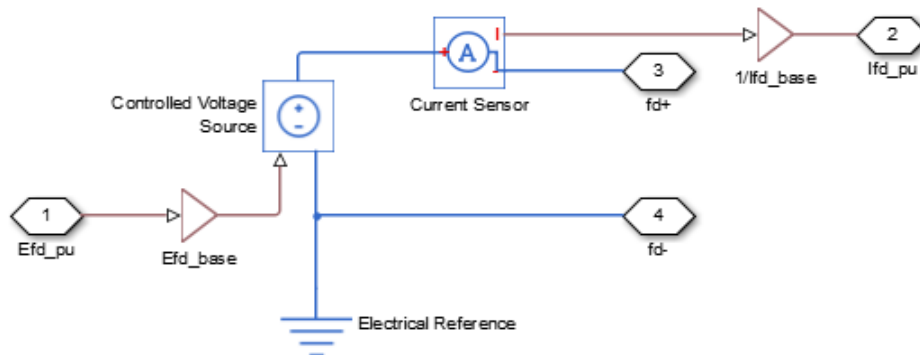
The Synchronous Machine Field Circuit block applies specified voltage to, and measures current through, the field circuit of the synchronous machine that it is connected to.

The SI model converts the SI values that you enter in the dialog box to per-unit values for simulation. For information on the relationship between SI and per-unit machine parameters, see “Per-Unit Conversion for Machine Parameters”. For information on per-unit parameterization, see “Per-Unit System of Units”.

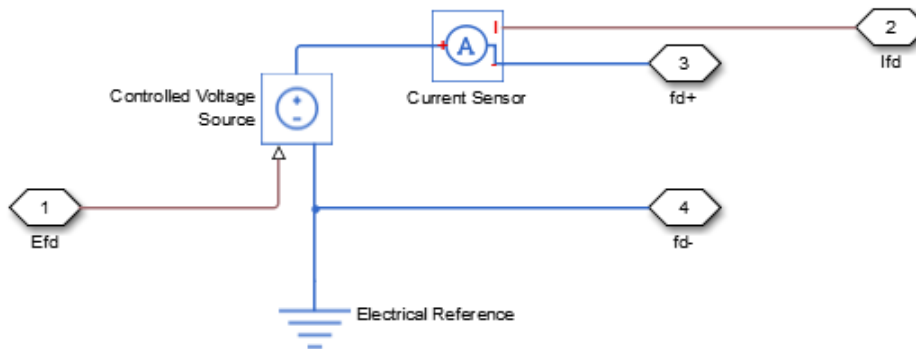
The block includes an electrical reference. The physical signal input **Efd\_pu** defines the voltage and the physical signal output **Ifd\_pu** provides the current, both in per-unit. The physical signal input **Efd** defines the voltage, in V, and the physical signal output **Ifd** provides the current, in A.

The per-unit bases are the nonreciprocal per-unit system, *Efd* and *Ifd*, rather than the reciprocal per-unit system, *efd* and *ifd*.

This figure shows the pu model of the Synchronous Machine Field Circuit block.



This figure shows the SI model of the Synchronous Machine Field Circuit block.



## Ports

### Input

#### **Efd – Field voltage**

physical

Physical input port associated with the field voltage.

#### **Dependencies**

This port is visible only when you set the **PS input and output unit** parameter to SI.

#### **Efd\_pu – Per unit field voltage**

physical

Physical input port associated with the field voltage, in per-unit..

#### **Dependencies**

This port is visible only when you set the **PS input and output unit** parameter to Per unit.

### Output

#### **Ifd – Field current**

physical

Physical input port associated with the field current.

#### **Dependencies**

This port is visible only when you set the **PS input and output unit** parameter to SI.

#### **Ifd\_pu – Per-unit field current**

physical

Physical input port associated with the field current, in per-unit..

#### **Dependencies**

This port is visible only when you set the **PS input and output unit** parameter to Per unit.

## Conserving

### **fd+ – Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

### **fd- – Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

## Parameters

The additional parameters for this block are visible only if you set the **PS input and output unit** parameter to `Per unit`.

### Main

#### **PS input and output unit – Unit-system configuration**

SI (default) | `Per unit`

Unit-system configuration for the block.

Selecting:

- SI exposes the SI ports.
- `Per unit` exposes the per-unit ports and parameters.

#### **Rated apparent power – Rated apparent power**

555e6 V\*A (default)

Rated apparent power of the connected machine.

#### **Rated electrical frequency – Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency at which rated apparent power of the connected machine is quoted.

#### **Specify field circuit input required to produce rated terminal voltage at no load by – Specify field circuit input required to produce rated terminal voltage at no load by**

`Field circuit current (default)` | `Field circuit voltage`

Choose between `Field circuit voltage` and `Field circuit current`.

#### **Field circuit current – Field circuit current**

1300 A (default)

This value is used to calculate the per-unit bases for the field circuit (nonreciprocal per-unit system).

### Dependencies

This parameter is visible only when you set the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter to `Field circuit current`.

**Field circuit voltage — Field circuit voltage**

92.95 V (default)

This value is used to calculate the per-unit bases for the field circuit (nonreciprocal per-unit system).

**Dependencies**

This parameter is visible only when you set the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter to Field circuit voltage.

**Machine Parameters****Specify parameterization by — Machine parameterization**

Fundamental Parameters (default) | Standard Parameters

Choose between Fundamental Parameters and Standard Parameters.

**Stator d-axis mutual inductance (unsaturated),  $L_{adu}$  — Stator d-axis mutual inductance (unsaturated)**

1.66 (default)

Unsaturated stator d-axis mutual inductance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Fundamental Parameters.

**Rotor field circuit resistance,  $R_{fd}$  — Rotor field circuit resistance**

0.0006 (default)

Rotor field-circuit resistance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Fundamental Parameters.

**Stator leakage reactance,  $X_l$  — Stator leakage reactance**

0.15 (default)

Stator leakage reactance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters.

**d-axis synchronous reactance,  $X_d$  — d-axis synchronous reactance**

1.81 (default)

The  $d$ -axis synchronous reactance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters.

**d-axis transient reactance,  $X_d'$  – d-axis transient reactance**

0.3 (default)

The  $d$ -axis transient reactance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters.

**d-axis subtransient reactance,  $X_d''$  – d-axis subtransient reactance**

0.23 (default)

$d$ -axis subtransient reactance.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters.

**Specify d-axis time constant – d-axis time constant specification**

Open-circuit value (default) | Short-circuit value

Choose between Open-circuit value and Short-circuit value.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters.

**d-axis transient open-circuit,  $T_d0'$  – d-axis transient open-circuit**

8 s (default)

The  $d$ -axis transient open-circuit time constant. This parameter is visible only when **Specify d-axis transient time constant** is set to Open-circuit value.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters and the **Specify d-axis transient time constant** to Open-circuit value.

**d-axis transient short-circuit,  $T_d'$  – d-axis transient short-circuit**

1.3260 s (default)

The  $d$ -axis transient short-circuit time constant. This parameter is visible only when **Specify d-axis transient time constant** is set to Short-circuit value.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters and the **Specify d-axis transient time constant** to Short-circuit value.

**d-axis subtransient open-circuit,  $T_d0''$  – d-axis subtransient open-circuit**

0.03 (default)

$d$ -axis subtransient open-circuit.



**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters and the **Specify d-axis transient time constant** to Open-circuit value.

**d-axis subtransient short-circuit,  $T_d''$  — d-axis subtransient short-circuit**  
0.0230 (default)

*d*-axis subtransient short-circuit.

**Dependencies**

This parameter is visible only when you set the **Specify parameterization by** parameter to Standard Parameters and the **Specify d-axis transient time constant** to Short-circuit value.

## Version History

**Introduced in R2014b**

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Synchronous Machine Measurement | Synchronous Machine Model 1.0 | Synchronous Machine Model 2.1 | Synchronous Machine Round Rotor | Synchronous Machine Salient Pole

# Synchronous Machine Measurement

Per-unit measurement from synchronous machine

**Library:** Simscape / Electrical / Electromechanical / Synchronous



## Description

The Synchronous Machine Measurement block outputs a per-unit measurement associated with a connected Synchronous Machine Round Rotor or Synchronous Machine Salient Pole block. The input of the Synchronous Machine Measurement block connects to the pu output port of the synchronous machine block.

You set the **Output** parameter to a per-unit measurement associated with the synchronous machine. Based on the value you select, the Synchronous Machine Measurement block:

- Directly outputs the value of an element in the input signal vector
- Calculates the per-unit measurement by using values of elements in the input signal vector in mathematical expressions

The Synchronous Machine Measurement block outputs a per-unit measurement from the synchronous machine according to the output value expressions in the table. For example, when you set **Output** to **Stator d-axis voltage**, the block directly outputs the value of the pu\_ed element in the input signal vector. However, when you set **Output** to **Reactive power**, the block calculates the value from the pu\_ed, pu\_eq, pu\_id, and pu\_iq elements.

Output Parameter Setting	Output Value Expression
Field voltage (field circuit base, Efd)	pu_fd_Efd
Field current (field circuit base, Ifd)	pu_fd_Ifd
Electrical torque	pu_torque
Rotor velocity	pu_velocity
Stator d-axis voltage	pu_ed
Stator q-axis voltage	pu_eq
Stator zero-sequence voltage	pu_e0
Stator d-axis current	pu_id
Stator q-axis current	pu_iq
Stator zero-sequence current	pu_i0
Apparent power	$\sqrt{pu\_Pt^2 + pu\_Qt^2}$
Real power	$pu\_Pt = (pu\_ed*pu\_id) + (pu\_eq*pu\_iq) + 2(pu\_e0*pu\_i0)$
Reactive power	$pu\_Qt = (pu\_eq*pu\_id) - (pu\_ed*pu\_iq)$

Output Parameter Setting	Output Value Expression
Terminal voltage	$\sqrt{(pu\_ed^2 + pu\_eq^2)}$
Terminal current	$\sqrt{(pu\_id^2 + pu\_iq^2)}$
Power factor angle (rad)	<code>power_factor_angle = atan2(pu_Qt, pu_Pt)</code>
Power factor	<code>cos(power_factor_angle)</code>
Load angle (rad)	<code>load_angle(rad) = atan2(pu_ed, pu_eq)</code>
Rotor electrical angle	<code>y = pi+2*atan(tan((u-pi)/2))</code>

## Ports

### Input

#### **pu** — Per-unit measurements from a connected synchronous machine

physical

Physical signal vector port associated with per-unit measurements from a connected synchronous machine. The vector elements are:

- pu\_fd\_Efd
- pu\_fd\_Ifd
- pu\_torque
- pu\_velocity
- pu\_ed
- pu\_eq
- pu\_e0
- pu\_id
- pu\_iq
- pu\_i0

### Output

#### **o** — Per-unit measurement output port

physical

Physical signal output port associated with the per-unit measurement.

## Parameters

#### **Output** — Per-unit measurement from synchronous machine

Field voltage (field circuit base, Efd) (default) | Field current (field circuit base, Ifd) | Electrical torque | Rotor velocity | Stator d-axis voltage | Stator q-axis voltage | Stator zero-sequence voltage | Stator d-axis current | Stator q-axis current | Stator zero-sequence current | Apparent power | Real power | Reactive power | Terminal voltage | Terminal current | Power factor angle | Power factor | Load angle | Rotor electrical angle

Per-unit measurement from synchronous machine. The default value is `Field voltage (field circuit base, Efd)`.

## **Version History**

**Introduced in R2013b**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

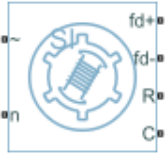
## **See Also**

[Synchronous Machine Field Circuit](#) | [Synchronous Machine Model 1.0](#) | [Synchronous Machine Model 2.1](#) | [Synchronous Machine Round Rotor](#) | [Synchronous Machine Salient Pole](#)

# Synchronous Machine Model 1.0

Synchronous machine with field circuit and no damper

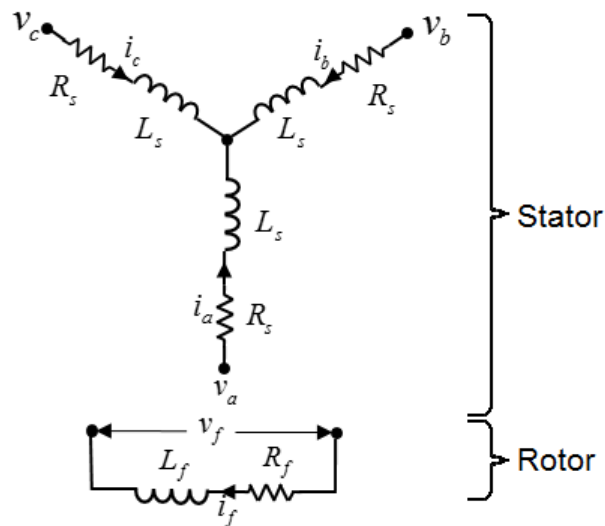
**Library:** Simscape / Electrical / Electromechanical / Synchronous



## Description

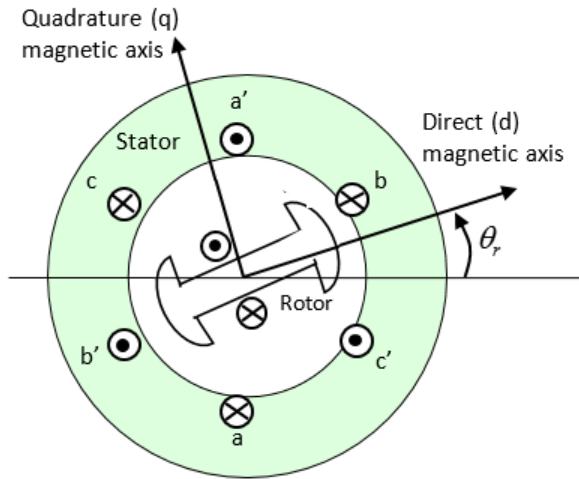
The Synchronous Machine Model 1.0 block uses a simplified parameterization model for synchronous machines. Use the block to model synchronous machines with a field winding and no dampers.

The figure shows the equivalent electrical circuit for the stator and rotor windings.



## Motor Construction

The diagram shows the motor construction with a single pole pair on the rotor. For the axes convention, when rotor mechanical angle  $\theta_r$  is zero, the  $a$ -phase and rotor magnet fluxes are aligned. The block supports a second rotor axis definition for which rotor mechanical angle is defined as the angle between the  $a$ -phase magnetic axis and the rotor  $q$ -axis.



### Equations

Voltages across the stator windings are defined by

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\frac{d\psi_a}{dt}$ ,  $\frac{d\psi_b}{dt}$ , and  $\frac{d\psi_c}{dt}$  are the rates of change of magnetic flux in each stator winding.

The voltage across the field winding is expressed as

$$v_f = R_f i_f + \frac{d\psi_f}{dt},$$

where:

- $v_f$  is the individual phase voltage across the field winding.
- $R_f$  is the equivalent resistance of the field winding.
- $i_f$  is the current flowing in the field winding.
- $\frac{d\psi_f}{dt}$  is the rate of change of magnetic flux in the field winding.

The excitation winding and the three star-wound stator windings contribute to the flux linking each winding. The total flux is defined by

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix} i_f,$$

where:

- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the total fluxes linking each stator winding.
- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ ,  $L_{bc}$ ,  $L_{ca}$ , and  $L_{cb}$  are the mutual inductances of the stator windings.
- $L_{amf}$ ,  $L_{bmf}$ , and  $L_{cmf}$  are the mutual inductances of the field winding.

The inductances in the stator windings are functions of rotor electrical angle and are defined by

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e),$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3)),$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_e + \pi/6)),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/6 - 2\pi/3)),$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_e + \pi/6 + 2\pi/3)),$$

where:

- $N$  is the number of rotor pole pairs.
- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the d-axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the q-axis.
- $L_s$  is the stator self-inductance per phase. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the fluctuation in self-inductance and mutual inductance with changing rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

The magnetization flux linking winding,  $a$ - $a'$  is a maximum when  $\theta_e = 0^\circ$  and zero when  $\theta_e = 90^\circ$ . Therefore:

$$L_{mf} = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix} = \begin{bmatrix} L_{mf} \cos \theta_e \\ L_{mf} \cos(\theta_e - 2\pi/3) \\ L_{mf} \cos(\theta_e + 2\pi/3) \end{bmatrix}$$

and

$$\Psi_f = L_f i_f + L_{mf}^T \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix},$$

where:

- $L_{mf}$  is the mutual field armature inductance.
- $\psi_f$  is the flux linking the field winding.
- $L_f$  is the field winding inductance.
- $[L_{mf}]^T$  is the transform of the  $L_{mf}$  vector, that is,

$$[L_{mf}]^T = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix}^T = [L_{amf} \ L_{bmf} \ L_{cmf}].$$

### Simplified Equations

Applying the Park transformation to the block electrical defining equations produces an expression for torque that is independent of rotor angle.

The Park transformation is defined by

$$P = 2/3 \begin{bmatrix} \cos\theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin\theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

Applying the Park transformation to the first two electrical defining equations produces equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} + L_{mf} \frac{di_f}{dt} - N\omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega(i_d L_d + i_f L_{mf}),$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt}$$

$$v_f = R_f i_f + L_f \frac{di_f}{dt} + \frac{3}{2} L_{mf} \frac{di_d}{dt},$$

$$T = \frac{3}{2} N (i_q (i_d L_d + i_f L_{mf}) - i_d i_q L_q),$$

and

$$J \frac{d\omega}{dt} = T = T_L - B_m \omega.$$

where:

- $v_d$ ,  $v_q$ , and  $v_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence voltages. These voltages are defined by



$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

- $L_d$  is the stator  $d$ -axis inductance.  $L_d = L_s + M_s + 3/2 L_m$ .
- $\omega$  is the mechanical rotational speed.
- $L_q$  is the stator  $q$ -axis inductance.  $L_q = L_s + M_s - 3/2 L_m$ .
- $L_0$  is the stator zero-sequence inductance.  $L_0 = L_s - 2M_s$ .
- $T$  is the rotor torque. For the Synchronous Machine Model 1.0 block, torque flows from the machine case (block conserving port **C**) to the machine rotor (block conserving port **R**).
- $J$  is the rotor inertia.
- $T_L$  is the load torque.
- $B_m$  is the rotor damping.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions

Flux distribution is sinusoidal.

## Ports

### Conserving

#### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

#### **~ — Three-phase composite**

electrical

Expandable three-phase port associated with the stator windings.

### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

### Dependencies

To enable this port, set **Zero sequence** to Include.

**fd+ – Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

**fd- – Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

**HA – Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB – Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC – Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR – Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Modeling fidelity — Modeling fidelity**Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$  (default) | Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$ 

Select the modeling fidelity:

- Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$  —  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$ , and  $PM$  values are constant and defined by their respective parameters.
- Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$  —  $L_d$ ,  $L_q$ ,  $L_{mf}$ ,  $L_f$ , and  $PM$  values are computed online from DQ and field currents look-up tables as follows:

$$L_d = f_1(i_d, i_q, i_f)$$

$$L_q = f_2(i_d, i_q, i_f)$$

$$L_{mf} = f_3(i_d, i_q, i_f)$$

$$L_f = f_4(i_f)$$

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_\theta$ .

**Number of pole pairs — Rotor pole pairs**

4 (default) | integer

Number of permanent magnet pole pairs on the rotor.

**Stator parameterization — Parameterization method**Specify  $L_d$ ,  $L_q$  and  $L_\theta$  (default) | Specify  $L_s$ ,  $L_m$ , and  $M_s$ 

Stator parameterization method.

**Dependencies**

The **Stator parameterization** setting affects the visibility of other parameters.

**Stator d-axis inductance,  $L_d$  — Inductance**

0.00015 H (default)

Direct-axis inductance of the machine stator.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_\theta$  and the **Modeling fidelity** parameter to Constant  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$ .

**Stator q-axis inductance,  $L_q$  — Inductance**

0.00021 H (default)

Quadrature-axis inductance of the machine stator.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Constant Ld, Lq, Lmf and Lf.

**Direct-axis current vector, iD – Direct-axis current vector**

[-200, 0, 200] A (default)

Direct-axis current vector, iD.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf and Lf.

**Quadrature-axis current vector, iQ – Quadrature-axis current vector**

[-200, 0, 200] A (default)

Quadrature-axis current vector, iQ.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf and Lf.

**Excitation current vector, iF – Excitation current vector**

[-20, 0, 20] A (default)

Excitation current vector, iF.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf and Lf.

**Ld matrix, Ld(id,iq,if) – Ld matrix**

0.00015 \* ones(3, 3, 3) H (default)

Ld matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf and Lf.

**Lq matrix, Lq(id,iq,if) – Lq matrix**

0.00021 \* ones(3, 3, 3) H (default)

Lq matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld, Lq, Lmf and Lf.

**Mutual field armature inductance,  $L_{mf}(i_d, i_q, i_f)$  – Mutual field armature inductance**

0.007 \* ones(3, 3, 3) H (default)

Mutual field armature inductance.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$ .

**Lf vector,  $L_f(i_f)$  – Lf vector**

0.5 \* ones(1, 3) H (default)

Lf vector.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$ , the **Modeling fidelity** parameter to Tabulated  $L_d$ ,  $L_q$ ,  $L_{mf}$  and  $L_f$ .

**Stator zero-sequence inductance,  $L_0$  – Inductance**

0.000012 H (default)

Zero-axis inductance for the machine stator.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_d$ ,  $L_q$  and  $L_0$  and **Zero sequence** is set to Include.

**Stator self-inductance per phase,  $L_s$  – Inductance**

0.00016 H (default)

Average self-inductance of the three stator windings. This parameter must be:

- Greater than 0.
- Greater than the magnitude of the **Stator inductance fluctuation,  $L_m$** .
- Greater than the magnitude of the **Stator mutual inductance,  $M_s$** .

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator inductance fluctuation,  $L_m$  – Inductance**

-0.00002 H (default)

Fluctuation in self-inductance and mutual inductance with the rotor angle.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

**Stator mutual inductance,  $M_s$  – Inductance**

0.00002 H (default)

Average mutual inductance between the stator windings.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify Ls, Lm, and Ms.

**Field winding inductance, Lf – Inductance**

0.5 H (default)

Inductance of the field winding.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Constant Ld, Lq, Lmf and Lf or **Stator parameterization** parameter to Specify Ls, Lm, and Ms

**Mutual field armature inductance, Lmf – Inductance**

0.007 H (default)

Armature-field mutual inductance.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Constant Ld, Lq, Lmf and Lf or **Stator parameterization** parameter to Specify Ls, Lm, and Ms

**Stator resistance per phase, Rs – Resistance**

0.08 Ohm (default)

Resistance of each of the stator windings.

**Field winding resistance, Rf – Resistance**

3 Ohm (default)

Resistance of the field winding.

**Zero sequence – Zero-sequence model**

Include (default) | Exclude

Zero-sequence model:

- **Include** — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Exclude** — Prioritize simulation speed for desktop simulation or real-time deployment.

**Dependencies**

If this parameter is set to:

- **Include** and **Stator parameterization** is set to **Specify Ld, Lq, and L0** — The **Stator zero-sequence inductance, L0** parameter is visible.
- **Exclude** — The **Stator zero-sequence inductance, L0** parameter is not visible.

## Mechanical

### **Rotor inertia – Inertia**

0.01 kg\*m<sup>2</sup> (default)

Inertia of the rotor.

### **Rotor Damping – Damping**

0 N\*m/(rad/s) (default)

Damping of the rotor.

### **Rotor angle definition – Angle**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. If you select the default value, the rotor and a-phase fluxes are aligned for a zero-rotor angle. Otherwise, an a-phase current generates the maximum torque value for a zero-rotor angle.

## Thermal

To enable these parameters, set **Modeling option** to Show thermal port.

### **Measurement temperature – Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

### **Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

### **Thermal mass for each stator winding – Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

### **Rotor thermal mass – Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

## Version History

### **Introduced in R2018a**

### **Electrical connection ports update**

*Behavior changed in R2021b*



From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. IEEE Press, Power Systems Engineering, 1995.
- [3] Retif, J. M., X. Lin-Shi, A. M. Llor, and F. Morand “New hybrid direct-torque control for a winding rotor synchronous machine.” *2004 IEEE 35th Annual Power Electronics Specialists Conference*. Vol. 2 (2004): 1438-1442.
- [4] IEEE Power Engineering Society. IEEE Std 1110-2002. *IEEE Guide for Synchronous Generator Modeling Practices and Applications in Power System Stability Analyses*. Piscataway, NJ: IEEE, 2002.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

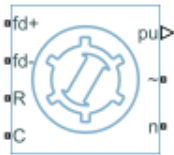
## See Also

BLDC | Switched Reluctance Machine | Synchronous Reluctance Machine | Synchronous Machine Field Circuit | Synchronous Machine Measurement | Synchronous Machine Model 2.1 | Synchronous Machine Round Rotor | Synchronous Machine Salient Pole

## Synchronous Machine Model 2.1

Synchronous machine with simplified transformation, simplified representation, and fundamental or standard parameterization

**Library:** Simscape / Electrical / Electromechanical / Synchronous



### Description

The Synchronous Machine Model 2.1 block models a synchronous machine with one field winding and one damper on the  $d$ -axis and one damper on the  $q$ -axis. You use fundamental or standard parameters to define the characteristics of the machine. This block contains a  $dq$  Park transformation, so use it only for balanced operation.

### Synchronous Machine Initialization Using Load-Flow Target Values

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize the machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”. For an example that shows how initialize an synchronous machine using data from a load flow analysis, see “Synchronous Machine Initialization with Loadflow”.

### Equations

The synchronous machine equations are expressed with respect to a rotating reference frame, defined by

$$\theta_e(t) = N\theta_r(t),$$

where:

- $\theta_e$  is the electrical angle.
- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor angle.

The Park transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. The Park transformation is defined by

$$P_s = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \end{bmatrix}.$$

The Park transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = e_d'' - R_a i_d + x_q'' i_q$$

and

$$e_q = e_q'' - x_d'' i_d - R_a i_q,$$

where:

- $e_d''$  and  $e_q''$  are the  $d$ -axis and  $q$ -axis voltages behind subtransient reactances.
- $R_a$  is the stator resistance.
- $i_d$  and  $i_q$  are the  $d$ -axis and  $q$ -axis stator currents, defined by

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing from port  $\sim$  to port  $n$ .

- $x_d''$  and  $x_q''$  are the  $d$ -axis and  $q$ -axis subtransient reactances.
- $e_d$  and  $e_q$  are the  $d$ -axis and  $q$ -axis stator voltages, defined by

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages measured from port  $\sim$  to neutral port  $n$ .

The rotor voltage equation is defined by

$$e_{fd} = R_{fd} \cdot i_{fd},$$

where:

- $R_{fd}$  is the resistance of rotor field circuit.
- $i_{fd}$  is the per-unit field current using the synchronous machine model reciprocal per-unit system.
- $e_{fd}$  is the per-unit field voltage using the synchronous machine model reciprocal per-unit system.

The voltage-behind-transient-reactance equations are defined by

$$\frac{de_d''}{dt} = \frac{(x_q - x_q'')i_q - e_d''}{T_{q0}'},$$

$$\frac{de_q'}{dt} = \frac{E_{fd} - (x_d - x_d')i_d - e_q'}{T_{d0}'},$$

and

$$\frac{de_q''}{dt} = \frac{e_q' - (x_d - x_d'')i_d - e_q''}{T_{d0}''},$$

where:

- $x_d$  and  $x_q$  are the  $d$ -axis and  $q$ -axis synchronous reactances.
- $T''_{d0}$  and  $T''_{q0}$  are the  $d$ -axis and  $q$ -axis subtransient open-circuit time constants.
- $E_{fd}$  is the per-unit field voltage using the exciter model nonreciprocal per-unit system.
- $x'_d$  is the  $d$ -axis transient reactance.
- $e'_q$  is the  $q$ -axis voltage behind transient reactance.
- $T'_{d0}$  is the  $d$ -axis transient open-circuit time constant.

The rotor torque is defined by

$$T_e = e_d'' i_d + e_q'' i_q - (x_d'' - x_q'') i_d i_q.$$

These defining equations do not describe the parameters you can set in the dialog box. To see their relationship with the equation coefficients, see the book of P. Kundur about understanding, modeling, analyzing, and mitigating power system stability and control problems [1].

### Display Options

You can perform display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select an option:

- **Display Base Values** displays the machine per-unit base values in the MATLAB Command Window.
- **Display Associated Base Values** displays associated per-unit base values in the MATLAB Command Window.
- **Display Associated Initial Conditions** displays associated initial conditions in the MATLAB Command Window.

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and machine temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- `No thermal port` — The block does not contain thermal ports.
- `Show thermal port` — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

For this block, the **Initial Targets** settings are visible only if, in the **Initial Conditions** section, you set the **Initialization option** parameter to `Set targets for rotor angle and Park's transform variables`.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Output

#### **pu — Machine per-unit measurements**

physical

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- Field voltage (field circuit base,  $E_{fd}$ ),  $pu\_fd\_E_{fd}$
- Field current (field circuit base,  $I_{fd}$ ),  $pu\_fd\_I_{fd}$
- Electrical torque,  $pu\_torque$
- Rotor velocity,  $pu\_velocity$
- Stator d-axis voltage,  $pu\_ed$
- Stator q-axis voltage,  $pu\_eq$
- Stator zero-sequence voltage,  $pu\_e0$  — This port is provided to maintain a compatible interface with other machine models. Its value is always zero.
- Stator d-axis current,  $pu\_id$
- Stator q-axis current,  $pu\_iq$
- Stator zero-sequence current,  $pu\_i0$  — This port is provided to maintain a compatible interface with other machine models. Its value is always zero.
- Rotor electrical angle,  $electrical\_angle\_out$

To connect to this port, use the Synchronous Machine Measurement block.

### Conserving

#### **fd+ — Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

#### **fd- — Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

#### **R — Machine rotor**

mechanical

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical

Mechanical rotational conserving port associated with the machine case.

**~ — Stator windings**

electrical

Expandable three-phase port associated with the stator windings.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**n — Neutral phase**

electrical

Electrical conserving port associated with the neutral point of the wye winding configuration. This port is provided to maintain a compatible interface for existing machine models. The voltage and current on this port are ignored.

**a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and machine temperature.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power — Rated apparent power**

555e6 V\*A (default)

Rated apparent power.

**Rated voltage — RMS rated line-line voltage**

24e3 V (default)

RMS rated line-line voltage.

**Rated electrical frequency — Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency at which rated apparent power is quoted.

**Number of pole pairs — Number of pole pairs**

1 (default)

Number of machine pole pairs.

**Specify parameterization by — Block parameterization**

Fundamental parameters (default) | Standard parameters

Block parameterization method. Options are:

- **Fundamental parameters** — Fundamental parameters are visible in the **Impedances** settings and the **Time Constants** settings are not visible.
- **Standard parameters** — Standard parameters are visible in the **Impedances** and the **Time Constants** settings are visible.

This parameter affects the visibility of the **Time Constant** settings and the parameters in the **Impedances** settings.

**Specify field circuit input required to produce rated terminal voltage at no load by — Specify field circuit input required to produce rated terminal voltage at no load by**

Field circuit current (default) | Field circuit voltage

Field circuit parameterization method. Options are:

- **Field circuit voltage** — Specify the field circuit in terms of voltage.
- **Field circuit current** — Specify the field circuit in terms of current. This method is the default field-circuit parameterization method.

This parameter affects the visibility of the **Field circuit voltage** and **Field circuit current** parameters.

**Field circuit voltage — Field circuit voltage**

92.95 V (default)

Voltage across field circuit which produces rated voltage at machine terminals.

**Dependencies**

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to **Field circuit voltage**.

**Field circuit current — Field circuit current**

1300 A (default)

Current in field circuit which produces rated voltage at machine terminals.

**Dependencies**

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to **Field circuit current**.

**Rotor angle definition — Reference point for the rotor angle measurement**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement.

When you select the default value, the rotor *d*-axis and stator *a*-phase magnetic axis are aligned when the rotor angle is zero.



The other value you can choose for this parameter is **Angle between the a-phase magnetic axis and the q-axis**. When you select this value, the rotor  $q$ -axis and stator  $a$ -phase magnetic axis are aligned when the rotor angle is zero.

### Impedances

#### **Stator d-axis mutual inductance (unsaturated), $L_{adu}$ — Stator d-axis mutual inductance (unsaturated)**

1.66 (default) | positive scalar

Unsaturated stator  $d$ -axis mutual inductance. This parameter must be greater than 0.

#### **Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

#### **Stator q-axis mutual inductance (unsaturated), $L_{aqu}$ — Stator q-axis mutual inductance (unsaturated)**

1.61 (default) | positive scalar

Unsaturated stator  $q$ -axis mutual inductance. This parameter must be greater than 0.

#### **Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

#### **Stator leakage inductance, $L_l$ — Stator leakage inductance**

0.15 (default) | positive scalar

Stator leakage inductance. This parameter must be greater than 0.

#### **Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

#### **Stator resistance, $R_a$ — Stator resistance**

0.003 (default) | positive scalar

Stator resistance. This parameter must be greater than 0.

#### **Rotor field circuit inductance, $L_{fd}$ — Rotor field circuit inductance**

0.165 (default) | positive scalar

Rotor field circuit inductance. This parameter must be greater than 0.

#### **Rotor field circuit resistance, $R_{fd}$ — Rotor field circuit resistance**

0.0006 (default) | positive scalar

Rotor field circuit resistance. This parameter must be greater than 0.

#### **Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

**Rotor d-axis damper winding 1 inductance, L1d — Rotor d-axis damper winding 1 inductance**

0.1713 (default) | positive scalar

Rotor *d*-axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor d-axis damper winding 1 resistance, R1d — Rotor d-axis damper winding 1 resistance**

0.0284 (default) | positive scalar

Rotor *d*-axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor q-axis damper winding 1 inductance, L1q — Rotor q-axis damper winding 1 inductance**

0.1066 (default) | positive scalar

Rotor *q*-axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor q-axis damper winding 1 resistance, R1q — Rotor q-axis damper winding 1 resistance**

0.0650 (default) | positive scalar

Rotor *q*-axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Stator leakage reactance, Xl — Stator leakage reactance**

0.15 (default) | positive scalar

Stator leakage reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**d-axis synchronous reactance, Xd — d-axis synchronous reactance**

1.81 (default) | positive scalar

*d*-axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**q-axis synchronous reactance,  $X_q$  – q-axis synchronous reactance**

1.76 (default) | positive scalar

*q*-axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**d-axis transient reactance,  $X_d'$  – d-axis transient reactance**

0.3 (default) | positive scalar

*d*-axis transient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**d-axis subtransient reactance,  $X_d''$  – d-axis subtransient reactance**

0.23 (default) | positive scalar

*d*-axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**q-axis subtransient reactance,  $X_q''$  – q-axis subtransient reactance**

0.25 (default) | positive scalar

*q*-axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**Time Constants****Specify d-axis time constant – Specify d-axis time constant**

Open circuit (default) | Short circuit

Select between Open circuit and Short circuit.

The setting for this parameter affects the visibility of the *d*-axis time constant parameters.

**d-axis transient open-circuit,  $Td\theta'$  – d-axis transient open-circuit**

8 s (default) | positive scalar

*d*-axis transient open-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **d-axis subtransient open-circuit, Td0''**.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Open circuit`.

**d-axis transient short-circuit, Td' — d-axis transient short-circuit**

1.3260 s (default) | positive scalar

*d*-axis transient short-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **d-axis subtransient short-circuit, Td''**.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Short circuit`.

**d-axis subtransient open-circuit, Td0'' — d-axis subtransient open-circuit**

0.03 s (default) | positive scalar

*d*-axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Open circuit`.

**d-axis subtransient short-circuit, Td'' — d-axis subtransient short-circuit**

0.0230 s (default) | positive scalar

*d*-axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Short circuit`.

**Specify q-axis time constant — Specify q-axis time constant**

`Open circuit` (default) | `Short circuit`

Select between `Open circuit` and `Short circuit`.

The setting for this parameter affects the visibility of the *q*-axis time constant parameters.

**q-axis subtransient open-circuit, Tq0'' — q-axis subtransient open-circuit**

0.07 s (default) | positive scalar

*q*-axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to `Open circuit`.

**q-axis subtransient short-circuit,  $Tq''$  – q-axis subtransient short-circuit**

0.0269 s (default) | positive scalar

q-axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Short circuit.

**Initial Conditions****Initialization option – Initialization option**

Set real power, reactive power, terminal voltage, and terminal phase (default) | Set targets for rotor angle and Park's transform variables | Set targets for load flow variables

Model for specifying values for certain parameters and variables at the start of simulation. To:

Set an operating point regardless of the connected network, select Set real power, reactive power, terminal voltage and terminal phase.

- Specify the priority and initial target values for block variables before simulation using the **Variables** settings, select Set targets for rotor angle and Park's transform variables. For more information, see “Set Priority and Initial Target for Block Variables”.
- Select a bus type and specify the related parameters for a load-flow analysis in the **Initial Conditions** settings, select Set targets for load flow variables.

**Dependencies**

If you set this parameter to:

- Set targets for rotor angle and Park's transform variables – The **Variables** settings become visible.
- Set real power, reactive power, terminal voltage, and terminal phase – Related parameters become visible.
- Set targets for load flow variables – Related parameters become visible.

**Source type – Voltage source model**

Swing bus (default) | Voltage source with series impedance | PV bus | PQ bus

Type of voltage source that the block models.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.

The visibility of **Terminal voltage magnitude**, **Terminal voltage angle**, **Active power generated**, **Reactive power generated**, **Minimum terminal voltage magnitude (pu)**, **Phase search range at terminals**, and **Phase search range at terminals** depend on the value that you choose for this parameter.

**Terminal voltage magnitude – Terminal voltage magnitude**

24e3 V (default)

Terminal voltage magnitude.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.

**Terminal voltage angle — Terminal voltage angle**

0 deg (default)

Terminal voltage angle.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus.

**Active power generated — Active power generated**

500e6 V\*A (default)

Active power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PV bus or PQ bus.

**Reactive power generated — Reactive power generated**

0 V\*A (default)

Reactive power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Minimum terminal voltage magnitude (pu) — Minimum terminal voltage magnitude**

0.95 (default)

Per-unit minimum steady-state voltage magnitude.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Phase search range at terminals — Phase search range at terminals**

[-inf, inf] deg (default)

Vector that defines the phase angle search range.

#### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to `Set targets for load flow variables` and the **Source type** parameter is set to `PV bus` or `PQ bus`.

#### Parasitic conductance to electrical reference – Parasitic conductance to electrical reference

1e-6 S (default)

Parasitic conductance to the electrical reference.

#### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to `Set targets for load flow variables`.

#### Thermal

To enable these parameters, set **Modeling option** to `Show thermal port`.

#### Measurement temperature – Rated temperature

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

#### Resistance temperature coefficient – Temperature coefficient

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

#### Thermal mass for each stator winding – Winding thermal mass

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

#### Rotor thermal mass – Rotor thermal mass

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

### Introduced in R2015a

### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`. Also, the simlog format is identical for `Composite three-phase ports` and `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York: McGraw Hill, 1993.
- [2] Lyshevski, S. E. *Electromechanical Systems, Electric Machines and Applied Mechatronics*. Boca Raton, FL: CRC Press, 1999.
- [3] Pal, M. K. *Lecture Notes on Power System Stability*. June, 2007.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

[Synchronous Machine Field Circuit](#) | [Synchronous Machine Measurement](#) | [Synchronous Machine Model 1.0](#) | [Synchronous Machine Round Rotor](#) | [Synchronous Machine Salient Pole](#)

## Topics

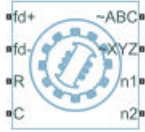
[“Perform a Load-Flow Analysis Using Simscape Electrical”](#)  
[“Expand and Collapse Three-Phase Ports on a Block”](#)



# Synchronous Machine (Six-Phase)

Six-phase synchronous machine

**Library:** Simscape / Electrical / Electromechanical / Synchronous

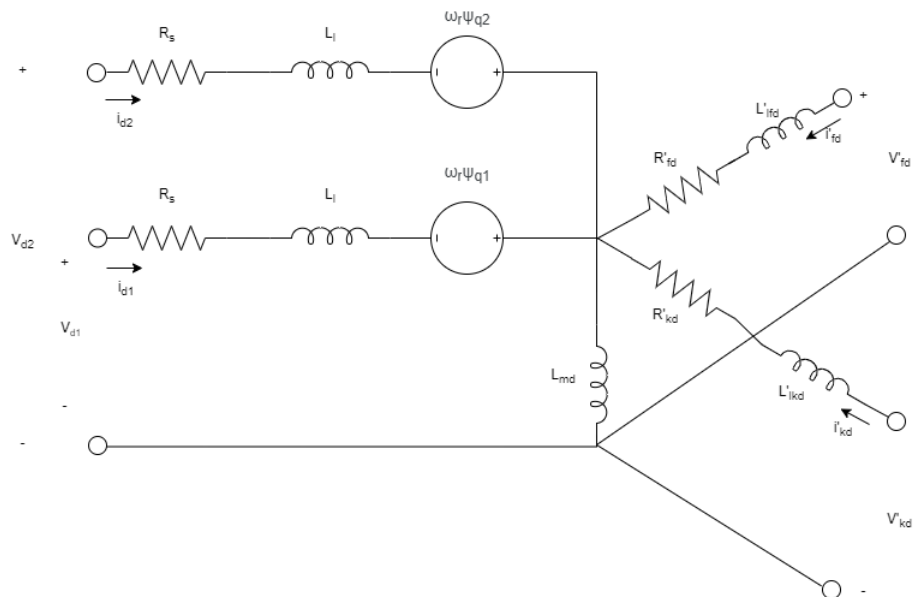


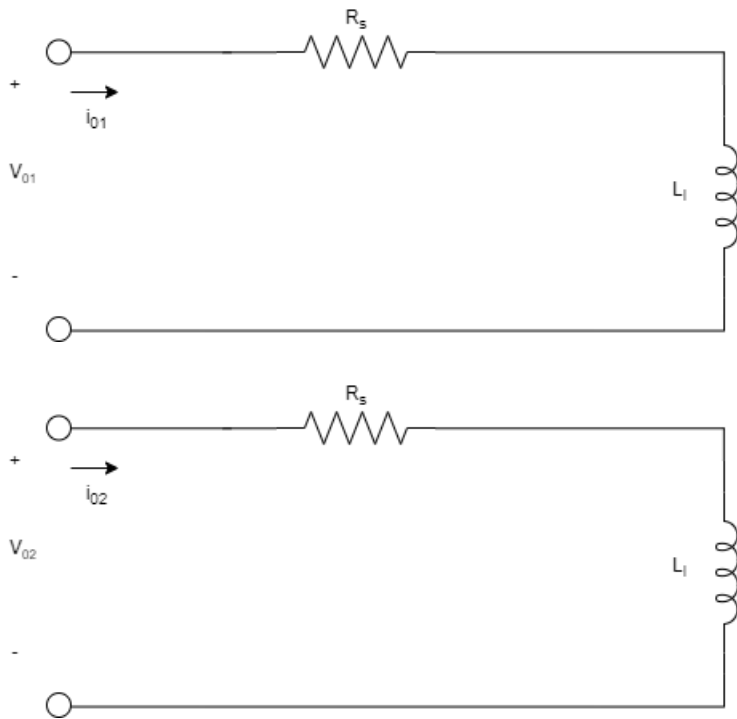
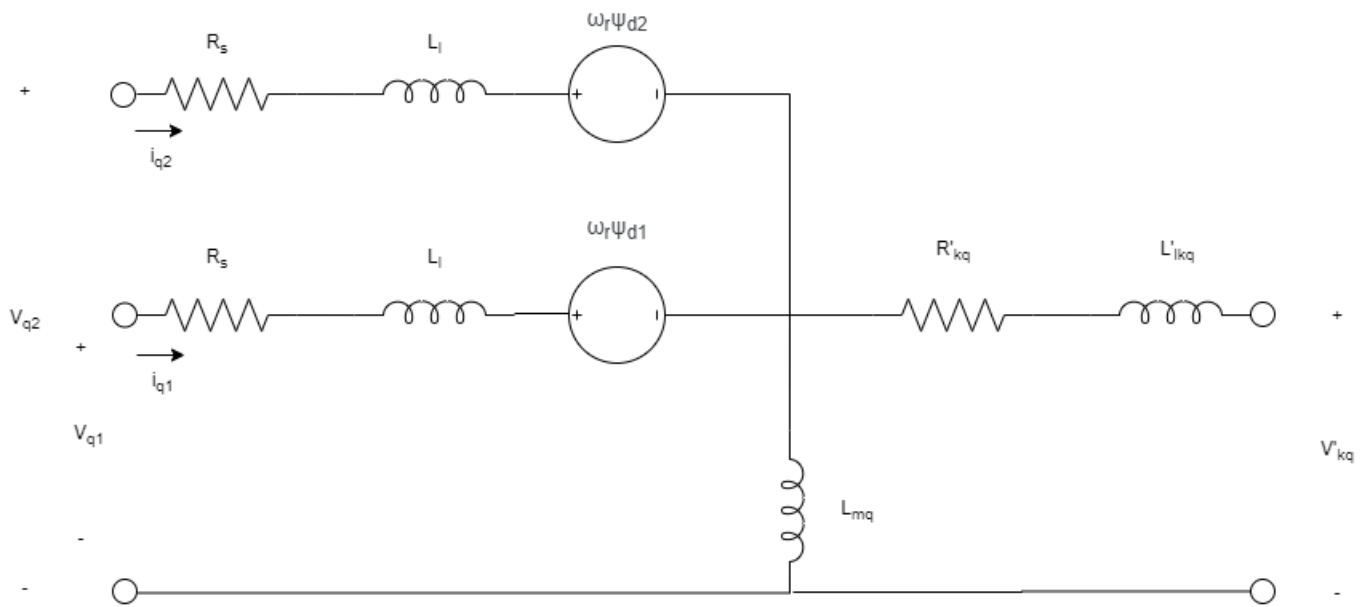
## Description

The Synchronous Machine (Six-Phase) block models a six-phase synchronous machine, also known as a dual-star machine.

A six-phase synchronous machine has two groups of stator windings: the ABC group and the XYZ group. These two groups have a 30 degree phase shift.

The equivalent circuits of the six-phase synchronous machine for the direct axis, the quadrature axis, and the two zero sequence are:





**Equations**

The synchronous machine equations are expressed with respect to a synchronously rotating reference frame, defined by:

$$\theta_e(t) = N\theta_r(t) + x\_rotor\_offset,$$

where:

- $\theta_e$  is the rotor electrical angle.
- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor mechanical angle.
- $x_{rotor\_offset}$  is  $\theta$  if you define the rotor electrical angle with respect to the  $d$ -axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the  $q$ -axis.

Two Park transformations map the synchronous machine equations to the rotating reference frame with respect to the electrical angle. The Park transformation for the first group of stator windings, the ABC group, is defined by:

$$P_{s1} = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos\left(\theta_e - \frac{2\pi}{3}\right) & \cos\left(\theta_e + \frac{2\pi}{3}\right) \\ -\sin\theta_e & -\sin\left(\theta_e - \frac{2\pi}{3}\right) & -\sin\left(\theta_e + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformation for the second group of stator windings, the XYZ group, is defined by:

$$P_{s2} = \frac{2}{3} \begin{bmatrix} \cos\left(\theta_e - \frac{\pi}{6}\right) & \cos\left(\theta_e - \frac{5\pi}{6}\right) & \cos\left(\theta_e + \frac{\pi}{2}\right) \\ -\sin\left(\theta_e - \frac{\pi}{6}\right) & -\sin\left(\theta_e - \frac{5\pi}{6}\right) & \sin\left(\theta_e + \frac{\pi}{2}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformations are used to define the per-unit synchronous machine equations.

The stator voltage equations for the ABC group are defined by:

$$v_{d1} = R_s i_{d1} - \psi_{q1} \omega_r + \frac{1}{\omega_{base}} \frac{d\psi_{d1}}{dt}$$

$$v_{q1} = R_s i_{q1} + \psi_{d1} \omega_r + \frac{1}{\omega_{base}} \frac{d\psi_{q1}}{dt}$$

$$v_{01} = R_s i_{01} + \frac{1}{\omega_{base}} \frac{d\psi_{01}}{dt}$$

where:

- $v_{d1}$ ,  $v_{q1}$ , and  $v_{01}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence ABC stator voltages, defined by:

$$\begin{bmatrix} v_{d1} \\ v_{q1} \\ v_{01} \end{bmatrix} = P_{s1} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the ABC stator voltages measured from port  $\sim\mathbf{ABC}$  to neutral port  $\mathbf{n1}$ .

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_{d1}$ ,  $\psi_{q1}$ , and  $\psi_{01}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages for the ABC group.

- $\omega_r$  is the per-unit rotor rotational speed.
- $R_s$  is the stator resistance.
- $i_{d1}$ ,  $i_{q1}$ , and  $i_{01}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence ABC stator currents, defined by:

$$\begin{bmatrix} i_{d1} \\ i_{q1} \\ i_{01} \end{bmatrix} = P_{s1} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the ABC stator currents flowing from port  $\sim\mathbf{ABC}$  to port  $\mathbf{n1}$ .

The stator voltage equations for the XYZ group are defined by:

$$v_{d2} = R_s i_{d2} - \psi_{q2} \omega_r + \frac{1}{\omega_{base}} \frac{d\psi_{d2}}{dt}$$

$$v_{q2} = R_s i_{q2} + \psi_{d2} \omega_r + \frac{1}{\omega_{base}} \frac{d\psi_{q2}}{dt}$$

$$v_{02} = R_s i_{02} + \frac{1}{\omega_{base}} \frac{d\psi_{02}}{dt}$$

where:

- $v_{d2}$ ,  $v_{q2}$ , and  $v_{02}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence XYZ stator voltages, defined by:

$$\begin{bmatrix} v_{d2} \\ v_{q2} \\ v_{02} \end{bmatrix} = P_{s2} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}.$$

$v_x$ ,  $v_y$ , and  $v_z$  are the XYZ stator voltages measured from port  $\sim\mathbf{XYZ}$  to neutral port  $\mathbf{n2}$ .

- $\psi_{d2}$ ,  $\psi_{q2}$ , and  $\psi_{02}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages for the XYZ group.
- $i_{d2}$ ,  $i_{q2}$ , and  $i_{02}$  are the  $d$ -axis,  $q$ -axis, and zero-sequence XYZ stator currents, defined by:

$$\begin{bmatrix} i_{d2} \\ i_{q2} \\ i_{02} \end{bmatrix} = P_{s2} \begin{bmatrix} i_X \\ i_Y \\ i_Z \end{bmatrix}.$$

$i_x$ ,  $i_y$ , and  $i_z$  are the XYZ stator currents flowing from port  $\sim\mathbf{XYZ}$  to port  $\mathbf{n2}$ .

The rotor voltage equations are defined by:

$$v'_{fd} = R'_{fd} i'_{fd} + \frac{1}{\omega_{base}} \frac{d\psi'_{fd}}{dt}$$

$$v'_{kd} = R'_{kd} i'_{kd} + \frac{1}{\omega_{base}} \frac{d\psi'_{kd}}{dt} = 0$$

$$v'_{kq} = R'_{kq} i'_{kq} + \frac{1}{\omega_{base}} \frac{d\psi'_{kq}}{dt} = 0$$

where:

- $v'_{fd}$  is the field winding voltage referred to the stator side.
- $v'_{kd}$  and  $v'_{kq}$  are the  $dq$ -axes damper winding voltages referred to the stator side. They are all equal to 0.
- $\psi'_{fd}$ ,  $\psi'_{kd}$ , and  $\psi'_{kq}$  are the magnetic fluxes linking the field circuit, the  $d$ -axis damper winding, and the  $q$ -axis damper winding.
- $R'_{fd}$ ,  $R'_{kd}$ , and  $R'_{kq}$  are the resistances of the rotor field circuit,  $d$ -axis damper winding, and  $q$ -axis damper winding.
- $i'_{fd}$ ,  $i'_{kd}$ , and  $i'_{kq}$  are the field and  $dq$ -axes damper winding currents referred to the stator side.

The stator flux linkage equations are defined by:

$$\psi_{d1} = L_l i_{d1} + L_{md}(i_{d1} + i_{d2} + i'_{fd} + i'_{kd})$$

$$\psi_{q1} = L_l i_{q1} + L_{mq}(i_{q1} + i_{q2} + i'_{kq})$$

$$\psi_{01} = L_l i_{01}$$

$$\psi_{d2} = L_l i_{d2} + L_{md}(i_{d1} + i_{d2} + i'_{fd} + i'_{kd})$$

$$\psi_{q2} = L_l i_{q2} + L_{mq}(i_{q1} + i_{q2} + i'_{kq})$$

$$\psi_{02} = L_l i_{02}$$

where:

- $L_l$  is the stator leakage inductance.
- $L_{md}$  and  $L_{mq}$  are the mutual inductances of the stator  $d$ -axis and  $q$ -axis.

The rotor flux linkage equations are defined by:

$$\psi'_{fd} = L'_{lfd} i'_{fd} + L_{md}(i_{d1} + i_{d2} + i'_{fd} + i'_{kd})$$

$$\psi'_{kd} = L'_{lkd} i'_{kd} + L_{md}(i_{d1} + i_{d2} + i'_{fd} + i'_{kd})$$

$$\psi'_{kq} = L'_{lkq} i'_{kq} + L_{mq}(i_{q1} + i_{q2} + i'_{kq})$$

where:

- $L'_{lfd}$  is the rotor field winding inductance.
- $L'_{lkd}$  is the rotor  $d$ -axis damper winding inductance.
- $L'_{lkq}$  is the rotor  $q$ -axis damper winding inductance.

The rotor torque is defined by:

$$T_e = \psi_{d1} i_{q1} - \psi_{q1} i_{d1} + \psi_{d2} i_{q2} - \psi_{q2} i_{d2}.$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block contains expanded electrical conserving ports associated with the stator windings, but does not contain thermal ports.
- **Show thermal port** — The block contains expanded electrical conserving ports associated with the stator windings and thermal conserving ports for each of the windings and for the rotor.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

For this block, the **Initial Targets** settings are visible only if, in the **Initial Conditions** section, you set the **Initialization option** parameter to **Set targets for rotor angle and Park's transform variables**.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **fd+** — Field winding positive terminal

electrical

Electrical conserving port associated with the field winding positive terminal.

#### **fd-** — Field winding negative terminal

electrical

Electrical conserving port associated with the field winding negative terminal.

#### **R** — Machine rotor

mechanical

Mechanical rotational conserving port associated with the machine rotor.

#### **C** — Machine case

mechanical

Mechanical rotational conserving port associated with the machine case.

#### **~ABC** — Stator ABC windings

electrical

Three-phase electrical port associated with the stator ABC windings.

#### **~XYZ** — Stator XYZ windings

electrical

Three-phase electrical port associated with the stator XYZ windings.

**n1 — ABC winding neutral point**

electrical

Electrical conserving port associated with the neutral point of the ABC winding configuration.

**Dependencies**

To enable this port, set **Zero sequence** to Include.

**n2 — XYZ winding neutral point**

electrical

Electrical conserving port associated with the neutral point of the XYZ winding configuration.

**Dependencies**

To enable this port, set **Zero sequence** to Include.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HX — x-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *x*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HY — y-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding y.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HZ — z-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding z.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Nominal power — Nominal power**

100e3 V\*A (default) | scalar

Nominal power.

**Nominal voltage (line-to-line, rms) — Nominal root mean square voltage**

240 V (default) | scalar

Nominal root mean square (RMS) line-to-line voltage.

**Nominal electrical frequency — Nominal electrical frequency**

60 Hz (default) | scalar

Nominal electrical frequency at which the nominal power is quoted.

**Number of pole pairs — Rotor pole pairs**

2 (default) | positive scalar

Number of machine pole pairs.

**Zero sequence — Zero sequence model**

Include (default) | Exclude



Zero-sequence model with two options:

- **Incl**ude — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Excl**ude — Prioritize simulation speed for desktop simulation or real-time deployment.

### **Rotor angle definition — Rotor angle reference point**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. If you select the default value, the rotor and *a*-phase fluxes are aligned for a zero-rotor angle. Otherwise, an *a*-phase current generates the maximum torque value for a zero-rotor angle.

### **Impedances**

#### **Stator resistance, $R_s$ — Stator resistance**

0.0288 (default) | positive scalar

Resistance of the stator. This parameter must be greater than 0.

#### **Stator leakage inductance, $L_l$ — Stator leakage inductance**

0.0969 (default) | positive scalar

Leakage inductance of the stator. This parameter must be greater than 0.

#### **Stator d-axis mutual inductance, $L_{md}$ — Stator d-axis mutual inductance**

1.9583 (default) | positive scalar

*d*-axis mutual inductance of the stator. This parameter must be greater than 0.

#### **Stator q-axis mutual inductance, $L_{mq}$ — Stator q-axis mutual inductance**

0.9045 (default) | positive scalar

*q*-axis mutual inductance of the stator. This parameter must be greater than 0.

#### **Rotor field winding resistance, $R_{fd}'$ — Rotor field winding resistance**

0.0028 (default) | positive scalar

Resistance of the field winding of the rotor. This parameter must be greater than 0.

#### **Rotor field winding inductance, $L_{fd}'$ — Rotor field winding inductance**

0.0813 (default) | positive scalar

Inductance of the field winding of the rotor. This parameter must be greater than 0.

#### **Rotor d-axis damper winding resistance, $R_{kd}'$ — Rotor d-axis damper winding resistance**

0.0041 (default) | positive scalar

Resistance of the damper winding of the rotor in the *d*-axis. This parameter must be greater than 0.

#### **Rotor d-axis damper winding inductance, $L_{kd}'$ — Rotor d-axis damper winding inductance**

0.0918 (default) | positive scalar

Inductance of the damper winding of the rotor in the  $d$ -axis. This parameter must be greater than 0.

**Rotor  $q$ -axis damper winding resistance,  $Rkq'$  — Rotor  $q$ -axis damper winding resistance**

0.0043 (default) | positive scalar

Resistance of the damper winding of the rotor in the  $q$ -axis. This parameter must be greater than 0.

**Rotor  $q$ -axis damper winding inductance,  $Llkq'$  — Rotor  $q$ -axis damper winding inductance**

0.1174 (default) | positive scalar

Inductance of the damper winding of the rotor in the  $q$ -axis. This parameter must be greater than 0.

**Initial Conditions**

**Initialization option — Initialization option**

Set real power, reactive power, terminal voltage, and terminal phase (default) | Set targets for rotor angle and Park's transform variables

Model for specifying values for certain parameters and variables at the start of simulation:

- To set an operating point independent of the connected network, select **Set real power, reactive power, terminal voltage and terminal phase**.
- To specify the priority and initial target values for block variables before simulation using the **Variables** settings, select **Set targets for rotor angle and Park's transform variables**. For more information, see “Set Priority and Initial Target for Block Variables”.

**Terminal voltage magnitude — Terminal voltage magnitude**

240 V (default) | scalar

Terminal voltage magnitude.

**Dependencies**

To enable this parameter, set **Initialization option** to **Set real power, reactive power, terminal voltage, and terminal phase**.

**Terminal voltage angle corresponding to ABC group — Terminal voltage angle corresponding to ABC group**

0 deg (default) | scalar

Terminal voltage angle corresponding to the ABC stator.

**Dependencies**

To enable this parameter, set **Initialization option** to **Set real power, reactive power, terminal voltage, and terminal phase**.

**Terminal active power (ABC group) — ABC terminal active power**

50e3 V\*A (default) | scalar

Terminal active power generated by the stator ABC windings.

**Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase.

**Terminal reactive power (ABC group) – ABC terminal reactive power**

0 V\*A (default) | scalar

Terminal reactive power generated by the stator ABC windings.

**Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase.

**Terminal active power (XYZ group) – XYZ terminal active power**

20e3 V\*A (default) | scalar

Terminal active power generated by the stator XYZ windings.

**Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase.

**Terminal reactive power (XYZ group) – XYZ terminal reactive power**

0 V\*A (default) | scalar

Terminal reactive power generated by the stator XYZ windings.

**Dependencies**

To enable this parameter, set **Initialization option** to Set real power, reactive power, terminal voltage, and terminal phase.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.

**Measurement temperature – Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

**Introduced in R2020b**

## References

- [1] Kieferndorf, F., Burzanowska , H., Kanerva S., Sario P. "Modeling of rotor based harmonics in dual-star, wound field, synchronous machines." *2008 18th International Conference on Electrical Machines*: Vilamoura, 1-6.
- [2] Burzanowska , H., Sario P, Stulz C., Joerg P. "Redundant Drive with Direct Torque Control (DTC) and dual-star synchronous machine, simulations and verifications." *2007 European Conference on Power Electronics and Applications*: Aalborg, 1-10.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

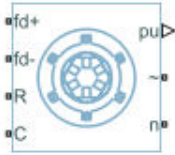
## See Also

[Synchronous Machine Field Circuit](#) | [Synchronous Machine Measurement](#) | [Synchronous Machine Model 1.0](#) | [Synchronous Machine Model 2.1](#) | [Synchronous Machine Salient Pole](#) | [Synchronous Machine Round Rotor](#)

# Synchronous Machine Round Rotor

Round-rotor synchronous machine with fundamental or standard parameterization

**Library:** Simscape / Electrical / Electromechanical / Synchronous



## Description

The Synchronous Machine Round Rotor block models a round-rotor synchronous machine using fundamental or standard parameters.

### Synchronous Machine Initialization Using Load-Flow Target Values

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize the machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”. For an example that shows how initialize an synchronous machine using data from a load flow analysis, see “Synchronous Machine Initialization with Loadflow”.

### Equations

The synchronous machine equations are expressed with respect to a rotating reference frame, defined by

$$\theta_e(t) = N\theta_r(t),$$

where:

- $\theta_e$  is the electrical angle.
- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor angle.

The Park transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. The Park transformation is defined by

$$P_s = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}} \frac{d\psi_d}{dt} - \Psi_q \omega_r - R_a i_d,$$

$$e_q = \frac{1}{\omega_{base}} \frac{d\psi_q}{dt} + \Psi_d \omega_r - R_a i_q,$$

and

$$e_0 = \frac{1}{\omega_{base}} \frac{d\Psi_0}{dt} - R_a i_0,$$

where:

- $e_d$ ,  $e_q$ , and  $e_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator voltages, defined by

$$\begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages measured from port  $\sim$  to neutral port  $\mathbf{n}$ .

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_d$ ,  $\psi_q$ , and  $\psi_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages.
- $\omega_r$  is the per-unit rotor rotational speed.
- $R_a$  is the stator resistance.
- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing from port  $\sim$  to port  $\mathbf{n}$ .

The rotor voltage equations are defined by

$$e_{fd} = \frac{1}{\omega_{base}} \frac{d\Psi_{fd}}{dt} + R_{fd} i_{fd},$$

$$e_{1d} = \frac{1}{\omega_{base}} \frac{d\Psi_{1d}}{dt} + R_{1d} i_{1d} = 0,$$

$$e_{1q} = \frac{1}{\omega_{base}} \frac{d\Psi_{1q}}{dt} + R_{1q} i_{1q} = 0,$$

and

$$e_{2q} = \frac{1}{\omega_{base}} \frac{d\Psi_{2q}}{dt} + R_{2q} i_{2q} = 0,$$

where:

- $e_{fd}$  is the field voltage.
- $e_{1d}$ ,  $e_{1q}$ , and  $e_{2q}$  are the voltages across the  $d$ -axis damper winding 1,  $q$ -axis damper winding 1, and  $q$ -axis damper winding 2. They are all equal to 0.
- $\psi_{fd}$ ,  $\psi_{1d}$ ,  $\psi_{1q}$ , and  $\psi_{2q}$  are the magnetic fluxes linking the field circuit,  $d$ -axis damper winding 1,  $q$ -axis damper winding 1, and  $q$ -axis damper winding 2.
- $R_{fd}$ ,  $R_{1d}$ ,  $R_{1q}$ , and  $R_{2q}$  are the resistances of rotor field circuit,  $d$ -axis damper winding 1,  $q$ -axis damper winding 1, and  $q$ -axis damper winding 2.
- $i_{fd}$ ,  $i_{1d}$ ,  $i_{1q}$ , and  $i_{2q}$  are the currents flowing in the field circuit,  $d$ -axis damper winding 1,  $q$ -axis damper winding 1, and  $q$ -axis damper winding 2.

The saturation equations are defined by

$$\psi_{ad} = \psi_d + L_l i_d,$$

$$\psi_{aq} = \psi_q + L_l i_q,$$

$$\psi_{at} = \sqrt{\psi_{ad}^2 + \psi_{aq}^2},$$

$K_s = 1$  (If saturation is disabled),

$K_s = f(\psi_{at})$  (If saturation is enabled),

$$L_{ad} = K_s * L_{adu},$$

and

$$L_{aq} = K_s * L_{aqu},$$

where:

- $\psi_{ad}$  is the  $d$ -axis air-gap or mutual flux linkage.
- $\psi_{aq}$  is the  $q$ -axis air-gap or mutual flux linkage.
- $\psi_{at}$  is the air-gap flux linkage.
- $K_s$  is the saturation factor.
- $L_{adu}$  is the unsaturated mutual inductance of the stator  $d$ -axis.
- $L_{ad}$  is the mutual inductance of the stator  $d$ -axis.
- $L_{aqu}$  is the unsaturated mutual inductance of the stator  $q$ -axis.
- $L_{aq}$  is the mutual inductance of the stator  $q$ -axis.

The saturation factor function,  $f$ , is calculated from the per-unit open-circuit lookup table as:

$$L_{ad} = \frac{d\psi_{at}}{di_{fd}},$$

$$V_{ag} = g(i_{fd}),$$

and

$$L_{ad} = \frac{dg(i_{fd})}{di_{fd}} = \frac{dV_{ag}}{di_{fd}},$$

where  $V_{ag}$  is the per-unit air-gap voltage.

In per-unit,

$$K_s = \frac{L_{ad}}{L_{adu}},$$

and

$$\psi_{at} = V_{ag}$$

can be rearranged to

$$K_s = f(\psi_{at}).$$

The stator flux linkage equations are defined by

$$\Psi_d = -(L_{ad} + L_l)i_d + L_{ad}i_{fd} + L_{ad}i_{1d},$$

$$\Psi_q = -(L_{aq} + L_l)i_q + L_{aq}i_{1q} + L_{aq}i_{2q},$$

and

$$\Psi_0 = -L_0i_0,$$

where:

- $L_l$  is the stator leakage inductance.
- $L_{ad}$  and  $L_{aq}$  are the mutual inductances of the stator  $d$ -axis and  $q$ -axis.

The rotor flux linkage equations are defined by

$$\psi_{fd} = L_{ffd}i_{fd} + L_{f1d}i_{1d} - L_{ad}i_d,$$

$$\psi_{1d} = L_{f1d}i_{fd} + L_{11d}i_{1d} - L_{ad}i_d,$$

$$\psi_{1q} = L_{11q}i_{1q} + L_{aq}i_{2q} - L_{aq}i_q,$$

and

$$\psi_{2q} = L_{aq}i_{1q} + L_{22q}i_{2q} - L_{aq}i_q,$$

where:

- $L_{ffd}$  is the self-inductances of the rotor field circuit.
- $L_{ffq}$  is the self-inductance of the rotor field circuit.
- $L_{11d}$  is the self-inductance of the  $d$ -axis damper winding 1.
- $L_{11q}$  is the self-inductance of the  $q$ -axis damper winding 1.
- $L_{22q}$  is the self-inductance of the  $q$ -axis damper winding 2.
- $L_{f1d}$  is the rotor field circuit and  $d$ -axis damper winding 1 mutual inductance.

The inductances are defined by these equations:



$$L_{ffd} = L_{ad} + L_{fd}$$

$$L_{f1d} = L_{ffd} - L_{fd}$$

$$L_{11d} = L_{f1d} + L_{1d}$$

$$L_{11q} = L_{aq} + L_{1q}$$

$$L_{22q} = L_{aq} + L_{2q}$$

The inductance equations assume that per-unit mutual inductance  $L_{12q} = L_{aq}$ , that is, the stator and rotor currents in the  $q$ -axis all link a single mutual flux represented by  $L_{aq}$ .

The rotor torque is defined by

$$T_e = \Psi_d i_q - \Psi_q i_d.$$

### Plotting and Display Options

You can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select an option:

- **Display Base Values** — Displays the machine per-unit base values in the MATLAB Command Window.
- **Display Associated Base Values** — Displays associated per-unit base values in the MATLAB Command Window.
- **Display Associated Initial Conditions** — Displays associated initial conditions in the MATLAB Command Window.
- **Plot Open-Circuit Saturation (pu)** — Plots air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , both measured in per-unit, in a MATLAB figure window. The plot contains three traces:
  - Unsaturated — **Stator d-axis mutual inductance (unsaturated),  $L_{ad}$**  you specify
  - Saturated — **Per-unit open-circuit lookup table (Vag versus ifd)** you specify
  - Derived — Open-circuit lookup table (per-unit) derived from the **Per-unit open-circuit lookup table (Vag versus ifd)** you specify. This data is used to calculate the saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_{at}$ , characteristic.
- **Plot Saturation Factor (pu)** — Plots saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_{at}$ , both measured in per-unit, in a MATLAB figure window using the machine parameters. This parameter is derived from other parameters that you specify:
  - **Stator d-axis mutual inductance (unsaturated),  $L_{ad}$**
  - **Per-unit field current saturation data, ifd**
  - **Per-unit air-gap voltage saturation data, Vag**

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and machine temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.

- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

For this block, the **Initial Targets** settings are visible only if, in the **Initial Conditions** section, you set the **Initialization option** parameter to `Set targets for rotor angle and Park's transform variables`.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Output

#### **pu — Machine per-unit measurements**

physical

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- Field voltage (field circuit base,  $E_{fd}$ ), *pu\_fd\_Efd*
- Field current (field circuit base,  $I_{fd}$ ), *pu\_fd\_Ifd*
- Electrical torque, *pu\_torque*
- Rotor velocity, *pu\_velocity*
- Stator d-axis voltage, *pu\_ed*
- Stator q-axis voltage, *pu\_eq*
- Stator zero-sequence voltage, *pu\_e0*
- Stator d-axis current, *pu\_id*
- Stator q-axis current, *pu\_iq*
- Stator zero-sequence current, *pu\_i0*
- Rotor electrical angle, *electrical\_angle\_out*

To connect to this port, use the Synchronous Machine Measurement block.

### Conserving

#### **fd+ — Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

**fd- — Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

**R — Machine rotor**

mechanical

Mechanical rotational conserving port associated with the machine rotor.

**C — Machine case**

mechanical

Mechanical rotational conserving port associated with the machine case.

**~ — Stator windings**

electrical

Expandable three-phase port associated with the stator windings.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**n — Neutral phase**

electrical

Electrical conserving port associated with the neutral point of the wye winding configuration.

**Dependencies**

To enable this port, set **Zero sequence** to Include.

**a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and machine temperature.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power — Rated apparent power**

555e6 V\*A (default)

Rated apparent power.

**Rated voltage — RMS rated line-line voltage**

24e3 V (default)

RMS rated line-line voltage.

**Rated electrical frequency — Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency at which rated apparent power is quoted.

**Number of pole pairs — Number of pole pairs**

1 (default)

Number of machine pole pairs.

**Specify parameterization by — Block parameterization**

Fundamental parameters (default) | Standard parameters

Block parameterization method. Options are:

- **Fundamental parameters** — Specify impedance using fundamental parameters.
- **Standard parameters** — Specify impedance using fundamental parameters and specify d-axis and q-axis time constants.

This parameter affects the visibility of the **Time Constant** settings and the parameters in the **Impedances** settings.

**Specify field circuit input required to produce rated terminal voltage at no load by — Specify field circuit input required to produce rated terminal voltage at no load by**

Field circuit current (default) | Field circuit voltage

Field circuit parameterization method. Options are:

- **Field circuit voltage** — Specify the field circuit in terms of voltage.
- **Field circuit current** — Specify the field circuit in terms of current.

This parameter affects the visibility of the **Field circuit voltage** and **Field circuit current** parameters.

**Field circuit voltage — Field circuit voltage**

95.95 V (default)

Voltage across field circuit which produces rated voltage at machine terminals.

**Dependencies**

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to **Field circuit voltage**.

**Field circuit current — Field circuit current**

1300 A (default)

Current in field circuit which produces rated voltage at machine terminals.

### Dependencies

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to `Field circuit current`.

### Zero sequence — Zero-sequence model

`Include (default)` | `Exclude`

Zero-sequence model:

- `Include` — Prioritize model fidelity. This model is the default zero-sequence model. An error occurs if you `Include` zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- `Exclude` — Prioritize simulation speed for desktop simulation or real-time deployment.

### Dependencies

If this parameter is set to:

- `Include` and **Specify parameterization by** is set to **Fundamental parameters** — The **Stator zero-sequence inductance, L0** parameter in the **Impedances** settings is visible.
- `Include` and **Specify parameterization by** is set to **Standard parameters** — The **zero-sequence reactance, X0** parameter in the **Impedances** settings is visible.
- `Exclude` — The zero-sequence parameter in the **Impedances** settings is not visible.

### Rotor angle definition — Reference point for the rotor angle measurement

`Angle between the a-phase magnetic axis and the d-axis (default)` | `Angle between the a-phase magnetic axis and the q-axis`

Reference point for the rotor angle measurement.

When you select the default value, the rotor *d*-axis and stator *a*-phase magnetic axis are aligned when the rotor angle is zero.

The other value you can choose for this parameter is `Angle between the a-phase magnetic axis and the q-axis`. When you select this value, the rotor *q*-axis and stator *a*-phase magnetic axis are aligned when the rotor angle is zero.

### Impedances

#### Stator d-axis mutual inductance (unsaturated), Ladu — Stator d-axis mutual inductance (unsaturated)

`1.66 (default)` | `positive scalar`

Unsaturated stator *d*-axis mutual inductance. If **Magnetic saturation representation** is set to `None`, this is equivalent to the stator *d*-axis mutual inductance. This parameter must be greater than 0.

### Dependencies

This parameter is visible only if the **Specify parameterization by** parameter is set to `Fundamental parameters`.

**Stator q-axis mutual inductance (unsaturated),  $L_{aq}$  — Stator q-axis mutual inductance (unsaturated)**

1.61 (default) | positive scalar

Unsaturated stator  $q$ -axis mutual inductance. If **Magnetic saturation representation** is set to **None**, this is equivalent to the stator  $q$ -axis mutual inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

**Stator zero-sequence inductance,  $L_0$  — Stator zero-sequence inductance**

0.15 (default) | positive scalar

Stator zero-sequence inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters** and the **Zero Sequence** parameter to **Include**.

**Stator leakage inductance,  $L_l$  — Stator leakage inductance**

0.15 (default) | positive scalar

Stator leakage inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

**Stator resistance,  $R_a$  — Stator resistance**

0.003 (default) | positive scalar

Stator resistance. This parameter must be greater than 0.

**Rotor field circuit inductance,  $L_{fd}$  — Rotor field circuit inductance**

0.165 (default) | positive scalar

Rotor field circuit inductance. This parameter must be greater than 0.

**Rotor field circuit resistance,  $R_{fd}$  — Rotor field circuit resistance**

0.0006 (default) | positive scalar

Rotor field circuit resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to **Fundamental parameters**.

**Rotor d-axis damper winding 1 inductance,  $L_{1d}$  — Rotor d-axis damper winding 1 inductance**

0.1713 (default) | positive scalar

Rotor  $d$ -axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor  $d$ -axis damper winding 1 resistance, R1d — Rotor  $d$ -axis damper winding 1 resistance**

0.0284 (default) | positive scalar

Rotor  $d$ -axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor  $q$ -axis damper winding 1 inductance, L1q — Rotor  $q$ -axis damper winding 1 inductance**

0.7252 (default) | positive scalar

Rotor  $q$ -axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor  $q$ -axis damper winding 1 resistance, R1q — Rotor  $q$ -axis damper winding 1 resistance**

0.00619 (default) | positive scalar

Rotor  $q$ -axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor  $q$ -axis damper winding 2 inductance, L2q — Rotor  $q$ -axis damper winding 2 inductance**

0.125 (default) | positive scalar

Rotor  $q$ -axis damper winding 2 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor  $q$ -axis damper winding 2 resistance, R2q — Rotor  $q$ -axis damper winding 2 resistance**

0.02368 (default) | positive scalar

Rotor  $q$ -axis damper winding 2 resistance. This parameter must be greater than 0.



**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Stator leakage reactance,  $X_l$  — Stator leakage reactance**

0.15 (default) | positive scalar

Stator leakage reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**d-axis synchronous reactance,  $X_d$  — d-axis synchronous reactance**

1.81 (default) | positive scalar

d-axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**q-axis synchronous reactance,  $X_q$  — q-axis synchronous reactance**

1.76 (default) | positive scalar

q-axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**zero-sequence reactance,  $X_0$  — Zero-sequence reactance**

0.15 (default) | positive scalar

Zero-sequence reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters and the **Zero Sequence** parameter to Include.

**d-axis transient reactance,  $X_d'$  — d-axis transient reactance**

0.3 (default) | positive scalar

d-axis transient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**q-axis transient reactance,  $X_q'$  — q-axis transient reactance**

0 (default) | positive scalar

$q$ -axis transient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

 **$d$ -axis subtransient reactance,  $X_d''$  —  $d$ -axis subtransient reactance**

0.23 (default) | positive scalar

$d$ -axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

 **$q$ -axis subtransient reactance,  $X_q''$  —  $q$ -axis subtransient reactance**

0.25 (default) | positive scalar

$q$ -axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**Time Constants****Specify  $d$ -axis time constant — Specify  $d$ -axis time constant**

Open circuit (default) | Short circuit

Select between Open circuit and Short circuit.

The setting for this parameter affects the visibility of the  $d$ -axis time constant parameters.

 **$d$ -axis transient open-circuit,  $T_{d0}'$  —  $d$ -axis transient open-circuit**

8 s (default) | positive scalar

$d$ -axis transient open-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than  **$d$ -axis subtransient open-circuit,  $T_{d0}''$** .

**Dependencies**

This parameter is visible only if the **Specify  $d$ -axis transient time constant** parameter is set to Open circuit.

 **$d$ -axis transient short-circuit,  $T_d'$  —  $d$ -axis transient short-circuit**

1.3260 s (default) | positive scalar

$d$ -axis transient short-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than  **$d$ -axis subtransient short-circuit,  $T_d''$** .

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to Short circuit.

**d-axis subtransient open-circuit,  $Td0''$  – d-axis subtransient open-circuit**

0.03 s (default) | positive scalar

$d$ -axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to Open circuit.

**d-axis subtransient short-circuit,  $Td''$  – d-axis subtransient short-circuit**

0.0230 s (default) | positive scalar

$d$ -axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to Short circuit.

**Specify q-axis time constant – Specify q-axis time constant**

Open circuit (default) | Short circuit

Select between Open circuit and Short circuit.

The setting for this parameter affects the visibility of the  $q$ -axis time constant parameters.

**q-axis transient open-circuit,  $Tq0'$  – q-axis transient open-circuit**

1 s (default) | positive scalar

$q$ -axis transient open-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **q-axis subtransient open-circuit,  $Tq0''$** .

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Open circuit.

**q-axis transient short-circuit,  $Tq'$  – q-axis transient short-circuit**

0.3693 s (default) | positive scalar

$q$ -axis transient short-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **q-axis subtransient short-circuit,  $Tq''$** .

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Short circuit.

**q-axis subtransient open-circuit,  $Tq0''$  – q-axis subtransient open-circuit**

0.07 s (default) | positive scalar

q-axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Open circuit.

**q-axis subtransient short-circuit,  $Tq''$  – q-axis subtransient short-circuit**

0.0269 s (default) | positive scalar

q-axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Short circuit.

**Saturation****Magnetic saturation representation – Magnetic saturation representation**

None (default) | Open-circuit lookup table (v versus i)

Block magnetic saturation model:

- None
- Open-circuit lookup table (v versus i)

**Dependencies**

If you set this parameter to Open-circuit lookup table (v versus i), related parameters are visible.

**Per-unit field current saturation data,  $ifd$  – Per-unit field current saturation data**

[0, .48, .76, 1.38, 1.79] (default)

Field current,  $i_{fd}$ , data that populates the air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , lookup table. This parameter must contain a vector with at least five elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i).

**Per-unit air-gap voltage saturation data,  $Vag$  – Per-unit air-gap voltage saturation data**

[0, .8, 1.08, 1.31, 1.4] (default)

Air-gap voltage,  $V_{ag}$ , data that populates the air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , lookup table. This parameter must contain a vector with at least five elements.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i).

## Initial Conditions

### Initialization option — Initialization option

Set real power, reactive power, terminal voltage, and terminal phase (default) | Set targets for rotor angle and Park's transform variables | Set targets for load flow variables

Model for specifying values for certain parameters and variables at the start of simulation. To:

Set an operating point regardless of the connected network, select Set real power, reactive power, terminal voltage and terminal phase.

- Specify the priority and initial target values for block variables before simulation using the **Variables** settings, select Set targets for rotor angle and Park's transform variables. For more information, see “Set Priority and Initial Target for Block Variables”.
- Select a bus type and specify the related parameters for a load-flow analysis in the **Initial Conditions** settings, select Set targets for load flow variables.

### Dependencies

If you set this parameter to:

- Set targets for rotor angle and Park's transform variables — The **Variables** settings become visible.
- Set real power, reactive power, terminal voltage, and terminal phase — Related parameters become visible.
- Set targets for load flow variables — Related parameters become visible.

### Source type — Voltage source model

Swing bus (default) | Voltage source with series impedance | PV bus | PQ bus

Type of voltage source that the block models.

### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.

The visibility of **Terminal voltage magnitude**, **Terminal voltage angle**, **Active power generated**, **Reactive power generated**, **Minimum terminal voltage magnitude (pu)**, **Phase search range at terminals**, and **Phase search range at terminals** depend on the value that you choose for this parameter.

### Terminal voltage magnitude — Terminal voltage magnitude

24e3 V (default)

Terminal voltage magnitude.

### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.

**Terminal voltage angle – Terminal voltage angle**

0 deg (default)

Terminal voltage angle.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus.

**Active power generated – Active power generated**

500e6 V\*A (default)

Active power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PV bus or PQ bus.

**Reactive power generated – Reactive power generated**

0 V\*A (default)

Reactive power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Minimum terminal voltage magnitude (pu) – Minimum terminal voltage magnitude**

0.95 (default)

Per-unit minimum steady-state voltage magnitude.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Phase search range at terminals – Phase search range at terminals**

[-inf, inf] deg (default)

Vector that defines the phase angle search range.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PV bus or PQ bus.

**Parasitic conductance to electrical reference – Parasitic conductance to electrical reference**

1e-6 S (default)

Parasitic conductance to the electrical reference.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to `Set targets for load flow variables`.

**Thermal**

To enable these parameters, set **Modeling option** to `Show thermal port`.

**Measurement temperature – Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

**Introduced in R2013b****Electrical connection ports update***Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`. Also, the simlog format is identical for `Composite three-phase ports` and `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Lyshevski, S. E. *Electromechanical Systems, Electric Machines and Applied Mechatronics*. Boca Raton, FL: CRC Press, 1999.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Synchronous Machine Field Circuit | Synchronous Machine Measurement | Synchronous Machine Model 1.0 | Synchronous Machine Model 2.1 | Synchronous Machine Salient Pole

### Topics

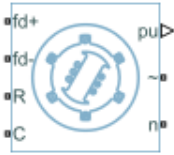
“Perform a Load-Flow Analysis Using Simscape Electrical”  
“Expand and Collapse Three-Phase Ports on a Block”



# Synchronous Machine Salient Pole

Salient-pole synchronous machine with fundamental or standard parameterization

**Library:** Simscape / Electrical / Electromechanical / Synchronous



## Description

The Synchronous Machine Salient Pole block models a salient-pole synchronous machine using fundamental or standard parameters.

### Synchronous Machine Initialization Using Load-Flow Target Values

If the block is in a network that is compatible with the frequency-time simulation mode, you can perform a load-flow analysis on the network. A load-flow analysis provides steady-state values that you can use to initialize the machine.

For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical” and “Frequency and Time Simulation Mode”. For an example that shows how initialize an synchronous machine using data from a load flow analysis, see “Synchronous Machine Initialization with Loadflow”.

### Equations

The synchronous machine equations are expressed with respect to a rotating reference frame, defined by

$$\theta_e(t) = N\theta_r(t),$$

where:

- $\theta_e$  is the electrical angle.
- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor angle.

The Park transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. The Park transformation is defined by

$$P_s = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The Park transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}} \frac{d\psi_d}{dt} - \Psi_q \omega_r - R_a i_d,$$

$$e_q = \frac{1}{\omega_{base}} \frac{d\psi_q}{dt} + \Psi_d \omega_r - R_a i_q,$$

and

$$e_0 = \frac{1}{\omega_{base}} \frac{d\Psi_0}{dt} - R_a i_0,$$

where:

- $e_d$ ,  $e_q$ , and  $e_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator voltages, defined by

$$\begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

$v_a$ ,  $v_b$ , and  $v_c$  are the stator voltages measured from port  $\sim$  to neutral port  $\mathbf{n}$ .

- $\omega_{base}$  is the per-unit base electrical speed.
- $\psi_d$ ,  $\psi_q$ , and  $\psi_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator flux linkages.
- $\omega_r$  is the per-unit rotor rotational speed.
- $R_a$  is the stator resistance.
- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence stator currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

$i_a$ ,  $i_b$ , and  $i_c$  are the stator currents flowing from port  $\sim$  to port  $\mathbf{n}$ .

The rotor voltage equations are defined by

$$e_{fd} = \frac{1}{\omega_{base}} \frac{d\Psi_{fd}}{dt} + R_{fd} i_{fd},$$

$$e_{1d} = \frac{1}{\omega_{base}} \frac{d\Psi_{1d}}{dt} + R_{1d} i_{1d} = 0,$$

and

$$e_{1q} = \frac{1}{\omega_{base}} \frac{d\Psi_{1q}}{dt} + R_{1q} i_{1q} = 0,$$

where:

- $e_{fd}$  is the field voltage.
- $e_{1d}$ , and  $e_{1q}$  are the voltages across the  $d$ -axis damper winding 1 and  $q$ -axis damper winding 1. They are equal to 0.

- $\psi_{fd}$ ,  $\psi_{1d}$ , and  $\psi_{1q}$  are the magnetic fluxes linking the field circuit,  $d$ -axis damper winding 1, and  $q$ -axis damper winding 1.
- $R_{fd}$ ,  $R_{1d}$ , and  $R_{1q}$  are the resistances of rotor field circuit,  $d$ -axis damper winding 1, and  $q$ -axis damper winding 1.
- $i_{fd}$ ,  $i_{1d}$ , and  $i_{1q}$  are the currents flowing in the field circuit,  $d$ -axis damper winding 1, and  $q$ -axis damper winding 1.

The saturation equations are defined by

$$\psi_{ad} = \psi_d + L_1 i_d,$$

$$\psi_{aq} = \psi_q + L_1 i_q,$$

$$\psi_{at} = \sqrt{\psi_{ad}^2 + \psi_{aq}^2},$$

$K_s = 1$  (if saturation is disabled),

$K_s = f(\psi_{at})$  (if saturation is enabled),

and

$$L_{ad} = K_s * L_{adu},$$

where:

- $\psi_{ad}$  is the  $d$ -axis air-gap or mutual flux linkage.
- $\psi_{aq}$  is the  $q$ -axis air-gap or mutual flux linkage.
- $\psi_{at}$  is the air-gap flux linkage.
- $K_s$  is the saturation factor.
- $L_{adu}$  is the unsaturated mutual inductance of the stator  $d$ -axis.
- $L_{ad}$  is the mutual inductance of the stator  $d$ -axis.

The saturation factor function,  $f$ , is calculated from the per-unit open-circuit lookup table as:

$$L_{ad} = \frac{d\psi_{at}}{di_{fd}},$$

$$V_{ag} = g(i_{fd}),$$

and

$$L_{ad} = \frac{dg(i_{fd})}{di_{fd}} = \frac{dV_{ag}}{di_{fd}},$$

where:

- $V_{ag}$  is the per-unit air-gap voltage.

In per-unit,

$$K_s = \frac{L_{ad}}{L_{adu}},$$

and

$$\psi_{at} = V_{ag}$$

can be rearranged to

$$K_s = f(\psi_{at}).$$

The stator flux linkage equations are defined by

$$\Psi_d = -(L_{ad} + L_l)i_d + L_{ad}i_{fd} + L_{ad}i_{1d},$$

$$\Psi_q = -(L_{aq} + L_l)i_q + L_{aq}i_{1q},$$

and

$$\Psi_0 = -L_0i_0,$$

where:

- $L_l$  is the stator leakage inductance.
- $L_{ad}$  and  $L_{aq}$  are the mutual inductances of the stator  $d$ -axis and  $q$ -axis.

The rotor flux linkage equations are defined by

$$\psi_{fd} = L_{ffd}i_{fd} + L_{f1d}i_{1d} - L_{ad}i_d,$$

$$\psi_{1d} = L_{f1d}i_{fd} + L_{11d}i_{1d} - L_{ad}i_d,$$

and

$$\psi_{1q} = L_{11q}i_{1q} - L_{aq}i_q,$$

where:

- $L_{ffd}$  is the self-inductance of the rotor field circuit.
- $L_{11d}$  is the self-inductance of the  $d$ -axis damper winding 1.
- $L_{11q}$  is the self-inductance of the  $q$ -axis damper winding 1.
- $L_{f1d}$  is the rotor field circuit and  $d$ -axis damper winding 1 mutual inductance.

The inductances are defined by these equations:

$$L_{ffd} = L_{ad} + L_{fd}$$

$$L_{f1d} = L_{ffd} - L_{fd}$$

$$L_{11d} = L_{f1d} + L_{1d}$$

$$L_{11q} = L_{aq} + L_{1q}$$

The inductance equations assume that per-unit mutual inductance  $L_{12q} = L_{aq}$ , that is, the stator and rotor currents in the  $q$ -axis all link a single mutual flux represented by  $L_{aq}$ .

The rotor torque is defined by

$$T_e = \Psi_d i_q - \Psi_q i_d.$$

### Plotting and Display Options

You can perform plotting and display actions using the **Electrical** menu on the block context menu.

Right-click the block and, from the **Electrical** menu, select one of these options:

- **Display Base Values** — Displays the machine per-unit base values in the MATLAB Command Window.
- **Display Associated Base Values** — Displays associated per-unit base values in the MATLAB Command Window.
- **Display Associated Initial Conditions** — Displays associated initial conditions in the MATLAB Command Window.
- **Plot Open-Circuit Saturation (pu)** — Plots air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , both measured in per-unit, in a MATLAB figure window. The plot contains three traces:
  - Unsaturated — **Stator d-axis mutual inductance (unsaturated),  $L_{ad}$**  you specify
  - Saturated — **Per-unit open-circuit lookup table (Vag versus ifd)** you specify
  - Derived — Open-circuit lookup table (per-unit) derived from the **Per-unit open-circuit lookup table (Vag versus ifd)** you specify. This data is used to calculate the saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_{at}$ , characteristic.
- **Plot Saturation Factor (pu)** — Plots saturation factor,  $K_s$ , versus magnetic flux linkage,  $\psi_{at}$ , both measured in per-unit, in a MATLAB figure window using the present machine parameters. This parameter is derived from other parameters that you specify:
  - **Stator d-axis mutual inductance (unsaturated),  $L_{ad}$**
  - **Per-unit field current saturation data, ifd**
  - **Per-unit air-gap voltage saturation data, Vag**

### Model Thermal Effects

You can expose thermal ports to model the effects of generated heat and machine temperature. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

For this block, the **Initial Targets** settings are visible only if, in the **Initial Conditions** section, you set the **Initialization option** parameter to **Set targets for rotor angle and Park's transform variables**.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Output

#### **pu — Machine per-unit measurements**

physical

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- Field voltage (field circuit base,  $E_{fd}$ ),  $pu\_fd\_E_{fd}$
- Field current (field circuit base,  $I_{fd}$ ),  $pu\_fd\_I_{fd}$
- Electrical torque,  $pu\_torque$
- Rotor velocity,  $pu\_velocity$
- Stator d-axis voltage,  $pu\_ed$
- Stator q-axis voltage,  $pu\_eq$
- Stator zero-sequence voltage,  $pu\_e0$
- Stator d-axis current,  $pu\_id$
- Stator q-axis current,  $pu\_iq$
- Stator zero-sequence current,  $pu\_i0$
- Rotor electrical angle,  $electrical\_angle\_out$

To connect to this port, use the Synchronous Machine Measurement block.

### Conserving

#### **fd+ — Field winding positive terminal**

electrical

Electrical conserving port associated with the field winding positive terminal.

#### **fd- — Field winding negative terminal**

electrical

Electrical conserving port associated with the field winding negative terminal.

#### **R — Machine rotor**

mechanical

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical

Mechanical rotational conserving port associated with the machine case.

#### **~ — Stator windings**

electrical

Expandable three-phase port associated with the stator windings.

**Dependencies**

To enable this port, set **Electrical connection** to Composite three-phase ports.

**n — Neutral phase**

electrical

Electrical conserving port associated with the neutral point of the wye winding configuration.

**Dependencies**

To enable this port, set **Zero sequence** to Include.

**a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**HA — a-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *a*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — b-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *b*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — c-phase terminal thermal port**

thermal

Thermal conserving port associated with stator winding *c*.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of generated heat and machine temperature.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Rated apparent power — Rated apparent power**

300e6 V\*A (default)

Rated apparent power.

**Rated voltage — RMS rated line-line voltage**

24e3 V (default)

RMS rated line-line voltage.

**Rated electrical frequency — Nominal electrical frequency**

60 Hz (default)

Nominal electrical frequency at which rated apparent power is quoted.

**Number of pole pairs — Number of pole pairs**

10 (default)

Number of machine pole pairs.

**Specify parameterization by — Block parameterization**

Fundamental parameters (default) | Standard parameters



Block parameterization method. Options are:

- **Fundamental parameters** — Specify impedance using fundamental parameters.
- **Standard parameters** — Specify impedance using fundamental parameters and specify d-axis and q-axis time constants.

#### Dependencies

If this parameter is set to **Fundamental parameters**:

- Fundamental parameters in the **Impedances** settings are visible.
- **Time Constant** settings are visible.

If this parameter is set to **Standard parameters**:

- Standard parameters in the **Impedances** settings are visible.
- **Time Constant** settings are not visible.

**Specify field circuit input required to produce rated terminal voltage at no load by** — Specify field circuit input required to produce rated terminal voltage at no load by

Field circuit current (default) | Field circuit voltage

Field circuit parameterization method. Options are:

- **Field circuit voltage** — Specify the field circuit in terms of voltage.
- **Field circuit current** — Specify the field circuit in terms of current.

This parameter affects the visibility of the **Field circuit voltage** and **Field circuit current** parameters.

#### Dependencies

If this parameter is set to **Field circuit voltage**:

- The **Field circuit voltage** parameter is visible.
- The **Field circuit current** parameter is not visible.

If this parameter is set to **Field circuit current**:

- The **Field circuit current** parameter is visible.
- The **Field circuit voltage** parameter is not visible.

**Field circuit voltage — Field circuit voltage**

216.54 V (default)

Voltage across field circuit which produces rated voltage at machine terminals.

#### Dependencies

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to **Field circuit voltage**.

**Field circuit current — Field circuit current**

1000 A (default)

Current in field circuit which produces rated voltage at machine terminals.

#### Dependencies

This parameter is visible only if the **Specify field circuit input required to produce rated terminal voltage at no load by** parameter is set to `Field circuit current`.

#### Zero sequence — Zero-sequence model

Include (default) | Exclude

Zero-sequence model:

- **Include** — Prioritize model fidelity. This model is the default zero-sequence model. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Exclude** — Prioritize simulation speed for desktop simulation or real-time deployment.

#### Dependencies

If this parameter is set to:

- **Include** and **Specify parameterization by** is set to **Fundamental parameters** — The **Stator zero-sequence inductance, L0** parameter in the **Impedances** settings is visible.
- **Include** and **Specify parameterization by** is set to **Standard parameters** — The **zero-sequence reactance, X0** parameter in the **Impedances** settings is visible.
- **Exclude** — The zero-sequence parameter in the **Impedances** settings is not visible.

#### Rotor angle definition — Reference point for the rotor angle measurement

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement.

When you select the default value, the rotor *d*-axis and stator *a*-phase magnetic axis are aligned when the rotor angle is zero.

The other value you can choose for this parameter is `Angle between the a-phase magnetic axis and the q-axis`. When you select this value, the rotor *q*-axis and stator *a*-phase magnetic axis are aligned when the rotor angle is zero.

#### Impedances

#### Stator d-axis mutual inductance (unsaturated), Ladu — Stator d-axis mutual inductance (unsaturated)

0.9 (default) | positive scalar

Unsaturated stator *d*-axis mutual inductance. If **Magnetic saturation representation** is set to `None`, this is equivalent to the stator *d*-axis mutual inductance. This parameter must be greater than 0.

#### Dependencies

This parameter is visible only if the **Specify parameterization by** parameter is set to `Fundamental parameters`.

**Stator q-axis mutual inductance,  $L_{aq}$  — Stator q-axis mutual inductance**

0.55 (default) | positive scalar

Stator  $q$ -axis mutual inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Stator zero-sequence inductance,  $L_0$  — Stator zero-sequence inductance**

0.15 (default) | positive scalar

Stator zero-sequence inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible if the **Specify parameterization by** parameter is set to Fundamental parameters and the **Zero Sequence** parameter is set to Include.

**Stator leakage inductance,  $L_l$  — Stator leakage inductance**

0.15 (default) | positive scalar

Stator leakage inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Stator resistance,  $R_a$  — Stator resistance**

0.011 (default) | positive scalar

Stator resistance. This parameter must be greater than 0.

**Rotor field circuit inductance,  $L_{fd}$  — Rotor field circuit inductance**

0.2571 (default) | positive scalar

Rotor field circuit inductance. This parameter must be greater than 0.

**Rotor field circuit resistance,  $R_{fd}$  — Rotor field circuit resistance**

0.0006 (default) | positive scalar

Rotor field circuit resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor d-axis damper winding 1 inductance,  $L_{1d}$  — Rotor d-axis damper winding 1 inductance**

0.2 (default) | positive scalar

Rotor  $d$ -axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter to Fundamental parameters.

**Rotor d-axis damper winding 1 resistance, R1d — Rotor d-axis damper winding 1 resistance**

0.0354 (default) | positive scalar

Rotor *d*-axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor q-axis damper winding 1 inductance, L1q — Rotor q-axis damper winding 1 inductance**

0.2567 (default) | positive scalar

Rotor *q*-axis damper winding 1 inductance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Rotor q-axis damper winding 1 resistance, R1q — Rotor q-axis damper winding 1 resistance**

0.0428 (default) | positive scalar

Rotor *q*-axis damper winding 1 resistance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Fundamental parameters.

**Stator leakage reactance, Xl — Stator leakage reactance**

0.15 (default) | positive scalar

Stator leakage reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter to Standard parameters.

**d-axis synchronous reactance, Xd — d-axis synchronous reactance**

1.05 (default) | positive scalar

*d*-axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter to Standard parameters.

**q-axis synchronous reactance,  $X_q$  — q-axis synchronous reactance**

0.7 (default) | positive scalar

$q$ -axis synchronous reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**zero-sequence reactance,  $X_0$  — Zero-sequence reactance**

0.15 (default) | positive scalar

Zero-sequence reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters and the **Zero Sequence** parameter is set to Include.

**d-axis transient reactance,  $X_d'$  — d-axis transient reactance**

0.35 (default) | positive scalar

$d$ -axis transient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**d-axis subtransient reactance,  $X_d''$  — d-axis subtransient reactance**

0.25 (default) | positive scalar

$d$ -axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**q-axis subtransient reactance,  $X_q''$  — q-axis subtransient reactance**

0.325 (default) | positive scalar

$q$ -axis subtransient reactance. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify parameterization by** parameter is set to Standard parameters.

**Time Constants****Specify d-axis time constant — Specify d-axis time constant**

Open circuit (default) | Short circuit

Select between Open circuit and Short circuit.

The setting for this parameter affects the visibility of the  $d$ -axis time constant parameters.

**d-axis transient open-circuit, Td0' – d-axis transient open-circuit**

5.25 s (default) | positive scalar

*d*-axis transient open-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **d-axis subtransient open-circuit, Td0''**.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Open circuit`.

**d-axis transient short-circuit, Td' – d-axis transient short-circuit**

1.75 s (default) | positive scalar

*d*-axis transient short-circuit time constant. This parameter must be:

- Greater than 0.
- Greater than **d-axis subtransient short-circuit, Td''**.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Short circuit`.

**d-axis subtransient open-circuit, Td0'' – d-axis subtransient open-circuit**

0.03 s (default) | positive scalar

*d*-axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Open circuit`.

**d-axis subtransient short-circuit, Td'' – d-axis subtransient short-circuit**

0.0214 s (default) | positive scalar

*d*-axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify d-axis transient time constant** parameter is set to `Short circuit`.

**Specify q-axis time constant – Specify q-axis time constant**`Open circuit` (default) | `Short circuit`

Select between `Open circuit` and `Short circuit`.

The setting for this parameter affects the visibility of the *q*-axis time constant parameters.

**q-axis subtransient open-circuit, Tq0' – q-axis subtransient open-circuit**

0.05 s (default) | positive scalar

*q*-axis subtransient open-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Open circuit.

**q-axis subtransient short-circuit,  $Tq'$  — q-axis subtransient short-circuit**

0.0232 s (default) | positive scalar

q-axis subtransient short-circuit time constant. This parameter must be greater than 0.

**Dependencies**

This parameter is visible only if the **Specify q-axis transient time constant** parameter is set to Short circuit.

**Saturation****Magnetic saturation representation — Magnetic saturation representation**

None (default) | Open-circuit lookup table (v versus i)

Block magnetic saturation model:

- None
- Open-circuit lookup table (v versus i)

**Dependencies**

If you set this parameter to Open-circuit lookup table (v versus i), related parameters are visible.

**Per-unit field current saturation data,  $ifd$  — Per-unit field current saturation data**

[0, .48, .76, 1.38, 1.79] (default)

Field current,  $i_{fd}$ , data that populates the air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , lookup table. This parameter must contain a vector with a number of elements between 5 and 10.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i).

**Per-unit air-gap voltage saturation data,  $Vag$  — Per-unit air-gap voltage saturation data**

[0, .8, 1.08, 1.31, 1.4] (default)

Air-gap voltage,  $V_{ag}$ , data that populates the air-gap voltage,  $V_{ag}$ , versus field current,  $i_{fd}$ , lookup table. This parameter must contain a vector with a number of elements between 5 and 10.

**Dependencies**

This parameter is visible only if the **Magnetic saturation representation** parameter is set to Open-circuit lookup table (v versus i).

## Initial Conditions

### Initialization option — Initialization option

Set real power, reactive power, terminal voltage, and terminal phase (default) | Set targets for rotor angle and Park's transform variables | Set targets for load flow variables

Model for specifying values for certain parameters and variables at the start of simulation. To:

Set an operating point regardless of the connected network, select Set real power, reactive power, terminal voltage and terminal phase.

- Specify the priority and initial target values for block variables before simulation using the **Variables** settings, select Set targets for rotor angle and Park's transform variables. For more information, see “Set Priority and Initial Target for Block Variables”.
- Select a bus type and specify the related parameters for a load-flow analysis in the **Initial Conditions** settings, select Set targets for load flow variables.

### Dependencies

If you set this parameter to:

- Set targets for rotor angle and Park's transform variables — The **Variables** settings become visible.
- Set real power, reactive power, terminal voltage, and terminal phase — Related parameters become visible.
- Set targets for load flow variables — Related parameters become visible.

### Source type — Voltage source model

Swing bus (default) | Voltage source with series impedance | PV bus | PQ bus

Type of voltage source that the block models.

### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.

The visibility of **Terminal voltage magnitude**, **Terminal voltage angle**, **Active power generated**, **Reactive power generated**, **Minimum terminal voltage magnitude (pu)**, **Phase search range at terminals**, and **Phase search range at terminals** depend on the value that you choose for this parameter.

### Terminal voltage magnitude — Terminal voltage magnitude

24e3 V (default)

Terminal voltage magnitude.

### Dependencies

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus or PV bus.



**Terminal voltage angle – Terminal voltage angle**

0 deg (default)

Terminal voltage angle.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to Swing bus.

**Active power generated – Active power generated**

270e6 V\*A (default)

Active power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PV bus or PQ bus.

**Reactive power generated – Reactive power generated**

0 V\*A (default)

Reactive power generated.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set real power, reactive power, terminal voltage, and terminal phase or if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Minimum terminal voltage magnitude (pu) – Minimum terminal voltage magnitude**

0.95 (default)

Per-unit minimum steady-state voltage magnitude.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PQ bus.

**Phase search range at terminals – Phase search range at terminals**

[-inf, inf] deg (default)

Vector that defines the phase angle search range.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to Set targets for load flow variables and the **Source type** parameter is set to PV bus or PQ bus.

**Parasitic conductance to electrical reference – Parasitic conductance to electrical reference**

1e-6 S (default)

Parasitic conductance to the electrical reference.

**Dependencies**

This parameter is visible only if the **Initialization option** parameter is set to `Set targets for load flow variables`.

**Thermal**

To enable these parameters, set **Modeling option** to `Show thermal port`.

**Measurement temperature – Rated temperature**

298.15 k (default) | real scalar

Temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Temperature coefficient**

3.93e-3 1/K (default) | positive scalar

Coefficient  $\alpha$  in the equation relating resistance to temperature for all three windings, as described in “Thermal Model for Actuator Blocks”. The default value, 3.93e-3 1/K, is for copper.

**Thermal mass for each stator winding – Winding thermal mass**

100 J/K (default) | positive scalar

Thermal mass value for each stator winding. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default) | positive scalar

Thermal mass of the rotor. The thermal mass is the energy required to raise the temperature of the rotor by one degree.

## Version History

**Introduced in R2013b****Electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either `Composite three-phase ports` or `Expanded three-phase ports`. Also, the simlog format is identical for `Composite three-phase ports` and `Expanded three-phase ports`.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Lyshevski, S. E. *Electromechanical Systems, Electric Machines and Applied Mechatronics*. Boca Raton, FL: CRC Press, 1999.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

[Synchronous Machine Field Circuit](#) | [Synchronous Machine Measurement](#) | [Synchronous Machine Model 1.0](#) | [Synchronous Machine Model 2.1](#) | [Synchronous Machine Round Rotor](#)

## Topics

[“Perform a Load-Flow Analysis Using Simscape Electrical”](#)  
[“Expand and Collapse Three-Phase Ports on a Block”](#)

# Synchronous Reluctance Machine

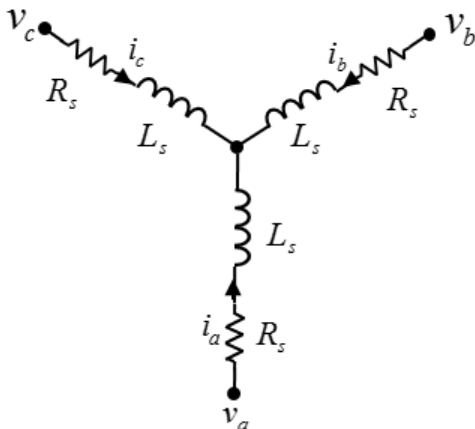
Synchronous reluctance machine with sinusoidal flux distribution

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



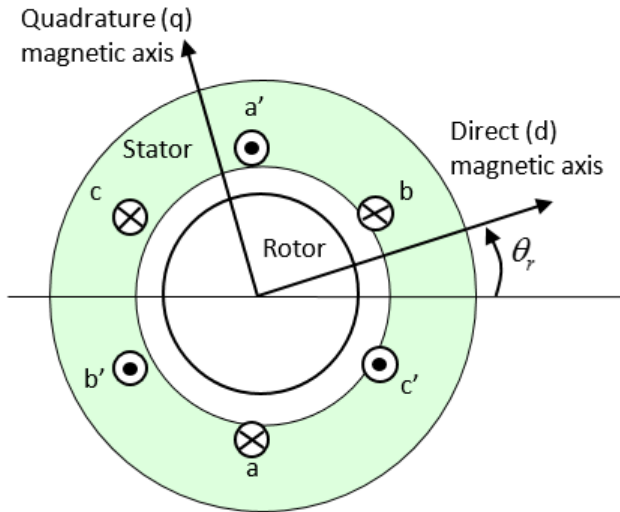
## Description

The Synchronous Reluctance Machine block represents a synchronous reluctance machine (SynRM) with sinusoidal flux distribution. The figure shows the equivalent electrical circuit for the stator windings.



## Motor Construction

The diagram shows the motor construction with a single pole-pair on the rotor. For the axes convention shown, when rotor mechanical angle  $\theta_r$  is zero, the  $a$ -phase and permanent magnet fluxes are aligned. The block supports a second rotor axis definition for which rotor mechanical angle is defined as the angle between the  $a$ -phase magnetic axis and the rotor  $q$ -axis.



### Equations

The combined voltage across the stator windings is

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \frac{d\psi_a}{dt} \\ \frac{d\psi_b}{dt} \\ \frac{d\psi_c}{dt} \end{bmatrix},$$

where:

- $v_a$ ,  $v_b$ , and  $v_c$  are the individual phase voltages across the stator windings.
- $R_s$  is the equivalent resistance of each stator winding.
- $i_a$ ,  $i_b$ , and  $i_c$  are the currents flowing in the stator windings.
- $\psi_a$ ,  $\psi_b$ , and  $\psi_c$  are the magnetic fluxes that link each stator winding.

The permanent magnet, excitation winding, and the three stator windings contribute to the flux that links each winding. The total flux is defined as

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

where:

- $L_{aa}$ ,  $L_{bb}$ , and  $L_{cc}$  are the self-inductances of the stator windings.
- $L_{ab}$ ,  $L_{ac}$ ,  $L_{ba}$ ,  $L_{bc}$ ,  $L_{ca}$ , and  $L_{cb}$  are the mutual inductances of the stator windings.

$$\theta_e = N\theta_r + \text{rotor offset}$$

$$L_{aa} = L_s + L_m \cos(2\theta_e)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3))$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(2(\theta_e + \pi/6))$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(2(\theta_e + \pi/6 - 2\pi/3))$$

and

$$L_{ca} = L_{ac} = -M_s - L_m \cos(2(\theta_e + \pi/6 + 2\pi/3))$$

where:

- $\theta_r$  is the rotor mechanical angle.
- $\theta_e$  is the rotor electrical angle.
- *rotor offset* is  $\theta$  if you define the rotor electrical angle with respect to the d-axis, or  $-\pi/2$  if you define the rotor electrical angle with respect to the q-axis.
- $L_s$  is the stator self-inductance per phase. This value is the average self-inductance of each of the stator windings.
- $L_m$  is the stator inductance fluctuation. This value is the amplitude of the fluctuation in self-inductance and mutual inductance with changing rotor angle.
- $M_s$  is the stator mutual inductance. This value is the average mutual inductance between the stator windings.

### Simplified Equations

Applying the Park transformation to the block electrical defining equations produces an expression for torque that is independent of rotor angle.

The Park transformation,  $P$ , is defined as

$$P = \frac{2}{3} \begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

where  $\theta_e$  is the electrical angle. The electrical angle depends on the rotor mechanical angle and the number of pole pairs such that

$$\theta_e = N\theta_r + \text{rotor offset}$$

where:

- $N$  is the number of pole pairs.
- $\theta_r$  is the rotor mechanical angle.

Applying the Park transformation to the first two electrical defining equations produces equations that define the behavior of the block:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N\omega_i q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N \omega i_d L_d,$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt},$$

$$T = \frac{3}{2} N (i_q i_d L_d - i_d i_q L_q)$$

$$J \frac{d\omega}{dt} = T - T_L - B_m \omega,$$

where:

- $i_d$ ,  $i_q$ , and  $i_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence currents, defined by

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix},$$

where  $i_a$ ,  $i_b$ , and  $i_c$  are the stator currents.

- $v_d$ ,  $v_q$ , and  $v_0$  are the  $d$ -axis,  $q$ -axis, and zero-sequence currents, defined by

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix},$$

where  $v_a$ ,  $v_b$ , and  $v_c$  are the stator currents.

- The  $dq0$  inductances are defined, respectively as
  - $L_d = L_s + M_s + \frac{3}{2} L_m$
  - $L_q = L_s + M_s - \frac{3}{2} L_m$
  - $L_0 = L_s - 2M_s$ .
- $R_s$  is the stator resistance per phase.
- $N$  is the number of rotor pole pairs.
- $T$  is the rotor torque. For the Synchronous Reluctance Machine block, torque flows from the machine case (block conserving port **C**) to the machine rotor (block conserving port **R**).
- $T_L$  is the load torque.
- $B_m$  is the rotor damping.
- $\omega$  is the rotor mechanical rotational speed.
- $J$  is the rotor inertia.

### Calculating Iron Losses

To calculate iron losses in the Synchronous Reluctance Machine block, choose whether you want to model them by specifying magnetizing resistance, parameterize them based on the Steinmetz equation, or if you want to tabulate the losses as a function of currents and rotor speed.

If you set the **Stator parameterization** parameter to Specify Ls, Lm, and Ms or Specify Ld, Lq, and L $\theta$  and **Modeling fidelity** to Constant Ld and Lq, then specify the **Magnetizing resistance** parameter to model iron losses.

In the **Iron Losses** settings, if you set the **Model** parameter to Specify tabulated Steinmetz coefficients or Tabulate with current and speed, the iron losses are applied to the mechanical side as a reduction in the electrical torque:

$$\tau_{loss} = \text{sign}(w) \frac{P_{loss, stator} + P_{loss, rotor}}{\text{abs}(w) + w_{min}},$$

where  $w_{min}$  is minimum rotor speed to start taking iron losses into account, and it is equal to 1 rad/s.

The total power dissipated is equal to the sum of the iron losses and the copper losses:

$$P_{dissipated} = \tau_{loss}w + R_a i_a^2 + R_b i_b^2 + R_c i_c^2.$$

### Steinmetz parameterization method

To model the iron losses based on the Steinmetz equation, set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Tabulated Ld and Lq. The Steinmetz method scales for different motor speeds or electrical frequencies so that iron loss data is only required as function of motor currents.

If, in the **Iron losses** settings, you set the **Model** parameter to Specify tabulated Steinmetz coefficients, the iron losses are given by:

$$P_{loss, rotor}(f) = k_{hr}(i_d, i_q)f + k_{Jr}(i_d, i_q)f^2 + k_{er}(i_d, i_q)f^{1.5}$$

$$P_{loss, stator}(f) = k_{hs}(i_d, i_q)f + k_{Js}(i_d, i_q)f^2 + k_{es}(i_d, i_q)f^{1.5}$$

where:

- $k_{hr}(i_d, i_q)$  is the **Rotor hysteresis loss coefficient, k\_hr(id,iq)**.
- $k_{Jr}(i_d, i_q)$  is the **Rotor eddy current loss coefficient, k\_Jr(id,iq)**.
- $k_{er}(i_d, i_q)$  is the **Rotor excess current loss coefficient, k\_er(id,iq)**.
- $k_{hs}(i_d, i_q)$  is the **Stator hysteresis loss coefficient, k\_hs(id,iq)**.
- $k_{Js}(i_d, i_q)$  is the **Stator eddy current loss coefficient, k\_Js(id,iq)**.
- $k_{es}(i_d, i_q)$  is the **Stator excess current loss coefficient, k\_es(id,iq)**.
- $f = w \frac{nPolePairs}{2\pi}$  is the electrical frequency, in Hz.

### Tabulate with current and speed

To model the iron losses by tabulating them independently for rotor and stator, set the **Stator parameterization** parameter to Specify Ld, Lq, and L $\theta$  and the **Modeling fidelity** parameter to Tabulated Ld and Lq.

Then, if in the **Iron losses** settings, you set the **Model** parameter to Tabulate with current and speed, the iron losses are given by:

$$P_{loss}(i_d, i_q, w) = \text{tablelookup}(P_{table}, i_d, i_q, w).$$



where  $w$  is the **Rotor speed vector**,  $w$  parameter.

### Iron losses with thermal ports

If you expose the thermal ports of the block, the power dissipated in each phase act as heat sources in the stator and in the rotor:

$$P_{dissA} = R_a i_a^2 + \frac{P_{loss, stator}}{3}$$

$$P_{dissB} = R_b i_b^2 + \frac{P_{loss, stator}}{3}$$

$$P_{dissC} = R_c i_c^2 + \frac{P_{loss, stator}}{3}$$

$$P_{dissR} = P_{loss, rotor}$$

If you specify the magnetizing resistance, the iron loss in the rotor is equal to a percentage of the total iron losses. This percentage remains constant and does not depend on other states:

$$P_{loss, rotor} = \frac{percent_{rotor}}{100} P_{iron loss, total} \cdot$$

Then the remaining iron loss is distributed among the different stator phases:

$$P_{loss, stator} = \left(1 - \frac{percent_{rotor}}{100}\right) P_{iron loss, total} \cdot$$

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions

The flux distribution is sinusoidal.

## Ports

### Conserving

#### **R — Machine rotor**

mechanical rotational

Mechanical rotational conserving port associated with the machine rotor.

#### **C — Machine case**

mechanical rotational

Mechanical rotational conserving port associated with the machine case.

#### **~ — Three-phase composite**

electrical

Expandable three-phase port associated with the stator windings.

### Dependencies

To enable this port, set **Electrical connection** to Composite three-phase ports.

#### **a — a-phase**

electrical

Electrical conserving port associated with *a*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **b — b-phase**

electrical

Electrical conserving port associated with *b*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **c — c-phase**

electrical

Electrical conserving port associated with *c*-phase.

### Dependencies

To enable this port, set **Electrical connection** to Expanded three-phase ports.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

### Dependencies

To enable this port, set **Zero sequence** to Include.

**HA — Winding A thermal port**

thermal

Thermal conserving port associated with winding A.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HB — Winding B thermal port**

thermal

Thermal conserving port associated with winding B.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HC — Winding C thermal port**

thermal

Thermal conserving port associated with winding C.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**HR — Rotor thermal port**

thermal

Thermal conserving port associated with the rotor.

**Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

**Parameters****Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

**Main****Electrical connection — Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Number of pole pairs — Rotor pole pairs**

6 (default) | integer

Number of permanent magnet pole pairs on the rotor.

**Stator parameterization – Parameterization model**Specify  $L_d$ ,  $L_q$  and  $L_0$  (default) | Specify  $L_s$ ,  $L_m$ , and  $M_s$ 

Stator parameterization model.

**Dependencies**The **Stator parameterization** setting affects the visibility of other parameters.**Modeling fidelity – Modeling fidelity**Constant  $L_d$  and  $L_q$  (default) | Tabulated  $L_d$  and  $L_q$ 

Select the modeling fidelity:

- Constant  $L_d$  and  $L_q$  —  $L_d$  and  $L_q$  values are constant and defined by their respective parameters.
- Tabulated  $L_d$  and  $L_q$  —  $L_d$  and  $L_q$  values are computed online from DQ currents look-up tables as follows:

$$L_d = f_1(i_d, i_q)$$

$$L_q = f_2(i_d, i_q)$$

**Dependencies**This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$ .**Stator d-axis inductance,  $L_d$  – Inductance**

0.0031 H (default)

Direct-axis inductance of the machine stator.

**Dependencies**This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Constant  $L_d$  and  $L_q$ .**Stator q-axis inductance,  $L_q$  – Inductance**

0.004 H (default)

Quadrature-axis inductance of the machine stator.

**Dependencies**This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Constant  $L_d$  and  $L_q$ .**Direct-axis current vector,  $i_D$  – Direct-axis current vector**

[-200, 0, 200] A (default)

Direct-axis current vector,  $i_D$ .**Dependencies**This parameter is visible only when you set the **Stator parameterization** parameter to Specify  $L_d$ ,  $L_q$ , and  $L_0$  and the **Modeling fidelity** parameter to Tabulated  $L_d$  and  $L_q$ .

**Quadrature-axis current vector,  $i_Q$  – Quadrature-axis current vector**

[-200, 0, 200] A (default)

Quadrature-axis current vector,  $i_Q$ .**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld and Lq.

**Ld matrix,  $L_d(i_d, i_q)$  – Ld matrix**

0.0031 \* ones(3, 3) H (default)

Ld matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld and Lq.

**Lq matrix,  $L_q(i_d, i_q)$  – Lq matrix**

0.004 \* ones(3, 3) H (default)

Lq matrix.

**Dependencies**

This parameter is visible only when you set the **Stator parameterization** parameter to Specify Ld, Lq, and L0 and the **Modeling fidelity** parameter to Tabulated Ld and Lq.

**Stator zero-sequence inductance,  $L_0$  – Inductance**

0.0005 H (default)

Zero-axis inductance for the machine stator.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify Ld, Lq and L0 and **Zero sequence** is set to Include.

**Stator self-inductance per phase,  $L_s$  – Inductance**

0.0025 H (default)

Average self-inductance of the three stator windings.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify Ls, Lm, and Ms.

**Stator inductance fluctuation,  $L_m$  – Inductance**

-0.0003 H (default)

Amplitude of the fluctuation in self-inductance and mutual inductance with the rotor angle.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify Ls, Lm, and Ms.

**Stator mutual inductance, Ms — Inductance**

0.0010 H (default)

Average mutual inductance between the stator windings.

**Dependencies**

This parameter is visible only if **Stator parameterization** is set to Specify Ls, Lm, and Ms.

**Stator resistance per phase, Rs — Resistance**

0.7 Ohm (default)

Resistance of each of the stator windings.

**Zero sequence — Zero-sequence model**

Include (default) | Exclude

Zero-sequence model:

- **Include** — Prioritize model fidelity. An error occurs if you Include zero-sequence terms for simulations that use the Partitioning solver. For more information, see “Increase Simulation Speed Using the Partitioning Solver”.
- **Exclude** — Prioritize simulation speed for desktop simulation or real-time deployment.

**Dependencies**

If this parameter is set to:

- **Include** and **Stator parameterization** is set to **Specify Ld, Lq, and L0** — The **Stator zero-sequence inductance, L0** parameter is visible.
- **Exclude** — The **Stator zero-sequence inductance, L0** parameter is not visible.

**Iron Losses**

For more information on calculating iron losses, see “Calculating Iron Losses” on page 1-2385.

**Model — Iron losses modeling option**

Specify magnetizing resistance (default) | Specify tabulated Steinmetz coefficients | Tabulate with current and speed

Whether to enable iron losses modeling.

**Dependencies**

To enable this parameter, set **Modeling fidelity** to Tabulate Ld and Lq.

**Magnetizing resistance — Magnetizing resistance for iron losses**

inf Ohm (default)

Magnetizing resistance. The value must be greater than zero. The default value is inf, which implies that there are no iron losses.

**Dependencies**

To enable this parameter, either set **Model** to Specify magnetizing resistance or set **Modeling fidelity** to Constant Ld and Lq.

**Rotor hysteresis loss coefficient,  $k_{hr}(id,iq)$  – Rotor hysteresis loss coefficient**  
`zeros(3, 3) W/Hz (default)`

Hysteresis loss coefficient of the rotor depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Rotor eddy current loss coefficient,  $k_{Jr}(id,iq)$  – Rotor eddy current loss coefficient**  
`zeros(3, 3) W/Hz^2 (default)`

Eddy current loss coefficient of the rotor depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Rotor excess current loss coefficient,  $k_{er}(id,iq)$  – Rotor excess current loss coefficient**  
`zeros(3, 3) W/Hz^1.50000 (default)`

Excess current loss coefficient of the rotor depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Stator hysteresis loss coefficient,  $k_{hs}(id,iq)$  – Stator hysteresis loss coefficient**  
`zeros(3, 3) W/Hz (default)`

Hysteresis loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Stator eddy current loss coefficient,  $k_{Js}(id,iq)$  – Stator eddy current loss coefficient**  
`zeros(3, 3) W/Hz^2 (default)`

Eddy current loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Stator excess current loss coefficient,  $k_{es}(id,iq)$  – Stator excess current loss coefficient**  
`zeros(3, 3) W/Hz^1.50000 (default)`

Excess current loss coefficient of the stator depending on the  $d$ -axis and  $q$ -axis currents. It is used by the Steinmetz equation.

**Dependencies**

To enable this parameter, set **Model** to Specify tabulated Steinmetz coefficients.

**Rotor speed vector,  $w$  — Rotor speed**

[1000, 2000] rpm (default)

Vector of rotor speed at which the iron loss is tabulated.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed.

**Rotor iron loss table,  $Pr(id,iq,w)$  — Rotor iron loss table**

zeros(3, 3, 2) W (default)

Iron losses of the rotor tabulated on the  $d$ -axis and  $q$ -axis currents, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed.

**Stator iron loss table,  $Ps(id,iq,w)$  — Stator iron loss table**

zeros(3, 3, 2) W (default)

Iron losses of the stator tabulated on the  $d$ -axis and  $q$ -axis currents, and rotor speed.

**Dependencies**

To enable this parameter, set **Model** to Tabulate with current and speed.

**Mechanical****Rotor inertia — Inertia**

0.01 kg\*m<sup>2</sup> (default)

Inertia of the rotor attached to mechanical translational port **R**.

**Rotor Damping — Damping**

0 N\*m/(rad/s) (default)

Rotary damping.

**Rotor angle definition — Angle**

Angle between the a-phase magnetic axis and the d-axis (default) | Angle between the a-phase magnetic axis and the q-axis

Reference point for the rotor angle measurement. If you select the default value, the rotor and  $a$ -phase fluxes are aligned for a zero-rotor angle. Otherwise, an  $a$ -phase current generates the maximum torque value for a zero-rotor angle.

**Thermal**

To enable these parameters, set **Modeling option** to Show thermal port.



**Measurement temperature – Measurement temperature**

298.15 K (default)

The temperature for which motor parameters are quoted.

**Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Thermal mass for each stator winding – Thermal mass for each stator winding**

100 J/K (default)

The thermal mass value for the A, B, and C windings. The thermal mass is the energy required to raise the temperature by one degree.

**Rotor thermal mass – Rotor thermal mass**

200 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Percentage of total iron losses associated with the rotor – Percentage of total iron losses**

50 (default)

Percentage of total iron losses associated with the rotor.

**Dependencies**

To enable this parameter, either set **Modeling fidelity** to Constant  $L_d$  and  $L_q$  or **Stator parameterization** to Specify  $L_s$ ,  $L_m$ , and  $M_s$ .

## Version History

**Introduced in R2017b****Electrical connection ports update***Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

## References

- [1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.
- [2] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.

[3] Moghaddam, R. *Synchronous Reluctance Machine (SynRM) in Variable Speed Drives (VSD) Applications - Theoretical and Experimental Reevaluation*. KTH School of Electrical Engineering, Stockholm, Sweden, 2011.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Simscape Blocks**

BLDC | Hybrid Excitation PMSM | PMSM | Switched Reluctance Machine | Synchronous Machine Field Circuit | Synchronous Machine Measurement

# Tap-Changing Transformer

Single-phase tap-changing transformer

**Library:** Simscape / Electrical / Passive / Transformers



## Description

The Tap-Changing Transformer block represents a single-phase tap-changing transformer. You can vary the turns-ratio of the transformer during the simulation using the control input.

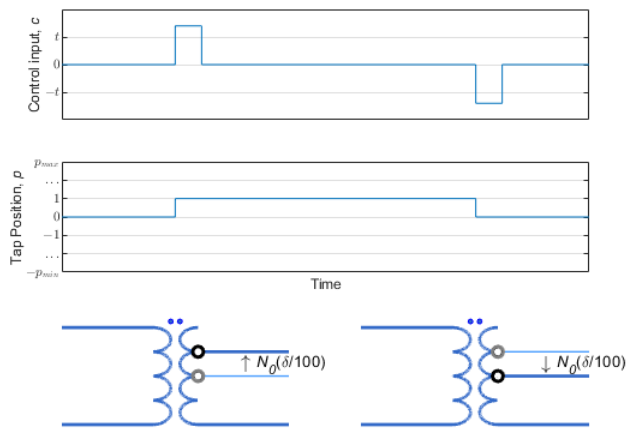
Use this component to regulate or change the output voltage of a linear transformer during a simulation. To model the effects of saturation, consider using the Nonlinear Transformer block.

## Operating Principle

Use the control input  $c$  to change the tap position of the transformer.

- Increase  $c$  above the **Control threshold**,  $t$ , to increase the tap position  $p$  by one.
- Decrease  $c$  below the negative of the **Control threshold** to decrease the tap position  $p$  by one.

This figure shows the tap-response to a control input.



In the diagram:

- $N_0$  is the nominal number of turns for the tap-changer winding.
- $\delta$  is the percent change in turns per tap position step. Specify this value using the **Change per tap (%)** parameter.

- $p_{min}$  and  $p_{max}$  are the minimum and maximum allowable tap indices. Specify these values with the **Minimum tap index (nominal=0)** and **Maximum tap index (nominal=0)** parameters.

To select the tap-changer winding, use the **Tap-changer location** parameter. The overall turns ratio,  $n$ , depends on this location:

- Primary

$$n = \frac{N_p(N_0, \delta, p)}{N_s} = n_0 \left( 1 + \frac{\delta p}{100} \right)$$

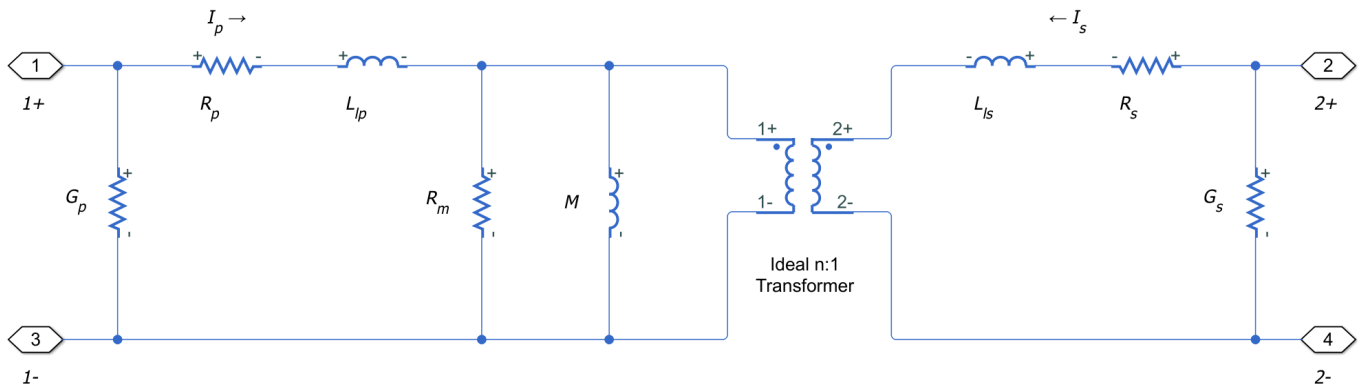
- Secondary

$$n = \frac{N_p}{N_s(N_0, \delta, p)} = n_0 \left( 1 + \frac{\delta p}{100} \right)^{-1}$$

where  $n_0$  is the nominal turns ratio for the transformer.

### Equivalent Circuit

The equivalent circuit is shown in the diagram.



Here:

- $R_p$  and  $R_s$  are the primary and secondary series resistances, respectively. Changes in the tap position affect these values.
- $L_{lp}$  and  $L_{ls}$  are the primary and secondary leakage inductances, respectively. Changes in the tap position affect these values.
- $R_m$  and  $M$  are the magnetization resistance and inductance, respectively. Changes in the tap position affect these values.
- $n$  is the turns ratio of the transformer.
- $G_p$  and  $G_s$  are the primary and secondary leakage conductances, respectively. Changes in the tap position do not affect these values.

## Ports

### Conserving

#### **1+ – Line 1 positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line 1.

#### **1- – Line 1 negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line 1.

#### **2+ – Line 2 positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line 2.

#### **2- – Line 2 negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line 2.

#### **c – Control input**

physical

Input physical signal that specifies the input control value.

## Parameters

### Nominal

#### **Turns ratio (primary/secondary) – Turns ratio**

1 (default) | positive number

Turns ratio of the transformer in the nominal tap position. The turns ratio is defined as the primary number of turns divided by the secondary number of turns.

#### **Primary leakage inductance – Line 1 leakage inductance**

1e-6 H (default) | positive number

Leakage inductance of the primary winding.

#### **Secondary leakage inductance – Line 2 leakage inductance**

1e-6 H (default) | positive number

Leakage inductance of the secondary winding.

#### **Core-loss resistance – Mutual resistance**

1e6 Ohm (default) | zero or positive number

Mutual resistance of the transformer.

**Magnetization inductance — Magnetization inductance**

100e-6 H (default) | positive number

Magnetization inductance of the transformer.

**Primary series resistance — Line 1 series resistance**

0 Ohm (default) | zero or positive number

Series resistance of the primary winding.

**Secondary series resistance — Line 2 series resistance**

0 Ohm (default) | zero or positive number

Series resistance of the secondary winding.

**Primary leakage conductance — Line 1 leakage conductance**

0 1/Ohm (default) | zero or positive number

Leakage conductance of the primary winding. Set this value to be nonzero for achieving numerical convergence in some circuit topologies.

**Secondary leakage conductance — Line 2 leakage conductance**

0 1/Ohm (default) | zero or positive number

Leakage conductance of the secondary winding. Set this value to be nonzero for achieving numerical convergence in some circuit topologies.

**Tap****Tap-changer location — Tap-changer winding**

Primary (default) | Secondary

Specify whether the tap-changer is on the primary or secondary winding.

**Minimum tap index (nominal=0) — Minimum tap position**

-5 (default) | zero or negative integer

Minimum allowable position of the tap-changer.

**Maximum tap index (nominal=0) — Maximum tap position**

5 (default) | zero or positive integer

Maximum allowable position of the tap-changer.

**Change per tap (%) — Tap change step**

1 % (default) | positive number

Percent change in the number of turns per tap position step of the tap-changer winding. Set this value such that the absolute percent change at the minimum and maximum tap indices is less than 100 %.

**Control threshold — Shift threshold**

0.5 (default)

Control value at which the tap changes position. To lower the tap position, apply a control signal **c** smaller than the negative of this value. To increase the tap position, apply a control signal **c** greater than this value.

## **Version History**

**Introduced in R2018a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Earthing Transformer | Ideal Transformer | Nonlinear Transformer | Three-Winding Transformer (Three-Phase) | Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer

# Thermal Resistor

Heat transfer by conduction through a layer of material

**Library:** Simscape / Electrical / Passive / Thermal



## Description

The Thermal Resistor block represents heat transfer by conduction through a layer of material. The heat transfer is:

- Governed by Fourier's law
- Proportional to the temperature difference across the layer of material
- Inversely proportional to the thermal resistance of the material

The equation for conductive heat transfer is:

$$Q_{AB} = \frac{T_{AB}}{R_{thermal}}$$

where:

- $Q_{AB}$  is the heat flow through the material.
- $T_{AB}$  is the temperature difference across the layer of material.
- $R_{thermal}$  is the thermal resistance of the material.

Thermal resistance can be calculated as:

$$R_{thermal} = \frac{D}{kA}$$

where:

- $D$  is the thickness of the layer of material.
- $k$  is the thermal conductivity of the material.
- $A$  is the area normal to the heat flow direction.

Use the Thermal Resistor block to parameterize an equivalent component in terms of thermal resistance of the material layer. To parameterize an equivalent component in terms of the thickness, thermal conductivity, and area of the material layer, use the Conductive Heat Transfer block from the Simscape Foundation library.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see "Set Priority and Initial Target for Block Variables".

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can



come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### A — Surface A

thermal

Thermal conserving port associated with surface A of the material that the heat flows through.

#### B — Surface B

thermal

Thermal conserving port associated with surface B of the material that the heat flows through.

## Parameters

### Thermal resistance — Thermal resistance

1e-3 K/W (default)

Thermal resistance,  $R_{thermal}$ .

## Version History

Introduced in R2016a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Cauer Thermal Model | Foster Thermal Model

# Thermistor

Negative temperature coefficient thermistor using B-parameter equation

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Thermistor block represents an NTC thermistor using the B-parameter equation. The resistance at temperature  $T$  is

$$R = R_0 e^{B(1/T - 1/T_0)}$$

where:

- $R_0$  is the nominal resistance at the reference temperature  $T_0$ .
- $B$  is the characteristic temperature constant.

The following equation describes the thermal behavior of the block:

$$Q = K_d t_c \frac{dT}{dt}$$

where:

- $Q$  is the net heat flow into port A.
- $K_d$  is the **Dissipation factor** parameter value.
- $t_c$  is the **Thermal time constant** parameter value.
- $dT/dt$  is the rate of change of the temperature.

To model the thermistor in free space:

- 1 Connect the thermistor to the B port of a Simscape Convective Heat Transfer block.
- 2 Connect the A port of the Convective Heat Transfer block to a Simscape Ideal Temperature Source block whose temperature is set to the ambient temperature.
- 3 Set the **Area** parameter of the Convective Heat Transfer block to an approximate area  $A_{\text{nom}}$ .
- 4 Set the **Heat transfer coefficient** parameter of the Convective Heat Transfer block to  $K_d/A_{\text{nom}}$ .

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### A — Thermal port

thermal

Thermal port.

#### + — Positive port

electrical

Electrical conserving port associated with the thermistor positive port.

#### - — Negative port

electrical

Electrical conserving port associated with the thermistor negative port.

## Parameters

### Electrical

#### Nominal resistance $R_0$ at reference temperature $T_0$ — Nominal resistance at reference temperature

1000 Ohm (default)

Nominal resistance of the thermistor at the reference temperature. Many datasheets quote the nominal resistance at 25°C and list it as R25.

#### Characteristic temperature constant $B$ — Characteristic temperature constant

3500 K (default)

Coefficient  $B$  in the equation that describes resistance as a function of temperature.

#### Reference temperature $T_0$ — Reference temperature

298.15 K (default)

Temperature at which the nominal resistance was measured.

### Thermal

#### Thermal time constant — Thermal time constant

5.0 s (default)

Time it takes the sensor temperature to reach 63% of the final temperature change when a step change in ambient temperature occurs.

#### Dissipation factor — Dissipation factor

0.75e-3 W/K (default)

Thermal power required to raise the thermistor temperature by one K.

## **Version History**

**Introduced in R2008a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

PTC Thermistor | Resistor | Thermal Resistor

# Thermocouple

Sensor that converts thermal potential difference into electrical potential difference

**Library:** Simscape / Electrical / Sensors & Transducers



## Description

The Thermocouple block represents a thermocouple using the standard polynomial parameterization defined in the NIST ITS-90 Thermocouple Database [1].

For thermocouples Type B, E, J, K ( $t \leq 0$  degC), N, R, S or T, the voltage  $E$  across the device in mV is

$$E(\text{mV}) = c_0 + c_1 t + \dots + c_n t^n$$

where:

- $c_i$  is the  $i^{\text{th}}$  element of the **Coefficients [c0 c1 ... cn]** parameter value.
- $t$  is the temperature difference in degrees Celsius between the temperature at the thermal port A and the **Reference temperature** parameter value.

---

**Note** The equation for voltage across the device as a function of temperature difference is defined in mV. The units of the voltage across the actual device is V.

---

For thermocouples Type K ( $t \geq 0$  degC), the equation contains an additional exponential term:

$$E(\text{mV}) = c_0 + c_1 t + \dots + c_n t^n + a_0 e^{a_1(t - a_2)^2}$$

where  $a_0$ ,  $a_1$ , and  $a_2$  are additional coefficients, required only by the Type K thermocouple, defined by the **Coefficients [a0 a1 a2]** parameter value.

The following equation describes the thermal behavior of the block:

$$Q = K_d t_c \frac{dT}{dt}$$

where:

- $T$  is the temperature at port A.
- $Q$  is the net heat flow into port A.
- $K_d$  is the **Dissipation factor** parameter value.
- $t_c$  is the **Thermal time constant** parameter value.
- $dT/dt$  is the rate of change of the temperature.

To model the thermocouple in free space:

- 1 Connect the thermocouple to the B port of a Simscape Convective Heat Transfer block.
- 2 Connect the A port of the Convective Heat Transfer block to a Simscape Ideal Temperature Source block whose temperature is set to the ambient temperature.
- 3 Set the **Area** parameter of the Convective Heat Transfer block to an approximate area  $A_{nom}$ .
- 4 Set the **Heat transfer coefficient** parameter of the Convective Heat Transfer block to  $K_d/A_{nom}$ .

## Assumptions and Limitations

- The high-order polynomials this block uses are very sensitive to the number of significant figures used for computation. Use all available significant figures when specifying the **Coefficients [c0 c1 ... cn]** parameter.
- **Coefficients [c0 c1 ... cn]** are defined for use over a specified temperature range.
- The maximum supported value for n in the **Coefficients [c0 c1 ... cn]** parameter is 14, that is, the vector cannot have more than 15 elements.

## Ports

### Conserving

#### A — Thermal port

thermal

Thermocouple thermal port.

#### + — Positive port

electrical

Electrical conserving port associated with the thermocouple positive port.

#### - — Negative port

electrical

Electrical conserving port associated with the thermocouple negative port.

## Parameters

### Electrical

#### Thermocouple type — Thermocouple type

Type B, E, J, K ( $t \leq 0$  degC), N, R, S or T (default) | Type K ( $t \geq 0$  degC)

Select one of the modeling options:

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

#### Coefficients [c0 c1 ... cn] — Coefficients [c0 c1 ... cn]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0] (default)

Vector of coefficients  $c$  in the equation that describes voltage as a function of temperature. The maximum length of the vector is 15 elements.

---

**Note** You can download parameters for standard thermocouple types from the NIST database [1]. For information on how to do this, see the Simulink “Approximating Nonlinear Relationships: Type S Thermocouple” example.

---

### **Coefficients [a0 a1 a2] – Coefficients [a0 a1 a2]**

[0, 0, 0] (default)

Vector of additional coefficients  $a_0$ ,  $a_1$ , and  $a_2$ , required only by the Type K thermocouple.

### **Dependencies**

This parameter is visible only when you select Type K ( $t \geq 0$  degC) for the **Thermocouple model** parameter.

### **Thermal**

#### **Reference temperature – Reference temperature**

0 degC (default)

Temperature the block subtracts from the temperature at the thermal port in calculating the voltage across the device.

#### **Thermal time constant – Thermal time constant**

1 s (default)

Time it takes the thermocouple temperature to reach 63% of the final temperature change when a step change in ambient temperature occurs.

#### **Dissipation factor – Dissipation factor**

1e-3 W/K (default)

Thermal power required to raise the thermocouple temperature by one K.

#### **Initial temperature – Initial temperature**

25 degC (default)

Temperature of the thermocouple at the start of the simulation.

## **Version History**

**Introduced in R2008a**

## **References**

[1] Dean C. Ripple, NIST ITS-90 Thermocouple Database, December 21, 1999 (<https://srdata.nist.gov/its90/main/>)

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

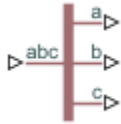
Thermal Resistor



# Three Element Demux

Convert three-element physical signal vector into scalar physical signals

**Library:** Simscape / Electrical / Connectors & References



## Description

The Three Element Demux block splits a three-element physical signal vector into three scalar physical signals.

## Ports

### Input

**abc — Three-element physical signal**  
physical signal

Three-element physical signal input port.

### Output

**a — Scalar physical signal**  
physical signal

Scalar physical signal output port.

**b — Scalar physical signal**  
physical signal

Scalar physical signal output port.

**c — Scalar physical signal**  
physical signal

Scalar physical signal output port.

## Version History

Introduced in R2013b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

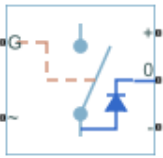
**See Also**

Phase Permute | Phase Splitter

## Three-Level Converter (Three-Phase)

Twelve-pulse three-phase three-level neutral-point clamped controlled converter

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



### Description

The Three-Level Converter (Three-Phase) block models a twelve-pulse three-phase three-level neutral-point clamped controlled converter. You can use this block to connect a three-phase AC network to a three-level DC network.

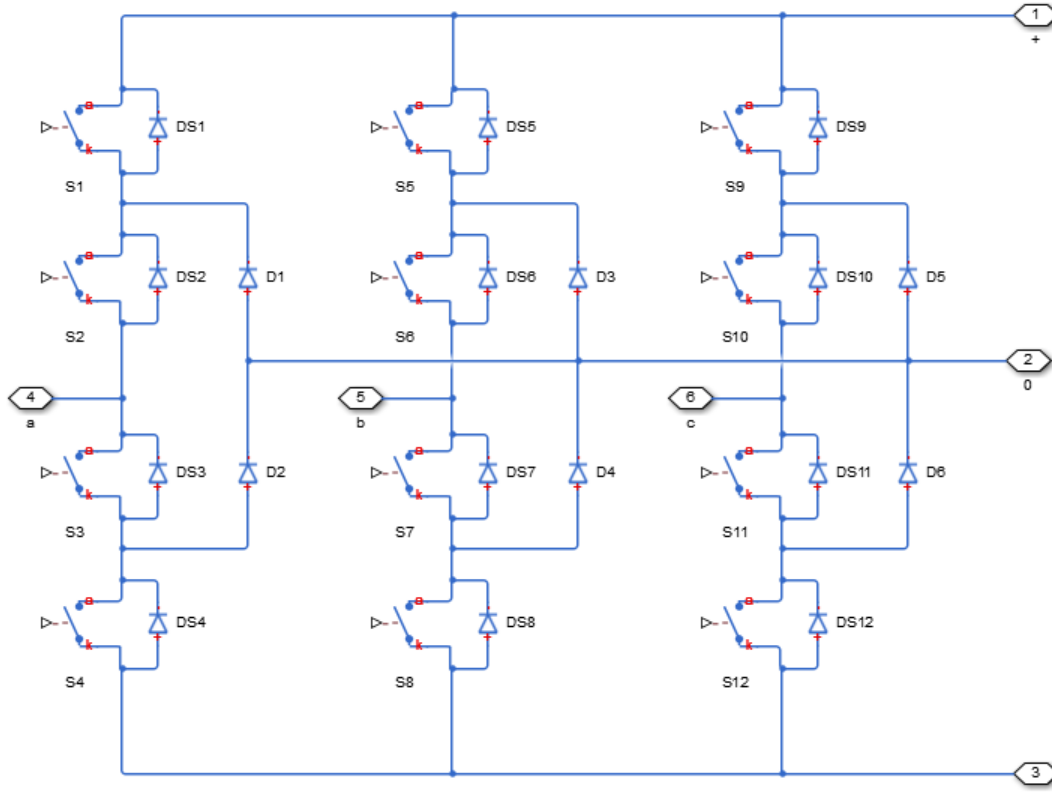
### Model

The block contains three bridge arms, each of which has four switching devices and the associated anti-parallel diodes. Options for the type of switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

Each component in the three-arm circuit is the same switching device that you specify. The switching devices are the same as the devices in the **Semiconductors > Fundamental Components** sublibrary.

The figure shows the equivalent circuit for the block using an Ideal Semiconductor block as the switching device.



You control the gate ports of the 12 switching devices via an input to the Three-Level Converter (Three-Phase) block **G** port.

- 1 Use a Twelve-Pulse Gate Multiplexer block to multiplex all 12 gate signals into a single vector.
- 2 Connect the output of the Twelve-Pulse Gate Multiplexer block to the Three-Level Converter (Three-Phase) block **G** port.

You use the **Diodes** tab of the block dialog box to include an integral protection diode for each switching device. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

The table shows how to set the **Integral protection diode** parameter based on your goals.

Goals	Value to Select	Integral Protection Diode
Prioritize simulation speed.	Diode with no dynamics	The Diode block
Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You use the **Snubbers** tab of the block dialog box to include a snubber circuit for each switching device. Each snubber consists of a resistor and capacitor connected in series. Typically, a snubber circuit protects a switching device against very high voltages produced by an inductive load when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of change of current when a switching device turns on.

## Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to `Averaged switch` and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input  $\mathbf{G}$ . To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Ports

### Conserving

#### **G — Gate terminals**

electrical

Vector input port associated with the gate terminals of the switching devices. Connect this port to a Twelve-Pulse Gate Multiplexer block.

#### **~ — Three-phase port**

electrical

Expandable three-phase port.

### Dependencies

To enable this port, set **Electrical connection** to `Composite three-phase ports`.

#### **a — a-phase**

electrical

Electrical conserving port associated with  $a$ -phase.

### Dependencies

To enable this port, set **Electrical connection** to `Expanded three-phase ports`.

#### **b — b-phase**

electrical

Electrical conserving port associated with  $b$ -phase.

### Dependencies

To enable this port, set **Electrical connection** to `Expanded three-phase ports`.

**c – c-phase**

electrical

Electrical conserving port associated with c-phase.

**Dependencies**

To enable this port, set **Electrical connection** to Expanded three-phase ports.

**+ – Positive terminal**

electrical

Electrical conserving port associated with the DC positive terminal.

**0 – Neutral terminal**

electrical

Electrical conserving port associated with the DC neutral terminal.

**- – Negative terminal**

electrical

Electrical conserving port associated with the DC negative terminal.

**Parameters****Electrical connection – Electrical connection**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to have composite or expanded three-phase ports.

**Switching device – Switching device**

Ideal Semiconductor Switch (default) | GTO | IGBT | MOSFET | Averaged Switch

Converter switching device. The default value is Ideal Semiconductor Switch.

The switching devices you can select are:

- “Switching Devices: GTO” on page 1-2416
- “Switching Devices: Ideal Semiconductor Switch” on page 1-2417
- “Switching Devices: IGBT” on page 1-2418
- “Switching Devices: MOSFET” on page 1-2418
- “Switching Devices: Averaged Switch” on page 1-2419

**Dependencies**

Multiple additional parameters will become visible depending on the choice of the specific switching device.

**Switching Devices: GTO**

For more information, see GTO.

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**Forward voltage, Vf — Forward voltage**

0.8 V (default)

Minimum voltage required across the anode and cathode block ports for the gradient of the device i-v characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

**On-state resistance — On-state resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the forward voltage.

**Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Gate trigger voltage, Vgt — Gate trigger voltage**

1 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Gate turn-off voltage, Vgt\_off — Gate turn-off voltage**

-1 V (default)

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Holding current — Holding current**

1 A (default)

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Switching Devices: Ideal Semiconductor Switch**

For more information, see Ideal Semiconductor Switch.

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**On-state resistance — On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

**Threshold voltage, Vth — Threshold voltage**

6 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

### **Switching Devices: IGBT**

For more information, see IGBT (Ideal, Switching).

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

#### **Forward voltage, Vf — Forward voltage**

0.8 V (default)

Minimum voltage required across the collector and emitter block ports for the gradient of the diode  $i$ - $v$  characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**.

#### **On-state resistance — On-state resistance**

0.001 Ohm (default)

Collector-emitter resistance when the device is on.

#### **Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Collector-emitter conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

#### **Threshold voltage, Vth — Threshold voltage**

6 V (default)

Gate-emitter voltage at which the device turns on.

### **Switching Devices: MOSFET**

For more information, see MOSFET (Ideal, Switching).

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

#### **Drain-source on resistance, R\_DS(on) — Drain-source on resistance**

0.001 Ohm (default)

Drain-source resistance when the device is on.

#### **Off-state Conductance — Off-state conductance**

1e-6 1/Ohm (default)

Drain-source conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

#### **Threshold voltage, Vth — Threshold voltage**

6 V (default)

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value.



**Switching Devices: Averaged Switch**

The parameters for this switching device will be visible only if you select it in the **Switching device** parameter.

**On-state resistance — On-state resistance**

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Integer for piecewise constant approximation of gate input (0 for disabled) — Integer mode for FPGA deployment**

0 (default) | nonnegative scalar

Integer used to perform piecewise constant approximation of the gate input for FPGA deployment.

**Dependencies**

To enable this parameter, set **Switching device** to Averaged Switch.

**Integral Diodes****Integral protection diode — Integral protection diode**

None (default) | Diode with no dynamics | Diode with charge dynamics

Integral protection diode for each switching device.

The diodes you can select are:

- Diode with no dynamics
- Diode with charge dynamics

---

**Note** If you select Averaged Switch for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and Diode with no dynamics is automatically selected.

---

**Forward voltage — Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

**Off conductance — Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with no dynamics or Diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** 

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.

If you select Specify stretch factor or Specify reverse recovery charge, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics.

#### Reverse recovery time, $t_{rr}$ — Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery time directly.

#### Reverse recovery time stretch factor — Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify stretch factor.

#### Reverse recovery charge, $Q_{rr}$ — Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

### Dependencies

To enable this parameter, set **Integral protection diode** to Diode with charge dynamics and **Reverse recovery time parameterization** to Specify reverse recovery charge.

For more information on these parameters, see Diode.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

#### Snubber — Snubber

None (default) | RC snubber

Snubber for each switching device:

- None - This is the default value.
- RC snubber

#### Snubber resistance — Snubber resistance

0.1 Ohm (default)

Snubber resistance.

#### Dependencies

This parameter is visible only when the **Snubber** parameter is set to RC snubber.

#### Snubber capacitance — Snubber capacitance

1e-7 F (default)

Snubber capacitance.

#### Dependencies

This parameter is visible only when the **Snubber** parameter is set to RC snubber.

## Version History

### Introduced in R2014b

#### Electrical connection ports update

*Behavior changed in R2021b*

From R2021b forward, to switch between composite and expanded ports, set the **Electrical connection** parameter to either Composite three-phase ports or Expanded three-phase ports.

As a result of these changes, inside a model saved in an earlier release, review the **Electrical connection** parameter of this block.

### Extended Capabilities

#### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

[Average-Value Inverter \(Three-Phase\)](#) | [Average-Value Rectifier \(Three-Phase\)](#) | [Converter \(Three-Phase\)](#) | [Rectifier \(Three-Phase\)](#) | [Twelve-Pulse Gate Multiplexer](#)

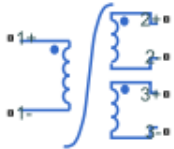
## **Topics**

[“Expand and Collapse Three-Phase Ports on a Block”](#)

# Three-Winding Nonlinear Transformer

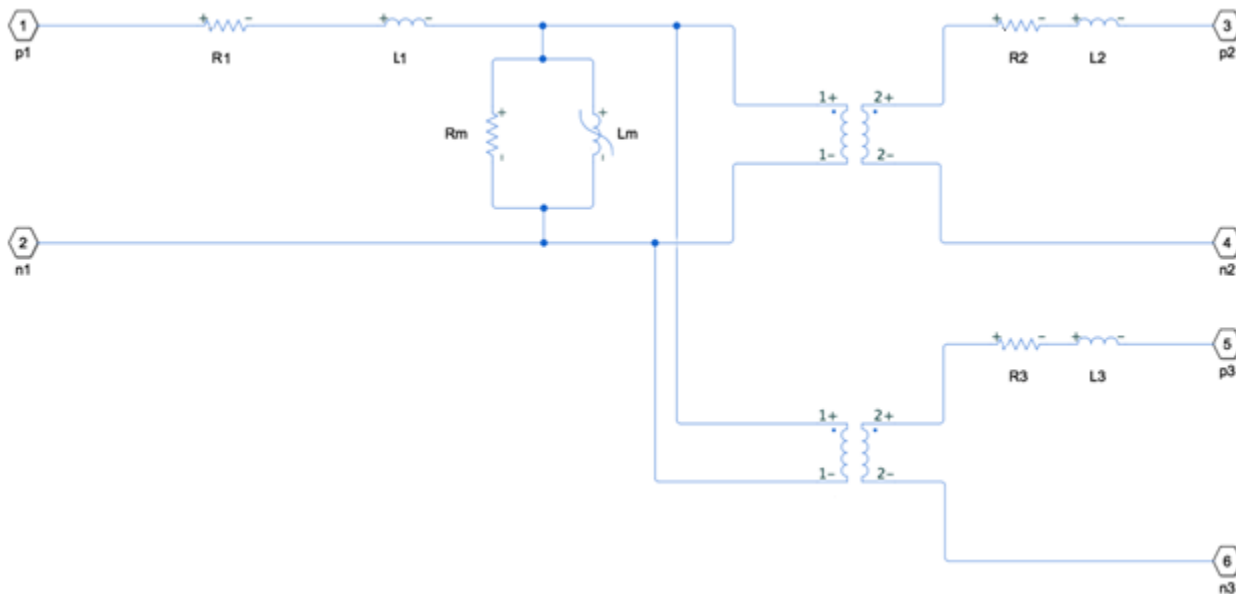
Single-phase nonlinear three-winding transformer

**Library:** Simscape / Electrical / Passive / Transformers



## Description

The Three-Winding Nonlinear Transformer block represents a single-phase, nonlinear, three-winding transformer with a nonideal core. A core may be nonideal due to its magnetic properties or dimensions. This figure shows the equivalent circuit topology:



where:

- $R1$  is the primary winding resistance.
- $L1$  is the primary leakage inductance.
- $R2$  is the first secondary winding resistance.
- $L2$  is the first secondary leakage inductance.
- $R3$  is the second secondary winding resistance.
- $L3$  is the second secondary leakage inductance.
- $Rm$  is the magnetization resistance.
- $Lm$  is the magnetization inductance.

The block provides the following parameterization options for the nonlinear magnetization inductance:

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

For more information, see the Nonlinear Inductor block reference page.

## Ports

### Conserving

#### **1+ — Line 1 positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line 1.

#### **1- — Line 1 negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line 1.

#### **2+ — Line 2 positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line 2.

#### **2- — Line 2 negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line 2.

#### **3+ — Line 3 positive terminal**

electrical

Electrical conserving port associated with the positive terminal of line 3.

#### **3- — Line 3 negative terminal**

electrical

Electrical conserving port associated with the negative terminal of line 3.

## Parameters

### Main

#### **Primary number of turns — Winding 1 primary turns number**

100 (default) | positive scalar integer

Number of turns of wire on the first primary winding of the transformer.

**First secondary number of turns — Winding 2 turns number**

200 (default) | positive scalar integer

Number of turns of wire on the first secondary winding of the transformer.

**Second secondary number of turns — Winding 3 turns number**

200 (default) | positive scalar integer

Number of turns of wire on the second secondary winding of the transformer.

**Primary winding resistance — Winding 1 resistance**

0.01 Ohm (default) | nonnegative scalar

Resistance for  $R1$ , which represents the power loss of the primary winding.

**Primary leakage inductance — Winding 1 leakage inductance**

0.0001 H (default) | nonnegative scalar

Inductance for  $L1$ , which represents the magnetic flux loss of the primary winding.

**First secondary winding resistance — Winding 2 resistance**

0.01 Ohm (default) | nonnegative scalar

Resistance for  $R2$ , which represents the power loss of the first secondary winding.

**First secondary leakage inductance — Winding 2 leakage inductance**

0.0001 H (default) | nonnegative scalar

Inductance for  $L2$ , which represents the magnetic flux loss of the first secondary winding.

**Second secondary winding resistance — Winding 3 resistance**

0.01 Ohm (default) | nonnegative scalar

Resistance for  $R3$ , which represents the power loss of the second secondary winding.

**Second secondary leakage inductance — Winding 3 leakage inductance**

0.0001 H (default) | nonnegative scalar

Inductance for  $L3$ , which represents the magnetic flux loss of the second secondary winding.

**Magnetization****Magnetization resistance — Magnetization resistance**

100 Ohm (default)

Resistance for  $Rm$ , which represents the magnetic losses in the transformer core.

**Magnetization inductance parameterized by — Nonlinear magnetization inductance parameterization**

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



Method of the nonlinear magnetization inductance parameterization:

- **Single inductance (linear)** — Provide the unsaturated inductance value.
- **Single saturation point** — Provide the values for the unsaturated and saturated inductances, as well as saturation magnetic flux. This is the default option.
- **Magnetic flux versus current characteristic** — Provide the current vector and the magnetic flux vector, and then populate the magnetic flux versus current lookup table.
- **Magnetic flux density versus field strength characteristic** — Provide the values for effective core length, cross-sectional area, magnetic field strength vector and the magnetic flux density vector, and then populate the magnetic flux density versus magnetic field strength lookup table.
- **Magnetic flux density versus field strength characteristic with hysteresis** — Define magnetic flux density as a function of both the current value and the history of the field strength by providing the number of turns, the effective core length and cross-sectional area, the initial anhysteretic B-H curve gradient, the magnetic flux density and field strength at a certain point on the B-H curve, the coefficient for the reversible magnetization, the bulk coupling coefficient, and the inter-domain coupling factor.

#### **Unsaturated inductance — Unsaturated inductance**

4e-2 H (default)

Value of inductance used when the magnetization inductance  $L_m$  is operating in its linear region.

#### **Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Single inductance (linear) or Single saturation point.

#### **Saturated inductance — Saturated inductance**

1e-2 H (default)

Value of inductance used when the magnetization inductance  $L_m$  is operating beyond its saturation point.

#### **Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Single saturation point.

#### **Saturation magnetic flux — Saturation magnetic flux**

1.6e-04 Wb (default)

Value of magnetic flux at which the magnetization inductance  $L_m$  saturates.

#### **Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Single saturation point.

#### **Current vector, $\mathbf{i}$ — Current data**

[0, .4, .8, 1.2, 1.6, 2] A (default)

Current data used to populate the magnetic flux versus current lookup table.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux versus current characteristic.

**Magnetic flux vector, phi – Magnetic flux vector**

[0, .161, .25, .284, .295, .299] .\* 1e-3 Wb (default)

Magnetic flux data used to populate the magnetic flux versus current lookup table.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux versus current characteristic.

**Magnetic field strength vector, H – Magnetic field strength vector**

[0, 200, 400, 600, 800, 1000] A/m (default)

Magnetic field strength data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic.

**Magnetic flux density vector, B – Magnetic flux density vector**

[0, .81, 1.25, 1.42, 1.48, 1.49] T (default)

Magnetic flux density data used to populate the magnetic flux density versus magnetic field strength lookup table.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic.

**Effective length – Effective length**

0.2 m (default)

Effective core length, also defined as the average distance of the magnetic path around the transformer core.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic.

**Effective cross-sectional area – Effective cross-sectional area**

2e-4 m<sup>2</sup> (default)

Effective core cross-sectional area, also defined as the average area of the magnetic path around the transformer core.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic.

**Anhyysteretic B-H gradient when H is zero – Anhyysteretic B-H gradient around zero field strength**

0.005 m\*T/A (default)

Gradient of the anhyysteretic B-H curve around zero field strength. Set this parameter to the average gradient of the positive-going and negative-going hysteresis curves.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Flux density point on anhyysteretic B-H curve – Flux density point on anhyysteretic B-H curve**

1.49 T (default)

Specify a point on the anhyysteretic curve by providing its flux density value. Picking a point at high field strength where the positive-going and negative-going hysteresis curves align is the most accurate option.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Corresponding field strength – Corresponding field strength**

1000 A/m (default)

Corresponding field strength for the point defined by the **Flux density point on anhyysteretic B-H curve** parameter.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Coefficient for reversible magnetization, c – Coefficient for reversible magnetization**

0.1 (default)

Proportion of the magnetization that is reversible. The value should be greater than zero and less than one.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Bulk coupling coefficient, K – Bulk coupling coefficient**

200 A/m (default)

Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Inter-domain coupling factor, alpha — Inter-domain coupling factor**

1e-4 (default)

Jiles-Atherton parameter that primarily affects the points at which the B-H curves intersect the zero field strength line. Typical values are in the range of 1e-4 to 1e-3.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis.

**Interpolation option — Interpolation option**

Linear (default) | Smooth

Lookup table interpolation option. Select one of the following interpolation methods:

- **Linear** — Select this option to get the best performance.
- **Smooth** — Produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

**Dependencies**

To enable this parameter, set the **Magnetization inductance parameterized by** parameter to Magnetic flux versus current characteristic or Magnetic flux density versus field strength characteristic.

**Initial Conditions****Primary leakage inductance initial current — Primary leakage inductance initial current**

0 A (default)

Value of the current through the primary leakage inductance  $L1$  at time zero.

**First secondary leakage inductance initial current — First secondary leakage inductance initial current**

0 A (default)

Value of the current through the first secondary leakage inductance  $L2$  at time zero.

**Second secondary leakage inductance initial current — Second secondary leakage inductance initial current**

0 A (default)

Value of the current through the second secondary leakage inductance  $L3$  at time zero.

**Specify magnetization inductance initial state by — Initial state specification option**

Current (default) | Magnetic flux

Initial state specification.

- **Current** — Specify the initial state of the magnetization inductance  $Lm$  by the initial current.

- **Magnetic flux** — Specify the initial state of the magnetization inductance  $L_m$  by the magnetic flux.

#### Dependencies

To enable this parameter, on the **Magnetization** setting, set the **Magnetization inductance parameterized by** parameter to:

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

#### Magnetization inductance initial current — Magnetization inductance initial current

0 A (default)

Initial current value used to calculate the value of magnetic flux within the magnetization inductance  $L_m$  at time zero. This is the current passing through the magnetization inductance  $L_m$ . Total magnetization current consists of current passing through the magnetization resistance  $R_m$  and current passing through the magnetization inductance  $L_m$ .

#### Dependencies

To enable this parameter, set the **Specify magnetization inductance initial state by** parameter to Current

#### Magnetization inductance initial magnetic flux — Magnetization inductance initial magnetic flux

0 Wb (default)

Value of the magnetic flux in the magnetization inductance  $L_m$  at time zero.

#### Dependencies

To enable this parameter, set the **Specify magnetization inductance initial state by** parameter to Magnetic flux

#### Magnetization inductance initial magnetic flux density — Magnetization inductance initial magnetic flux density

0 T (default)

Value of magnetic flux density at time zero.

#### Dependencies

To enable this parameter, on the **Magnetization** setting, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis

#### Magnetization inductance initial field strength — Magnetization inductance initial field strength

0 A/m (default)

Value of magnetic field strength at time zero.

### Dependencies

To enable this parameter, on the **Magnetization** setting, set the **Magnetization inductance parameterized by** parameter to Magnetic flux density versus field strength characteristic with hysteresis

### Parasitics

#### **Primary leakage inductance parasitic parallel conductance – Winding 1 leakage inductance parasitic parallel conductance**

1e-9 1/0hm (default) | nonnegative scalar

Small parasitic effects in parallel to the primary leakage inductance  $L1$ . A small parallel conductance may be required for the simulation of some circuit topologies.

#### **First secondary leakage inductance parasitic parallel conductance – Winding 2 leakage inductance parasitic parallel conductance**

1e-9 1/0hm (default) | nonnegative scalar

Small parasitic effects in parallel to the first secondary leakage inductance  $L2$ . A small parallel conductance may be required for the simulation of some circuit topologies.

#### **Second secondary leakage inductance parasitic parallel conductance – Winding 3 leakage inductance parasitic parallel conductance**

1e-9 1/0hm (default) | nonnegative scalar

Small parasitic effects in parallel to the second secondary leakage inductance  $L3$ . A small parallel conductance may be required for the simulation of some circuit topologies.

## Version History

Introduced in R2019b

### Extended Capabilities

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

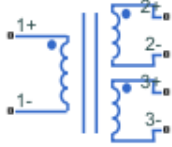
### See Also

Nonlinear Transformer

# Three-Winding Mutual Inductor

Three coupled inductors

**Library:** Simscape / Electrical / Passive / Transformers



## Description

The Three-Winding Mutual Inductor block represents a set of three coupled inductors or windings. The voltage across the three windings is

$$V_1 = L_1 \frac{dI_1}{dt} + M_{12} \frac{dI_2}{dt} + M_{13} \frac{dI_3}{dt}$$

$$V_2 = M_{12} \frac{dI_1}{dt} + L_2 \frac{dI_2}{dt} + M_{23} \frac{dI_3}{dt}$$

$$V_3 = M_{13} \frac{dI_1}{dt} + M_{23} \frac{dI_2}{dt} + L_3 \frac{dI_3}{dt}$$

where:

- $V_i$  is voltage across the  $i$ th winding.
- $I_i$  is current through the  $i$ th winding.
- $L_i$  is self inductance of the  $i$ th winding.
- $M_{ij}$  is mutual inductance of the  $i$ th and  $j$ th windings,  $M_{ij} = K_{ij}\sqrt{L_i L_j}$ .

In the preceding equations, currents are positive when flowing into the positive node of their respective inductor terminals.

When you run a simulation that includes this block, the software checks the specified parameter values to ensure that the resulting device is passive. If it is not, the software issues an error.

## Ports

### Conserving

#### 1+ — First inductor positive voltage

electrical

Electrical conserving port associated with the positive voltage of the first mutual inductor.

#### 1- — First inductor negative voltage

electrical

Electrical conserving port associated with the negative voltage of the first mutual inductor.

**2+ — Second inductor positive voltage**

electrical

Electrical conserving port associated with the positive voltage of the second mutual inductor.

**2- — Second inductor negative voltage**

electrical

Electrical conserving port associated with the negative voltage of the second mutual inductor.

**3+ — Third inductor positive voltage**

electrical

Electrical conserving port associated with the positive voltage of the third mutual inductor.

**3- — Third inductor negative voltage**

electrical

Electrical conserving port associated with the negative voltage of the third mutual inductor.

**Parameters****Inductance L1 — First winding self inductance**

0.001 H (default)

Self inductance of the first winding.

**Inductance L2 — Second winding self inductance**

0.001 H (default)

Self inductance of the second winding.

**Inductance L3 — Third winding self inductance**

0.001 H (default)

Self inductance of the third winding.

**Coefficient of coupling, K12 — Coefficient of coupling between the first and second windings**

0.9 (default)

Coefficient that defines the mutual inductance between the first and second windings. The absolute value must be between 0 and 1, exclusive.

**Coefficient of coupling, K13 — Coefficient of coupling between the first and third windings**

0.9 (default)

Coefficient that defines the mutual inductance between the first and third windings. The absolute value must be between 0 and 1, exclusive.

**Coefficient of coupling, K23 — Coefficient of coupling between the second and third windings**

0.9 (default)



Coefficient that defines the mutual inductance between the second and third windings. The absolute value must be between 0 and 1, exclusive.

### **Specify initial condition? — Initial condition option**

No (default) | Yes

Select one of the following options for specifying an initial condition:

- No — Do not specify an initial condition for the model. This is the default option.
- Yes — Specify the initial inductor currents.

### **Initial current port 1, IC1 — First winding initial current**

0 A (default)

The current flowing through the first winding at the start of the simulation.

#### **Dependencies**

This parameter is visible only when you select Yes for the **Specify initial condition** parameter.

### **Initial current port 2, IC2 — Second winding initial current**

0 A (default)

The current flowing through the second winding at the start of the simulation.

#### **Dependencies**

This parameter is visible only when you select Yes for the **Specify initial condition** parameter.

### **Initial current port 3, IC3 — Third winding initial current**

0 A (default)

The current flowing through the third winding at the start of the simulation.

#### **Dependencies**

This parameter is visible only when you select Yes for the **Specify initial condition** parameter.

## **Version History**

**Introduced in R2008a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

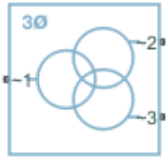
## **See Also**

Inductor | Mutual Inductor | Three-Winding Mutual Inductor | Variable Inductor

## Three-Winding Transformer (Three-Phase)

Three-phase linear nonideal wye- and delta-configurable three-winding transformer with saturation capability

**Library:** Simscape / Electrical / Passive / Transformers



### Description

The Three-Winding Transformer (Three-Phase) block represents a linear nonideal three-phase three-winding transformer that transfers electrical energy between two or more circuits through electromagnetic induction. The block includes linear winding leakage and linear core magnetization effects. You can parameterize the block impedance using per-unit values. The primary, first secondary, and second secondary winding types, delta-wye phase angle, and core types are configurable.

The configuration options for the primary and secondary winding types are:

- Wye with floating neutral — Star or T configuration with Floating Neutral (Three-Phase)
- Wye with neutral port — Star or T configuration with Neutral Port (Three-Phase)
- Wye with grounded neutral — Star or T configuration with Grounded Neutral (Three-Phase)
- Delta 1 o'clock — Mesh configuration with a lagging 30 degree phase shift relative to the voltage of a connected wye configuration
- Delta 11 o'clock — Mesh configuration with leading 30 degree phase shift relative to the voltage of a connected wye configuration

Options for the core type are:

- Three-phase five-limb
- Three-phase three-limb

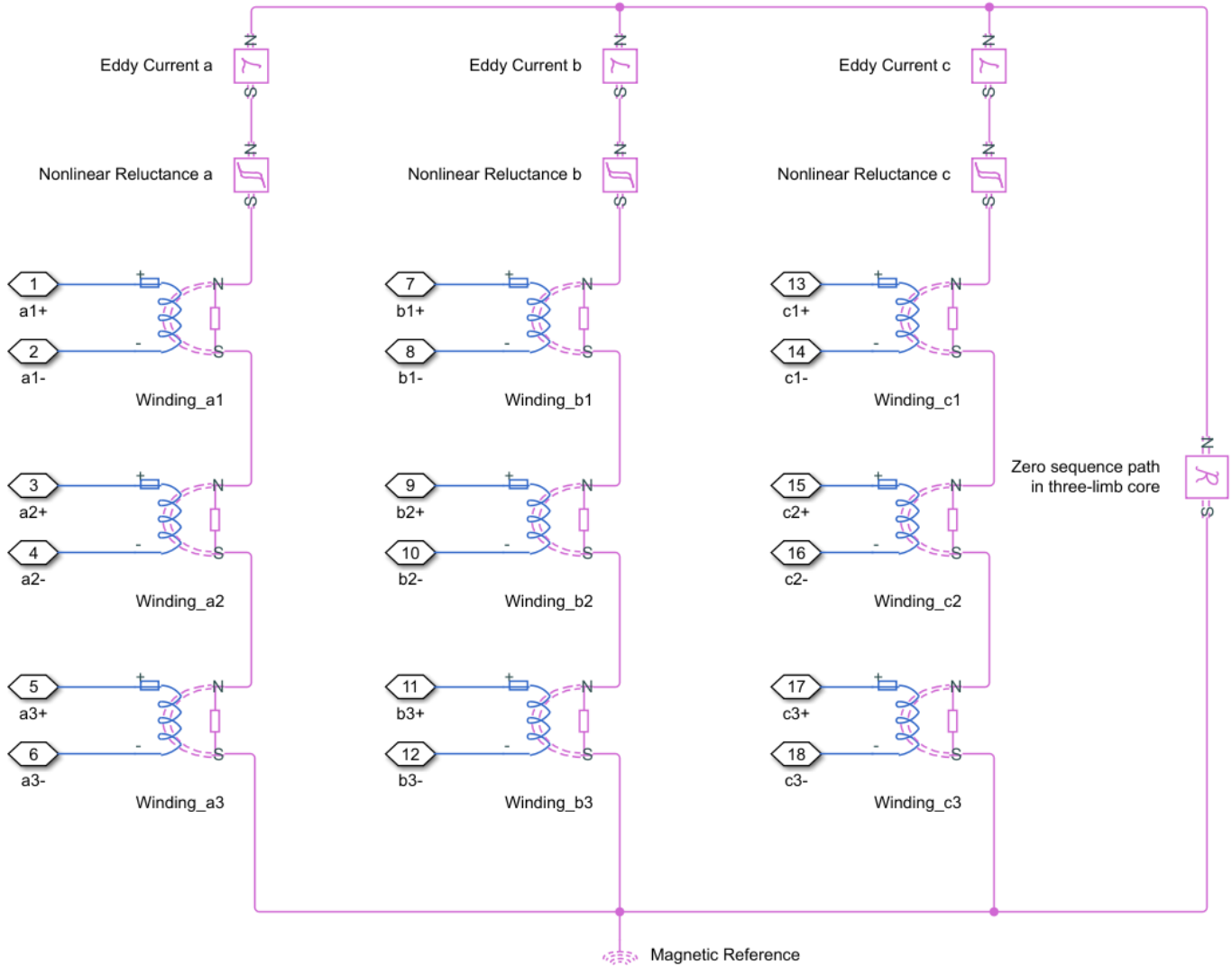
Although a three-limbs core is typically less expensive, a five-limb core offers these advantages:

- Lower impedance for the zero-sequence component of current, that is between the line and neutral, in the case of an unbalanced load
- Greater heat dissipation

### Equations

#### Three-Limb Core

This block is implemented in the magnetic domain using basic magnetic reluctances, windings, and eddy currents blocks.



It is important to determine the relation between the electrical domain parameters from the block mask and the magnetic domain parameters used in the model:

- $n_1$  is the number of the primary winding turns.
- $n_2$  is the number of the first secondary winding turns.
- $n_3$  is the number of the second secondary winding turns.
- $L_m$  is the shunt magnetizing inductance.
- $L_0$  is the zero-sequence inductance.
- $L_p$  is the primary winding leakage inductance.
- $L_{s1}$  is the first secondary winding leakage inductance.
- $L_{s2}$  is the second secondary winding leakage inductance.
- $R_m$  is the shunt magnetizing resistance.
- $R$  is the magnetizing reluctance between phases.

$$R = \frac{n_1^2}{L_m}$$

- $R_0$  is the zero sequence reluctance.

$$R_0 = \frac{1}{3} \frac{n_1^2}{L_0 - L_p}$$

- $RL_1$  is the primary winding leakage reluctance.

$$RL_1 = \frac{n_1^2}{L_p}$$

- $RL_2$  is the first winding secondary leakage reluctance.

$$RL_2 = \frac{n_2^2}{L_{s1}}$$

- $RL_3$  is the second winding secondary leakage reluctance.

$$RL_3 = \frac{n_3^2}{L_{s2}}$$

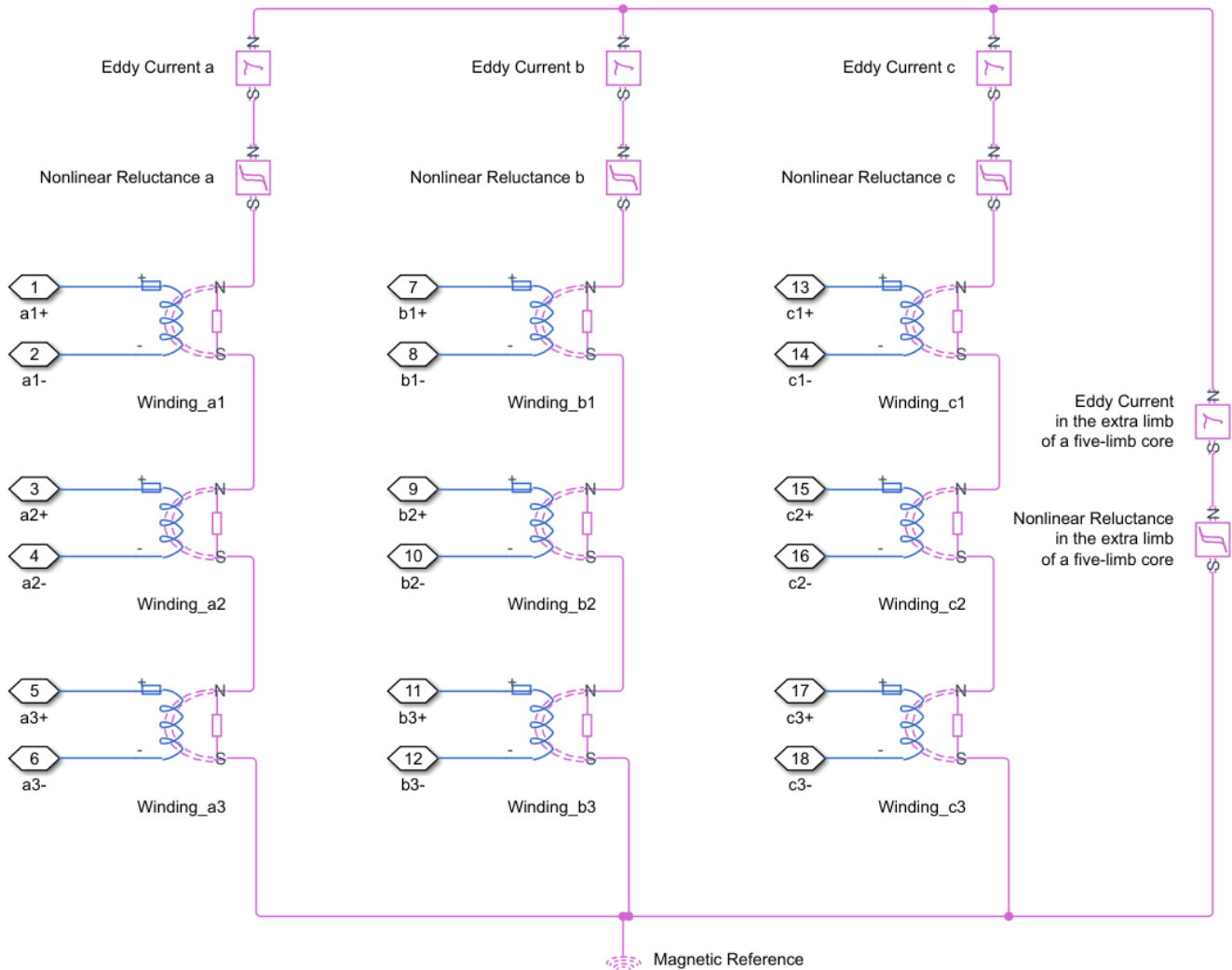
- $L_{eddy}$  is the conductance of eddy current loop

$$L_{eddy} = \frac{n_1^2}{R_m}$$

For three-winding transformers (three-phase), the coupling between different windings in each phase is identical.

### **Five-Limb Core**

In the case of a five-limb transformer, the extra magnetic flux paths provided by the extra limbs can be represented by zero sequence reluctances, which are originally designed for magnetic paths through the air in the three-limb transformer.



In a five-limb model, the magnetic reluctances from the phases to the extra limbs are supposed to be equal to the magnetic reluctances between phases.

$$R = R_0$$

Therefore we deduce:

$$R = R_0 = \frac{n_1^2}{L_m}$$

### Display Options

You can display the transformer per-unit base values in the MATLAB command window using the block context menu. To display the values, right-click the block and select **Electrical > Display Base Values**.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **~1 — Primary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a1\ b1\ c1]$ , voltage of winding 1.

#### **n1 — Primary winding neutral point**

electrical

Electrical conserving port associated with the primary winding neutral point.

### Dependencies

This port is only visible when the **Main** parameter **Winding 1 connection type** is set to **Wye** with **neutral** port.

#### **~2 — First secondary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a2\ b2\ c2]$ , voltage of the first secondary winding.

#### **n2 — First secondary winding neutral point**

electrical

Electrical conserving port associated with the first secondary winding neutral point.

### Dependencies

This port is only visible when the **Main** parameter **Winding 2 connection type** is set to **Wye** with **neutral** port.

#### **~3 — Second secondary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a3\ b3\ c3]$ , voltage of the second secondary winding.

#### **n3 — Second secondary winding neutral point**

electrical

Electrical conserving port associated with the second secondary winding neutral point.

### Dependencies

This port is only visible when the **Main** parameter **Winding 3 connection type** is set to **Wye with neutral port**.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### Rated apparent power — Apparent power at rated capacity

100e6 V\*A (default) | positive scalar

Apparent power flowing through the transformer when operating at rated capacity. The value must be greater than 0.

#### Rated electrical frequency — Connected network electrical frequency

60 Hz (default) | positive scalar

Rated or nominal frequency of the AC network to which the transformer is connected. The value must be greater than 0.

#### Primary winding connection type — Primary winding configuration

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta 1 o'clock | Delta 11 o'clock

Primary winding type.

#### Primary rated voltage — RMS line voltage applied to the primary winding

4160 V (default) | positive scalar

RMS line voltage applied to the primary winding under normal operating conditions. The value must be greater than 0.

#### First secondary connection type — First secondary winding configuration

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta 1 o'clock | Delta 11 o'clock

First secondary winding type.

**First secondary rated voltage – RMS line voltage applied to the first secondary winding**

24e3 V (default) | positive scalar

RMS line voltage applied to the first secondary winding under normal operating conditions. The value must be greater than 0.

**Second secondary connection type – Second secondary winding configuration**

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta 1 o'clock | Delta 11 o'clock

Second secondary winding type.

**Second secondary rated voltage – RMS line voltage applied to the second secondary winding**

24e3 (default) | positive scalar

RMS line voltage applied to the second secondary winding under normal operating conditions. The value must be greater than 0.

**Core type – Number of limbs**

Three-phase three-limb (default) | Three-phase five-limb

Number of limbs that comprise the magnetic circuit.

**Dependencies**

**Zero sequence reactance (pu)**, an **Impedances** parameter, is only visible when this parameter is set to Three-phase three-limb.

**Impedances****Primary winding resistance (pu) – Primary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the primary winding. The value must be greater than 0.

**First secondary winding resistance (pu) – First secondary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the first secondary winding. The value must be greater than 0.

**Second secondary winding resistance (pu) – Second secondary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the second secondary winding. The value must be greater than 0.

**Leakage reactance – Leakage reactance modeling**

Exclude (default) | Include

Whether to model magnetic flux losses.

**Primary leakage reactance (pu) – Primary winding magnetic flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the primary winding. The value must be greater than 0.



**Dependencies**

To enable this parameter, set **Leakage reactance** to Include.

**First secondary leakage reactance (pu) – First secondary winding magnetic flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the first secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to Include.

**Second secondary leakage reactance (pu) – Second secondary winding magnetic flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the second secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to Include.

**Magnetizing resistance – Magnetizing resistance modeling**

Exclude (default) | Include

Whether to model the transformer core losses.

**Shunt magnetizing resistance (pu) – Transformer core losses**

500 (default) | positive scalar

Per-unit losses in the transformer core. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing resistance** to Include.

**Magnetizing reactance – Magnetizing reactance modeling**

Exclude (default) | Include

Whether to model magnetic effects of the transformer core.

**Magnetic saturation representation – Representation of the magnetic saturation**

None (default) | Lookup table (phi versus i)

Choose if and how you want to represent the magnetic saturation.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include.

**Current vector (pu), i – Per unit vector of currents**

[0, .002, 1, 2] (default)

Per unit vector of currents. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Magnetic flux vector (pu), phi — Per unit vector of magnetic flux**

[0, 1.2, 1.5, 1.51] (default)

Per unit vector of magnetic flux. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Shunt magnetizing reactance (pu) — Transformer core magnetic effects**

500 (default) | positive scalar

Per-unit magnetic effects of the transformer core when operating in its linear region. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to None.

**Zero sequence reactance (pu) — Zero sequence reactance**

0.5 (default) | positive scalar

Per-unit zero sequence reactance. The value must be greater than or equal to the primary winding magnetic flux loss.

**Dependencies**

To enable this parameter, set **Core type**, a **Main** parameter, to Three-phase three-limb.

## Version History

Introduced in R2019a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

Earthing Transformer | Ideal Transformer | Nonlinear Transformer | Tap-Changing Transformer | Two-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer

# Thyristor

Thyristor using NPN and PNP transistors

**Library:** Simscape / Electrical / Semiconductors & Converters



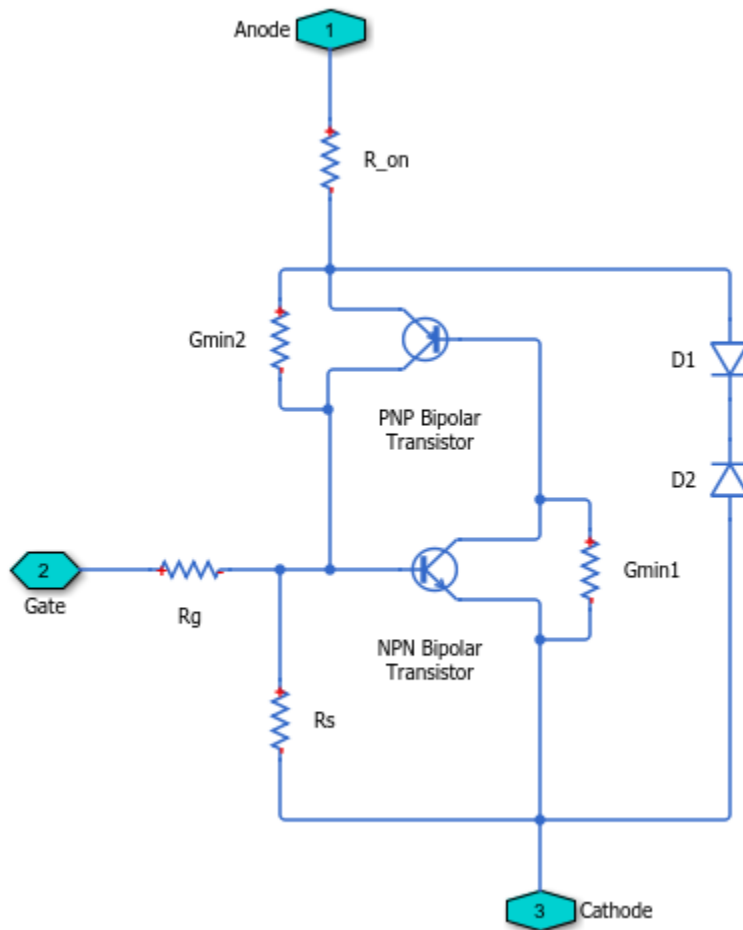
## Description

The Thyristor block provides two ways of modeling a thyristor:

- As an equivalent circuit based on NPN and PNP bipolar transistors
- By a lookup table approximation to the on-state I-V (current-voltage) curve

### Representation by Equivalent Circuit

The equivalent circuit contains a pair of NPN and PNP bipolar transistors, as shown in the following illustration.



The P-N-P-N structure of a thyristor is matched by the P-N-P and N-P-N structures of the bipolar transistors, the base of each device being connected to the collector of the other device. To ensure that this circuit behaves like a thyristor, it is necessary to pick suitable parameter values of the NPN and PNP devices, plus external resistors. For example, for the circuit to latch into the on-state, once triggered by a suitable gate current, the total gain of the two transistors must be greater than one. This model structure replicates the behavior of a thyristor in typical application circuits, while at the same time presenting a minimum number of equations to the solver, to improve simulation speed.

---

**Note** It is extremely important that you parameterize the thyristor component correctly before using it in your model. To help you do this, there are two test harnesses in the Simscape Electrical examples, Thyristor Static Behavior Validation and Thyristor Dynamic Behavior Validation. Follow the help text for these two examples, plus a datasheet for your device, to re-parameterize the thyristor component so that it replicates the required behavior. You can then copy the parameterized component into your model. Take care to model the gate drive circuitry correctly, including circuit series resistance. Connecting a controlled voltage source directly to the thyristor gate gives nonphysical results because it clamps the gate to the cathode voltage when the gate demand is zero.

---

The model captures the following thyristor behaviors:

- Off-state currents,  $I_{DRM}$  and  $I_{RRM}$ . These are typically quoted for the maximum off-state voltages  $V_{DRM}$  and  $V_{RRM}$ . It is assumed, as is the case for most thyristors, that  $I_{DRM} = I_{RRM}$  and  $V_{DRM} = V_{RRM}$ .
- The gate trigger voltage is equal to the **Corresponding gate voltage, V\_GT** parameter value when the gate current is equal to the **Gate trigger current, I\_GT** parameter value.
- The thyristor latches on when the gate current is equal to the **Gate trigger current, I\_GT**. The thyristor does not latch on until the gate current reaches this value. To ensure this is the case, you must set the **Internal shunt resistor, Rs** parameter correctly. If the resistance is too high, then the gate triggers before the gate current reaches  $i_{GT}$ . If the resistance is too small, then the gate does not trigger.

You can determine the value of the internal shunt resistor  $R_s$  by running the simulation. To see how this can be done, refer to the Thyristor Static Behavior Validation example. Alternatively, if you are using the thyristor in a circuit where there is an external resistor  $R_{GK}$  connected from gate to cathode, then the effect of  $R_s$  is usually very small, and it can be set to `inf`.

- With the thyristor in the on-state, if the gate current is removed, the thyristor stays in the on-state, provided that the load current is higher than the holding current. You do not specify the holding current directly because its value is primarily determined by other block parameters.

However, the holding current can be influenced by the **Product of NPN and PNP forward current gains** parameter on the **Advanced** tab. Reducing the gain increases the holding current.

- The on-state voltage is equal to the **On-state voltage, V\_T** parameter value when the load current is equal to the **On-state current, I\_T** parameter value. This is ensured by the  $R_{on}$  resistance value, which takes into account the voltage drop seen across the PNP and NPN devices.
- Triggering by rate of rise of off-state voltage. A rapid change in anode-cathode voltage induces a current in the base-collector capacitance terms. If this current is large enough, it triggers the thyristor into the on-state. The thyristor initialization routine calculates a suitable value for the base-collector capacitance, so that when the rate of change of voltage is equal to the **Critical rate of rise of off-state voltage, dV/dt** parameter value, the thyristor triggers on. This calculation is based on the approximation that the required current is  $v_{GT} / R_{GK}$  where  $R_{GK}$  is the gate-cathode resistance value used when quoting the critical  $dV/dt$  value.
- A nonzero gate-controlled turn-on time, primarily influenced by the **NPN device forward transit time, TF**. You either specify this parameter directly, or calculate an approximate value for TF from the turn-on time.
- A nonzero commuted turn-off time, primarily influenced by the **PNP device forward transit time, TF**. You can either specify this parameter directly, or set it to be equal to the forward transit time for the NPN transistor.

Resistors  $G_{min1}$  and  $G_{min2}$  improve numerical robustness at large forward and reverse voltages. Their values influence the off-state currents by no more than 1% at the maximum off-state forward and reverse voltages.

---

**Note** Because this block implementation includes a charge model, you must model the impedance of the circuit driving the gate to obtain representative turn-on and turn-off dynamics. Therefore, if you are simplifying the gate drive circuit by representing it as a controlled voltage source, you must include a suitable series resistor between the voltage source and the gate.

---

### Representation by Lookup Table

If using the lookup table representation, you provide tabulated values for anode-cathode current as a function of anode-cathode voltage when in the on state. The main advantages of using this option are

simulation speed and ease of parameterization. To further simplify the underlying model, this representation does not model:

- Device triggering due to rate of rise of off-state voltage
- Commuted turn-off time

The turn-on delay is represented by an input gate-cathode capacitor, the value of which is calculated so that the delay between gate voltage rising and the device starting to turn on is equal to the value specified by the **Turn-on delay time** parameter. The turn-on rise time for the load current is implemented by ramping nonlinearly between zero and the current determined by on-state current-voltage profile over a time period specified by the value of the **Turn-on rise time** parameter. Note that the resulting turn-on current profile is an approximation to an actual device.

### Thermal Port

You can expose the thermal port to model the effects of generated heat and device temperature. To expose the thermal port, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain a thermal port and does not simulate heat generation in the device.
- **Show thermal port** — The block contains a thermal port that allows you to model the heat that conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block.

For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

### Assumptions and Limitations

- This block does not model temperature-dependent effects. This block is simulated at the temperature specified by the **Measurement temperature** parameter value. All parameters must be quoted for this temperature.
- If you use the equivalent circuit representation:
  - In sensitive gate circuits (that is, where there is no external gate-cathode resistor  $R_{GK}$ ), you must set the value of the **Internal shunt resistor,  $R_s$**  parameter to ensure correct triggering. If the internal shunt resistance is too high, then the thyristor triggers for currents less than  $i_{GT}$ . If the internal shunt resistance is too low, the thyristor does not trigger for an input current of  $i_{GT}$ . For details on using simulation to determine the acceptable internal shunt resistance value, see the Thyristor Static Behavior Validation example.
  - Triggering by exceeding the breakover voltage is not modeled.

- Numerically the thyristor can be demanding to simulate, given the very small gate currents in comparison to the load current, and also the very steep current gradients during switching. However, for most typical thyristor-based circuits, you can use the default simulation parameters. In some cases you may need to tighten the **Absolute Tolerance** and **Relative Tolerance** parameters on the **Solver** tab of the Configuration Parameters dialog box, to ensure convergence. In such cases, changing the default value of **Absolute Tolerance** from `auto` to `1e-4` or `1e-5` is usually sufficient, because it prevents adaptive changing of this parameter during simulation.
- The leakage currents are approximated by the diodes D1 and D2, as shown in the equivalent circuit. This approach assumes that the leakage via the two transistors is small in comparison. This assumption is not valid for values of  $v_{GT}$  that are significantly smaller than the typical forward voltage drop of 0.6 V.
- If you use the lookup table representation:
  - Triggering by exceeding the breakover voltage or by rate of change of off-state voltage is not modeled.
  - Commutated turn-off time is not modeled. Check that your circuit does not violate the stated commutated turn-off time for the thyristor.
  - When you specify a turn-on rise time, the resulting current-time profile is an approximation.

## Ports

### Conserving

#### G — Gate port

electrical

Electrical conserving port associated with the gate.

#### A — Anode port

electrical

Electrical conserving port associated with the anode.

#### K — Cathode port

electrical

Electrical conserving port associated with the cathode.

#### H — Thermal port

thermal

Thermal conserving port.

### Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option — Whether to enable thermal port

No thermal port (default) | Show thermal port

Whether to enable the thermal port of the block and model the effects of generated heat and device temperature.

**Main****I-V characteristics defined by — Thyristor representation**

Fundamental nonlinear equations (default) | Lookup table

Represent the thyristor either by using an equivalent circuit based on NPN and PNP bipolar transistors or use a lookup table approximation to the on-state I-V curve.

**On-state voltage,  $V_T$  — On-state Anode-Cathode static voltage**

1.2 V (default)

The anode-cathode static voltage drop when in the on-state, and the current flowing is equal to the on-state current  $I_T$ .

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter.

**On-state current,  $I_T$  — On-state static load current**

1 A (default)

Static load (or equivalently anode) current that flows when the anode-cathode voltage is equal to the on-state voltage  $V_T$ .

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter.

**Vector of on-state voltages,  $V_T$  — Vector of on-state voltages**

[.75, 1, 1.25, 1.5, 1.75, 2, 2.25] V (default)

The vector of on-state voltages, to be used for table lookup. The vector values must be strictly increasing, and the first value must be greater than zero. The values can be nonuniformly spaced.

**Dependencies**

This parameter is visible only when you select Lookup table for the **I-V characteristics defined by** parameter.

**Vector of corresponding currents,  $I_T$  — Vector of corresponding currents**

[.015, .22, .75, 1.4, 2, 2.75, 3.45] A (default)

The vector of currents corresponding to the on-state voltages vector values, to be used for 1D table lookup. The two vectors must be of the same size.

**Dependencies**

This parameter is visible only when you select Lookup table for the **I-V characteristics defined by** parameter.

**Off-state current,  $I_{DRM}$  — Off-state anode current**

0.01 mA (default)



The off-state anode current  $I_{DRM}$  that flows when the anode-cathode voltage is equal to the off-state voltage  $V_{DRM}$ .

**Corresponding off-state voltage, V\_DRM — Off-state anode-cathode voltage**

400 V (default)

The anode-cathode voltage  $V_{DRM}$  applied with the thyristor in the off-state when quoting the off-state current  $I_{DRM}$

**Holding current — Holding current**

1 mA (default)

This is the minimum current for which the thyristor remains in the on-state. For the Lookup table option, the latching current is assumed equal to the holding current, so this is also the maximum current for which the thyristor remains in the off-state.

**Dependencies**

This parameter is visible only when you select Lookup table for the **I-V characteristics defined by** parameter.

**Measurement temperature — Device simulation temperature**

25 degC (default)

Device simulation temperature. You must specify all block parameter values for this temperature.

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter.

**Gate Triggering**

**Gate trigger current, I\_GT — Critical gate current**

3 uA (default)

Critical gate current  $i_{GT}$  required to turn the transistor on, resulting in a gate voltage equal to the corresponding gate voltage  $v_{GT}$ . You must set the value of the **Internal shunt resistor, Rs** parameter on the **Advanced** tab to ensure that the gate triggers at  $i_{GT}$ , and not for currents less than  $i_{GT}$ .

**Corresponding gate voltage, V\_GT — Gate-cathode voltage**

0.6 V (default)

Gate-cathode voltage  $v_{GT}$  when the gate current is equal to the gate trigger current  $i_{GT}$ .

**Test voltage, V\_D — Test supply voltage**

12 V (default)

Supply voltage used when quoting values for  $v_{GT}$  and  $i_{GT}$ .

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter.

**Test load resistor — Test load resistor**

120 Ohm (default)

Load resistor used when quoting values for  $v_{GT}$  and  $i_{GT}$ .

**Dependencies**

This parameter is visible only when you select **Fundamental nonlinear equations for the I-V characteristics defined by** parameter.

**dV/dt Triggering****Critical rate of rise of off-state voltage, dV/dt — Critical rate of rise of off-state voltage**

150 V/us (default)

If the anode-cathode voltage increases faster than this rate, the thyristor will experience a parasitic turn-on due to capacitive effects.

**Dependencies**

This parameter is visible only when you select **Fundamental nonlinear equations for the I-V characteristics defined by** parameter.

**Test gate-cathode resistor, R\_GK — Test gate-cathode resistor**

1000 Ohm (default)

Gate-cathode resistor used when quoting the critical rate of rise off off-state voltage.

**Dependencies**

This parameter is visible only when you select **Fundamental nonlinear equations for the I-V characteristics defined by** parameter.

**Time Constants****NPN device forward transit time parameterization — NPN device forward transit time parameterization**

Derive approximate value from gate-controlled turn-on time (default) | Specify directly

Select one of the following options:

- **Derive approximate value from gate-controlled turn-on time** — The block calculates the NPN device forward transit time based on the values for the gate-controlled turn-on time and corresponding gate current that you specify.
- **Specify directly** — Provide the value directly by using the **NPN device forward transit time** parameter.

**Dependencies**

This parameter is visible only when you select **Fundamental nonlinear equations for the I-V characteristics defined by** parameter.

**Gate-controlled turn-on time — Gate-controlled turn-on time**

2 us (default)

Time for the gate to turn from the off to the on state when a gate current is applied.

**Corresponding gate current — Test gate current**

10 mA (default)

The gate current used when quoting the gate-controlled turn-on time. The gate current and turn-on time are used to calculate an approximate value for the NPN device forward transit time on the assumption that all of the input charge is used to raise the gate voltage to the corresponding gate voltage  $v_{GT}$ . The default value is 10 mA.

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter and Derive approximate value from gate-controlled turn-on time for **NPN device forward transit time parameterization**.

**NPN device forward transit time — Mean forward transit time**

0.3 us (default)

Mean time for the minority carriers to cross the base region from the emitter to the collector of the NPN device [1].

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter and Specify directly for **NPN device forward transit time parameterization**.

**PNP device forward transit time parameterization — PNP device forward transit time parameterization**

Set equal to NPN device forward transit time (default) | Specify directly

Select one of the following options:

- Set equal to NPN device forward transit time — The block uses the NPN device forward transit time value.
- Specify directly — Provide the value directly by using the **PNP device forward transit time** parameter.

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter.

**PNP device forward transit time — Mean forward transit time**

0.3 us (default)

Mean time for the minority carriers to cross the base region from the emitter to the collector of the PNP device [1].

**Dependencies**

This parameter is visible only when you select Fundamental nonlinear equations for the **I-V characteristics defined by** parameter and Specify directly for **PNP device forward transit time parameterization**.

**Turn-on delay time — Time delay**

0 s (default)

Time delay before the device starts to turn on following a step in current on the gate from zero to the value specified by the **Gate current for turn-on delay time** parameter.

**Dependencies**

This parameter is visible only when you select `Lookup` table for the **I-V characteristics defined by**.

**Gate current for turn-on delay time — Gate current for turn-on delay time**

1 mA (default)

The gate current used when measuring the turn-on delay time.

**Dependencies**

This parameter is visible only when you select `Lookup` table for the **I-V characteristics defined by**.

**Turn-on rise time — Turn-on rise time**

0 s (default)

Time it takes for the thyristor to turn on fully following the turn-on delay event.

**Dependencies**

This parameter is visible only when you select `Lookup` table for the **I-V characteristics defined by**.

**Advanced****Internal shunt resistor,  $R_s$  — Gate-Cathode shunt resistance**

87 kOhm (default)

Gate-cathode shunt resistance. It is important to set this parameter value to ensure that the gate triggers at  $i_{GT}$ , and not for currents less than  $i_{GT}$ . For details, see the Thyristor Static Behavior Validation example. If you are using the thyristor in a circuit where there is an external gate-cathode resistor  $R_{GK}$ , then usually the effect of  $R_s$  is small, and it can be set to `inf`.

**Dependencies**

This parameter is visible only when you select `Fundamental nonlinear` equations for the **I-V characteristics defined by**.

**Internal series gate resistor,  $R_g$  — Resistance associated with gate connection**

10 Ohm (default)

Resistance associated with the gate connection. A typical value is of the order of a few ohms, and its impact on static and dynamic characteristics is small. Therefore, its precise value is not important, but its presence helps avoid numerical simulation issues if the gate is driven directly by a voltage source. You can specify any positive value.

**Product of NPN and PNP forward current gains — Product of NPN and PNP forward current gains**

10 (default)

This is the product of the NPN forward gain  $BF_{NPN}$  and the PNP forward gain  $BF_{PNP}$ . The value must be greater than one for latching to occur. The smaller the value, the larger the latching current

becomes. However, latching current is primarily set by other block parameters, and the total gain has only a small effect.

### **Dependencies**

This parameter is visible only when you select **Fundamental nonlinear equations** for the **I-V characteristics defined by**.

## **Version History**

### **Introduced in R2010b**

### **Thermal network simlog paths**

*Behavior changed in R2019b*

From R2019b forward, it is possible to model the thermal network of this block either internally or externally.

As a result, the simlog paths corresponding to the thermal network have changed. If you have a script that programmatically access the simlog, you have to manually update it to reflect these changes.

## **References**

[1] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE*. 2nd Edition, McGraw-Hill, 1993.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

NPN Bipolar Transistor | PNP Bipolar Transistor | Thyristor (Piecewise Linear)

# Thyristor (Piecewise Linear)

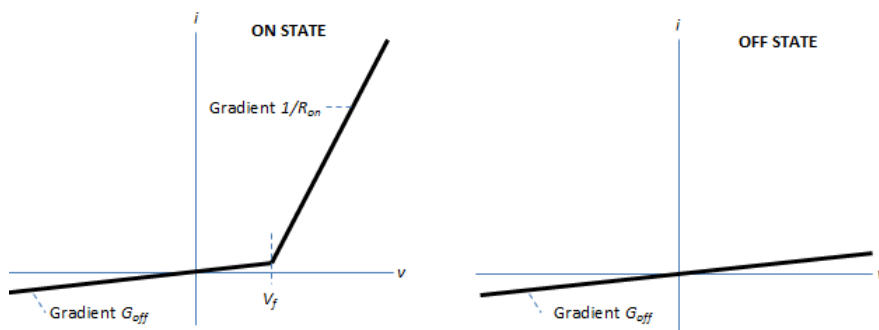
Thyristor

**Library:** Simscape / Electrical / Semiconductors & Converters



## Description

The Thyristor (Piecewise Linear) block models a thyristor. The I-V characteristic for a thyristor is such that the thyristor turns on if the gate-cathode voltage exceeds the specified gate trigger voltage. The device turns off if the load current falls below the specified holding-current value.



To define the I-V characteristic of the thyristor, set the **On-state behaviour and switching losses** parameter to either Specify constant values or Tabulate with temperature and current. The Tabulate with temperature and current option is available only if you expose the thermal port of the block.

In the on state, the anode-cathode path behaves like a linear diode with forward-voltage drop,  $V_f$ , and on-resistance,  $R_{on}$ . However, if you expose the thermal port of the block and parameterize the device using tabulated I-V data, the tabulated resistance is a function of the temperature and current.

In the off state, the anode-cathode path behaves like a linear resistor with a low off-state conductance,  $G_{off}$ .

The defining Simscape equations for the block are:

```

if (v > Vf)&&((G>Vgt) || (i>Ih))
    i == (v - Vf*(1-Ron*Goff))/Ron;
else
    i == v*Goff;
end

```

where:

- $v$  is the anode-cathode voltage.

- $V_f$  is the forward voltage.
- $G$  is the gate voltage.
- $V_{gt}$  is the gate trigger voltage.
- $i$  is the anode-cathode current.
- $I_h$  is the holding current.
- $R_{on}$  is the on-state resistance.
- $G_{off}$  is the off-state conductance.

Using the Integral Diode tab of the block dialog box, you can include an integral cathode-anode diode. An integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

The table shows you how to set the **Integral protection diode** parameter based on your goals.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	Protection diode with no dynamics	The block includes an integral copy of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.
Precisely specify reverse-mode charge dynamics.	Protection diode with charge dynamics	The block includes an integral copy of the dynamic model of the Diode block. To parameterize the internal Diode block, use the <b>Protection</b> parameters.

### Model Gate Port and Thermal Effects

You can choose between physical or electrical ports to control the gate terminal and expose the thermal port to model the heat that switching events and conduction losses generate. To choose the gate control port, set the **Gate control port** parameter to `PS` or `Electrical`. To expose the thermal port, set the **Modeling option** parameter to either `No thermal port` or `Show thermal port`.

For more information about using thermal ports, see “Simulating Thermal Effects in Semiconductors”.

### Thermal Losses

Switching losses are one of the main sources of thermal loss in semiconductors. During each on switching transition, the thyristor parasitics store and then dissipate energy.

Switching losses depend on the off-state voltage and the on-state current. When the switching device is turned on, the power losses depend on the initial off-state voltage across the device and the final on-state current once the device is fully in its on state. When the switching device is turned off, the power loss is defined by the **Natural commutation rectification loss** parameter value. This is the rectification loss applied at the point that the device switches off due to the current falling below the holding current. This loss is a fixed value and it is not scaled by the off-state voltage or the on-state current.

In this block, switching losses are applied by stepping up the junction temperature with a value equal to the switching loss divided by the total thermal mass at the junction. The **Switch-on loss,  $E_{on}(T_j, I_{ak})$**  parameter value sets the sizes of the switching losses and they are either fixed or dependent on junction temperature and drain-source current. In both cases, losses are scaled by the off-state voltage prior to the latest device turn-on event.

---

**Note** As the final current after a switching event is not known during the simulation, the block records the on-state current once the current is greater than the holding current for a time longer than the value specified in the **Wait time before switch-on current measurement**. Similarly, the block records the off-state voltage at the point that the device is commanded on. For this reason, the simlog does not report the switching losses to the thermal network until one switching cycle later.

For all ideal switching devices, the switching losses are reported in the simlog as `lastTurnOffLoss` and `lastTurnOnLoss` (for the thyristor, this is the **Natural commutation rectification loss**) and recorded as a pulse with amplitude equal to the energy loss. If you use a script to sum the total losses over a defined simulation period, you must sum the pulse values at each pulse rising edge. Alternatively, you can use the `ee_getPowerLossSummary` and `ee_getPowerLossTimeSeries` functions to extract conduction and switching losses from logged data.

Note that the `power_dissipated` variable in the simlog does not include switching losses as they are modeled as instantaneous events. The `power_dissipated` variable therefore just reports instantaneous on-state losses.

---

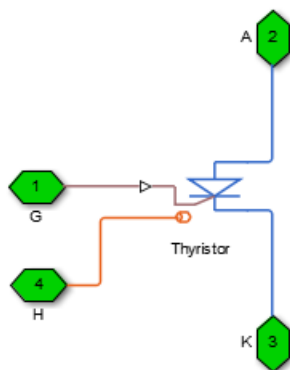
## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

The figure shows the block port names.





## Conserving

### G – Gate terminal

electrical

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

### A – Anode terminal

electrical

Electrical conserving port associated with the anode terminal.

### K – Cathode terminal

electrical

Electrical conserving port associated with the cathode terminal.

### H – Thermal port

thermal

Thermal conserving port.

## Dependencies

To enable this port, set **Modeling option** to Show thermal port.

## Parameters

### Modeling option – Thermal port visibility

No thermal port (default) | Show thermal port

Whether to enable the thermal port.

## Main

This table shows how the visibility of **Main** parameters depends on how you configure the **Modeling option** and **On-state behavior and switching losses** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

## Main Parameter Dependencies

Parameters and Options		
<b>Modeling option</b>		
<i>No thermal port</i>	<i>Show thermal port</i>	
Gate-control port	Gate-control port	
Forward voltage, Vf	Gate trigger voltage, Vgt	
On-state resistance	Holding current	
Off-state conductance	<b>On-state behaviour and switching losses</b>	
	<i>Specify constant values</i>	<i>Tabulate with temperature and current</i>
Gate trigger voltage, Vgt	Forward voltage, Vf	On-state voltage, Vak(Tj,Iak)
Holding current	On-state resistance	Temperature vector, Tj
	Off-state conductance	Anode-cathode current vector, Iak
		Off-state conductance

### Gate-control port – Whether to specify physical or electrical control port

PS (default) | Electrical

Whether to specify physical or electrical control port for the switch gate.

### On-state behaviour and switching losses – On-state current for switching loss data

Specify constant values (default) | Tabulate with temperature and current

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- **Specify constant values** — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data. This is the default parameterization method.
- **Tabulate with temperature and current** — Use vectors to specify the output current, switch-on loss, switch-off loss, and temperature data.

### Dependencies

See the Main Parameter Dependencies table.

### Forward voltage, Vf – Forward voltage

0.8 V (default)

Forward voltage at which the device turns on.

### Dependencies

See the Main Parameter Dependencies table.

### On-state resistance – On-state resistance

0.001 Ohm (default)

Anode-cathode resistance when the device is on.

**Dependencies**

See the Main Parameter Dependencies table.

**Off-state conductance — Off-state conductance**

1e-5 1/Ohm (default)

Anode-cathode conductance when the device is off. The value must be less than 1/R, where R is the value of **On-state resistance**.

**Dependencies**

See the Main Parameter Dependencies table.

**Gate trigger voltage, Vgt — Gate trigger voltage**

6 V (default)

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Main Parameter Dependencies table.

**Holding current — Holding current**

1 A (default)

Current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Main Parameter Dependencies table.

**On-state voltage, Vak(Tj, Iak) — On-state voltage**

[0, .1, .6, .8, 1, 1.3, 1.6, 2, 2.4; 0, .1, .7, 1, 1.2, 1.5, 1.9, 2.4, 2.8] V (default)

Voltage drop across the device while it is in a triggered conductive state. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Temperature vector, Tj — Temperature vector**

[298.15, 398.15] K (default)

Temperature values at which the on-state voltage is specified. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Anode-cathode current vector,  $I_{ak}$  — Anode-cathode current vector**

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Anode-cathode currents for which the on-state voltage is defined. The first element must be zero. Specify this parameter using a vector quantity.

**Dependencies**

See the Main Parameter Dependencies table.

**Switching Losses**

To enable these parameters, set **Modeling option** to Show thermal port.

**Switch-on loss — Switch-on loss**

22.86e-3 J (default)

Energy dissipated during a single switch-on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a scalar quantity.

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Natural commutation rectification loss — Natural commutation rectification loss**

10e-3 J (default)

Rectification loss applied at the point that the block switches off when the current falls below the **Holding current**. Specify this parameter using a scalar quantity.

**Off-state voltage for switching loss data — Off-state voltage for losses data**

300 V (default)

The output voltage of the device during the off state. This is the blocking voltage at which the switch-on loss and switch-off loss data are defined.

**On-state current for switching loss data — Output current**

600 A (default)

Output currents for which the switch-on loss, switch-off loss, and on-state voltage are defined. The first element must be zero. Specify this parameter using a scalar quantity.

---

**Note** This parameter is measured at the point that the gate voltage falls below the **Gate trigger voltage,  $V_{gt}$** . The turn-on pulse is longer than the time it takes the current to reach its maximum value.

---

**Dependencies**

To enable this parameter, set **On-state behavior and switching losses** to Specify constant values.

**Switch-on loss,  $E_{on}(T_j, I_{ak})$  — Switch-on loss**

[0, .0024, .024, .12, .2, .48, 1.04, 2.16, 3.24; 0, .003, .03, .15, .25, .6, 1.3, 2.7, 4.05] \* 1e-3 J (default)

Energy dissipated during a single switch on event. This parameter is defined as a function of temperature and final on-state output current. Specify this parameter using a vector quantity.

#### Dependencies

To enable this parameter, set **On-state behavior and switching losses** to `Tabulate` with temperature and current.

#### Temperature vector for switching losses, $T_j$ — Temperature vector for switching losses

[298.15, 398.15] K (default)

Temperature values at which the switch-on loss and switch-off loss are specified. Specify this parameter using a vector quantity.

#### Dependencies

To enable this parameter, set **On-state behavior and switching losses** to `Tabulate` with temperature and current.

#### Anode-cathode current vector for switching losses, $I_{ak}$ — Anode-cathode current vector for switching losses

[0, .1, 1, 5, 10, 20, 40, 70, 100] A (default)

Anode-cathode currents for which the switch-on loss and switch-off-loss are defined. The first element must be zero. Specify this parameter using a vector quantity.

#### Dependencies

To enable this parameter, set **On-state behavior and switching losses** to `Tabulate` with temperature and current.

#### Wait time before switch-on current measurement — Time before on-state current is measured

1e-4 s (default)

Time to wait before recording the on-state current

#### Integral Diode

##### Integral protection diode — Integral protection diode

None (default) | Protection diode with no dynamics | Protection diode with charge dynamics

Block integral protection diode. The default value is `None`.

The diodes you can select are:

- Protection diode with no dynamics
- Protection diode with charge dynamics

##### Diode model — Diode model

Piecewise Linear (default) | Tabulated I-V curve

Select one of these diode models:

- **Piecewise Linear** — Use a piecewise linear model for the diode, as described in “Piecewise Linear Diode” on page 1-448. This is the default method.
- **Tabulated I-V curve** — Use tabulated forward bias I-V data plus fixed reverse bias off conductance.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to **Protection diode with no dynamics** or **Protection diode with charge dynamics**.

**Table type — Tabulated function**

Table in  $I_f(T_j, V_f)$  form (default) | Table in  $V_f(T_j, I_f)$  form

Whether to tabulate the current as a function of temperature and voltage or the voltage as a function of temperature and current.

**Dependencies**

This parameter is visible only when the thermal port is exposed and the **Integral protection diode** parameter is set to **Protection diode with no dynamics** or **Protection diode with charge dynamics** and **Diode model** is set to **Tabulated I-V curve**.

**Forward voltage — Forward voltage**

0.8 V (default)

Minimum voltage required across the + and - block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to **Protection diode with no dynamics** or **Protection diode with charge dynamics**.
- If the thermal port is exposed, set **Integral protection diode** to **Protection diode with no dynamics** or **Protection diode with charge dynamics** and **Diode model** to **Piecewise linear**.

**On resistance — On resistance**

0.001 Ohm (default)

Rate of change of voltage versus current above the **Forward voltage**.

**Dependencies**

To enable this parameter:

- If the thermal port is hidden, set **Integral protection diode** to **Protection diode with no dynamics** or **Protection diode with charge dynamics**.
- If the thermal port is exposed, set **Integral protection diode** to **Protection diode with no dynamics** or **Protection diode with charge dynamics** and **Diode model** to **Piecewise linear**.

**Forward currents,  $I_f(T_j, V_f)$  – Vector of forward currents**

[.07, .12, .19, 1.75, 4.24, 7.32, 11.2; .16, .3, .72, 2.14, 4.02, 6.35, 9.12] A (default) | nonnegative vector

Forward currents. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

**Junction temperatures,  $T_j$  – Vector of junction temperatures**

[25, 125] degC (default)

Vector of junction temperatures. This parameter must be a vector of at least two elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve.

**Forward voltages,  $V_f$  – Vector of forward voltages**

[.5, .7, .9, 1.3, 1.7, 2.1, 2.5] V (default)

Vector of forward voltages. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $I_f(T_j, V_f)$  form.

**Forward voltages,  $V_f(T_j, I_f)$  – Vector of forward voltages**

[.9, 1.15, 1.25, 1.5, 1.75, 2.17, 2.6, 2.85; .58, .68, .75, 1.1, 1.38, 1.77, 2.27, 2.7] V (default) | nonnegative vector

Forward voltages. This parameter must be a vector of at least three nonnegative elements.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

**Forward currents,  $I_f$  – Vector of forward currents**

[.1, .2, .5, 1, 2, 4, 7, 10] A (default)

Vector of forward currents. This parameter must be a vector of at least three nonnegative values.

**Dependencies**

To enable this parameter, expose the thermal port and set **Diode model** to Tabulated I-V curve and **Table type** to Table in  $V_f(T_j, I_f)$  form.

**Off conductance – Off conductance**

1e-5 1/Ohm (default)

Conductance of the reverse-biased diode.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with no dynamics or Protection diode with charge dynamics.

**Junction capacitance — Junction capacitance**

50e-9 F (default)

Diode junction capacitance.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Peak reverse current,  $i_{RM}$  — Peak reverse current**

-235 A (default) | negative scalar

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is -235 A.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Initial forward current when measuring  $i_{RM}$  — Initial forward current when measuring  $i_{RM}$** 

300 A (default) | positive scalar

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Rate of change of current when measuring  $i_{RM}$  — Rate of change of current when measuring  $i_{RM}$** 

-50A/ $\mu$ s (default) | negative scalar

Rate of change of current when measuring peak reverse current. This value must be less than zero.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics.

**Reverse recovery time parameterization — Reverse recovery time parameterization**

Specify reverse recovery time directly (default) | Specify stretch factor | Specify reverse recovery charge

Determines how you specify reverse recovery time in the block. The default value is Specify reverse recovery time directly.



If you select **Specify stretch factor** or **Specify reverse recovery charge**, you specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates TM and Tau” on page 1-452.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics**.

#### Reverse recovery time, $t_{rr}$ — Reverse recovery time

15  $\mu$ s (default)

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10% of the peak reverse current. The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify reverse recovery time directly**.

#### Reverse recovery time stretch factor — Reverse recovery time stretch factor

3 (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . This value must be greater than 1. Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

#### Dependencies

This parameter is visible only when the **Integral protection diode** parameter is set to **Protection diode with charge dynamics** and the **Reverse recovery time parameterization** parameter is set to **Specify stretch factor**.

#### Reverse recovery charge, $Q_{rr}$ — Reverse recovery charge

1500  $\mu$ As (default)

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current,  $i_{RM}$** .
- $a$  is the value specified for **Rate of change of current when measuring  $i_{RM}$** .

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery charge.

**Diode turn-off voltage when measuring Erec — Diode turn-off voltage when measuring Erec**

-600 V (default)

Voltage between the diode in steady-state.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Stray inductance when measuring Erec — Stray inductance when measuring Erec**

150 nH (default)

Total unintended inductance in the measurement circuit. The block uses this value to calculate **Reverse recovery energy, Erec**.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

**Reverse recovery energy, Erec — Reverse recovery energy**

0.03 J (default)

Total switching losses due to the diode reverse recovery.

**Dependencies**

This parameter is visible only when the **Integral protection diode** parameter is set to Protection diode with charge dynamics and the **Reverse recovery time parameterization** parameter is set to Specify reverse recovery energy.

For more information on these parameters, see Diode.

**Thermal Port**

Use the thermal port to simulate the effects of generated heat and device temperature. For more information on using thermal ports and on the **Thermal Port** parameters, see “Simulating Thermal Effects in Semiconductors”.

## Version History

**Introduced in R2013b****Update on switching losses and thermal modelling options**

*Behavior changed in R2020b*

From R2020b forward, the Thyristor (Piecewise Linear) block has improved losses and thermal modelling options.

As a result of these changes:

- Electrical and thermal on-state losses are now always identical. The **Thermal loss dependent on** parameter and its options, **Voltage** and **current** and **Voltage, current, and temperature**, have been renamed to **On-state behavior and switching losses**, **Specify constant values**, and **Tabulate with temperature and current**:
  - If you selected **Voltage** and **current** for **Thermal loss dependent on**, then the electrical on-state losses are unchanged and their values are determined using the on-state resistance. However, the thermal on-state losses are now also determined by the on-state resistance. Prior to R2020b, the thermal on-state losses were defined by the product of the **On-state voltage** and **Output current, Iout** parameters.
  - If you selected **Voltage, current, and temperature** for **Thermal loss dependent on**, then the thermal on-state losses are unchanged and the **On-state voltage, Vak(Tj,Iak)** parameter sets their values. However, the electrical on-state losses are now equal to the thermal on-state losses. Prior to R2020b, the electrical on-state losses were defined by the value of the on-state resistance.
- The **On-state voltage** and the switch-off parameters are no longer used.

### Removal of energy dissipation time constant parameter

*Behavior changed in R2021a*

From R2021a forward, the **Energy dissipation time constant** parameter of the Thyristor (Piecewise Linear) block is no longer used. A step in junction temperature now reflects the switching losses. If your model contains a thermal mass directly connected to this block thermal port, remove it and model the thermal mass inside the component itself.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Diode | GTO | IGBT (Ideal, Switching) | Ideal Semiconductor Switch | MOSFET (Ideal, Switching)

### Topics

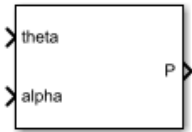
“Simulating Thermal Effects in Semiconductors”

“Switch Between Physical Signal and Electrical Ports”

## Thyristor 6-Pulse Generator

Generate thyristor 6-pulse waveform in single-pulsing mode

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



### Description

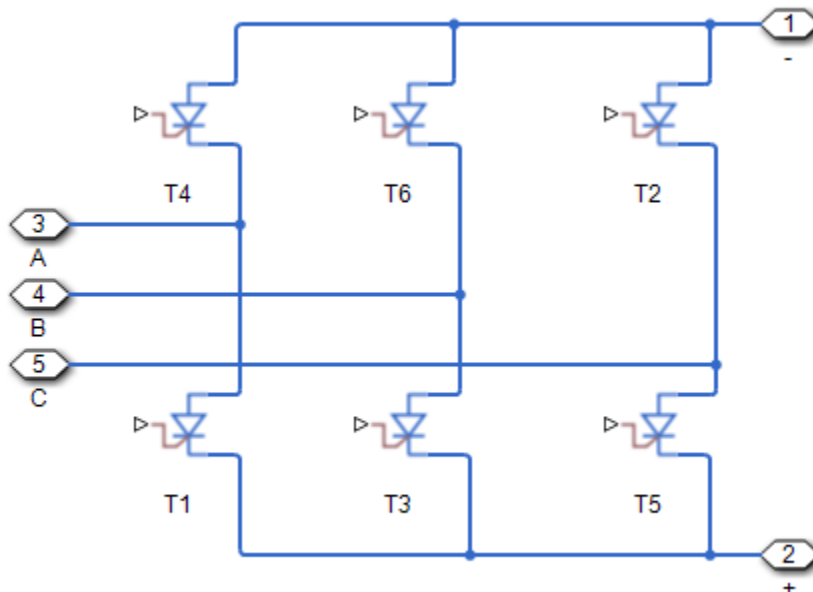
The Thyristor 6-Pulse Generator block implements a thyristor 6-pulse waveform generator in single-pulsing mode.

You can use this block to perform phase-controlled AC-to-DC conversion by:

- Measuring the synchronization angle of the AC signal with a phase-locked loop
- Controlling a thyristor converter network with the pulses generated by this block

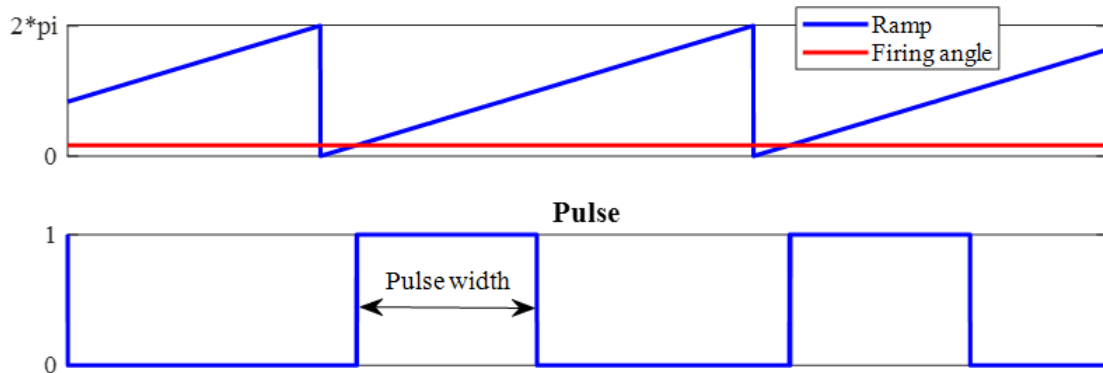
### Model

The figure shows the equivalent circuit for the Thyristor 6-Pulse Generator.



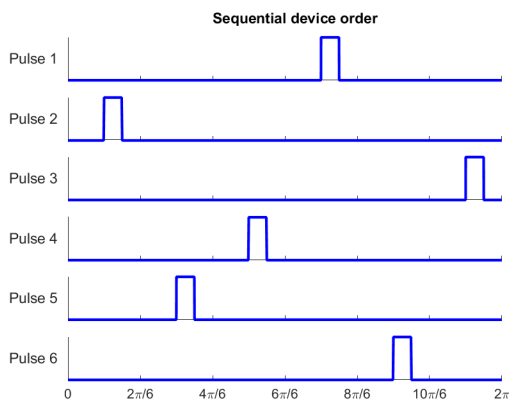
Based on the synchronization angle,  $\theta$ , and the firing angle,  $\alpha$ , the block internally generates six ramps, one for each of the pulse elements in its output vector.

The block generates a pulse at one of the outputs when the associated ramp meets or crosses the specified firing angle in the upward direction. This figure shows such a pulse generation mechanism for one of the outputs.

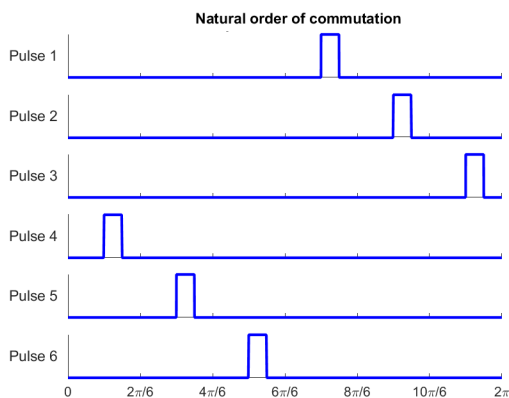


Set the pulse ordering strategy to modify the distinct phase-shift of each ramp, and as a result, the order of generated pulses:

- Set the Pulse ordering property to Sequential device order to generate pulses in sequential order. Use this strategy to generate pulses for the Converter (Three-Phase) block or other thyristor networks that use sequential ordering.



- Set the Pulse ordering property to Natural order of commutation to generate pulses in the natural order. Use this strategy to generate pulses for thyristor networks that use natural ordering.



## Ports

### Input

#### **theta — Synchronization angle, radians**

scalar

Synchronization angle in the range  $[0, 2\pi]$ , in radians.

Data Types: `single` | `double`

#### **alpha — Firing angle, radians**

scalar

Thyristor firing angle in radians.

Data Types: `single` | `double`

### Output

#### **P — Pulse vector**

vector

Thyristor pulse vector.

Data Types: `single` | `double`

## Parameters

#### **Pulse ordering — Pulse ordering strategy**

`Sequential device order (default)` | `Natural order of commutation`

Specify the rule for pulse ordering based on the configuration of the thyristor network you are controlling. Use the `Sequential device order` strategy to generate pulses for the Converter block.

#### **Pulse width (rad) — Pulse width**

$5\pi/6$  rad (default) | positive scalar in range  $[0, \pi]$

Specify the width of each pulse in the range  $[0, \pi]$ .

#### **Sample time (-1 for inherited) — Block sample time**

$1e-5$  (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

## Version History

**Introduced in R2017b**

## References

- [1] Pelly, B. R. *Thyristor Phase-Controlled Converters and Cycloconverters: Operation, Control, and Performance*. New York, NY: John Wiley & Sons, Inc., 1971.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Sinusoidal Power Measurement (PLL, Three-Phase) | Thyristor 12-Pulse Generator | Thyristor Rectifier Voltage Controller (Three-Phase)

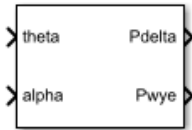
### Simscape Blocks

Converter (Three-Phase) | Six-Pulse Gate Multiplexer | Thyristor (Piecewise Linear)

# Thyristor 12-Pulse Generator

Generate thyristor 12-pulse waveform in single-pulsing mode

**Library:** Simscape / Electrical / Control / Pulse Width Modulation



## Description

The Thyristor 12-Pulse Generator block implements a thyristor 12-pulse waveform generator in single-pulsing mode.

You can use this block to perform phase-controlled AC to DC conversion by:

- Measuring the synchronization angle of the AC signal with a phase-locked loop
- Controlling a thyristor converter network with the pulses generated by this block

## Model

The Thyristor 12-Pulse Generator outputs six pulses for a delta gate driver and six pulses for a wye gate driver. The delta connection can lead (Delta11) or lag (Delta1). For information on the control model, see the Thyristor 6-Pulse Generator block.

## Ports

### Input

#### **theta** — Synchronization angle, rad

scalar

Synchronization angle in the range  $[0, 2\pi]$ , in rad.

Data Types: single | double

#### **alpha** — Firing angle, rad

scalar

Thyristor firing angle in rad.

Data Types: single | double

### Output

#### **Pdelta** — Delta pulse vector

vector

Thyristor pulse vector for the delta connection.

Data Types: single | double



**Pwye — Wye pulse vector**

vector

Thyristor pulse vector for the wye connection.

Data Types: `single` | `double`

**Parameters****Pulse ordering — Pulse ordering strategy**

`Sequential device order (default)` | `Natural order of commutation`

Specify the rule for pulse ordering based on the configuration of the thyristor network you are controlling. Use the `Sequential device order` strategy to generate pulses for the Converter (Three-Phase) block.

**Delta connection — Delta sinusoidal phase variance model**

`Leading (Delta11) (default)` | `Lagging (Delta1)`

Sinusoidal phase variance model for the delta connection.

**Pulse width (rad) — Pulse width**

`5*pi/6 rad (default)` | `positive scalar in range [0, pi]`

Specify the width of each pulse in the range `[0, pi]`.

**Sample time (-1 for inherited) — Block sample time**

`1e-5 (default)` | `positive scalar`

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to `-1`. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**Version History**

**Introduced in R2018a**

**References**

[1] Pelly, B. R. *Thyristor Phase-Controlled Converters and Cycloconverters: Operation, Control, and Performance*. New York, NY: John Wiley & Sons, Inc., 1971.

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Sinusoidal Power Measurement (PLL, Three-Phase) | Thyristor 6-Pulse Generator | Thyristor Rectifier Voltage Controller (Three-Phase)

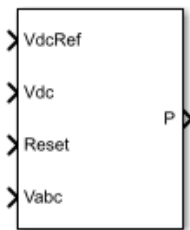
### **Simscape Blocks**

Converter (Three-Phase) | Twelve-Pulse Gate Multiplexer | Thyristor (Piecewise Linear) | Three-Winding Transformer (Three-Phase)

# Thyristor Rectifier Voltage Controller (Three-Phase)

Discrete-time DC-link voltage PI control for thyristor rectifiers

**Library:** Simscape / Electrical / Control / Converter Control

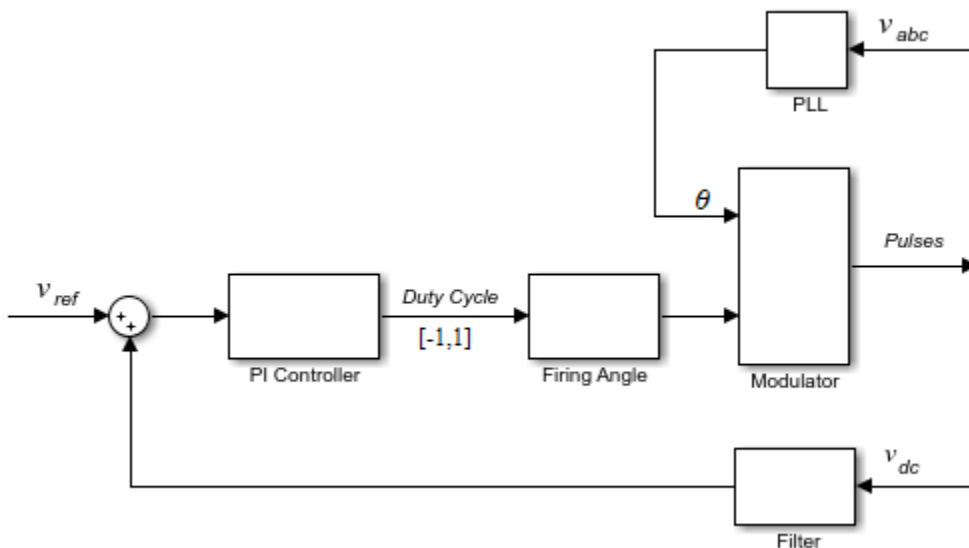


## Description

The Thyristor Rectifier Voltage Controller (Three-Phase) block implements a discrete-time proportional-integral (PI) based DC-link voltage controller for thyristor rectifiers.

## Model

To regulate the output DC-link voltage of a thyristor rectifier, the Thyristor Rectifier Voltage Controller (Three-Phase) block determines firing angles using the cosine wave-crossing method. The figure shows the control structure, which includes a three-phase phase-locked loop (PLL).



## Ports

### Input

#### **VdcRef — Reference DC voltage, V**

scalar

Desired DC-link voltage.

Data Types: single | double

#### **Vdc — DC-link voltage, V**

scalar

Measured DC-link voltage.

Data Types: single | double

#### **Reset — External reset**

scalar

External reset signal (rising edge) for integrators.

Data Types: single | double

#### **Vabc — Three-phase voltage, V**

vector

Three-phase voltage specified as a vector of that contains the *a*-, *b*-, and *c*-phase voltages.

Data Types: single | double

### Output

#### **P — Pulses**

vector

Pulse control signal.

Data Types: single | double

## Parameters

#### **Loop filter proportional gain — Loop filter $K_p$**

2 (default) | positive scalar

Loop filter proportional gain for the phase-locked loop (PLL).

#### **Loop filter integral gain — Loop filter $K_i$**

20 (default) | positive scalar

Loop filter integral gain for the phase-locked loop (PLL).

#### **Proportional gain — Controller $K_p$**

1 (default) | positive scalar

Proportional gain for the PI-controller that generates the reference phase voltage.

**Integral gain – Controller  $K_i$** 

1 (default) | positive scalar

Integral gain for the PI-controller that generates the reference phase voltage.

**Thyristor pulse width (rad) – Pulse width** $5\pi/6$  (default) | positive scalar

Angular width of pulses.

**Pulse ordering – Pulse ordering rule**

Sequential device order (default) | Natural order of commutation

Model for the ordering the generated pulses.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**Time constant voltage filter – DC voltage filter  $\tau$** 

0.001 (default) | positive scalar

Time constant for the DC voltage filter.

**Dependencies**

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** check box is selected.

**Filter DC voltage – DC voltage filter option**

on (default) | off

When the check box is:

- Selected — The block filters the DC voltage signal.
- Cleared — The block does not filter the DC voltage signal.

**Dependencies**

The **Time constant voltage filter** parameter is only visible when the **Filter DC voltage** check box is selected.

## Version History

Introduced in R2018a

## References

[1] Pelly, B. R. *Thyristor Phase-Controlled Converters and Cycloconverters: Operation, Control, and Performance*. New York, NY: John Wiley & Sons, Inc., 1971.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

### Blocks

Sinusoidal Power Measurement (PLL, Three-Phase) | Thyristor 6-Pulse Generator | Thyristor 12-Pulse Generator

### Simscape Blocks

Converter (Three-Phase) | Six-Pulse Gate Multiplexer | Twelve-Pulse Gate Multiplexer

# Timer

Behavioral model of a timer integrated circuit

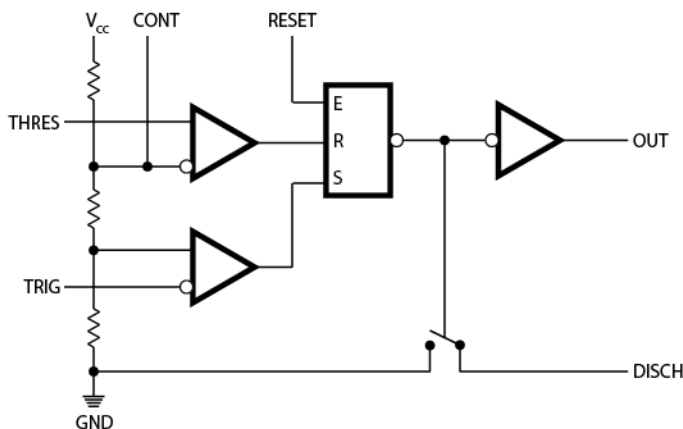
**Library:** Simscape / Electrical / Integrated Circuits



## Description

The Timer block is a behavioral model of a timer integrated circuit such as the NE555.

The following figure shows the implementation structure.



The **Potential divider component resistance** parameter sets the values of the three resistors creating the potential divider. The two comparator inputs have infinite input resistance and zero input capacitance. The S-R Latch block provides the functionality of the set-reset latch. It includes an output capacitor and a resistor with values set to match the **Propagation delay** parameter value. The block models the output stage inverter using a CMOS NOT block. You define the output resistance, low-level output voltage, and high-level output voltage for the CMOS gate in the Timer block dialog box. The discharge switch approximates the NPN bipolar transistor on a real timer as a switch with defined switch on-resistance and off-resistance values.

## Assumptions and Limitations

- The behavior is abstracted. Results are not as accurate as a transistor-level model.
- Delay in response to changing inputs depends solely upon the RC time constant of the resistor-capacitor network at the output of the latch. In practice, the delay has a more complex dependency on the device structure. Set this value based on the output-pulse rise and fall times.
- The drop in output voltage is a linear function of output current. In practice, the relationship is that of a bipolar transistor push-pull pair.
- The controlled switch arrangement used by the block is an approximation of an open-collector arrangement.

- The power supply connects internally within the component, and the block assumes that the GND pin is grounded.

## Ports

### Conserving

#### **THRES — Threshold pin**

electrical

Electrical conserving port associated to the timer threshold pin

#### **TRIG — Trigger pin**

electrical

Electrical conserving port associated to the timer trigger pin

#### **CONT — Control pin**

electrical

Electrical conserving port associated to the timer control pin

#### **RESET — Reset pin**

electrical

Electrical conserving port associated to the timer reset pin

#### **OUT — Output pin**

electrical

Electrical conserving port associated to the timer output pin

#### **DISCH — Discharge pin**

electrical

Electrical conserving port associated to the timer discharge pin

## Parameters

### Supply

#### **Power supply voltage — Power supply voltage**

15 V (default)

The voltage value  $V_{CC}$  that the block applies internally to the timer component.

### Outputs

#### **Low level output voltage — Low level output voltage**

0 V (default)

The output voltage when the timer output is low and no output current is drawn.

#### **High level output voltage — High level output voltage**

14.1 V (default)



The output voltage  $V_{OH}$  when the timer output is high and no current is drawn.

### **Output resistance – Output resistance**

8 0hm (default)

The ratio of output voltage drop to output current. Set this parameter to  $(V_{OH} - V_{OH1})/I_{OH1}$ , where  $V_{OH1}$  is the reduced output high voltage when the output current is  $I_{OH1}$ .

### **Propagation delay – Propagation delay**

100e-9 s (default)

Set this value to the input-pulse or output-pulse rise time.

### **Discharge**

#### **Discharge switch on-resistance – Discharge switch on-resistance**

12 0hm (default)

A representative value is the discharge pin saturation voltage divided by the corresponding current.

#### **Discharge switch off-resistance – Discharge switch off-resistance**

500e6 0hm (default)

A representative value is the discharge pin leakage current divided by the corresponding pin voltage.

### **Potential Divider**

#### **Potential divider component resistance – Potential divider component resistance**

5 k0hm (default)

A typical value for a 555-type timer is 5 k $\Omega$ . You can measure it directly across the positive supply and control pins when the chip does not connect to a circuit.

## **Version History**

Introduced in R2009b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

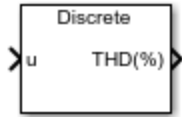
### **See Also**

S-R Latch | Comparator

# Total Harmonic Distortion

Total harmonic distortion measurement

**Library:** Simscape / Electrical / Control / Measurements



## Description

The Total Harmonic Distortion block calculates the total harmonic distortion (THD) percentage using these equations:

$$M = \frac{\text{harmonic magnitude}}{\sqrt{2}},$$

and

$$\%THD = 100 \frac{\sqrt{\sum_{i=2}^n M_i^2}}{M_1},$$

where:

- $M_i$  is the root mean squared (RMS) value of the harmonic magnitude that corresponds to the  $i^{\text{th}}$  harmonic order.
- $M$  is  $V_{RMS}$  or  $I_{RMS}$ .
- $\sum_{i=2}^n M_i^2$  is the sum of all harmonic components.

## Ports

### Input

#### **u — Harmonic magnitude**

scalar | vector

Harmonic magnitude.

Data Types: single | double

### Output

#### **THD (%) — Total harmonic distortion**

scalar | vector

Total harmonic distortion as a percentage.

Data Types: single | double

## Parameters

### Fundamental frequency — Fundamental frequency in Hz

60 Hz (default)

Fundamental frequency, in Hz.

### Sample time (-1 for inherited) — Block sample time

-1 (default) | nonnegative scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

To improve accuracy, set this value at least 10 times smaller than the **Time constant** parameter.

---

**Note** If this block is in a masked subsystem, or another variant subsystem that allows either continuous or discrete operation, promote the sample time parameter to ensure correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

---

## Version History

Introduced in R2019b

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

# Transmission Line

Delay-based or lumped parameter transmission line

**Library:** Simscape / Electrical / Passive / Lines



## Description

The Transmission Line block lets you choose between the following models of a transmission line:

- 1 Delay-based and lossless
- 2 Delay-based and lossy
- 3 Lumped parameter L-section
- 4 Lumped parameter pi-section
- 5 Distributed parameter line

The first option provides the best simulation performance, with options 2, 3 and 4 requiring progressively more computing power.

### Delay-Based and Lossless

This first option, **Delay-based and lossless**, models the transmission line as a fixed impedance, irrespective of frequency, plus a delay term. The defining equations are:

$$v_1(t) - i_1(t) Z_0 = v_2(t - \tau) + i_2(t - \tau) Z_0 \quad (1-54)$$

2

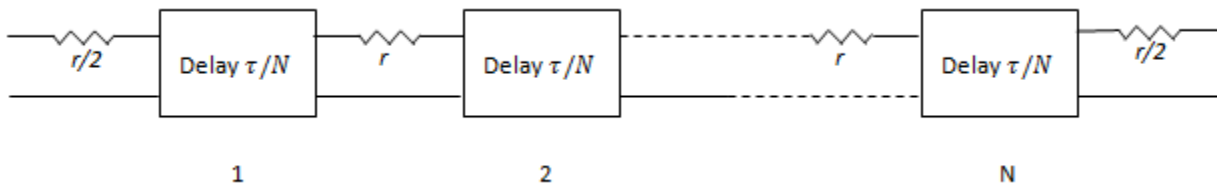
$$v_2(t) - i_2(t) Z_0 = v_1(t - \tau) + i_1(t - \tau) Z_0 \quad (1-55)$$

where:

- $v_1$  is the voltage across the left-hand end of the transmission line.
- $i_1$  is the current into the left-hand end of the transmission line.
- $v_2$  is the voltage across the right-hand end of the transmission line.
- $i_2$  is the current into the right-hand end of the transmission line.
- $\tau$  is the transmission line delay.
- $Z_0$  is the line characteristic impedance.

### Delay-Based and Lossy

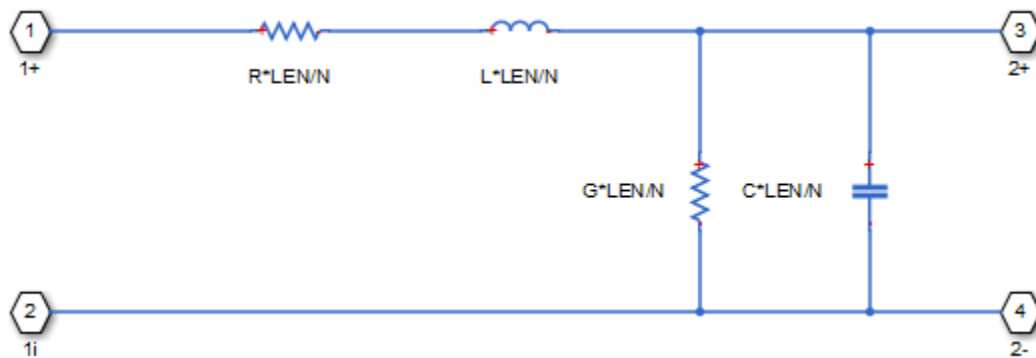
To introduce losses, the second option, **Delay-based and lossy**, connects  $N$  delay-based components, each defined by the above equations, in series via a set of resistors, as shown in the following illustration.



$N$  is an integer greater than or equal to 1.  $r = R \cdot LEN / N$ , where  $R$  is the line resistance per unit length and  $LEN$  is the line length.

### Lumped Parameter L-Section

The following block diagram shows the model of one L-line segment.



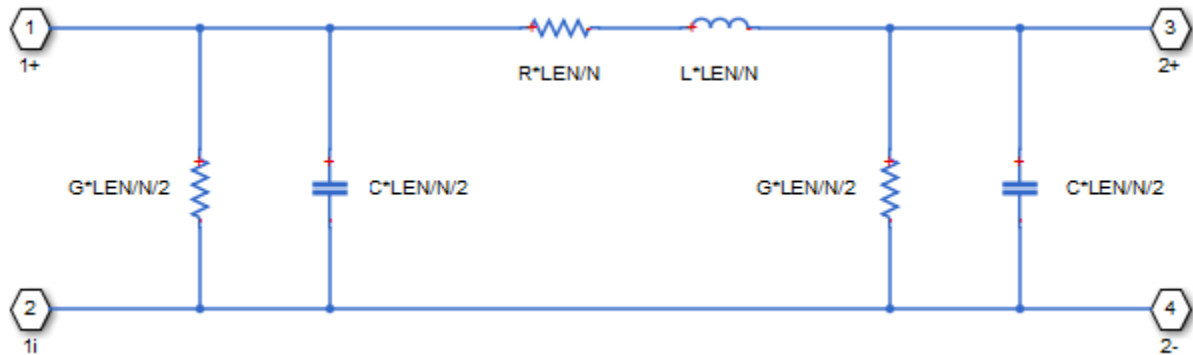
The lumped parameter parameterization uses  $N$  copies of the above segment model connected in series.

Parameters are as follows:

- $R$  is line resistance per unit length.
- $L$  is the line inductance per unit length.
- $C$  is the line capacitance per unit length.
- $G$  is the line conductance per unit length.
- $LEN$  is the length of the line.
- $N$  is the number of series segments.

### Lumped Parameter Pi-Section

The following block diagram shows the model of one pi-line segment.



The lumped parameter parameterization uses  $N$  copies of the above segment model connected in series. The parameters are as defined for the L-section transmission line model. Unlike the L-section model, the pi-section model is symmetric.

### Lumped Parameter Line Model Parameterization

The lumped-parameter models (L-section or pi-section) are the most challenging to simulate, typically needing many more segments (greater  $N$ ) than for the delay-based and lossy model [1 on page 1-2493].

Cable manufacturers do not typically quote an inductance value per unit length, but instead give the characteristic impedance. The inductance, capacitance, and characteristic impedance are related by:

$$L = C \cdot Z_0^2 \tag{1-56}$$

The block lets you specify either  $L$  or  $Z_0$  when using the lumped parameter model.

### Distributed parameter line

A distributed parameter line gives more accurate simulation at particular frequency compared to the other options provided by this block. However, this is not a frequency-dependent model. The accuracy of the model will drop outside the frequency point specified in the **Frequency used for rlcg specification** parameter.

For a frequency-dependent transmission line model, see Frequency-Dependent Overhead Line (Three-Phase).

The electromagnetic behavior of a multiconductor transmission line is described by the telegrapher's equation.

$$I_2 - Y_c V_2 = -H(I_1 + Y_c V_1) \tag{1-57}$$

$$I_1 - Y_c V_1 = -H(I_2 + Y_c V_2) \tag{1-58}$$

Define:

- $I_{sh,1} = Y_c V_1$  — Shunt current vector produced at terminal 1 by injected voltages  $V_1$
- $I_{sh,2} = Y_c V_2$  — Shunt current vector produced at terminal 2 by injected voltages  $V_2$
- $I_{rfl,1} = \frac{1}{2}(I_1 + Y_c V_1)$  — Reflected currents of terminal 1

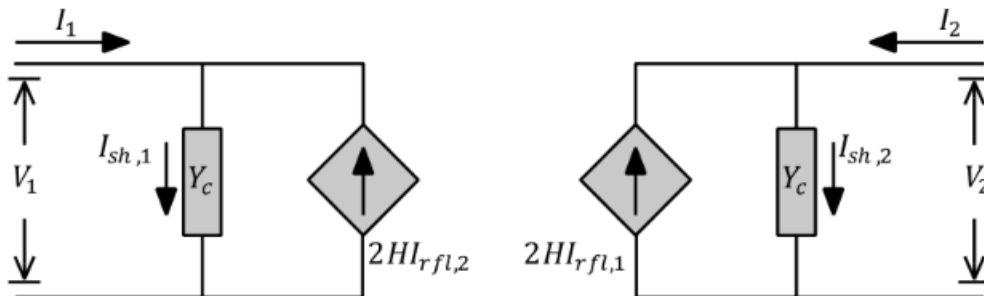
- $I_{rfl,2} = \frac{1}{2}(I_2 + Y_c V_2)$  — Reflected currents of terminal 2

You can then rewrite and solve equations 4 and 5:

$$I_1 = I_{sh,1} - 2HI_{rfl,2}$$

$$I_2 = I_{sh,2} - 2HI_{rfl,1}$$

The following block diagram shows the equivalent circuit for a distributed parameter line.

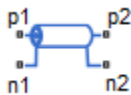


## Assumptions and Limitations

- For the lumped parameter options, MathWorks recommends that you use a trapezoidal solver such as ode23t. This is because lumped parameter transmission models have very lightly damped internal dynamics, which are best suited to trapezoidal solvers for numerical accuracy.
- The lumped parameter pi-section model has a parallel capacitor at both ends. This means that you should not connect it directly to an ideal voltage source, that is, a source with no internal resistance. The lumped parameter L-section model, however, has a series input resistor, and therefore you can connect it directly to an ideal voltage source.

## Ports

Refer to the figure for port locations.



## Conserving

### p1 — Inner conductor 1

electrical

Electrical conserving port associated with one end of the transmission line inner conductor.

### n1 — External shielding conductor 1

electrical

Electrical conserving port associated with one end of the transmission line external shielding conductor.

**p2 — Inner conductor 2**

electrical

Electrical conserving port associated with one end of the transmission line inner conductor.

**n2 — External shielding conductor 2**

electrical

Electrical conserving port associated with one end of the transmission line external shielding conductor.

**Parameters****Model Type — Transmission line models**

Delay-based and lossless (default) | Delay-based and lossy | Lumped parameter L-section | Lumped parameter pi-section | Distributed parameter line

Select one of the following transmission line models:

- **Delay-based and lossless** — Model the transmission line as a fixed impedance, irrespective of frequency, plus a delay term, as described in “Delay-Based and Lossless” on page 1-2486. This is the default method. It provides the best simulation performance.
- **Delay-based and lossy** — Model the transmission line as a number of delay-based components, connected in series via a set of resistors, as described in “Delay-Based and Lossy” on page 1-2486.
- **Lumped parameter L-section** — Model the transmission line as a number of L-line segments, connected in series, as described in “Lumped Parameter L-Section” on page 1-2487.
- **Lumped parameter pi-section** — Model the transmission line as a number of pi-line segments, connected in series, as described in “Lumped Parameter Pi-Section” on page 1-2487.
- **Distributed parameter line** — Model the transmission line as a distributed parameter line, as described in “Distributed parameter line” on page 1-2488 .

**Transmission delay — Total transmission delay**

5 ns (default) | positive scalar

The total transmission line delay. The parameter value must be greater than zero. The default value is 5 ns, which is a typical value for a one-meter coaxial cable.

**Dependencies**

This parameter is visible only when you select **Delay-based and lossless** or **Delay-based and lossy** for the **Model type** parameter.

**Parameterization — Model parameterization**

By characteristic impedance and capacitance (default) | By inductance and capacitance

Select the model parameterization method, as described in “Lumped Parameter Line Model Parameterization” on page 1-2488:

- **By characteristic impedance and capacitance** — Specify values for the **Characteristic impedance** and **Capacitance per unit length** parameters. This is the default method.



- By inductance and capacitance — Specify values for the **Inductance per unit length** and **Capacitance per unit length** parameters.

#### Dependencies

This parameter is visible only when you select Lumped parameter L-section or Lumped parameter pi-section for the **Model type** parameter.

#### Characteristic impedance — Characteristic impedance

50 Ohm (default) | positive scalar

The characteristic impedance of the transmission line. The parameter value must be greater than zero.

#### Dependencies

This parameter is visible only when you select either:

- Delay-based and lossless or Delay-based and lossy for the **Model type** parameter.
- Lumped parameter L-section or Lumped parameter pi-section for the **Model type** parameter and By characteristic impedance and capacitance for the **Parameterization** parameter.

#### Frequency used for rlcg specification — Frequency used for rlcg specification

60 Hz (default) | positive scalar

Frequency used for the R, L, C, G specification, where:

- $R$  is line resistance per unit length.
- $L$  is the line inductance per unit length.
- $C$  is the line capacitance per unit length.
- $G$  is the line conductance per unit length.

#### Dependencies

This parameter is visible only when you select Lumped parameter L-section, Lumped parameter pi-section, or Distributed parameter line for the **Model type** parameter.

#### Inductance per unit length — Inductance per unit length

220  $\mu$ H/m (default) | positive scalar

The effective inductance of the transmission line per unit length. The parameter value must be greater than zero.

#### Dependencies

This parameter is visible only when you select Lumped parameter L-section, Lumped parameter pi-section or Distributed parameter line for the **Model type** parameter and By inductance and capacitance for the **Parameterization** parameter.

#### Capacitance per unit length — Capacitance per unit length

90 pF/m (default)

The transmission line capacitance per unit length.

**Dependencies**

This parameter is visible only when you select Lumped parameter L-section, Lumped parameter pi-section or Distributed parameter line for the **Model type** parameter.

**Resistance per unit length — Resistance per unit length**

0.3 Ohm/m (default)

The total transmission line resistance (that is, the sum of the resistance for the two conducting paths) per unit length.

**Dependencies**

This parameter is visible only when you select Delay-based and lossy, Lumped parameter L-section, Lumped parameter pi-section or Distributed parameter line for the **Model type** parameter.

**Insulation conductance per unit length — Insulation conductance per unit length**

5e-6 S/m (default) | nonnegative scalar

The conductance between the two transmission line conductors per unit length. The parameter value must be greater than, or equal to, zero.

**Dependencies**

This parameter is visible only when you select Lumped parameter L-section, Lumped parameter pi-section or Distributed parameter line for the **Model type** parameter.

**Line length — Total transmission line length**

1 m (default) | positive scalar

The total transmission line length. The parameter value must be greater than zero.

**Dependencies**

This parameter is visible only when you select Delay-based and lossy, Lumped parameter L-section, Lumped parameter pi-section or Distributed parameter line for the **Model type** parameter.

**Number of segments — Number of model segments**

1 (default)

The number of model segments used to represent the transmission line. The parameter value must be an integer greater than, or equal to, 1.

**Dependencies**

This parameter is visible only when you select Delay-based and lossy, Lumped parameter L-section or Lumped parameter pi-section for the **Model type** parameter.

## Version History

Introduced in R2012a

## References

- [1] Sussman-Fort, S.E. and J.C. Hantgan. "SPICE Implementation of Lossy Transmission Line and Schottky Diode Models." *IEEE Transactions on Microwave Theory and Techniques*. Vol. 36, No. 1, January, 1988.

## Extended Capabilities

### **C/C++ Code Generation**

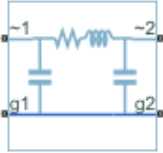
Generate C and C++ code using Simulink® Coder™.

## See Also

## Transmission Line (Three-Phase)

Three-phase transmission line using lumped-parameter pi-section line model

**Library:** Simscape / Electrical / Passive / Lines



### Description

The Transmission Line (Three-Phase) block models a three-phase transmission line using the lumped-parameter pi-line model. This model takes into account phase resistance, phase self-inductance, line-line mutual inductance and resistance, line-line capacitance, and line-ground capacitance.

To simplify the block-defining equations, Clarke's transformation is used. The resulting equations are:

$$V_1 - V_2 = \begin{bmatrix} R + 2R_m & & \\ & R - R_m & \\ & & R - R_m \end{bmatrix} I_1 + \begin{bmatrix} L + 2M & & \\ & L - M & \\ & & L - M \end{bmatrix} \frac{dI_1}{dt}$$

$$I_1 + I_2 = \begin{bmatrix} C_g & & \\ & C_g + 3C_l & \\ & & C_g + 3C_l \end{bmatrix} \frac{dV_2}{dt}$$

$$I_1 = T' I_1$$

$$I_2 = T' I_2$$

$$V_1 = T V_1$$

$$V_2 = T V_2$$

$$T = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & \sqrt{2} & 0 \\ 1 & -1/\sqrt{2} & \sqrt{3}/2 \\ 1 & -1/\sqrt{2} & -\sqrt{3}/2 \end{bmatrix}$$

where:

- $R$  is the line resistance for the segment.
- $R_m$  is the mutual resistance for the segment.
- $L$  is the line inductance for the segment.
- $M$  is the value of the **Mutual inductance** parameter.
- $C_g$  is the line-ground capacitance for the segment.

- $C_l$  is the line-line capacitance for the segment.
- $T$  is the Clarke's transformation matrix.
- $I_1$  is the three-phase current flowing into the ~1 port.
- $I_2$  is the three-phase current flowing into the ~2 port.
- $V_1$  is the three-phase voltage at the ~1 port.
- $V_2$  is the three-phase voltage at the ~2 port.

The positive and zero-sequence parameters are defined by the diagonal terms in the transformed equations:

$$R_0 = R + 2R_m$$

$$R_1 = R - R_m$$

$$L_0 = L + 2M$$

$$L_1 = L - M$$

$$C_0 = C_g$$

$$C_1 = C_g + 3C_l$$

Rearranging these equations gives the physical line quantities in terms of positive and zero-sequence parameters:

$$R = \frac{2R_1 + R_0}{3}$$

$$R_m = \frac{R_0 - R_1}{3}$$

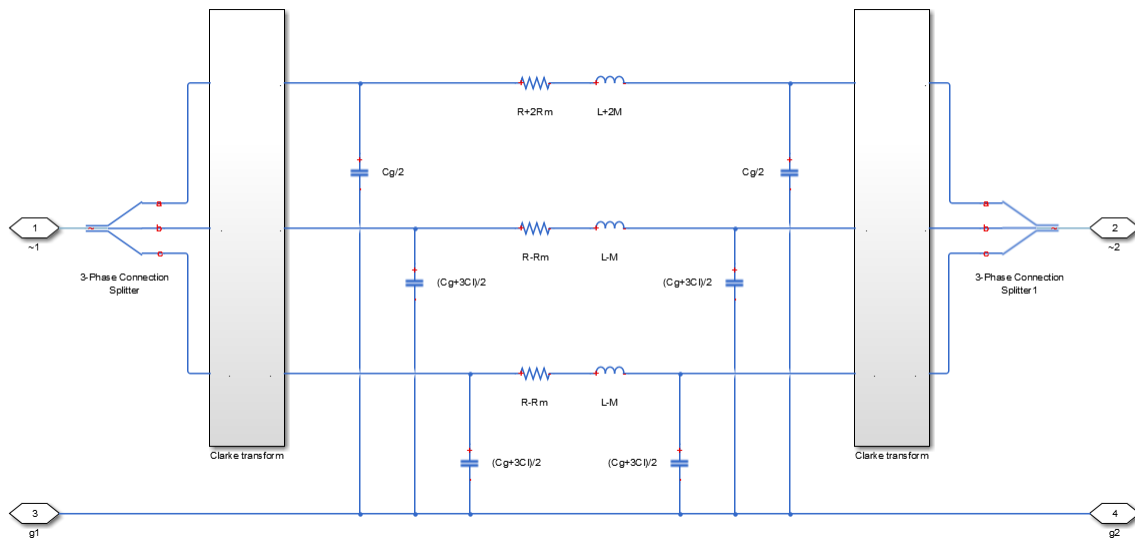
$$L = \frac{2L_1 + L_0}{3}$$

$$M = \frac{L_0 - L_1}{3}$$

$$C_l = \frac{C_1 - C_0}{3}$$

$$C_g = C_0$$

The figure shows the equivalent electrical circuit for a single-segment pi-line model using Clarke's transformation.



To increase fidelity, you can use the **Number of segments** parameter to repeat the pi-section N times, resulting in an N-segment transmission line model. More segments significantly slows down your simulation.

To improve numerical performance, you can add parasitic resistance and conductance components. Choosing large values for these components improves simulation speed but decreases simulation accuracy.

### Propagation Speeds

The Transmission Line (Three-Phase) calculates the positive-sequence propagation speeds by using these equations:

- $Z_1 = R_1 + j\omega L_1$  is the series impedance, in  $\Omega/\text{Km}$ .
- $Y_1 = j\omega C_1$  is the shunt admittance, in  $\text{s}/\text{Km}$ .
- $\gamma_1 = \sqrt{Z_1 Y_1}$  is the propagation constant.
- $v_1 = \frac{\omega}{\text{Imag}(\gamma_1)}$  is the propagation speed, in  $\text{Km}/\text{s}$ , where  $\omega = 2\pi f$ .

For zero-sequence propagation speed, the equations are the same but  $R_1, L_1, C_1, Z_1, Y_1, \gamma_1$ , and  $v_1$  are  $R_0, L_0, C_0, Z_0, Y_0, \gamma_0$ , and  $v_0$ .

### Faults

The Transmission Line (Three-Phase) block allows you to model these types of faults at specific position along the transmission line:

- Single-phase-to-ground fault (a-g, b-g, or c-g)
- Two-phase fault (a-b, b-c, or c-a)
- Two-phase-to-ground fault (a-b-g, b-c-g, or c-a-g)
- Three-phase fault (a-b-c)

- Three-phase-to-ground fault (a-b-c-g)

An input physical signal, **F**, triggers the fault. If the input physical signal is greater than or equal to 0.5, the fault is active. If the input physical signal is less than 0.5, the fault is inactive.

## Ports

### Input

#### **F — Fault triggering signal**

physical

Physical signal port associated with the fault control signal.

#### **Dependencies**

To enable this port, set **Enable fault** to Yes

### Conserving

#### **~1 — Three-phase port**

electrical

Expandable three-phase port.

#### **~2 — Three-phase port**

electrical

Expandable three-phase port.

#### **g1 — Ground connection at end ~1**

electrical

Electrical conserving port corresponding to ground connection at ~1 end of the transmission line.

#### **g2 — Ground connection at end ~2**

electrical

Electrical conserving port corresponding to ground connection at ~2 end of the transmission line.

#### **g1' — Ground connection at fault position 1**

electrical

Electrical conserving port corresponding to ground connection at fault position 1.

#### **Dependencies**

To enable this port, set **Accessible ground connections at fault position** to Yes

#### **g2' — Ground connection at fault position 2**

electrical

Electrical conserving port corresponding to ground connection at fault position 2.

#### **Dependencies**

To enable this port, set **Accessible ground connections at fault position** to Yes

## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### **Main**

#### **Line length — Length of the transmission line**

1 km (default)

Length of the transmission line.

#### **Frequency used for rlcg specification — Frequency used for rlcg specification**

60 Hz (default) | positive scalar

Frequency used for the R, L, C, G specification, where:

- **R** is line resistance per unit length.
- **L** is the line inductance per unit length.
- **C** is the line capacitance per unit length.
- **G** is the line conductance per unit length.

#### **Resistance — Resistance of the transmission line**

0.02 Ohm/km (default)

Resistance of the transmission line per phase per-unit length.

#### **Inductance — Self-inductance of the transmission line**

0.5 mH/km (default)

Self-inductance of the transmission line per phase per-unit length.

#### **Mutual inductance — Line-line mutual inductance**

0.1 mH/km (default)

Line-line mutual inductance per-unit length. Set this to 0 to remove mutual inductance.

#### **Line-line capacitance — Line-line capacitance**

0.3  $\mu$ F/km (default)

Line-line capacitance per-unit length.

#### **Line-ground capacitance — Line-ground capacitance**

0  $\mu$ F/km (default)

Line-ground capacitance per-unit length. The default value is 0  $\mu$ F/km (no line-ground capacitance).



**Mutual resistance — Line-line mutual resistance**

0 Ohm/km (default)

Line-line mutual resistance per unit length. The default value is 0 Ohm/km (no line-line mutual resistance).

**Number of segments — Number of segments**

1 (default)

Number of segments in the pi-line model.

**Parasitics****Parasitic series resistance — Parasitic series resistance**

1e-6 Ohm (default)

Resistance value, divided by the number of segments, that is added in series with every capacitor in the model.

**Parasitic parallel conductance — Parasitic parallel conductance**

1e-6 1/Ohm (default)

Conductance value, divided by the number of segments, that is added in parallel with every series resistor and inductor in the model.

**Faults****Enable fault — Fault enabling**

No (default) | Yes

Whether to enable faults.

**Fault type — Type of fault**

Single-phase to ground (a-g) (default) | Single-phase to ground (b-g) | Single-phase to ground (c-g) | Two-phase (a-b) | Two-phase (b-c) | Two-phase (c-a) | Two-phase to ground (a-b-g) | Two-phase to ground (b-c-g) | Two-phase to ground (c-a-g) | Three-phase (a-b-c) | Three-phase to ground (a-b-c-g)

Type of fault to model. The visibility of related parameters depends on the fault model. Options are:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)
- Single-phase to ground (c-g)
- Two-phase (a-b)
- Two-phase (b-c)
- Two-phase (c-a)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase (a-b-c)
- Three-phase to ground (a-b-c-g)

**Faulted phase-neutral resistance — Faulted phase-neutral resistance**

1e-3 Ohm (default)

Resistance between the phase connection and the neutral point when the fault is active.

**Dependencies**

To enable this parameter set the **Fault type** parameter to:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)
- Single-phase to ground (c-g)
- Two-phase (a-b)
- Two-phase (b-c)
- Two-phase (c-a)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase (a-b-c)
- Three-phase to ground (a-b-c-g)

**Faulted neutral-ground resistance — Faulted neutral-ground resistance**

1e-3 Ohm (default)

Resistance between the neutral point and the electrical reference when fault is active.

**Dependencies**

To enable this parameter set the **Fault type** parameter to:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)
- Single-phase to ground (c-g)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase to ground (a-b-c-g)

**Unfaulted phase-neutral conductance — Unfaulted phase-neutral conductance**

1e-6 1/Ohm (default)

Conductance between the phase connection and the neutral point when the fault is not active.

**Dependencies**

To enable this parameter set the **Fault type** parameter to:

- Single-phase to ground (a-g)
- Single-phase to ground (b-g)

- Single-phase to ground (c-g)
- Two-phase (a-b)
- Two-phase (b-c)
- Two-phase (c-a)
- Two-phase to ground (a-b-g)
- Two-phase to ground (b-c-g)
- Two-phase to ground (c-a-g)
- Three-phase (a-b-c)
- Three-phase to ground (a-b-c-g)

#### **Fault location — Location of fault**

1 Km (default)

Location where the fault occurs

#### **Dependencies**

To enable this parameter set **Enable fault** to Yes.

#### **Number of segments before fault — Number of segments before fault**

1 (default)

Number of segments in the pi-line model before the fault occurs

#### **Dependencies**

To enable this parameter set **Enable fault** to Yes.

#### **Accessible ground connections at fault position — Accessible ground connections at fault position**

No (default) | Yes

Select Yes to access the ground connections at the fault position, otherwise the ground connections at the fault position are connected to the electrical reference internally

#### **Dependencies**

To enable this parameter set **Enable fault** to Yes.

## **Version History**

Introduced in R2013b

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

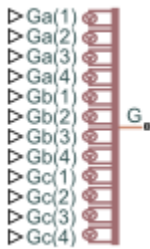
### **Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Twelve-Pulse Gate Multiplexer

Multiplex gate input signals to Three-Level Converter (Three-Phase) block

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

The Twelve-Pulse Gate Multiplexer block routes gate voltage signals to the 12 switching devices in a Three-Level Converter (Three-Phase) block. The block multiplexes the 12 gate signals into a single vector. Gate signals are ordered as *a*-phase, *b*-phase, and then *c*-phase, with four gate signals per phase.

When you switch the block inputs to electrical ports, the block shows 12 pairs of electrical connections, each pair corresponding to the gate and cathode of a switching device.

## Ports

### Input

**Ga(1), Ga(2), Ga(3), Ga(4) — Ga(1),Ga(2),Ga(3),Ga(4)**

physical signal

Physical signal ports associated with the gate terminals of the Three-Level Converter (Three-Phase)*a*-phase switching devices.

### Dependencies

To enable this port, set **Gate-control port** to PS.

**Gb(1), Gb(2), Gb(3), Gb(4) — Gb(1),Gb(2),Gb(3),Gb(4)**

physical signal

Physical signal ports associated with the gate terminals of the Three-Level Converter (Three-Phase)*b*-phase switching devices.

### Dependencies

To enable this port, set **Gate-control port** to PS.

**Gc(1), Gc(2), Gc(3), Gc(4) — Gc(1),Gc(2),Gc(3),Gc(4)**

physical signal

Physical signal ports associated with the gate terminals of the Three-Level Converter (Three-Phase)c-phase switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

**Conserving**

**Ga(1), Ga(2), Ga(3), Ga(4) — Ga(1),Ga(2),Ga(3),Ga(4)**

electrical

Electrical conserving ports associated with the gate terminals of the Three-Level Converter (Three-Phase)a-phase switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Gb(1), Gb(2), Gb(3), Gb(4) — Gb(1),Gb(2),Gb(3),Gb(4)**

electrical

Electrical conserving ports associated with the gate terminals of the Three-Level Converter (Three-Phase)b-phase switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Gc(1), Gc(2), Gc(3), Gc(4) — Gc(1),Gc(2),Gc(3),Gc(4)**

electrical

Electrical conserving ports associated with the gate terminals of the Three-Level Converter (Three-Phase)c-phase switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G — G**

electrical

Vector output port associated with the multiplexed gate signals. Connect this port to the G port of the Three-Level Converter (Three-Phase) block.

**Ka(1), Ka(2), Ka(3), Ka(4) — Ka(1),Ka(2),Ka(3),Ka(4)**

electrical

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter (Three-Phase) block a-phase switching devices.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Kb(1), Kb(2), Kb(3), Kb(4) — Kb(1),Kb(2),Kb(3),Kb(4)**

electrical

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter (Three-Phase) block *b*-phase switching devices.

#### Dependencies

To enable this port, set **Gate-control port** to `Electrical`.

**Kc(1), Kc(2), Kc(3), Kc(4) — Kc(1), Kc(2), Kc(3), Kc(4)**  
`electrical`

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter (Three-Phase) block *c*-phase switching devices.

#### Dependencies

To enable this port, set **Gate-control port** to `Electrical`.

## Parameters

**Gate-control port — Gate control port**  
`PS (default) | Electrical`

Whether to expose PS or electrical control ports.

## Version History

### Introduced in R2014b

**Physical and electrical connection ports update**  
*Behavior changed in R2021b*

From R2021b forward, to switch between physical and electrical ports, set the **Gate-control port** parameter to either `PS` or `Electrical`.

As a result of these changes, inside a model saved in an earlier release, review the **Gate-control port** parameter of this block.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

Six-Pulse Gate Multiplexer | Three-Level Converter (Three-Phase)

### Topics

“Switch Between Physical Signal and Electrical Ports”

## Two-Pulse Gate Multiplexer

Multiplex gate input signals to two quadrant chopper

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



### Description

The Two-Pulse Gate Multiplexer block multiplexes two separate voltage signals into a single vector. The vectorized signal can control the gates of two switching devices in a converter, such as a Two-Quadrant Chopper block.

### Model

This block can model a two-pulse gate multiplexer through either physical signal ports or electrical ports:

- PS ports — Two-pulse gate multiplexer with physical signal ports. Set the **Gate-control port** parameter to PS to control switching device gates in a converter block using Simulink gate-control voltage signals. To multiplex and connect Simulink signals to the gate-control inport of a converter block:
  - 1 Convert each voltage signal using a Simulink-PS Converter block.
  - 2 Multiplex the converted gate signals into a single vector using the multiplexer block.
  - 3 Connect the vector signal to the **G** port of the converter.
- Electrical ports — Two-pulse gate multiplexer with electrical conserving ports. To control switching device gates in a converter block using Simscape Electrical Electronics and Mechatronics blocks, set the **Gate-control port** parameter to **Electrical**. The electrical ports include pairs of electrical connections. Each pair corresponds to the gate and cathode of a switching device in the connected converter block.

### Ports

#### Input

##### G1 — Gate-control voltage signal 1

physical signal

Physical signal port associated with the gate terminal of the first switching device in a connected converter block.

#### Dependencies

To enable this port, set **Gate-control port** to PS.

Data Types: double



**G2 — Gate-control voltage signal 2**

physical signal

Physical signal port associated with the gate terminal of the second switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to PS.

Data Types: double

**Conserving****G1 — Gate-control voltage signal 1**

electrical

Electrical conserving port associated with the gate terminal of the first switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**a — A-phase AC reference point**

electrical

Electrical conserving port associated with the A-phase for the high-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**G2 — Gate-control voltage signal 2**

electrical

Electrical conserving port associated with the gate terminal of the second switching device in a connected converter block.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**L — DC reference point**

electrical

Electrical conserving port associated with the DC negative connection for the low-side switching device.

**Dependencies**

To enable this port, set **Gate-control port** to Electrical.

**Parameters****Gate-control port — Gate control port**

PS (default) | Electrical

Whether to expose PS or electrical control ports.

## **Version History**

**Introduced in R2018a**

### **Physical and electrical connection ports update**

*Behavior changed in R2021b*

From R2021b forward, to switch between physical and electrical ports, set the **Gate-control port** parameter to either PS or Electrical.

As a result of these changes, inside a model saved in an earlier release, review the **Gate-control port** parameter of this block.

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## **See Also**

Four-Pulse Gate Multiplexer | Six-Pulse Gate Multiplexer | Twelve-Pulse Gate Multiplexer | Two-Quadrant Chopper

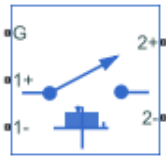
## **Topics**

Switch Between Physical Signal and Electrical Ports

# Two-Quadrant Chopper

Two-quadrant controlled DC-DC chopper

**Library:** Simscape / Electrical / Semiconductors & Converters / Converters



## Description

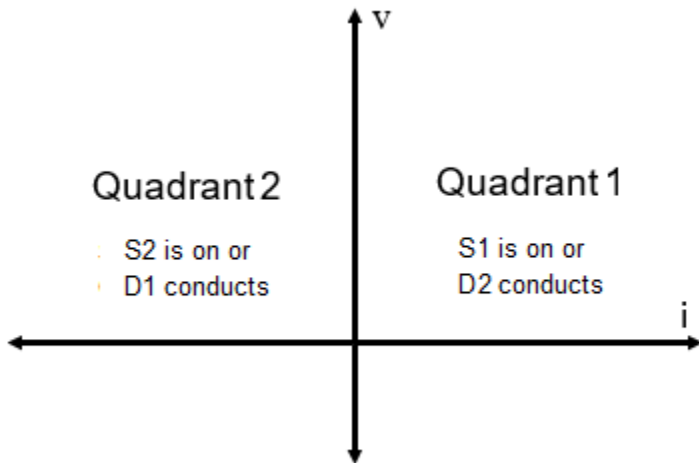
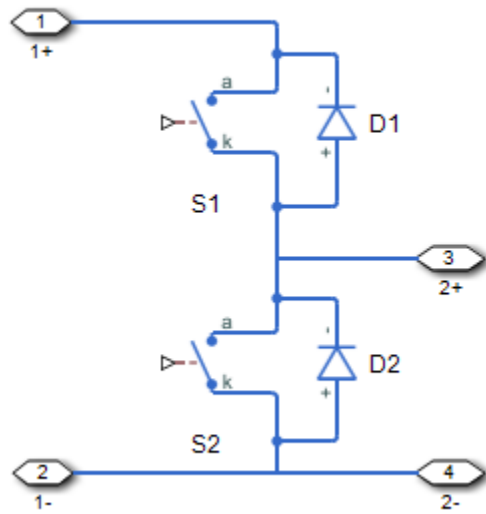
The Two-Quadrant Chopper block represents a two-quadrant controlled chopper for converting a fixed DC input to a variable DC output. The block contains two switching devices. Options for the type of switching devices are:

- GTO — Gate turn-off thyristor. For information about the I-V characteristic of the device, see GTO.
- Ideal semiconductor switch — For information about the I-V characteristic of the device, see Ideal Semiconductor Switch.
- IGBT — Insulated-gate bipolar transistor. For information about the I-V characteristic of the device, see IGBT (Ideal, Switching).
- MOSFET — N-channel metal-oxide-semiconductor field-effect transistor. For information about the I-V characteristic of the device, see MOSFET (Ideal, Switching).
- Thyristor — For information about the I-V characteristic of the device, see Thyristor (Piecewise Linear).
- Averaged Switch — Semiconductor switch with an anti-parallel diode. The control signal port, **G**, accepts values in the  $[0, 1]$  interval. When the value at port **G** is equal to 0 or 1, the averaged switch is either fully opened or fully closed, and it behaves similarly to the Ideal Semiconductor Switch block with an anti-parallel diode. When the value at port **G** is between 0 and 1, the averaged switch is partly opened. You can then average the PWM signal over a specified period. This allows for undersampling of the model or using modulation waveforms instead of PWM signals.

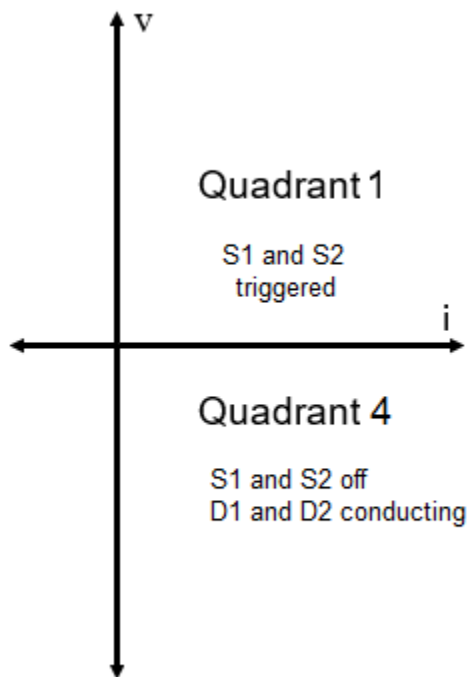
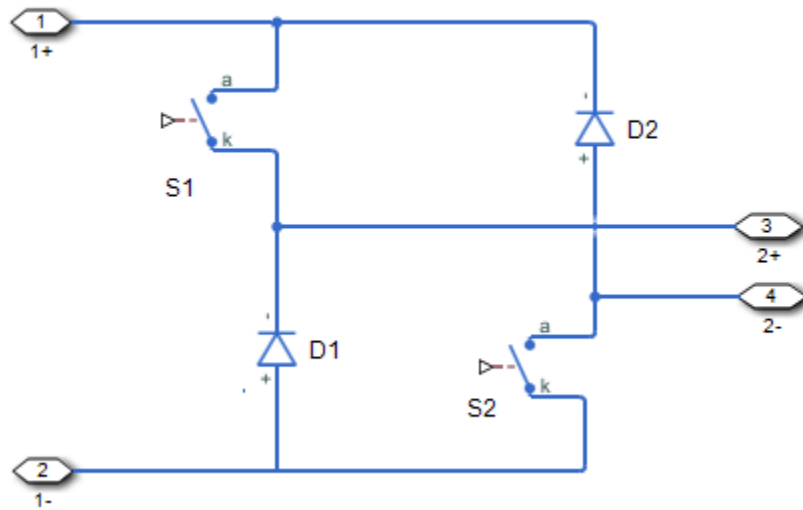
## Specify Chopper Quadrants

You can choose the quadrants where the Two-Quadrant Chopper operates. To select the quadrants, set the **Modeling option** parameter to either:

- **First and second quadrant.** The figures show the equivalent circuit and the operation for the first- and second- quadrant model.



- First and fourth quadrant. The figures show the equivalent circuit and the operation for the first- and fourth- quadrant model.



### Protection

The block contains an integral protection diode for each switching device. The integral diode protects the semiconductor device by providing a conduction path for reverse current. An inductive load can produce a high reverse-voltage spike when the semiconductor device suddenly switches off the voltage supply to the load.

To configure the internal protection diode block, use the **Diode** parameters. This table shows how to set the **Model dynamics** parameter based on your goals.

Goals	Value to Select	Integral Protection Diode
Prioritize simulation speed.	Diode with no dynamics	The Diode block
Prioritize model fidelity by precisely specifying reverse-mode charge dynamics.	Diode with charge dynamics	The dynamic model of the Diode block

You can also include a snubber circuit for each switching device. Snubber circuits contain a series-connected resistor and capacitor. They protect switching devices against high voltages that inductive loads produce when the device turns off the voltage supply to the load. Snubber circuits also prevent excessive rates of current change when a switching device turns on.

To include and configure a snubber circuit for each switching device, use the **Snubbers** parameters.

### Gate Control

To connect Simulink gate-control voltage signals to the gate ports of the internal switching devices:

- 1 Convert each voltage signal using a Simulink-PS Converter block.
- 2 Multiplex the converted gate signals into a single vector using a Two-Pulse Gate Multiplexer block.
- 3 Connect the vector signal to the **G** port.

### Piecewise Constant Approximation in Averaged Switch for FPGA Deployment

If you set the **Switching device** parameter to *Averaged switch* and your model uses a partitioning solver, this block produces nonlinear partitions because the average mode equations include modes,  $G_{sat}$  that are functions of the input **G**. To make these equations compatible with hardware description language (HDL) code generation, and therefore FPGA deployment, set the **Integer for piecewise constant approximation of gate input (0 for disabled)** parameter to a value greater than 0. This block then treats the  $G_{sat}$  mode as a piecewise constant integer with a fixed range. This turns the previously nonlinear partitions to linear time varying partitions.

An integer value in the range  $[0, K]$ , where  $K$  is the value of the **Integer for piecewise constant approximation of gate input (0 for disabled)**, is now associated with each real value mode in the range  $[0, 1]$ . The block computes the piecewise constant mode by dividing the original mode by  $K$  to normalize it back to the range  $[0, 1]$ :

$$u_I = (\text{floor}(u \cdot K))$$

$$\hat{u} = \frac{u_I}{K}$$

## Ports

### Conserving

#### **G** — Switching device gate control

electrical | vector

Electrical conserving port associated with the gate terminals of the switching devices.

Data Types: double

### **1+ — Positive DC voltage 1**

electrical | scalar

Electrical conserving port associated with the positive terminal of the first DC voltage.

Data Types: double

### **1- — Negative DC voltage 1**

electrical | scalar

Electrical conserving port associated with the negative terminal of the first DC voltage.

Data Types: double

### **2+ — Positive DC voltage 2**

electrical | scalar

Electrical conserving port associated with the positive terminal of the second DC voltage.

Data Types: double

### **2- — Negative DC voltage 2**

electrical | scalar

Electrical conserving port associated with the negative terminal of the second DC voltage.

Data Types: double

## **Parameters**

### **Modeling option — Chopper operating quadrants**

First and second quadrant (default) | First and fourth quadrant

Whether to model a chopper that operates in the first and second quadrant or in the first and fourth quadrant.

### **Switching Devices**

This table shows how the visibility of **Switching Devices** parameters depends on the **Switching device** that you select. To learn how to read the table, see “Parameter Dependencies” on page A-2.

## Switching Devices Parameter Dependencies

Parameters and Options					
Switching device					
<i>Ideal Semiconductor Switch</i>	<i>GTO</i>	<i>IGBT</i>	<i>MOSFET</i>	<i>Thyristor</i>	<i>Averaged Switch</i>
On-state resistance	Forward voltage	Forward voltage	Drain-source on resistance	Forward voltage	On-state resistance
Off-state conductance	On-state resistance	On-state resistance	Off-state conductance	On-state resistance	
Threshold voltage	Off-state conductance	Off-state conductance	Threshold voltage	Off-state conductance	
	Gate trigger voltage, Vgt	Threshold voltage		Gate trigger voltage, Vgt	Integer for piecewise constant approximation of gate input (0 for disabled)
	Gate turn-off voltage, Vgt_off			Gate turn-off voltage, Vgt_off	
	Holding current			Holding current	

### Switching device – Switch type

*Ideal Semiconductor Switch* (default) | *GTO* | *IGBT* | *MOSFET* | *Thyristor* | *Averaged Switch*

Switching device type for the converter.

### Dependencies

See the Switching Devices Parameter Dependencies table.

### Forward voltage – Voltage

0.8 Ohm (default) | scalar

For the different switching device types, the **Forward voltage** is taken as:

- GTO — Minimum voltage required across the anode and cathode block ports for the gradient of the device I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- IGBT — Minimum voltage required across the collector and emitter block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On-state resistance**
- Thyristor — Minimum voltage required for the device to turn on

### Dependencies

See the Switching Devices Parameter Dependencies table.

### On-state resistance – Resistance

0.001 Ohm (default) | scalar

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Rate of change of voltage versus current above the forward voltage
- Ideal semiconductor switch — Anode-cathode resistance when the device is on



- IGBT — Collector-emitter resistance when the device is on
- Thyristor — Anode-cathode resistance when the device is on
- Averaged switch — Anode-cathode resistance when the device is on

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Drain-source on resistance — Resistance

0.001 0hm (default) | scalar

Resistance between the drain and the source, which also depends on the gate-to-source voltage.

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Off-state conductance — Conductance

1e-5 1/0hm (default) | scalar

Conductance when the device is off. The value must be less than  $1/R$ , where  $R$  is the value of **On-state resistance**.

For the different switching device types, the **On-state resistance** is taken as:

- GTO — Anode-cathode conductance
- Ideal semiconductor switch — Anode-cathode conductance
- IGBT — Collector-emitter conductance
- MOSFET — Drain-source conductance
- Thyristor — Anode-cathode conductance

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Threshold voltage — Voltage threshold

6 V (default) | scalar

Gate voltage threshold. The device turns on when the gate voltage is above this value. For the different switching device types, the device voltage of interest is:

- Ideal semiconductor switch — Gate-emitter voltage
- IGBT — Gate-cathode voltage
- MOSFET — Gate-source voltage

#### Dependencies

See the Switching Devices Parameter Dependencies table.

#### Gate trigger voltage, Vgt — Voltage threshold

1 V (default) | scalar

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Gate turn-off voltage, Vgt\_off – Voltage threshold**

-1 V (default) | scalar

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Holding current – Current threshold**

1 A (default) | scalar

Gate current threshold. The device stays on when the current is above this value, even when the gate-cathode voltage falls below the gate trigger voltage.

**Dependencies**

See the Switching Devices Parameter Dependencies table.

**Protection Diode**

The visibility of **Diode** parameters depends on how you configure the protection diode **Model dynamics** and **Reverse recovery time parameterization** parameters. To learn how to read this table, see “Parameter Dependencies” on page A-2.

**Protection Diode Parameter Dependencies**

Parameters and Options			
<b>Model dynamics</b>			
<i>Diode with no dynamics</i>		<i>Diode with charge dynamics</i>	
Forward voltage	Forward voltage		
On resistance	On resistance		
Off conductance	Off conductance		
	Junction capacitance		
	Peak reverse current, iRM		
	Initial forward current when measuring iRM		
	Rate of change of current when measuring iRM		
	<b>Reverse recovery time parameterization</b>		
	<i>Specify stretch factor</i>	<i>Specify reverse recovery time directly</i>	<i>Specify reverse recovery charge</i>
Reverse recovery time stretch factor	Reverse recovery time, trr	Reverse recovery charge, Qrr	

**Model dynamics – Diode model**

Diode with no dynamics (default) | Diode with charge dynamics

Diode type. The options are:

- **Diode with no dynamics** — Select this option to prioritize simulation speed using the Diode block.
- **Diode with charge dynamics** — Select this option to prioritize model fidelity in terms of reverse mode charge dynamics using the commutation diode model of the Diode block.

---

**Note** If you select **Averaged Switch** for the **Switching Device** parameter in the **Switching Device** setting, this parameter is not visible and **Diode with no dynamics** is automatically selected.

---

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Forward voltage — Voltage

0.8 V (default) | scalar

Minimum voltage required across the positive and negative block ports for the gradient of the diode I-V characteristic to be  $1/R_{on}$ , where  $R_{on}$  is the value of **On resistance**.

### On resistance — Resistance

0.001 Ohm (default) | scalar

Rate of change of voltage versus current above the **Forward voltage**.

### Off conductance — Conductance

1e-5 1/Ohm (default) | scalar

Conductance of the reverse-biased diode.

### Junction capacitance — Capacitance

50 nF (default) | scalar

Diode junction capacitance.

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Peak reverse current, $i_{RM}$ — Current

-235 A (default) | scalar less than 0

Peak reverse current measured by an external test circuit.

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Initial forward current when measuring $i_{RM}$ — Current

300 A (default) | scalar greater than 0

Initial forward current when measuring peak reverse current. This value must be greater than zero.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Rate of change of current when measuring  $i_{RM}$  – Current change rate**

-50 A/us (default) | scalar

Rate of change of current when measuring peak reverse current.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery time parameterization – Recovery-time model**

Specify stretch factor (default) | Specify reverse recovery time directly | Specify reverse recovery charge

Model for parameterizing the recovery time. When you select `Specify stretch factor` or `Specify reverse recovery charge`, you can specify a value that the block uses to derive the reverse recovery time. For more information on these options, see “How the Block Calculates  $T_M$  and  $\tau$ ” on page 1-452.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery time stretch factor – Stretch factor**

3 (default) | scalar greater than 1

Value that the block uses to calculate **Reverse recovery time,  $t_{rr}$** . Specifying the stretch factor is an easier way to parameterize the reverse recovery time than specifying the reverse recovery charge. The larger the value of the stretch factor, the longer it takes for the reverse recovery current to dissipate.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery time,  $t_{rr}$  – Time**

15 us (default) | scalar

Interval between the time when the current initially goes to zero (when the diode turns off) and the time when the current falls to less than 10 percent of the peak reverse current.

The value of the **Reverse recovery time,  $t_{rr}$**  parameter must be greater than the value of the **Peak reverse current,  $i_{RM}$**  parameter divided by the value of the **Rate of change of current when measuring  $i_{RM}$**  parameter.

**Dependencies**

See the Protection Diode Parameter Dependencies table.

**Reverse recovery charge,  $Q_{rr}$  – Charge**

1500 s\* $\mu$ A (default) | scalar

Value that the block uses to calculate **Reverse recovery time, trr**. Use this parameter if the data sheet for your diode device specifies a value for the reverse recovery charge instead of a value for the reverse recovery time.

The reverse recovery charge is the total charge that continues to dissipate when the diode turns off.

The value must be less than  $-\frac{i_{RM}^2}{2a}$ ,

where:

- $i_{RM}$  is the value specified for **Peak reverse current, iRM**.
- $a$  is the value specified for **Rate of change of current when measuring iRM**.

### Dependencies

See the Protection Diode Parameter Dependencies table.

### Snubbers

The **Snubbers** parameters tab is not visible if you set **Switching device** to Averaged Switch.

The table summarizes the **Snubbers** parameter dependencies. To learn how to read the table, see “Parameter Dependencies” on page A-2.

### Snubbers Parameter Dependencies

Snubbers Parameter Dependencies	
Snubber	
None	RC Snubber
	Snubber resistance
	Snubber capacitance

### Snubber — Snubber model

None (default) | RC snubber

Switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber resistance — Resistance

0.1 Ohm (default) | scalar

Resistance of the switching device snubber.

### Dependencies

See the Snubbers Parameter Dependencies table.

### Snubber capacitance — Capacitance

1e-7 (default) | F | scalar

Capacitance of the switching device snubber.

### **Dependencies**

See the Snubbers Parameter Dependencies table.

## **Version History**

**Introduced in R2018a**

### **References**

[1] Trzynadlowski, A. M. *Introduction to Modern Power Electronics, 2nd Edition*. Hoboken, NJ: John Wiley & Sons Inc., 2010.

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

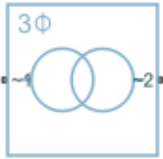
### **See Also**

Average-Value Chopper | Four-Quadrant Chopper | One-Quadrant Chopper | Two-Pulse Gate Multiplexer

## Two-Winding Transformer (Three-Phase)

Three-phase linear nonideal wye- and delta-configurable two-winding transformer with saturation capability

**Library:** Simscape / Electrical / Passive / Transformers



### Description

The Two-Winding Transformer (Three-Phase) block represents a linear nonideal three-phase two-winding transformer that transfers electrical energy between two or more circuits through electromagnetic induction. The block includes linear winding leakage and linear core magnetization effects. You can parameterize the block impedance using per-unit values. The primary and secondary winding types, delta-wye phase angle, and core types are configurable.

The configuration options for both the primary and secondary windings are:

- Wye with floating neutral — Star or T configuration with Floating Neutral (Three-Phase)
- Wye with neutral port — Star or T configuration with Neutral Port (Three-Phase)
- Wye with grounded neutral — Star or T configuration with Grounded Neutral (Three-Phase)
- Delta 1 o'clock — Mesh configuration with a lagging 30 degree phase shift relative to the voltage of a connected wye configuration
- Delta 11 o'clock — Mesh configuration with leading 30 degree phase shift relative to the voltage of a connected wye configuration

Options for the core type are:

- Three-phase five-limb
- Three-phase three-limb

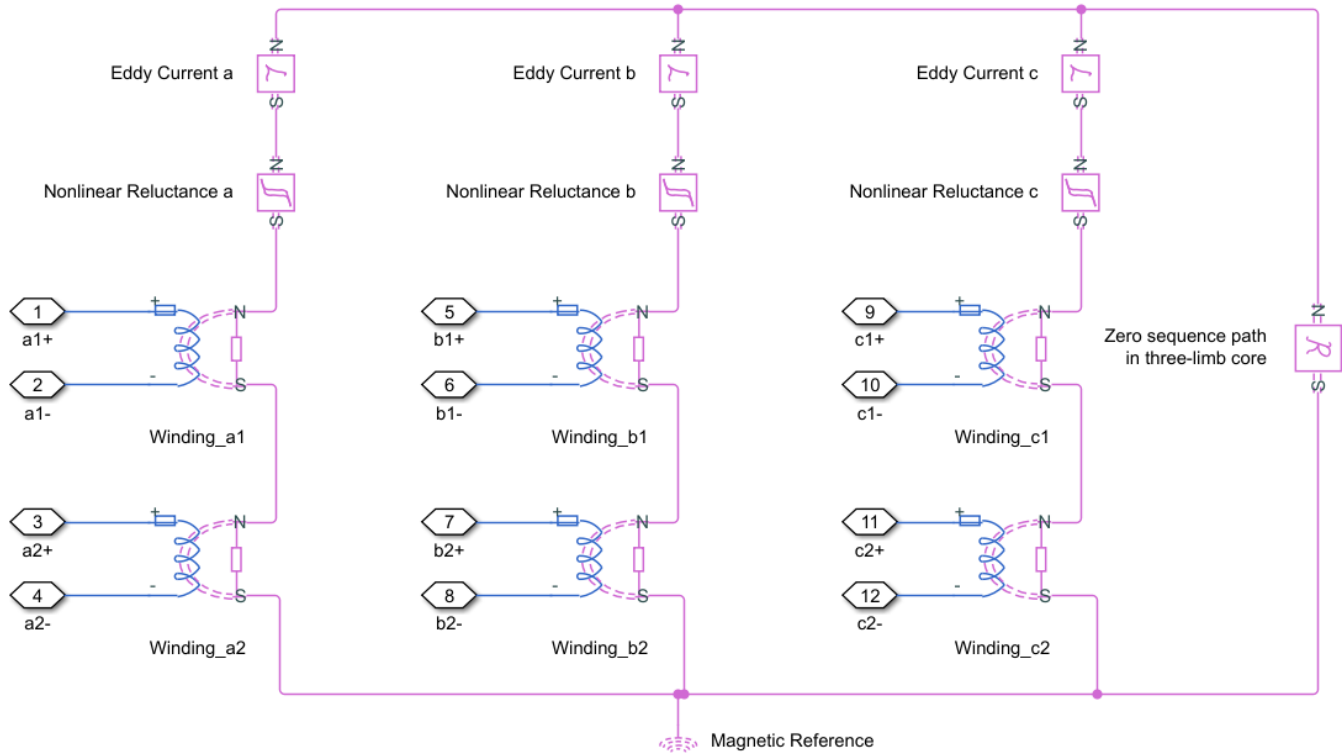
Although a three-limb core is typically less expensive, a five-limb core offers these advantages:

- Lower impedance for the zero-sequence component of current, that is between the line and neutral, in the case of an unbalanced load
- Greater heat dissipation

### Equations

#### Three-Limb Core

This block is implemented in the magnetic domain using basic magnetic reluctances, windings, and eddy currents blocks.



It is important to determine the relation between the electrical domain parameters from the block mask and the magnetic domain parameters used in the model:

- $n_1$  is the number of the primary winding turns.
- $n_2$  is the number of the secondary winding turns.
- $L_m$  is the shunt magnetizing inductance.
- $L_0$  is the zero-sequence inductance.
- $L_p$  is the primary winding leakage inductance.
- $L_s$  is the secondary winding leakage inductance.
- $R_m$  is the shunt magnetizing resistance.
- $R$  is the magnetizing reluctance between phases.

$$R = \frac{n_1^2}{L_m}$$

- $R_0$  is the zero sequence reluctance.

$$R_0 = \frac{1}{3} \frac{n_1^2}{L_0 - L_p}$$

- $RL_1$  is the primary winding leakage reluctance.

$$RL_1 = \frac{n_1^2}{L_p}$$

- $RL_2$  is the secondary winding leakage reluctance.



$$Rl_2 = \frac{n_2^2}{L_s}$$

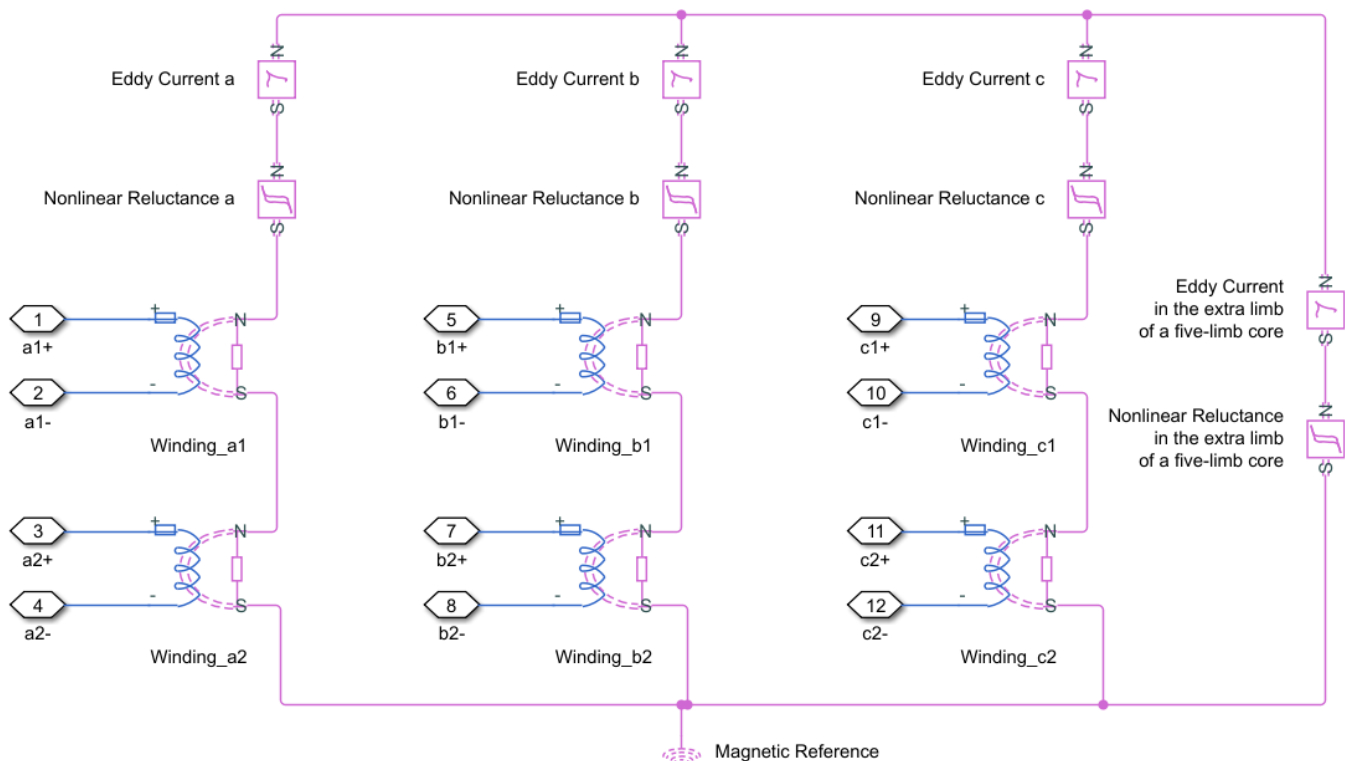
- $L_{eddy}$  is the conductance of eddy current loop

$$L_{eddy} = \frac{n_1^2}{R_m}$$

For two-winding transformers (three-phase), the coupling between different windings in each phase is identical.

### Five-Limb Core

In the case of a five-limb transformer, the extra magnetic flux paths provided by the extra limbs can be represented by zero sequence reluctances, which are originally designed for magnetic paths through the air in the three-limb transformer.



In a five-limb model, the magnetic reluctances from the phases to the extra limbs are supposed to be equal to the magnetic reluctances between phases.

$$R = R_0$$

Therefore we deduce:

$$R = R_0 = \frac{n_1^2}{L_m}$$

## Display Options

You can display the transformer per-unit base values in the MATLAB command window using the block context menu. To display the values, right-click the block and select **Electrical > Display Base Values**.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **~1 – Primary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a1\ b1\ c1]$ , voltage of winding 1.

#### **n1 – Primary winding neutral point**

electrical

Electrical conserving port associated with the primary winding neutral point.

### Dependencies

This port is only visible when the **Main** parameter **Winding 1 connection type** is set to **Wye** with **neutral** port.

#### **~2 – First secondary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a2\ b2\ c2]$ , voltage of the first secondary winding.

#### **n2 – First secondary winding neutral point**

electrical

Electrical conserving port associated with the first secondary winding neutral point.

### Dependencies

This port is only visible when the **Main** parameter **Winding 2 connection type** is set to **Wye** with **neutral** port.

## Parameters

### Modeling option — Whether to model composite or expanded three-phase ports

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### Rated apparent power — Apparent power at rated capacity

100e6 (default) | positive scalar

Apparent power flowing through the transformer when operating at rated capacity. The value must be greater than 0.

#### Rated electrical frequency — Connected network electrical frequency

60 (default) | positive scalar

Rated or nominal frequency of the AC network to which the transformer is connected. The value must be greater than 0.

#### Winding 1 connection type — Primary winding configuration

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta 1 o'clock | Delta 11 o'clock

Primary winding type.

#### Primary rated voltage — RMS line voltage applied to the primary winding

4160 (default) | positive scalar

RMS line voltage applied to the primary winding under normal operating conditions. The value must be greater than 0.

#### Winding 2 connection type — Secondary winding configuration

Wye with floating neutral (default) | Wye with neutral port | Wye with grounded neutral | Delta 1 o'clock | Delta 11 o'clock

Secondary winding type.

#### Secondary rated voltage — RMS line voltage applied to the secondary winding

24e3 (default) | positive scalar

RMS line voltage applied to the secondary winding under normal operating conditions. The value must be greater than 0.

#### Core type — Number of limbs

Three-phase three-limb (default) | Three-phase five-limb

Number of limbs that comprise the magnetic circuit.

**Impedances****Primary winding resistance (pu) – Primary winding power loss**`0.01` (default) | positive scalar

Per-unit power loss in the primary winding. The value must be greater than 0.

**Secondary winding resistance (pu) – Secondary winding power loss**`0.01` (default) | positive scalar

Per-unit power loss in the secondary winding. The value must be greater than 0.

**Leakage reactance – Leakage reactance modeling**`Exclude` (default) | `Include`

Whether to model magnetic flux losses.

**Primary leakage reactance (pu) – Primary winding magnetic flux loss**`0.001` (default) | positive scalar

Per-unit magnetic flux loss in the primary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to `Include`.

**Secondary leakage reactance (pu) – Secondary winding magnetic flux loss**`0.001` (default) | positive scalar

Per-unit magnetic flux loss in the secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to `Include`.

**Magnetizing resistance – Magnetizing resistance modeling**`Exclude` (default) | `Include`

Whether to model the transformer core losses.

**Shunt magnetizing resistance (pu) – Transformer core losses**`500` (default) | positive scalar

Per-unit losses in the transformer core. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing resistance** to `Include`.

**Magnetizing reactance – Magnetizing reactance modeling**`Exclude` (default) | `Include`

Whether to model magnetic effects of the transformer core.

**Magnetic saturation representation – Representation of the magnetic saturation**`None` (default) | `Lookup table (phi versus i)`

Choose if and how you want to represent the magnetic saturation.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include.

**Current vector (pu), i – Per unit vector of currents**

[0, .002, 1, 2] (default)

Per unit vector of currents. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Magnetic flux vector (pu), phi – Per unit vector of magnetic flux**

[0, 1.2, 1.5, 1.51] (default)

Per unit vector of magnetic flux. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Shunt magnetizing reactance (pu) – Transformer core magnetic effects**

500 (default) | positive scalar

Per-unit magnetic effects of the transformer core when operating in its linear region. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to None.

**Zero sequence reactance (pu) – Zero sequence reactance**

0.5 (default) | positive scalar

Per-unit zero sequence reactance. The value must be greater than or equal to the primary winding magnetic flux loss.

**Dependencies**

To enable this parameter, set **Core type**, a **Main** parameter, to Three-phase three-limb.

## Version History

Introduced in R2019a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

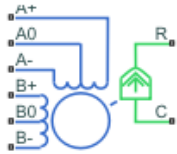
**See Also**

Earthing Transformer | Ideal Transformer | Nonlinear Transformer | Tap-Changing Transformer | Three-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer

# Unipolar Stepper Motor

Stepper motor with center taps on two-phase windings

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Unipolar Stepper Motor block represents a stepper motor that has center taps on the two phase windings. The winding currents and mechanical output are defined by the following equations:

$$e_{A+} = -K_m \omega \sin(N_r \theta)$$

$$e_{A-} = K_m \omega \sin(N_r \theta)$$

$$e_{B+} = K_m \omega \cos(N_r \theta)$$

$$e_{B-} = -K_m \omega \cos(N_r \theta)$$

$$\frac{di_{A+}}{dt} = (v_{A+} - R i_{A+} - e_{A+})/L$$

$$\frac{di_{A-}}{dt} = (v_{A-} - R i_{A-} - e_{A-})/L$$

$$\frac{di_{B+}}{dt} = (v_{B+} - R i_{B+} - e_{B+})/L$$

$$\frac{di_{B-}}{dt} = (v_{B-} - R i_{B-} - e_{B-})/L$$

$$J \frac{d\omega}{dt} + B\omega = T_e$$

$$T_e = -K_m \left( i_{A+} - i_{A-} - \frac{e_{A+} - e_{A-}}{R_m} \right) \sin(N_r \theta) + K_m \left( i_{B+} - i_{B-} - \frac{e_{B+} - e_{B-}}{R_m} \right) \cos(N_r \theta) - T_d \sin(4N_r \theta)$$

$$\frac{d\theta}{dt} = \omega$$

where:

- $e_{A+}$  is the back emf induced across the **A+** to **A0** half-winding.

- $e_{A-}$  is the back emf induced across the **A-** to **A0** half-winding.
- $e_{B+}$  is the back emf induced across the **B+** to **B0** half-winding.
- $e_{B-}$  is the back emf induced across the **B-** to **B0** half-winding.
- $i_{A+}$  is the current flowing from the **A+** port to the **A0** center tap port.
- $i_{A-}$  is the current flowing from the **A-** port to the **A0** center tap port.
- $i_{B+}$  is the current flowing from the **B+** port to the **B0** center tap port.
- $i_{B-}$  is the current flowing from the **B-** port to the **B0** center tap port.
- $v_{A+}$  is the voltage at the **A+** port relative to the **A0** center tap port.
- $v_{A-}$  is the voltage at the **A-** port relative to the **A0** center tap port.
- $v_{B+}$  is the voltage at the **B+** port relative to the **B0** center tap port.
- $v_{B-}$  is the voltage at the **B-** port relative to the **B0** center tap port.
- $K_m$  is the motor torque constant.
- $N_r$  is the number of teeth on each of the two rotor poles. The **Full step size** parameter is  $(\pi/2)/N_r$ .
- $R$  is the half-winding resistance. For example, it is the resistance between **A+** and **A0** ports.
- $L$  is the half-winding inductance. For example, it is the inductance between **A+** and **A0** ports.
- $R_m$  is the magnetizing resistance.
- $B$  is the rotational damping.
- $J$  is the inertia.
- $\omega$  is the rotor speed.
- $\theta$  is the rotor angle.
- $T_d$  is the detent torque amplitude.
- $T_e$  is the electrical torque.

If the initial rotor is zero or some multiple of  $(\pi/2)/N_r$ , the rotor is aligned with the A-phase winding. If a positive current flows from the **A+** port to the **A0** center tap port, then the stepper acts to stay aligned with the A-phase. Equivalently, a positive current flowing from the **A0** center tap port to the **A-** port also acts on the rotor to stay aligned with the A-phase.

The Unipolar Stepper Motor block produces a positive torque acting from the mechanical C to R ports for either of the following sequences. Both sequences assume the rotor initial angle is zero or some multiple of  $(\pi/2)/N_r$ .

Sequence	Center taps connected to ground	Center taps connected to positive supply
1	Positive current from <b>A+</b> to <b>A0</b>	Positive current from <b>A0</b> to <b>A-</b>
2	Positive current from <b>B+</b> to <b>B0</b>	Positive current from <b>B0</b> to <b>B-</b>
3	Positive current from <b>A-</b> to <b>A0</b>	Positive current from <b>A0</b> to <b>A+</b>
4	Positive current from <b>B-</b> to <b>B0</b>	Positive current from <b>B0</b> to <b>B+</b>

### Averaged Mode

If you set the **Simulation mode** parameter to Averaged, both for a Unipolar Stepper Motor block and for the Unipolar Stepper Motor Driver block that controls it, then the individual steps are not simulated. This can be a good way to speed up simulation. In Averaged mode, under non-slipping



conditions, the motor and driver are represented by a second-order linear system that tracks the specified step rate. The demanded step rate is determined directly from voltage across **A+** and **A-**. So, for example, a voltage of +10V across the **A+** and **A-** terminals is interpreted as a step rate demand of ten steps per second. See the Unipolar Stepper Motor Driver reference page for more information on how to connect the Unipolar Stepper Motor Driver to your step angle controller.

Averaged mode includes a slip estimator to predict whether the stepper motor would have slipped if running in Stepping simulation mode. Slip is predicted if the motor torque exceeds the **Vector of maximum torque values** parameter value for longer than one step period, the step period being determined from the current step rate demand. Upon detecting slip, the simulation will proceed or stop with an error, according to the **Action on slipping** parameter value. If you choose the action that lets the simulation continue, note that simulation results may be incorrect: when slipping occurs, the torque generated by the motor will not generally be the maximum available torque; the maximum torque is only achieved if the stepper controller detects slip and adjusts the step rate command accordingly.

The dynamics of the equivalent second-order system are determined from the values that you specify for the **Approximate total load inertia** and **Maximum step rate command** parameters. It is important that you set as accurate values as possible for these parameters, so that the step rate command is tracked, and the block does not generate false slipping warnings or errors.

If you run the motor in Averaged mode with the optional thermal ports exposed (see “Model Thermal Effects” on page 1-2531), then heat is added to the thermal ports assuming that the windings are always powered, even when the step rate command is zero. The block makes adjustments for half stepping and for reduced torque (and winding currents) at higher speeds. For these adjustments to be correct, the **Vector of maximum torque** parameter values must be correct. For half stepping, at zero speed the heat generated by the block is the average of that generated when stopped at a half step and at a full step.

If you simulate or predict slip, MathWorks recommends that you do some validation runs comparing Stepping and Averaged modes before using the averaged model representation for simulation studies.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- **No thermal port** — The block does not contain thermal ports.
- **Show thermal port** — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

### Assumptions and Limitations

- The model neglects magnetic saturation effects and any magnetic coupling between phases.
- When you select the **Start simulation from steady state** check box in the Simscape Solver Configuration block, this block will not initialize an **Initial rotor angle** value between  $-\pi$  and  $\pi$ .
- All four half-windings are assumed to be identical, and therefore have the same resistance temperature coefficient, alpha, and the same thermal mass.
- To use Averaged mode, the Unipolar Stepper Motor block must be directly connected to a Unipolar Stepper Motor Driver block also running in Averaged mode.

- The Averaged mode is an approximation, and exact step tracking compared to the Stepping mode should not be expected.
- Slip detection in Averaged mode is approximate, and depends on a good estimate for load inertia and maximum step rate. Incorrect values may result in false slip detection.
- When simulating slip in Averaged mode, it is assumed that the stepper motor controller adjusts the step rate command so as to achieve maximum possible torque.

## Ports

### Conserving

#### **A+ — Top A-phase electrical connection**

electrical

Electrical conserving port associated with the top A-phase electrical connection.

#### **A0 — A-phase center tap connection**

electrical

Electrical conserving port associated with the A-phase center tap connection.

#### **A- — Lower A-phase electrical connection**

electrical

Electrical conserving port associated with the lower A-phase electrical connection.

#### **B+ — Top B-phase electrical connection**

electrical

Electrical conserving port associated with the top B-phase electrical connection.

#### **B0 — B-phase center tap connection**

electrical

Electrical conserving port associated with the B-phase center tap connection.

#### **B- — Lower B-phase electrical connection**

electrical

Electrical conserving port associated with the lower B-phase electrical connection.

#### **C — Case**

mechanical

Mechanical rotational conserving port associated with the case.

#### **R — Rotor**

mechanical

Mechanical rotational conserving port associated with the rotor.

#### **HA+ — Thermal port for winding between A+ and A0**

thermal

Thermal port for winding between **A+** and **A0**. For more information, see “Model Thermal Effects” on page 1-2531.

#### **HA- — Thermal port for winding between A- and A0**

thermal

Thermal port for winding between **A-** and **A0**.

##### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

#### **HB+ — Thermal port for winding between B+ and B0**

thermal

Thermal port for winding between **B+** and **B0**.

##### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

#### **HB- — Thermal port for winding between B- and B0**

thermal

Thermal port for winding between **B-** and **B**.

##### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

#### **HR — Rotor thermal port**

thermal

Thermal port for rotor.

##### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

## **Parameters**

#### **Modeling option — Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

#### **Electrical Torque**

#### **Simulation mode — Simulation mode**

Stepping (default) | Averaged

Select Stepping or Averaged. Use Averaged only if the block is connected directly to a Unipolar Stepper Motor Driver block also running in Averaged mode.

#### **Vector of rotational speeds — Vector of rotational speeds**

[0, 1000, 3000] rpm (default)

Vector of rotational speeds at which to define maximum torque values, for slip prediction.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Averaged.

**Vector of maximum torque values – Vector of maximum torque values**

[2, 2, 1] N\*m (default)

Vector of maximum torque values, to be used for slip prediction in conjunction with the **Vector of rotational speeds** parameter. The maximum torque values are often given on a datasheet, and should correspond to the supply voltage and stepping type (half step or full step) specified in the driver.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Averaged.

**Action on slipping – Action upon detecting slip**

none (default) | warn | error

Select the action for the block to perform during simulation upon detecting slip:

- none — Continue simulation, limiting the load torque according to the **Vector of maximum torque values**.
- warn — Continue simulation, limiting the load torque according to the **Vector of maximum torque values**, and generate a warning that the rotor is slipping.
- error — Stop the simulation and generate an error message that the rotor is slipping.

Note that if you choose an action that lets the simulation continue, simulation results may be incorrect: when slipping occurs, the torque generated by the motor will not generally be the maximum available torque; the maximum torque is only achieved if the stepper controller detects slip and adjusts the step rate command accordingly.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Averaged.

**Approximate total load inertia – Approximate total load inertia**

1e-4 kg\*m<sup>2</sup> (default)

The approximate total load inertia, including the rotor inertia. This value is used to help predict when slipping will occur due to rapid acceleration demands.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Averaged.

**Maximum step rate command – Maximum step rate command**

10 Hz (default)

The maximum step rate that your system will command. It is used to determine a suitable bandwidth for the second order system approximation to the stepper motor and driver.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Averaged.

**Half-winding resistance — Half-winding resistance**

0.55 Ohm (default)

Half of the resistance of the A and B phase windings as measured between the **A+** and **A-**, and the **B+** and **B-** ports.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Stepping.

**Half-winding inductance — Half-winding inductance**

1.5e-3 H (default)

Half of the inductance of the A and B phase windings as measured between the **A+** and **A-**, and the **B+** and **B-** ports.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Stepping.

**Motor torque constant — Motor torque constant**

0.19 N\*m/A (default)

Motor torque constant  $K_m$ .

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Stepping.

**Detent torque — Detent torque**

0 N\*m (default)

The amplitude of the sinusoidal torque variation observed when rotating the shaft of the unpowered motor.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Stepping.

**Magnetizing resistance — Magnetizing resistance**

inf Ohm (default) | positive scalar

The total magnetizing resistance seen from each of the phase windings, for example across **A+** and **A0**. The value must be greater than zero. The default value is **Inf**, which implies that there are no iron losses.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter to Stepping.

**Full step size — Full step size**

1.8 deg (default)

Step size when changing the polarity of either the A or B phase current. The default value is 1.8°.

**Coefficient of coupling — Coefficient of coupling for mutual inductance**

0 (default)

Coefficient of coupling of the windings for mutual inductance.

### **Mechanical**

#### **Rotor inertia – Rotor inertia**

4.5e-5 kg\*m<sup>2</sup> (default)

Resistance of the rotor to change in motor motion. The value can be zero.

#### **Rotor damping – Rotor damping**

8.0e-4 N\*m/(rad/s) (default)

Energy dissipated by the rotor. The value can be zero.

#### **Initial rotor speed – Initial rotor speed**

0 rpm (default)

Speed of the rotor at the start of the simulation.

#### **Initial rotor angle – Initial rotor angle**

0 deg (default)

Angle of the rotor at the start of the simulation.

### **Temperature Dependence**

To enable these parameters, set **Modeling option** to Show thermal port.

#### **Resistance temperature coefficient – Resistance temperature coefficient**

3.93e-3 1/K (default)

Parameter  $\alpha$  in the equation defining resistance as a function of temperature, as described in “Thermal Model for Actuator Blocks”. It is assumed that all windings are made of the same material, and therefore have the same resistance temperature coefficient.

#### **Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter in the **Electrical Torque** setting to Stepping.

#### **Measurement temperature – Measurement temperature**

25 degC (default)

The temperature for which motor parameters are defined.

#### **Vector of maximum torque values at second measurement temperature – Vector of maximum torque values at second measurement temperature**

[1.7, 1.7, .85] N\*m (default)

Vector of maximum torque values at second measurement temperature.

#### **Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter in the **Electrical Torque** setting to Averaged.

**Second measurement temperature – Second measurement temperature**

125 degC (default)

Second measurement temperature.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter in the **Electrical Torque** setting to Averaged.

**Stepping mode – Stepping mode**

Full stepping (default) | Half stepping

Specify stepping mode.

**Dependencies**

This parameter is visible only when you set the **Simulation mode** parameter in the **Electrical Torque** setting to Averaged.

**Thermal Port**

To enable these parameters, set **Modeling option** to Show thermal port.

**Half-winding thermal mass – Half-winding thermal mass**

100 J/K (default)

The thermal mass for half of either the A or B winding. The thermal mass is the energy required to raise the temperature by one degree. It is assumed that all four half-windings have the same thermal mass.

**Half-winding initial temperatures, [T\_A+ T\_A- T\_B+ T\_B-] – Half-winding initial temperatures**

[25, 25, 25, 25] degC (default)

A 1 by 4 row vector defining the temperature of the four half-windings at the start of simulation.

**Rotor thermal mass – Rotor thermal mass**

50 J/K (default)

The thermal mass of the rotor, that is, the energy required to raise the temperature of the rotor by one degree.

**Rotor initial temperature – Rotor initial temperature**

25 degC (default)

The temperature of the rotor at the start of simulation.

**Percentage of magnetizing resistance associated with the rotor – Percentage of magnetizing resistance associated with the rotor**

90 (default)

The percentage of the magnetizing resistance associated with the magnetic path through the rotor. It determines how much of the iron loss heating is attributed to the rotor thermal port HR, and how much is attributed to the four winding thermal ports.

## **Version History**

**Introduced in R2012b**

### **References**

- [1] M. Bodson, J. N. Chiasson, R. T. Novotnak and R. B. Rekowski. "High-Performance Nonlinear Feedback Control of a Permanent Magnet Stepper Motor." IEEE Transactions on Control Systems Technology, Vol. 1, No. 1, March 1993.
- [2] P. P. Acarnley. *Stepping Motors: A Guide to Modern Theory and Practice*. New York: Peregrinus, 1982.
- [3] S.E. Lyshevski. *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC, 1999.

### **Extended Capabilities**

#### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

#### **See Also**

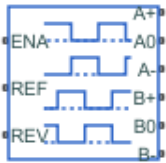
Stepper Motor | Unipolar Stepper Motor Driver



# Unipolar Stepper Motor Driver

Driver for unipolar stepper motor

**Library:** Simscape / Electrical / Electromechanical / Reluctance & Stepper



## Description

The Unipolar Stepper Motor Driver block 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 grounded in the appropriate sequence to create the stepping motion. The block initiates a step each time the voltage at the **ENA** port rises above the **Enable threshold voltage** parameter value.

If the voltage at the **REV** port is less than or equal to the **Reverse threshold voltage** parameter value, pulse *A* leads pulse *B* by 90 degrees. If the voltage at the **REV** port is greater than the **Reverse threshold voltage** value, pulse *B* leads pulse *A* by 90 degrees and the motor direction is reversed.

At time zero, **A-** and **B+** are grounded.

If you set the **Stepping mode** parameter to **Half stepping**, the Unipolar Stepper Motor Driver block can produce the output waveforms required for half stepping. In this mode, there is 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. At half steps, windings that are not powered are short-circuited. This approximates the effect of a freewheeling diode connected across the windings.

## Averaged Mode

If you set the **Simulation mode** parameter to **Averaged**, both for a Unipolar Stepper Motor Driver block and for the Unipolar Stepper Motor block connected to it, then the individual steps are not simulated. This can be a good way to speed up simulation. The **Averaged** mode assumes that the external controller provides a step rate demand. This step rate demand is determined from the voltage applied between the **ENA** and **REF** ports on the Unipolar Stepper Motor Driver block, by multiplying this voltage by the value of the **Step rate sensitivity** parameter. The rotation direction is set by the **REV** port in the same way as for the Stepping mode.

**Averaged** mode needs to communicate the step rate demand and also output voltage amplitude information to the Unipolar Stepper Motor block. To do this, the step rate demand is applied as an equivalent voltage across the **A+** and **A-** ports. Similarly the output voltage amplitude information is conveyed by applying a steady-state voltage across the **B+** and **B-** ports with value equal to the **Output voltage amplitude** parameter.

## Assumptions and Limitations

- To use Averaged mode, the Unipolar Stepper Motor Driver block must be directly connected to a Unipolar Stepper Motor block also running in Averaged mode.
- When changing from Stepping to Averaged mode and back, you will need to modify your upstream blocks that provide the input voltages to the Unipolar Stepper Motor Driver. One way to achieve this easily is to use Simulink variant subsystems.

## Ports

### Conserving

#### **A+** — Top A-phase electrical connection

electrical

Electrical conserving port associated with the top A-phase electrical connection.

#### **A0** — A-phase center tap connection

electrical

Electrical conserving port associated with the A-phase center tap connection.

#### **A-** — Lower A-phase electrical connection

electrical

Electrical conserving port associated with the lower A-phase electrical connection.

#### **B+** — Top B-phase electrical connection

electrical

Electrical conserving port associated with the top B-phase electrical connection.

#### **B0** — B-phase center tap connection

electrical

Electrical conserving port associated with the B-phase center tap connection.

#### **B-** — Lower B-phase electrical connection

electrical

Electrical conserving port associated with the lower B-phase electrical connection.

#### **ENA** — Triggering input step voltage

electrical

Electrical conserving port associated with the triggering input step voltage.

#### **REF** — Rotor

electrical

Electrical conserving port associated with the input floating reference voltage.

#### **REV** — Motor direction control voltage

electrical

Electrical conserving port associated with the voltage that controls motor direction.

## Parameters

### Simulation mode — Simulation mode

Stepping (default) | Averaged

Select **Stepping** or **Averaged**. Use **Averaged** only if the block is connected directly to a Unipolar Stepper Motor block also running in **Averaged** mode.

### Step rate sensitivity — Step rate sensitivity

10 Hz/V (default)

This parameter converts the voltage presented across the **ENA** and **REF** ports into a step rate demand.

#### Dependencies

This parameter is visible only when you set the **Simulation mode** parameter to **Averaged**.

### Enable threshold voltage — Enable threshold voltage

2.5 V (default)

When the voltage at the **ENA** port rises above this threshold, the Unipolar Stepper Motor Driver block initiates a step.

#### Dependencies

This parameter is visible only when you set the **Simulation mode** parameter to **Stepping**.

### Reverse threshold voltage — Reverse threshold voltage

2.5 V (default)

When the voltage at the **REV** port rises above this threshold, pulse *B* leads pulse *A* by 90 degrees, and the motor direction is reversed.

### Output voltage amplitude — Output voltage amplitude

10 V (default)

Amplitude of the output pulse trains.

### Stepping mode — Stepping mode

Full stepping (default) | Half stepping

Select **Full stepping** or **Half stepping**.

## Version History

Introduced in R2014a

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

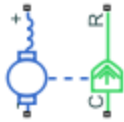
**See Also**

Controlled PWM Voltage | Unipolar Stepper Motor

# Universal Motor

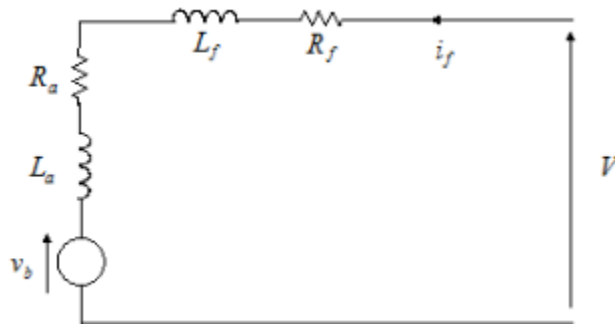
Universal (or series) motor with electrical and torque characteristics

**Library:** Simscape / Electrical / Electromechanical / Brushed Motors



## Description

The Universal Motor block represents the electrical and torque characteristics of a universal (or series) motor using the following equivalent circuit model.



Where:

- $R_a$  is the armature resistance.
- $L_a$  is the armature inductance.
- $R_f$  is the field winding resistance.
- $L_f$  is the field winding inductance.

When you set the **Model parameterization** parameter to **By equivalent circuit parameters**, you specify the equivalent circuit parameters for this model. The Universal Motor block computes the motor torque as follows:

- 1 The magnetic field in the motor induces the following back emf  $v_b$  in the armature:

$$v_b = L_{af} i_f \omega$$

where  $L_{af}$  is a constant of proportionality and  $\omega$  is the angular velocity.

- 2 The mechanical power is equal to the power reacted by the back emf:

$$P = v_b i_f = L_{af} i_f^2 \omega$$

- 3 The motor torque is:

$$T = P/\omega = L_{af} i_f^2$$

The torque-speed characteristic for the Universal Motor block model is related to the parameters in the preceding figure. When you set the **Model parameterization** parameter to By DC rated power, rated speed & maximum torque or By DC rated power, rated speed & electrical power, the block solves for the equivalent circuit parameters as follows:

- 1 For the steady-state torque-speed relationship when using a DC supply,  $L$  has no effect.
- 2 Sum the voltages around the loop:

$$V = (R_f + R_a)i_f + v_b = (R_f + R_a + L_{af}\omega)i_f$$

- 3 Solve the preceding equation for  $i_f$  and substitute this value into the equation for torque:

$$T = L_{af} \left( \frac{V}{R_f + R_a + L_{af}\omega} \right)^2$$

The block uses the rated speed and power to calculate the rated torque. The block uses the rated torque and rated speed values in the preceding equation plus the corresponding electrical power to determine values for  $R_f+R_a$  and  $L_{af}$ .

When you set the **Model parameterization** parameter to By AC rated power, rated speed, current & electrical power, then the block must include the inductive terms  $L_a$  and  $L_f$  in the model. This requires information about the RMS rated current and voltage for the total inductance.

The block models motor inertia  $J$  and damping  $B$  for all values of the **Model parameterization** parameter. The output torque is:

$$T_{load} = L_{af} \left( \frac{V}{R_f + R_a + L_{af}\omega} \right)^2 - J\dot{\omega} - B\omega$$

The block produces a positive torque acting from the mechanical C to R ports.

### Model Thermal Effects

You can expose thermal ports to model the effects of losses that convert power to heat. To expose the thermal ports, set the **Modeling option** parameter to either:

- No thermal port — The block does not contain thermal ports.
- Show thermal port — The block contains multiple thermal conserving ports.

For more information about using thermal ports in actuator blocks, see “Simulating Thermal Effects in Rotational and Translational Actuators”.

## Ports

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the universal motor positive terminal.

#### - — Negative terminal

electrical

Electrical conserving port associated with the universal motor negative terminal.

### **C – Motor case**

mechanical

Mechanical rotational conserving port associated with the universal motor case.

### **R – Motor rotor**

mechanical

Mechanical rotational conserving port associated with the universal motor rotor.

### **Hf – Field winding thermal port**

thermal

Field winding thermal port.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

### **Ha – Armature winding thermal port**

thermal

Armature winding thermal port.

#### **Dependencies**

To enable this port, set **Modeling option** to Show thermal port.

## **Parameters**

### **Modeling option – Whether to enable thermal ports**

No thermal port (default) | Show thermal port

Whether to enable the thermal ports of the block and model the effects of losses that convert power to heat.

### **Electrical Torque**

#### **Model parameterization – Block parameterization**

By DC rated power, rated speed & maximum torque (default) | By equivalent circuit parameters | By DC rated power, rated speed & electrical power | By AC rated power, rated speed, current & electrical power

Select one of the following methods for block parameterization:

- By equivalent circuit parameters – Provide electrical parameters for an equivalent circuit model of the motor.
- By DC rated power, rated speed & maximum torque – Provide DC power and speed parameters that the block converts to an equivalent circuit model of the motor.
- By DC rated power, rated speed & electrical power – Provide AC power and speed parameters that the block converts to an equivalent circuit model of the motor.
- By AC rated power, rated speed, current & electrical power – Provide AC power and speed parameters that the block converts to an equivalent circuit model of the motor.

**Total armature and field winding resistance – Total resistance of the armature and field winding**

132.8 Ohm (default)

Total resistance of the armature and field winding.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Back-emf constant – Back-emf constant**

0.1722 s\*V/rad/A (default)

The ratio of the voltage generated by the motor to the motor speed.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters.

**Total armature and field winding inductance – Total armature and field winding inductance**

0.525 H (default)

Total inductance of the armature and field winding. If you do not have information about this inductance, set the value of this parameter to a small, nonzero number.

---

**Note** You can set the **Total armature and field winding inductance** value to zero, but this only makes sense if you are driving the motor with a DC source.

---

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By equivalent circuit parameters, By DC rated power, rated speed & maximum torque, or By DC rated power, rated speed & electrical power.

**Rated speed (at rated load) – Rated speed**

6500 rpm (default)

Motor speed at the rated mechanical load.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By DC rated power, rated speed & maximum torque, By DC rated power, rated speed & electrical power, or By AC rated power, rated speed, current & electrical power.

**Rated load (mechanical power) – Rated load**

75 W (default)

The mechanical load for which the motor is rated to operate.



**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By DC rated power, rated speed & maximum torque, By DC rated power, rated speed & electrical power, or By AC rated power, rated speed, current & electrical power.

**Rated DC supply voltage — Rated DC supply voltage**

200 V (default)

The DC voltage at which the motor is rated to operate.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By DC rated power, rated speed & maximum torque or By DC rated power, rated speed & electrical power.

**Maximum (starting) torque — Maximum torque**

0.39 N\*m (default)

Maximum torque the motor produces.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By DC rated power, rated speed & maximum torque.

**Electrical power in at rated load — Electrical power at the rated mechanical power**

160 W (default)

The amount of electrical power the motor uses at the rated mechanical power.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By DC rated power, rated speed & electrical power or By AC rated power, rated speed, current & electrical power.

**RMS rated voltage — RMS rated voltage**

240 V (default)

RMS supply voltage when the motor operates on AC power.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By AC rated power, rated speed, current & electrical power.

**RMS current at rated load — RMS current at rated load**

0.8 A (default)

RMS current when the motor operates on AC power at the rated load.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By AC rated power, rated speed, current & electrical power.

**AC frequency – AC frequency**

50 Hz (default)

Frequency of the AC supply voltage.

**Dependencies**

This parameter is visible only when you set the **Model parameterization** parameter to By AC rated power, rated speed, current & electrical power.

**Mechanical****Rotor inertia – Rotor inertia**2e-4 kg\*m<sup>2</sup> (default)

Resistance of the rotor to change in motor motion. The value can be zero.

**Rotor damping – Rotor dissipated energy**

1e-6 N\*m/(rad/s) (default)

Energy dissipated by the rotor. The value can be zero.

**Initial rotor speed – Initial rotor speed**

0 rpm (default)

Speed of the rotor at the start of the simulation.

**Temperature Dependence**

This tab appears only for blocks with exposed thermal ports.

**Field to armature resistance ratio, R<sub>f</sub>/R<sub>a</sub> – Stator and rotor resistance temperature coefficients**

1 (default)

The ratio of the field to the armature resistance. This parameter is required only when showing the field and armature thermal ports. It is used to determine individual resistance values for the field and armature windings so that the thermal heat generated by the two resistors can be apportioned correctly.

**Resistance temperature coefficients, [alpha<sub>f</sub> alpha<sub>a</sub>] – Stator and rotor resistance temperature coefficients**

[.00393, .00393] 1/K (default)

A 1 by 2 row vector that defines the coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The first element corresponds to the field winding, and the second to the armature. The default value is for copper.

**Measurement temperature – Measurement temperature**

25 C (default)

The temperature for which motor parameters are defined.

**Thermal Port**

This tab appears only for blocks with exposed thermal ports.

**Thermal masses, [Mf Ma] – Thermal masses**

[100, 100] J/K (default)

A 1 by 2 row vector that defines the thermal mass for the field and armature windings. The thermal mass is the energy required to raise the temperature by one degree.

**Initial temperatures – Initial temperatures**

[25, 25] C (default)

A 1 by 2 row vector that defines the temperature of the field and armature thermal ports at the start of simulation.

## Version History

**Introduced in R2008a**

## References

- [1] Bolton, W. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 3rd edition Pearson Education, 2004..

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

DC Motor | Induction Machine (Single-Phase) | Motor & Drive (System Level) | Shunt Motor

# Variable Capacitor

Linear time-varying capacitor

**Library:** Simscape / Electrical / Passive



## Description

The Variable Capacitor block represents a linear time-varying capacitor. The block provides two options for the relationship between the current  $i$  through the capacitor and the voltage  $v$  across the device when the capacitance at port **C** is  $C$ . The **Equation** parameter determines which of the following equations the block uses:

- $$i = C \frac{dv}{dt}$$

Use the preceding equation when the capacitance is defined as the local gradient of the charge-voltage curve for a given voltage:

$$C(v) = \frac{dQ(v)}{dv}$$

- $$i = \frac{dC}{dt}v + C \frac{dv}{dt}$$

Use the preceding equation when the capacitance is defined as the ratio of the charge  $Q$  to the steady-state voltage:

$$C(v) = \frac{Q(v)}{v}$$

The block includes a resistor in series with the variable capacitor. You can use this resistor to represent the total ohmic connection resistance of the capacitor. You may need to use this resistor to prevent numerical issues for some circuit topologies, such as where a Variable Capacitor block is connected in parallel with another capacitor block that does not have a series resistance.

## Ports

### Input

#### **C — Capacitance**

physical signal

Physical signal input port associated with the capacitance.  $C$  must be finite and greater than zero

### Conserving

#### **+ — Positive terminal**

electrical

Electrical conserving port associated with the capacitor positive voltage.

#### - — Negative terminal

electrical

Electrical conserving port associated with the capacitor negative voltage.

## Parameters

### Equation — Equation for block capacitance

$I = C \cdot dV/dt$  (default) |  $I = C \cdot dV/dt + dC/dt \cdot V$

Select one of the following options for block capacitance:

- $I = C \cdot dV/dt$  — This equation assumes the capacitance is defined as the local gradient of the charge-voltage curve for a given voltage. This option is the default.
- $I = C \cdot dV/dt + dC/dt \cdot V$  — This equation assumes the capacitance is defined as the ratio of the charge to the steady-state voltage.

### Minimum capacitance $C > 0$ — Minimum capacitance

$1e-9$  F (default)

The lower limit on the value of the signal at port C. This limit prevents the signal from reaching a value that has no physical meaning. The default value is  $1e-09$  F.

### Series resistance — Series resistance

$1e-6$  Ohm (default)

The value of the resistance placed in series with the variable capacitor. The default value is  $1e-06$  Ohm.

### Initial voltage — Initial output voltage

$0$  V (default)

The output voltage at the start of the simulation.

#### Dependencies

This parameter is visible only when you select  $I = C \cdot dV/dt$  for the **Equation** parameter.

### Initial charge — Initial charge

$0$  C (default)

The charge at the start of the simulation.

#### Dependencies

This parameter is visible only when you select  $I = C \cdot dV/dt + dC/dt \cdot V$  for the **Equation** parameter.

## Version History

Introduced in R2008a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

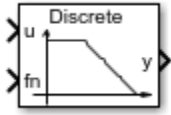
### **See Also**

Variable Inductor | Variable Resistor

# Variable-Frequency Second-Order Filter

Discrete-time or continuous-time variable-frequency second-order filter

**Library:** Simscape / Electrical / Control / General Control



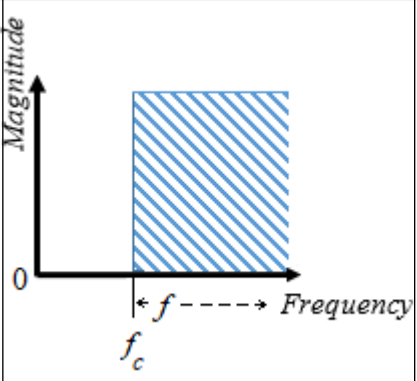
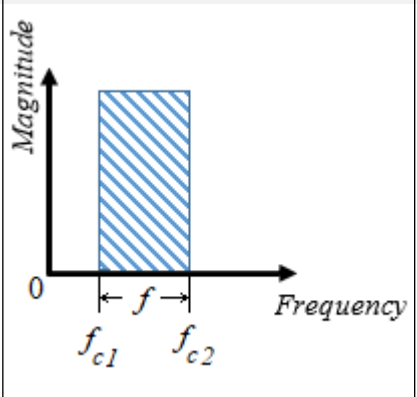
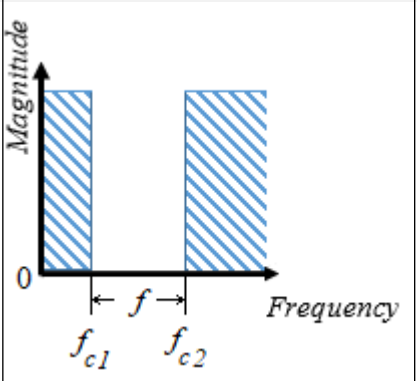
## Description

The Variable-Frequency Second-Order Filter block implements four different types of second-order filters, each with external frequency input.. Filters are useful for attenuating noise in measurement signals.

The block provides these filter types:

- Low pass — Allows signals,  $f$ , only in the range of frequencies below the cutoff frequency,  $f_c$ , to pass.
- High pass — Allows signals,  $f$ , only in the range of frequencies above the cutoff frequency,  $f_c$ , to pass.
- Band pass — Allows signals,  $f$ , only in the range of frequencies between two cutoff frequencies,  $f_{c1}$  and  $f_{c2}$ , to pass.
- Band stop — Prevents signals,  $f$ , only in the range of frequencies between two cutoff frequencies,  $f_{c1}$  and  $f_{c2}$ , from passing.

Filter Type	Frequency Range, $f$	
Low-Pass		$f < f_c$

Filter Type	Frequency Range, $f$	
<p><b>High-Pass</b></p>		$f > f_c$
<p><b>Band-Pass</b></p>		$f_{c1} < f < f_{c2}$
<p><b>Band-Stop</b></p>		$f_{c1} < f < f_{c2}$

**Equations**

The second order derivative state equation for the filter is:

$$\frac{d^2x}{dt^2} = u - 2\zeta\omega_n\frac{dx}{dt} - \omega_n^2x$$

Where:

- $x$  is the filter internal state.
- $u$  is the filter input.



- $\omega_n$  is the filter natural frequency.
- $\zeta$  is the filter damping factor.

For each filter type, the table maps the block output,  $y(x)$ , as a function of the internal state of the filter, to the  $s$ -domain transfer function,  $G(s)$ .

Filter Type	Output, $y(x)$	Transfer Function, $G(s)$
<b>Low-Pass</b>	$\omega_n^2 x$	$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>High-Pass</b>	$\frac{d^2 x}{dt^2}$	$\frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>Band-Pass</b>	$2\zeta\omega_n \frac{dx}{dt}$	$\frac{2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
<b>Band-Stop</b>	$\frac{d^2 x}{dt^2} + x$	$\frac{s^2 + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

For Initialization:

$$\dot{x}(0) = \left. \frac{dx}{dt} \right|_{t=0}$$

$$u(0) = u_1(0) + u_2(0)$$

$$u_1(0) = A_0 e^{j\varphi_0}$$

$$u_2(0) = b_0 e^{j\frac{\pi}{2}}$$

Where:

- $x(0)$  is the initial state of the filter.
- $u(0)$  is the initial input to the filter.
- $u_1(0)$  is the AC component of the steady-state initial input.
- $A_0$  is the initial amplitude.
- $\varphi_0$  is the initial phase.
- $u_2(0)$  is the DC component of the steady-state initial input.
- $b_0$  is the initial bias.

In the  $s$ -domain  $s = j\omega_0$ . Therefore, for the initial frequency,  $\omega_0$ :

$$\dot{x}(0) = \text{Im} \left( \frac{j\omega_0 u_1(0)}{-\omega_0^2 + j\omega_0 2\zeta\omega_n + \omega_n^2} \right)$$

$$x(0) = \text{Im} \left( \frac{\dot{x}(0)\omega_n^2}{j\omega_0} + u_2(0) \right)$$

## Ports

### Input

**u – Filter input**

scalar

Filter input.

Data Types: single | double

**fn – Natural frequency**

scalar

Natural frequency.

Data Types: single | double

### Output

**y – Filtered output**

scalar

Filtered output.

Data Types: single | double

## Parameters

### Main

**Filter type – Filter type**

Low-pass (default) | High-pass | Band-pass | Band-stop

Type of second-order filter.

**Initial natural frequency (Hz) – Initial natural frequency**

60 (default) | positive scalar

Natural frequency, in Hz, at the start of simulation.

### Initial Conditions

**Damping factor – Damping factor**

0.707 (default) | nonnegative scalar

Damping factor of the filter.

**Sample time (-1 for inherited) – Block sample time**

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

**Initial amplitude — Initial amplitude**

0 (default) | nonnegative scalar

Amplitude at the start of simulation.

**Initial phase (rad) — Initial phase**

0 (default) | nonnegative scalar

Phase, in rad, at the start of simulation.

**Initial frequency (Hz) — Initial frequency**

0 (default) | scalar

Frequency, in Hz, at the start of simulation.

**Initial bias — Initial bias**

0 (default) | nonnegative scalar

Bias at the start of simulation.

## Version History

Introduced in R2018b

## References

[1] Agarwal, A. and Lang, J. H. *Foundations of Analog and Digital Electronic Circuits*. New York: Elsevier, 2005.

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

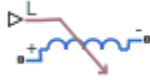
## See Also

Low-Pass Filter (Discrete or Continuous) | SM PSS1A | Second-Order Filter | Second-Order Low-Pass Filter (Discrete or Continuous) | Washout (Discrete or Continuous)

# Variable Inductor

Linear time-varying inductor

**Library:** Simscape / Electrical / Passive



## Description

The Variable Inductor block represents a linear time-varying inductor. The block provides two options for the relationship between the voltage  $v$  across the device and the current through the inductor  $i$  when the inductance at port L is  $L$ . The **Equation** parameter determines which of the following equations the block uses:

- $$v = \frac{dL}{dt}i + L\frac{di}{dt}$$

Use the preceding equation when the inductance is defined as the ratio of the magnetic flux  $\Phi$  to the steady-state current:

$$L(i) = \frac{\Phi(i)}{i}$$

- $$v = L\frac{di}{dt}$$

Use the preceding equation when the inductance is defined as the local gradient of the flux-current curve for a given current:

$$L(i) = \frac{d\Phi(i)}{di}$$

The block includes a conductance in parallel with the variable inductor. You can use the conductor to represent the total insulation conductance of the inductor. You may need to use the conductor to prevent numerical issues for some circuit topologies, such as where a Variable Inductor block is connected in series with another inductor block that does not have a parallel conductance.

## Ports

### Input

#### L — Inductance

physical signal

Physical signal input port associated with the inductance. L must be finite, greater than zero, and continuous (no step changes).

### Conserving

#### + — Positive terminal

electrical

Electrical conserving port associated with the inductor positive voltage.

**- – Negative terminal**

electrical

Electrical conserving port associated with the inductor negative voltage.

## Parameters

**Equation – Equation for block inductance**

$V = L \cdot dI/dt + dL/dt \cdot I$  (default) |  $V = L \cdot dI/dt$

Select one of the following options for block inductance:

- $V = L \cdot dI/dt + dL/dt \cdot I$  — This equation assumes the inductance is defined as the ratio of the magnetic flux to the steady-state current. This option is the default.
- $V = L \cdot dI/dt$  — This equation assumes the inductance is defined as the local gradient of the flux-current curve for a given current.

**Minimum inductance  $L > 0$  – Minimum inductance**

1e-6 H (default)

The lower limit on the value of the signal at port L. This limit prevents the signal from reaching a value that has no physical meaning.

**Parallel conductance – Parallel conductance**

1e-9 1/Ohm (default)

The value of the conductance placed in parallel with the variable inductor.

**Initial current – Initial current**

0 A (default)

The output current at the start of the simulation.

**Dependencies**

This parameter is visible only when you select  $V = L \cdot dI/dt$  for the **Equation** parameter.

**Initial magnetic flux – Initial magnetic flux**

0 Wb (default)

The magnetic flux at the start of the simulation.

**Dependencies**

This parameter is visible only when you select  $V = L \cdot dI/dt + dL/dt \cdot I$  for the **Equation** parameter.

## Version History

Introduced in R2008a

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

Variable Capacitor | Variable Resistor

# Varistor

Voltage-dependent resistor

**Library:** Simscape / Electrical / Passive



## Description

The Varistor block represents a voltage-dependent resistor (VDR). This component is also commonly known as a metal-oxide varistor (MOV). The block exhibits high resistance at low voltages and low resistance at high voltages.

You can protect parts of an electrical circuit from high-voltage surges by placing this block in parallel with them. When a surge occurs, the resistance of the varistor drops significantly, causing the current to be shunted through the varistor rather than through the circuit.

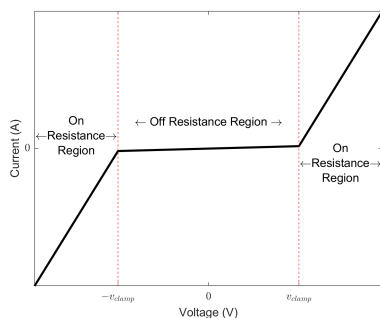
Use the **Parameterization** parameter to choose between two different behaviors for this block. The **Linear** option focuses on the on- and off-states of the varistor and uses a linear relationship between current and voltage in both regions. The **Power-law** option uses an exponential relationship between current and voltage in the initial on-state. This option also adds a third, linear region at higher voltages.

### Linear Parameterization

This parameterization option separates the voltage-current relationship into two linear regions:

- Off-region — resistance is high and current increases slowly with increasing voltage.
- On-region — resistance is low and current increases rapidly with increasing voltage.

This figure shows the voltage-current relationship across the on- and off-regions.



Use linear parameterization in one of these scenarios:

- You are modeling voltage surges close to the threshold voltage
- You expect your varistor to behave linearly in all regions

The voltage-current relationship for the linear varistor is:

$$i_{varistor} = \begin{cases} \frac{v_{varistor}}{R_{off}} & |v_{varistor}| < v_{clamp} \\ \frac{v_{varistor}}{R_{on}} + c_1 \text{sgn}(v_{varistor}) & |v_{varistor}| \geq v_{clamp} \end{cases}$$

where:

- $v_{varistor}$  and  $i_{varistor}$  are the varistor voltage and current, respectively.
- $v_{clamp}$  is the threshold voltage that separates the two regions of operation. Set this value using the **Clamping voltage** parameter.
- $R_{on}$  and  $R_{off}$  are the resistances in the on- and off-regions. Set these values using the **On resistance** and **Off resistance**, respectively.
- $c_1$  is a constant used to enforce current continuity between the two regions:

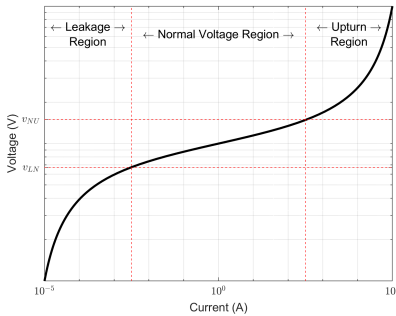
$$c_1 = v_{clamp} \left( \frac{1}{R_{off}} - \frac{1}{R_{on}} \right)$$

### Power-Law Parameterization

This parameterization option separates the voltage-current relationship into three regions:

- Leakage region — Resistance is high and current increases slowly with increasing voltage.
- Normal region — Resistance decreases exponentially with increasing voltage.
- Upturn region — Resistance is low and current increases rapidly with increasing voltage.

This figure shows the three regions of operation in log-log-scale.



Use power-law parameterization in one of these scenarios:

- You are modeling voltage surges across a large range of voltages
- You expect your varistor to behave exponentially in the first on-region

The voltage-current relationship for the power-law varistor is:

$$i_{varistor} = \begin{cases} \frac{v_{varistor}}{R_L} & |v_{varistor}| < v_{LN} \\ k(v_{varistor})^\alpha + c_1 & v_{LN} \leq |v_{varistor}| \leq v_{NU} \\ \frac{v_{varistor}}{R_U} + c_2 & |v_{varistor}| > v_{NU} \end{cases}$$



where:

- $v_{\text{varistor}}$  and  $i_{\text{varistor}}$  are the varistor voltage and current, respectively.
- $\alpha$  is the power-law exponent which determines the rate of current increase with voltage increase in the normal region. Set this value using the **Normal-mode power-law exponent** parameter.
- $v_{LN}$  and the  $v_{NU}$  are the threshold voltages corresponding to the leakage-normal and normal-upturn transition points. Set these values using the **Leakage to normal voltage transition** and **Normal to upton voltage transition** parameters, respectively.
- $R_L$  and  $R_U$  are the resistances in the leakage- and upton-regions. Set these values using the **Leakage-mode resistance** and **Upton-mode resistance** parameters, respectively.
- $k$ ,  $c_1$ , and  $c_2$  are constants used to enforce current continuity between the regions:

$$k = \frac{1}{\alpha R_U v_{NU}^{\alpha-1}},$$

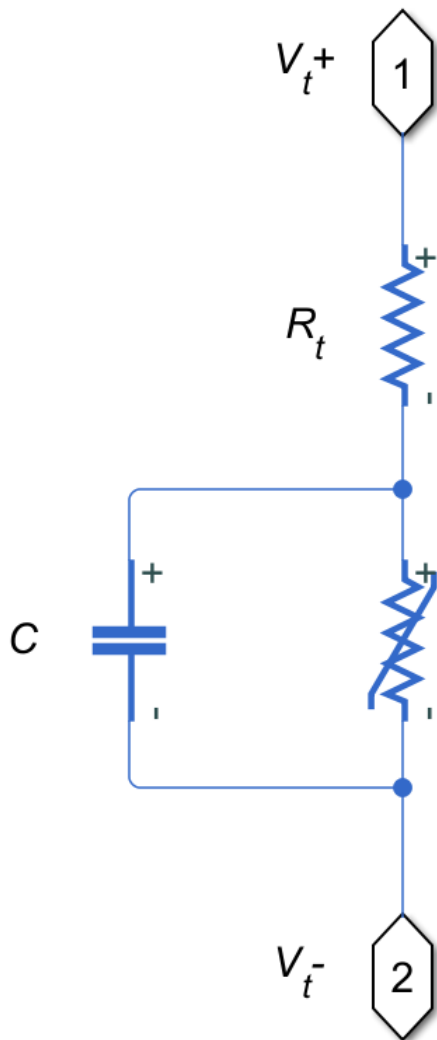
$$c_1 = \frac{v_{LN}}{R_L} - \frac{v_{LN}^\alpha}{\alpha R_U v_{NU}^{\alpha-1}},$$

and

$$c_2 = \frac{1}{\alpha R_U v_{NU}^{\alpha-1}} (v_{NU}^\alpha - v_{LN}^\alpha) - \frac{v_{NU}}{R_U} + \frac{v_{LN}}{R_L}.$$

### Equivalent Circuit

In addition to the varistor equations, you can also specify a constant terminal resistance  $R_t$  and device capacitance  $C$ . This figure shows the equivalent circuit for the varistor in either parameterization mode.



## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the varistor positive terminal.

#### - – Negative terminal

electrical

Electrical conserving port associated with the varistor negative terminal.

## Parameters

### Parameterization — Varistor operation mode

Linear (default) | Power-law

Choose how the varistor resistance changes with increasing voltages:

- **Linear** — Two regions. The low-voltage region has high resistance and the high-voltage region has low resistance.
- **Power-law** — Three regions. The leakage region has high resistance. The normal region has exponentially decreasing resistance. The upturn region has low resistance.

### Clamping voltage — Threshold voltage

260 (default) | positive number

Transition point voltage,  $v_{clamp}$ , between the off- and on-states of the linear varistor.

#### Dependencies

Enabled when the **Parameterization** parameter is set to **Linear**.

### Off resistance — Low-voltage resistance

3e8 (default) | positive number

Low-voltage resistance,  $R_{off}$ , of the varistor in the off-state.

#### Dependencies

Enabled when the **Parameterization** parameter is set to **Linear**.

### On resistance — High-voltage resistance

1 (default) | positive number

High-voltage resistance,  $R_{on}$ , of the varistor in the on-state.

#### Dependencies

Enabled when the **Parameterization** parameter is set to **Linear**.

### Leakage to normal voltage transition — First threshold voltage

130 (default) | positive number

Transition point voltage,  $v_{LN}$ , between the leakage and normal regions of the power-law varistor.

#### Dependencies

Enabled when the **Parameterization** parameter is set to **Power-law**.

### Normal to upturn voltage transition — Second threshold voltage

300 (default) | positive number

Transition point voltage,  $v_{NU}$ , between the normal and upturn regions of the power-law varistor.

#### Dependencies

Enabled when the **Parameterization** parameter is set to **Power-law**.

**Leakage-mode resistance — Low-voltage resistance**

3e8 (default) | positive number

Low-voltage resistance,  $R_L$ , of the varistor in the leakage region.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Power-law.

**Normal-mode power-law exponent — Mid-voltage resistance**

45 (default) | positive number

Exponent that determines rate of current increase with voltage increase of the varistor in the normal region.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Power-law.

**Upturn-mode resistance — High-voltage resistance**

0.07 (default) | positive number

High-voltage resistance,  $R_U$ , of the varistor in the upturn region.

**Dependencies**

Enabled when the **Parameterization** parameter is set to Power-law.

**Terminal resistance — Terminal resistance**

100e-6 (default) | non-negative number

Small, constant resistance in series with the varistor. Set this value to zero to remove the resistance from the equivalent circuit.

**Capacitance — Parallel capacitance**

4.4 (default) | non-negative number

Capacitor in parallel with the varistor. Set this value to zero to remove the capacitor from the equivalent circuit.

## Version History

Introduced in R2018a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

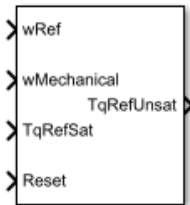
**See Also**

Variable Resistor | Resistor

# Velocity Controller

Discrete-time velocity controller

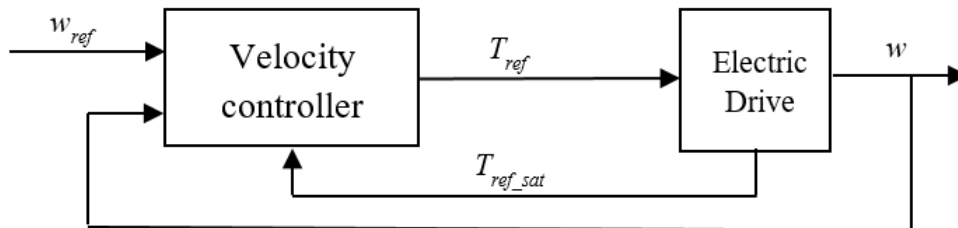
**Library:** Simscape / Electrical / Control / General Machine Control



## Description

The Velocity Controller block implements a velocity controller in discrete-time.

You provide measured and reference rotor velocities ( $w$  and  $w_{ref}$ ) as inputs to the block. The block then outputs a reference torque  $T_{ref}$  for an electric drive.



To prevent windup in the integrator, feed the saturated reference torque  $T_{ref\_sat}$  from the electric drive back to the velocity controller.

## Equations

You can control the rotor angular velocity with discrete sample time  $T_s$  using one of three common approaches:

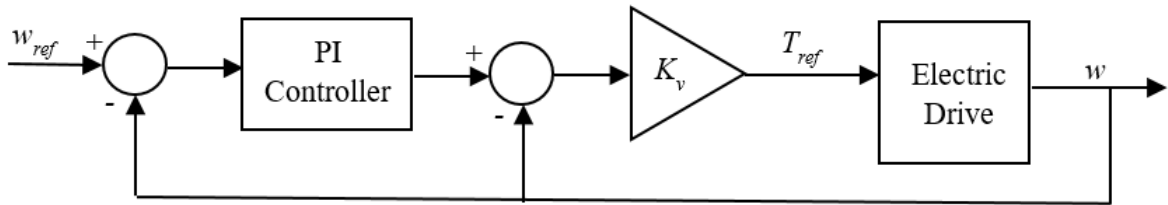
- Proportional-integral (PI) control, with proportional and integral gains  $K_{p\_w}$  and  $K_{i\_w}$ :

$$T_{ref} = \left( K_{p\_w} + K_{i\_w} \frac{T_s z}{z-1} \right) (w_{ref} - w)$$

- Proportional (P) control, with proportional gain  $K_{p\_w}$ :

$$T_{ref} = K_{p\_w} (w_{ref} - w)$$

- P-PI control characterized by a double velocity feedback loop as shown in the following figure:



Here, the PI Controller block is structured as in the PI control strategy, and  $K_v$  is the proportional gain for a P controller.

### Zero Cancellation

Using PI control results in a zero in the closed-loop transfer function, which can result in undesired overshoot in the closed-loop response. This zero can be canceled by introducing a zero-cancellation block in the feedforward path. The zero cancellation transfer function in discrete time is

$$G_{ZC\_w}(z) = \frac{\frac{T_s K_{i\_w}}{K_{p\_w}}}{z + \left( \frac{T_s - \frac{K_{p\_w}}{K_{i\_w}}}{\frac{K_{p\_w}}{K_{i\_w}}} \right)}$$

### Ports

#### Input

##### wRef – Desired velocity

scalar

Desired or reference velocity, in rad/s.

Data Types: single | double

##### wMechanical – Actual velocity

scalar

Measured mechanical velocity, in rad/s.

Data Types: single | double

##### TqRefSat – Saturated reference torque

scalar

Saturated torque reference used for integral anti-windup gain, in N\*m.

Data Types: single | double

##### Reset – Integral reset

scalar

External reset signal (rising edge) for the integrator.

Data Types: single | double

## Output

### TqRefUnsat — Unsaturated desired torque

scalar

Unsaturated reference torque, in N\*m.

Data Types: `single` | `double`

## Parameters

### Control type — Control model

`PI control` (default) | `P control` | `P-PI control`

Type of controller:

- `PI control` — Proportional-integral control using a single feedback loop
- `P control` — Proportional-integral control using a single feedback loop
- `P-PI control` — Proportional and proportional-integral control using a double feedback loop

### Dependencies

The **Control type** options affect the visibility or configurability of these parameters:

- **Controller integral gain**
- **P controller proportional gain**
- **Anti-windup gain**
- **Integral anti-windup gain**
- **Sample time (-1 for inherited)**
- **Enable zero cancellation**

### Controller proportional gain — Proportional gain

1 (default) | positive scalar

Proportional gain for the:

- PI controller
- P controller in the single-loop control model
- PI controller in the P-PI controller

### Controller integral gain — Integral gain

1 (default) | positive scalar

Integral gain for the PI or P-PI controller.

### Dependencies

This parameter is visible only when the **Control type** is set to `PI control` or `P-PI control`.

### P controller proportional gain — P proportional gain

1 (default) | positive scalar

Proportional gain for the P controller in the P-PI controller.

**Dependencies**

This parameter is visible only when the **Control type** is set to P-PI control.

**Integral anti-windup gain — PI anti-windup gain**

1 (default) | positive scalar

Anti-windup gain for the PI controller.

**Dependencies**

This parameter is visible only when the **Control type** is set to PI control or P-PI control.

**Sample time (-1 for inherited) — Block sample time**

-1 (default) | positive scalar

Time, in s, between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

If this block is inside a triggered subsystem, inherit the sample time by setting this parameter to -1. If this block is in a continuous variable-step model, specify the sample time explicitly using a positive scalar.

**Dependencies**

This parameter is visible only when the **Control type** is set to PI control or P-PI control.

**Discretization sample time — Sample time for discretization**

0.001 (default) | positive scalar

Time, in s, between consecutive discretizations. Discretization is required for zero cancellation.

**Dependencies**

This parameter is only visible when all these conditions are met:

- **Control type** is set to PI control or P-PI control.
- **Sample time** is set to -1.
- **Enable zero cancellation** is selected .

**Enable zero cancellation — Feedforward zero-cancellation**

off (default) | on

Option to use zero cancellation on the feedforward path.

**Dependencies**

The **Enable zero cancellation** parameter is visible only when **Control type** is set to PI control or P-PI control.

The **Discretization sample time** parameter is only visible when **Enable zero cancellation** is selected .



## Version History

Introduced in R2017b

## References

- [1] Naouar, M. W., A. A. Naassani, E. Monmasson, and I. Slama-Belkhodja. "FPGA-based predictive current controller for synchronous machine speed drive." *IEEE Transactions on Power Electronics*. Vol. 23, Number 4, 2008, pp. 2115-2126.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## See Also

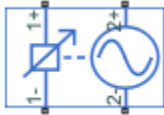
### Blocks

SM Current Reference Generator | SM Current Controller

# Voltage-Controlled Oscillator

Behavioral model of voltage-controlled oscillator

**Library:** Simscape / Electrical / Integrated Circuits



## Description

The Voltage-Controlled Oscillator block provides a behavioral model of a voltage-controlled oscillator (VCO). The output voltage is defined by the following equations:

$$v_{\text{lim}} = \begin{cases} v_{\text{min}} & \text{for } v_{\text{in}} < v_{\text{min}} \\ v_{\text{in}} & \text{for } v_{\text{min}} \leq v_{\text{in}} \leq v_{\text{max}} \\ v_{\text{max}} & \text{for } v_{\text{in}} > v_{\text{max}} \end{cases}$$

$$\dot{\Phi} = 2\pi F(v_{\text{lim}})$$

$$v_{\text{out}} = A \sin(2\pi f_{\text{nom}} t + \Phi) - i_{\text{out}} R_{\text{out}}$$

where:

- $v_{\text{in}}$  is the voltage applied across the 1+ and 1- ports.
- $v_{\text{out}}$  is the voltage across the 2+ and 2- ports.
- $f_{\text{nom}}$  is the oscillator frequency when the input control voltage is  $v_{\text{nom}}$ .
- $F$  is a linear function of  $v_{\text{lim}}$  or a lookup table function of  $v_{\text{lim}}$ .
- $A$  is the output voltage peak amplitude.
- $t$  is simulation time.
- $i_{\text{out}}$  is the output current.
- $R_{\text{out}}$  is the output resistance.

If you choose **Linear** for the **Frequency dependence on input voltage** parameter, then the function  $F$  is given by:

$$F = f_{\text{nom}} + k(v_{\text{lim}} - v_{\text{nom}})$$

where  $k$  is the rate of change of frequency with input voltage.

If you choose **Tabulated** for the **Frequency dependence on input voltage** parameter, then the function  $F$  is defined by the vectors of input voltages and corresponding output frequency deviations from nominal that you supply. The values for  $v_{\text{min}}$  and  $v_{\text{max}}$  are the first and the last values of the input voltage vector.

You can model the time delay between a change in the input control voltage and the oscillator frequency. Do this by modeling a first-order dynamic between  $v_{lim}$  and the value passed to the function  $F$ .

## Ports

### Conserving

#### 1+ — Positive input voltage

electrical

Electrical conserving port associated with the oscillator positive input voltage.

#### 1- — Negative input voltage

electrical

Electrical conserving port associated with the oscillator negative input voltage.

#### 2+ — Positive output voltage

electrical

Electrical conserving port associated with the oscillator positive output voltage.

#### 2- — Negative output voltage

electrical

Electrical conserving port associated with the oscillator negative output voltage.

## Parameters

### Frequency

#### Frequency dependence on input voltage — Block parameterization

Linear (default) | Tabulated

Select one of the following methods for block parameterization:

- **Linear** — Define a linear function by specifying the rate of change of frequency with input voltage. This is the default option.
- **Tabulated** — Provide the vectors of input voltages and corresponding output frequency deviations from nominal. The block determines the frequency deviation by table lookup based on these values.

#### Nominal frequency — Nominal frequency

1000 Hz (default)

The oscillator frequency when the input control voltage is at the nominal value.

#### Input voltage corresponding to nominal frequency — Input voltage corresponding to nominal frequency

0.5 V (default)

The input voltage corresponding to the oscillator nominal frequency.

**Dependencies**

This parameter is visible only when you select Linear for the **Frequency dependence on input voltage** parameter.

**Rate of change of frequency with input voltage — Rate of change of frequency with input voltage**

1 Hz/V (default)

The linear coefficient defining the rate of change of frequency depending on input voltage.

**Dependencies**

This parameter is visible only when you select Linear for the **Frequency dependence on input voltage** parameter.

**Minimum input voltage — Minimum input voltage**

0 V (default)

The minimum input voltage that affects VCO frequency.

**Dependencies**

This parameter is visible only when you select Linear for the **Frequency dependence on input voltage** parameter.

**Maximum input voltage — Maximum input voltage**

1 V (default)

The maximum input voltage that affects VCO frequency.

**Dependencies**

This parameter is visible only when you select Linear for the **Frequency dependence on input voltage** parameter.

**Input voltage vector — Input voltage vector**

[0, .2, .4, .6, .8, 1] V (default)

The vector of voltages for the tabulated VCO frequency.

**Dependencies**

This parameter is visible only when you select Tabulated for the **Frequency dependence on input voltage** parameter.

**Frequency deviation from nominal — Frequency deviation from nominal**

[-1000, -329, -51, 162, 342, 500] Hz (default)

The corresponding vector of VCO frequencies relative to the nominal frequency.

**Dependencies**

This parameter is visible only when you select Tabulated for the **Frequency dependence on input voltage** parameter.

## Electrical Characteristics

### Output voltage peak amplitude — Output voltage peak amplitude

1 V (default)

The peak amplitude of the voltage across the 2+ and 2- terminals.

### Input resistance — Input resistance

inf Ohm (default)

The resistance seen at the 1+ and 1- terminals.

### Output resistance — Output resistance

0 Ohm (default)

The value of the series output resistance.

## Dynamics

### Dynamics — Dynamics specification

No dynamics (default) | Model frequency tracking dynamics

Select one of the following methods for specifying dynamics:

- **No dynamics** — Do not model the time delay between a change in the input control voltage and the oscillator frequency. This is the default option.
- **Model frequency tracking dynamics** — Model a first order dynamic between the input control voltage and the oscillator frequency.

### Frequency tracking time constant — Frequency tracking time constant

1e-3 s (default)

Time constant for the first-order filter that delays the measured input control voltage, to model the lag between a change in VCO demanded frequency and the resulting VCO frequency.

### Dependencies

This parameter is visible only when you select **Model frequency tracking dynamics** for the **Dynamics** parameter.

### Initial frequency — Initial frequency

1000 Hz (default)

The initial VCO output frequency.

## Version History

Introduced in R2013b

## Extended Capabilities

### C/C++ Code Generation

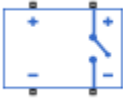
Generate C and C++ code using Simulink® Coder™.

## **See Also**

# Voltage-Controlled Switch

Voltage-controlled switch with hysteresis

**Library:** Simscape / Electrical / Additional Components / SPICE Passives



## Description

The Voltage-Controlled Switch block represents the electrical characteristics of a switch whose state is controlled by the voltage across the input ports (the controlling voltage).

This block models either a variable-resistance or a short-transition switch. For a variable-resistance switch, set the **Switch model** parameter to Smooth transition between  $V_{on}$  and  $V_{off}$ . For a short-transition switch, set **Switch model** to Abrupt transition after delay.

### Short-transition Switch

In a short-transition switch, the transition between the off and on states is instantaneous:

- When the controlling voltage is greater than or equal to the sum of the **Threshold voltage,  $V_T$**  and **Hysteresis voltage,  $V_H$**  parameter values, the switch is closed and has a resistance equal to the **On resistance,  $R_{ON}$**  parameter value.
- When the controlling voltage is less than the **Threshold voltage,  $V_T$**  parameter value minus the **Hysteresis voltage,  $V_H$**  parameter value, the switch is open and has a resistance equal to the **Off resistance,  $R_{OFF}$**  parameter value.
- When the controlling voltage is greater than or less than the **Threshold voltage,  $V_T$**  parameter value by an amount less than or equal to the **Hysteresis voltage,  $V_H$**  parameter value, the voltage is in the crossover region and the state of the switch remains unchanged.

---

**Note** If the **Hysteresis voltage,  $V_H$**  parameter value is less than 0, the block models a variable-resistance switch independently of the value you set for the **Switch model** parameter.

---

### Variable-resistance Switch

In a variable-resistance switch, the transition between the off and on states is exponential:

- When the **Control voltage for on state,  $V_{ON}$**  is greater than the **Control voltage for off state,  $V_{OFF}$** :
  - If the controlling voltage is greater than the **Control voltage for on state,  $V_{ON}$**  parameter value, the switch is closed and has a resistance equal to the **On resistance,  $R_{ON}$**  parameter value.

- If the controlling voltage is less than the **Control voltage for off state, VOFF** parameter value, the switch is open and has a resistance equal to the **Off resistance, ROFF** parameter value.
- If the controlling voltage is greater than the **Control voltage for off state, VOFF** parameter value or less than the **Control voltage for on state, VON** parameter value, the resistance is defined by:

$$R_s = \exp \left[ L_m + 3L_r \left( \frac{V_c - V_m}{2V_d} \right) - 2L_r \left( \frac{(V_c - V_m)^3}{V_d^3} \right) \right]$$

where:

- $R_s$  is the switch resistance.
  - $L_m = \ln \left[ (R_{ON} * R_{OFF})^{\frac{1}{2}} \right]$  is the log-mean of resistor values.
  - $L_r = \ln \left( \frac{R_{ON}}{R_{OFF}} \right)$  is the log-ratio of resistor values.
  - $V_c$  is the controlling voltage.
  - $V_m = \frac{V_{ON} + V_{OFF}}{2}$  is the mean of the control voltages.
  - $V_d = V_{ON} - V_{OFF}$  is the difference between the control voltages.
- When the **Control voltage for on state, VON** is less than the **Control voltage for off state, VOFF**:
    - If the controlling voltage is less than the **Control voltage for on state, VON** parameter value, the switch is closed and has a resistance equal to the **On resistance, RON** parameter value.
    - If the controlling voltage is greater than the **Control voltage for off state, VOFF** parameter value, the switch is open and has a resistance equal to the **Off resistance, ROFF** parameter value.
    - If the controlling voltage is less than the **Control voltage for off state, VOFF** parameter value or greater than the **Control voltage for on state, VON** parameter value, the resistance is defined by:

$$R_s = \exp \left[ L_m + 3L_r \left( \frac{I_c - I_m}{2I_d} \right) - 2L_r \left( \frac{(I_c - I_m)^3}{I_d^3} \right) \right]$$

## Assumptions and Limitations

The block output resistance model is discontinuous during switching. The discontinuity might cause numerical issues. Try the following actions to resolve the issues:

- Set the **On resistance, RON** and **Off resistance, ROFF** parameter values to keep  $R_{ON}$  divided by  $R_{OFF}$  as large as possible, and more than  $1e+12$ .
- Increase the **Hysteresis voltage, VH** parameter value to reduce switch chatter.
- If a variable-step solver is employed, decrease the **Max step size** parameter value (in the Configuration Parameters block dialog box).



---

**Note** This increases the simulation time.

---

## Ports

### Conserving

#### 1+ — Positive input port

electrical

Electrical conserving port associated with the voltage controlled switch positive input.

#### 1- — Negative input port

electrical

Electrical conserving port associated with the voltage controlled switch negative input.

#### 2+ — Positive output port

electrical

Electrical conserving port associated with the voltage controlled switch positive output.

#### 2- — Negative output port

electrical

Electrical conserving port associated with the voltage controlled switch negative output.

## Parameters

### Switch model — Switch model

Abrupt transition after delay (default) | Smooth transition between  $I_{on}$  and  $I_{off}$

Whether the transition between on state and off state current is instantaneous or not. To model this block as a variable-resistance switch, select Smooth transition between  $I_{on}$  and  $I_{off}$ . To model this block as a short-transition switch, select Abrupt transition after delay.

### Threshold voltage, VT — Threshold voltage

0 V (default)

Voltage above which the block interprets the controlling voltage as HIGH.

---

**Note** The controlling voltage must differ from the threshold voltage by at least the **Hysteresis voltage, VH** parameter value to change the state of the switch.

---

### Dependencies

To enable this parameter, set **Switch model** to Abrupt transition after delay.

### Hysteresis voltage, VH — Hysteresis voltage

0 V (default)

Amount by which the controlling voltage must exceed or fall below the **Threshold voltage, VT** parameter value to change the state of the switch.

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

**Control voltage for on state, VON – Control voltage for on state**

1e-3 V (default)

Switch control voltage for the on state.

**Dependencies**

To enable this parameter, set **Switch model** to Smooth transition between Von and Voff.

**Control voltage for off state, IOFF – Control voltage for off state**

0 V (default)

Switch control voltage for the off state.

**Dependencies**

To enable this parameter, set **Switch model** to Smooth transition between Von and Voff.

**On resistance, RON – Closed switch resistance**

1 Ohm (default)

Resistance of the switch when it is closed.

**Use 1/GMIN as Off resistance – Whether to use 1/GMIN as off resistance**

No (default) | Yes

Whether to automatically calculate the off resistance of this switch as 1/GMIN.

---

**Note** The value of the parallel conductance, *GMIN*, depends on whether your model contains an Environment Parameters block.

By default, *GMIN* matches the **GMIN** parameter of the Environment Parameters block, whose default value is 1e-12. To change *GMIN*, add an Environment Parameters block to your model and set the **GMIN** parameter to the desired value.

---

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

**Off resistance, ROFF – Open switch resistance**

1e12 Ohm (default) | 1e6

Resistance of the switch when it is open.

**Dependencies**

To enable this parameter either:

- Set **Switch model** to Abrupt transition after delay and **Use 1/GMIN as Off resistance** to No. The default value is 1e12.
- Set **Switch model** to Smooth transition between Von and Voff. The default value is 1e6.

, .

**Time delay, TD — Time delay**

0 s (default)

Time delay after which the switch transition takes place.

**Dependencies**

To enable this parameter, set **Switch model** to Abrupt transition after delay.

## Version History

Introduced in R2009a

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

## See Also

**Simscape Blocks**

Environment Parameters | Current-Controlled Switch

**Functions**

subcircuit2ssc

**Topics**

“Additional Parameterization Workflows”

“Converting a SPICE Netlist to Simscape Blocks”

“Parameterize an Exponential Diode from SPICE Netlist”

# Voltage Source

Voltage source with optional DC, AC and noise components

**Library:** Simscape / Electrical / Sources



## Description

The Voltage Source block implements a voltage source with DC, AC, and noise components. The voltage across the + and - terminals is given by:

$$v = v_{DC} + v_{AC}\sin(2\pi ft + \phi) + v_N$$

where:

- $v_{DC}$  is the steady-state DC voltage component.
- $v_{AC}$  is the amplitude of the AC voltage component.
- $f$  is the frequency of the AC component.
- $\phi$  is the phase offset of the AC component.
- $v_N$  is the noise voltage.

You can configure your source as DC-only, AC-only, or a combination of both. By default, both AC and DC components are set to 0. Define the AC/DC voltage by specifying nonzero parameter values after placing the block in your model.

The noise component is also optional. If you set the **Noise mode** parameter to Enabled, then the added noise voltage is given by:

$$v_N = \sqrt{P_v/2} \frac{N(0, 1)}{\sqrt{h}}$$

where:

- $P_v$  is the single-sided noise power spectral density for a 1 ohm load, in  $V^2/\text{Hz}$ .
- $N$  is a Gaussian random number with zero mean and standard deviation of one.
- $h$  is the sampling interval.

By default, the **Noise mode** parameter is set to Disabled, and the voltage source generates no thermal noise.

## Noise Options

The block generates Gaussian noise by using the PS Random Number source in the Simscape Foundation library. You can control the random number seed by setting the **Repeatability** parameter:

- **Not repeatable** — Every time you simulate your model, the block resets the random seed using the MATLAB random number generator:

```
seed = randi(2^32-1);
```

- **Repeatable** — The block automatically generates a seed value and stores it inside the block, to always start the simulation with the same random number. This auto-generated seed value is set when you add a Voltage Source block from the block library to the model. When you make a new copy of the Voltage Source block from an existing one in a model, a new seed value is generated. The block sets the value using the MATLAB random number generator command shown above.
- **Specify seed** — If you select this option, the additional **Seed** parameter lets you directly specify the random number seed value.

## Assumptions and Limitations

- Simulating with noise enabled slows down simulation. Choose the sample time ( $h$ ) so that noise is generated only at frequencies of interest, and not higher.

## Ports

### Conserving

#### + — Positive port

electrical

Electrical conserving port associated with the voltage source positive terminal

#### - — Negative port

electrical

Electrical conserving port associated with the voltage source negative terminal

## Parameters

### DC & AC Components

#### DC voltage — DC output voltage

0 V (default)

DC component of the output voltage. Specify a nonzero value to add a DC component to the voltage source.

#### AC voltage peak amplitude — AC voltage amplitude

0 V (default)

Amplitude of the AC component of the output voltage. Specify a nonzero value to add an AC component to the voltage source.

#### AC voltage phase shift — AC voltage phase offset

0 rad (default)

Phase offset of the AC component of the output voltage.

#### AC voltage frequency — AC voltage frequency

60 Hz (default)

Frequency of the AC component of the output voltage, in Hz.

## Noise

### Noise mode — Noise option

Disabled (default) | Enabled

Select the noise option:

- **Disabled** — No noise is produced by the voltage source.
- **Enabled** — The voltage source generates thermal noise, and the associated parameters become visible on the **Noise** tab.

### Power spectral density — Power spectral density

0 V<sup>2</sup>/Hz (default)

The single-sided spectrum noise power. Strictly-speaking, this is a density function for the square of the voltage, commonly thought of as a power into a 1 ohm load, and therefore the units are V<sup>2</sup>/Hz. To avoid this unit ambiguity, some datasheets quote noise voltage as a noise density with units of V/√Hz. In this case, you should enter the square of the noise density quoted in the datasheet as the parameter value.

### Repeatability — Noise control option

Not repeatable (default) | Repeatable | Specify seed

Select the noise control option:

- **Not repeatable** — The random sequence used for noise generation is not repeatable.
- **Repeatable** — The random sequence used for noise generation is repeatable, with a system-generated seed.
- **Specify seed** — The random sequence used for noise generation is repeatable, and you control the seed by using the **Seed** parameter.

### Auto-generated seed used for repeatable option — Auto-generated random number seed

1964622691 (default)

Random number seed stored inside the block to make the random sequence repeatable. The parameter value is automatically generated using the MATLAB random number generator command. You can modify this parameter value, but it gets overwritten by a new random value if you copy the block to another block in the model. Therefore, if you want to control the seed of the random sequence, use the **Specify seed** option for the **Repeatability** parameter and specify the desired seed value using the **Seed** parameter.

### Dependencies

This parameter is visible only when you select **Repeatable** for the **Repeatability** parameter.

### Seed — Random number seed

0 (default)

Random number seed used by the noise random number generator.

### Dependencies

This parameter is visible only when you select **Specify seed** for the **Repeatability** parameter.

**Sample time – Sample time**

1e-3 s (default)

Defines the rate at which the noise source is sampled. Choose it to reflect the frequencies of interest in your model. Making the sample time too small will unnecessarily slow down your simulation.

**Version History****Introduced in R2013a****Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

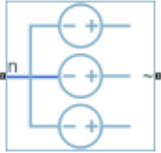
**See Also**

[Current Source](#) | [Resistor](#) | [Voltage Source \(Three-Phase\)](#)

## Voltage Source (Three-Phase)

Ideal three-phase voltage source with optional harmonics

**Library:** Simscape / Electrical / Sources



### Description

The Voltage Source (Three-Phase) block models an ideal three-phase voltage source or a three-phase voltage source with harmonics. You specify the configuration using the Source representation parameter.

When you select **None** for the **Source Impedance** parameter, the Voltage Source block models an ideal three-phase voltage source that maintains sinusoidal voltage of the specified magnitude across its terminals, independently of the current flowing through the source.

The source has a wye configuration, and port **n** provides a connection to the center of the wye. Port **~** is an expandable three-phase port representing the three phases, *a*, *b*, and *c*. The current is positive if it flows from the center of the wye to the positive terminal, and the voltage across each phase is equal to the difference between the voltage at the positive terminal and the center of the wye,  $V(+)$  -  $V_n$ .

### Equations

The output voltage for the Voltage Source block is defined by these equations:

$$v = v_{DC} + v_{AC}\sin(2\pi ft + \phi) + v_N$$

$$v_a = V_0\sin(2\pi ft + \varphi)$$

$$v_N = \sqrt{P_v/2} \frac{N(0, 1)}{\sqrt{h}}$$

$$v_c = V_0\sin(2\pi ft + \varphi + 120^\circ),$$

where:

- $V_0$  is the peak phase voltage.
- $v_{line\_rms}$  is the root-mean square (RMS) phase-to-phase voltage.
- $v_a$ ,  $v_b$ ,  $v_c$  are the respective phase voltages.
- $f$  is frequency.
- $\varphi$  is the phase shift.
- $t$  is time.



When you specify the three-phase voltage source with harmonics representation, the output voltage for the Voltage Source block is defined by these equations:

$$V_0 = \frac{\sqrt{2}}{\sqrt{3}} v_{line\_rms} H_{ratios}$$

$$v_a = V_0 \sin((2\pi f t + \varphi) H_{orders})$$

$$v_b = V_0 \sin((2\pi f t + \varphi - \theta) H_{orders})$$

$$v_c = V_0 \sin((2\pi f t + \varphi + \theta) H_{orders}),$$

where:

- $V_0$  is a row-vector containing the peak voltage of the fundamental and harmonic sinusoids.
- $v_{line\_rms}$  is the RMS phase-to-phase voltage.
- $H_{ratios}$  is a row-vector of harmonic ratios. The first element is 1 to represent the fundamental.
- $H_{orders}$  is a row-vector of harmonic orders. The first element is 1 to represent the fundamental.
- $v_a$ ,  $v_b$ ,  $v_c$  are the respective phase voltages.
- $f$  is a column-vector of harmonic frequencies. The first element is the fundamental frequency.
- $\varphi$  is a column-vector of harmonic phase shifts. The first element is the fundamental phase shift.
- $\theta$  is a column-vector of harmonic phase offsets. The first element is  $120^\circ$ .
- $t$  is the time.

When you select X/R ratio for the **Source Impedance** parameter, the equations for source impedance are:

$$R = \frac{v_{line\_rms}^2}{S_{sc} \sqrt{1 + \phi^2}}$$

$$X = R\phi$$

$$L = \frac{X}{2\pi f},$$

where:

- $S_{sc}$  is the **Short-circuit power level** that you specify.
- $\phi$  is the **Source X/R ratio** that you specify.
- $R$  is the calculated source resistance.
- $X$  is the calculated source reactance.
- $L$  is the calculated source inductance.

### Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can

come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### **~ — Three-phase port**

electrical

Expandable three-phase port associated with the three phases, *a*, *b*, and *c*.

#### **n — Wye center**

electrical

Electrical conserving port associated with the center of the wye.

## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### **Rated voltage (phase-to-phase RMS) — RMS line voltage**

$\sqrt{3} * 100 / \sqrt{2}$  V (default)

RMS phase-to-phase, or line, voltage.

#### **Phase shift — Phase shift**

0 deg (default)

Phase shift in angular units.

#### **Frequency — Voltage frequency**

60 Hz (default)

Voltage frequency, specified in Hz or units directly convertible to Hz (where Hz is defined as 1/s). For example, kHz and MHz are valid units, but rad/s is not.

#### **Source impedance — Source impedance method**

X/R Ratio (default) | None | Series R | Series L | Series RL

Choose a method for specifying source impedance. The default option is X/R Ratio. Selecting any other options enables other parameters. The options are:

- None
- X/R Ratio
- Series R
- Series L
- Series RL

### Short-circuit power level — Short-circuit power level

1e6 V\*A (default)

Short-circuit power level.

#### Dependencies

This parameter is visible only when you select X/R Ratio for the **Source Impedance** parameter.

### Source X/R ratio — Reactance-to-resistance ratio

15 (default)

Complex impedance, that is, the reactance-to-resistance ratio.

#### Dependencies

This parameter is visible only when you select X/R Ratio for the **Source Impedance** parameter.

### Source resistance — Source resistance

0.01 Ohm (default)

Source resistance.

#### Dependencies

This parameter is visible only when you select Series R or Series RL for the **Source Impedance** parameter.

### Source inductance — Source inductance

3.97e-4 H (default)

Source inductance.

#### Dependencies

This parameter is visible only when you select Series L or Series RL for the **Source Impedance** parameter.

#### Harmonics

### Source harmonics — Source Harmonics

None (default) | Generate harmonics

Choose between None and Generate harmonics. The default value is None.

### Harmonic orders — Harmonic orders

[5, 7, 11, 13] (default)

A row-vector of additional integer harmonic orders at which harmonics are to be generated.

**Dependencies**

This parameter is visible only when you select `Generate harmonics` for the **Source representation** parameter.

**Harmonic magnitude to peak magnitude ratios — Harmonic magnitude to peak magnitude ratios**

[.1, .1, .1, .1] (default)

A row-vector of ratios of harmonic magnitudes relative to the fundamental magnitude.

**Dependencies**

This parameter is visible only when you select `Generate harmonics` for the **Source representation** parameter.

**Parasitics****Source impedance parasitic parallel conductance — Source impedance parasitic parallel conductance**

0 1/Ohm (default)

Source impedance parasitic parallel conductance.

**Dependencies**

This parameter is visible only when you select `X/R Ratio`, `Series L` or `Series RL` for the **Source Impedance** parameter.

## Version History

Introduced in R2013b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Battery | Controlled Current Source (Three-Phase) | Current Source (Three-Phase) | Voltage Source

**Topics**

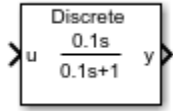
“Build and Simulate Composite and Expanded Three-Phase Models”

“Expand and Collapse Three-Phase Ports on a Block”

## Washout (Discrete or Continuous)

Discrete-time or continuous-time washout or high-pass filter

**Library:** Simscape / Electrical / Control / General Control



### Description

The Washout (Discrete or Continuous) block implements a washout filter in conformance with IEEE 421.5-2016<sup>[1]</sup>. The washout is also known as a high-pass filter.

You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

### Equations

#### Continuous

To configure the Washout (Discrete or Continuous) block for continuous time, set the **Sample time** property to 0. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{Ts}{Ts + 1},$$

where  $T$  is the time constant. From the preceding transfer function, the washout defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}(-x(t) + u(t)) \\ y(t) = -x(t) + u(t) \end{cases} \quad x(0) = u_0, \quad y(0) = 0,$$

where:

- $u$  is the washout input.
- $x$  is the washout state.
- $y$  is the washout output.
- $t$  is the simulation time.
- $u_0$  is the initial input to the block.

#### Discrete

To configure the washout Washout (Discrete or Continuous) for discrete time, set the **Sample time** property to a positive, nonzero value, or to -1 to inherit the sample time from an upstream block. The discrete representation is equivalent to the transfer function:

$$G(z) = \frac{z - 1}{z + T_s/T - 1},$$

where  $T_s$  is the sample time. From the discrete transfer function, the washout defining equations are defined using the forward Euler method:

$$\begin{cases} x(n+1) = \left(1 - \frac{T_s}{T}\right)x(n) + \left(\frac{T_s}{T}\right)u(n) \\ y(n) = u(n) - x(n) \end{cases} \quad x(0) = u_0, y(0) = 0,$$

where:

- $u$  is the washout input.
- $x$  is the washout state.
- $y$  is the washout output.
- $n$  is the simulation time step.
- $u_0$  is the initial input to the block.

### Initial Conditions

To specify the initial conditions of this block, set **Initialization** to:

- **Inherited from block input** — The block sets the state and output initial conditions to the initial input.
- **Specify as parameter** — The block sets the state initial condition to the value of **Initial state**.

### Bypass Filter Dynamics

Set the time constant to a value smaller than or equal to the sample time to ignore the dynamics of the filter. When bypassed, the block feeds the input directly to the output:

$$T \leq T_s \rightarrow y = u .$$

In the continuous case, the sample time and time constant must both be zero.

## Ports

### Input

#### **u — Washout input**

vector

Washout input signal. The block uses the input initial value to determine the state initial value.

Data Types: single | double

### Output

#### **y — Washout output**

vector

Washout output signal.

Data Types: single | double

## Parameters

### Time constant – Washout time constant

0.1 (default) | positive number

Washout time constant. Set this value less than the **Sample time** to bypass the dynamics of the filter.

### Initialization – Initial state specification

Inherited from block input (default) | Specify as parameter

Specify the initial state condition for this block. For more information, see “Initial Conditions” on page 1-2592.

### Initial state – Initial state

0 (default) | real number

Block initial state.

### Dependencies

To enable this parameter, set **Initialization** to Specify as parameter.

### Sample time (-1 for inherited) – Block sample time

-1 (default) | 0 | positive scalar

Time between consecutive block executions. During execution, the block produces outputs and, if appropriate, updates its internal state. For more information, see “What Is Sample Time?” and “Specify Sample Time”.

For inherited discrete-time operation, specify -1. For discrete-time operation, specify a positive integer. For continuous-time operation, specify 0.

If this block is in a masked subsystem, or other variant subsystem that allows you to switch between continuous operation and discrete operation, promote the sample time parameter. Promoting the sample time parameter ensures correct switching between the continuous and discrete implementations of the block. For more information, see “Promote Block Parameters on a Mask”.

## Version History

Introduced in R2017b

## References

- [1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

## Extended Capabilities

### C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

## **See Also**

### **Blocks**

Filtered Derivative (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)



# Winding

Electromagnetic converter with ohmic and magnetic flux losses

**Library:** Simscape / Electrical / Passive



## Description

The Winding block represents an electromagnetic converter with winding resistance and leakage reluctance. You can use this block as a base component for building custom transformers. For an ideal electromagnetic converter, see the Electromagnetic Converter.

When you apply a positive current across the electrical ports of the block, a positive magnetomotive force (MMF) is induced across the magnetic terminals.

$$\mathcal{F} = Ni$$

Where:

- $\mathcal{F}$  is the MMF across the magnetic terminals of the block
- $N$  is the number of winding turns
- $i$  is the current through the winding

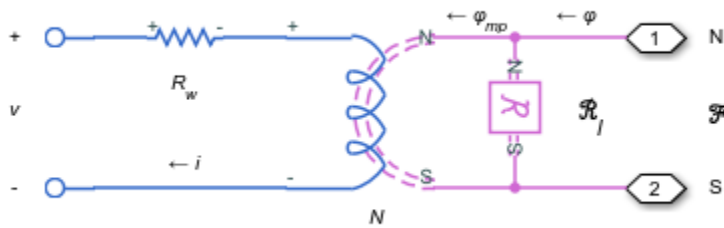
When you apply a positive time-varying flux across the magnetic terminals of the block, a negative voltage is induced across the electrical terminals of the block.

$$v = -N \frac{d\phi}{dt} + \frac{N^2}{\mathcal{R}_l} \frac{di}{dt} + R_w i$$

Where:

- $\phi$  is the flux through the magnetic terminals of the block
- $i$  is the current through the electrical terminals of the block
- $\mathcal{R}_l$  is the leakage reluctance
- $R_w$  is the winding resistance
- $\mathcal{F}$  is the magnetomotive force across the magnetic terminals of the block
- $v$  is the voltage drop across the electrical terminals of the block

This figure shows the equivalent circuit for the block.



In the diagram,  $\varphi_{mp}$  corresponds to the main-path flux and  $\varphi$  to the total flux. You can set the initial condition for the total flux in the block's **Variables** tab.

### Faults

Instantaneous changes in the winding parameters are unphysical. Therefore, when the Winding block enters the faulted state, short-circuit and open-circuit voltages transition to their faulted values over a period of time based on this formula:

$$CurrentValue = FaultedValue - (FaultedValue - UnfaultedValue) \cdot \operatorname{sech}(\Delta t / \tau)$$

where:

- $\Delta t$  is time since the onset of the fault condition.
- $\tau$  is user-defined time constant associated with the fault transition.

For short-circuit faults, the conductance of the short-circuit path also changes according to the  $\operatorname{sech}(\Delta t / \tau)$  function from a small value (representing an open-circuit path) to a large value.

The block can trigger the start of fault transition:

- At a specific time
- After voltage exceeds the maximum permissible value a certain number of times
- When current exceeds the maximum permissible value for longer than a specific time interval

You can enable or disable these trigger mechanisms separately, or use them together if more than one trigger mechanism is required in a simulation. When more than one mechanism is enabled, the first mechanism to trigger the fault transition takes precedence. In other words, a component fails no more than once per simulation.

You can also choose whether to issue an assertion when a fault occurs, by using the **Reporting when a fault occurs** parameter. The assertion can take the form of a warning or an error. By default, the block does not issue an assertion.

Faultable windings often require that you use the fixed-step local solver rather than the variable-step solver. In particular, if you model transitions to a faulted state that include short circuits, MathWorks recommends that you use the fixed-step local solver. For more information, see “Making Optimal Solver Choices for Physical Simulation”.

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Conserving

#### + – Positive terminal

electrical

Electrical conserving port associated with the positive terminal of the block.

#### - – Negative terminal

electrical

Electrical conserving port associated with the negative terminal of the block.

#### N – North terminal

magnetic

Magnetic conserving port associated with the north terminal of the block.

#### S – South terminal

magnetic

Magnetic conserving port associated with the south terminal of the block.

## Parameters

### Main

#### Number of winding turns – Winding turn count

10 (default) | positive number

Number of wire turns on the transformer winding.

#### Winding resistance – Series resistance

1e-3 Ohm (default) | positive number

Power loss in the winding.

#### Leakage reluctance – Parallel reluctance

1e5 1/H (default) | positive number

Magnetic flux loss in the winding. If you do not want to model the leakage internally to the Winding block, set this parameter to `inf`.

**Parallel conductance — Parallel conductance**

0 1/Ohm (default) | positive number

Parallel leakage path, Simulation of some circuits may require the presence of a small parallel conductance.

**Thermal Port****Thermal port — Whether to model thermal port**

Omit (default) | Model

Whether to model the thermal port of the winding. The thermal port reports the winding resistance losses.

**Measurement temperature — Measurement temperature**

25 K (default)

The temperature for which the winding parameters are quoted.

**Dependencies**

To enable this parameter, set **Thermal port** to Model.

**Resistance temperature coefficient — Resistance temperature coefficient**

3.93e-3 1/K (default)

Coefficient  $\alpha$  in the equation relating resistance to temperature, as described in “Thermal Model for Actuator Blocks”. The default value is for copper.

**Dependencies**

To enable this parameter, set **Thermal port** to Model.

**Winding Thermal mass — Thermal mass for winding**

100 J/K (default)

The thermal mass value for the winding. The thermal mass is the energy required to raise the temperature by one degree.

**Dependencies**

To enable this parameter, set **Thermal port** to Model.

**Faults****Enable faults — Whether to enable faults modeling**

Off (default) | On

Whether to enable faults modeling. The associated parameters in the **Faults** section become visible to let you select the reporting method and specify the trigger mechanism (temporal or behavioral). You can enable these trigger mechanisms separately or use them together.

**Location of fault node (% of total turns from - terminal) — Percentage of turns in the subinductor that is in contact with the - port of the faulted winding**

50 (default)

Faults are enabled by segmenting the faulted winding into two coupled subinductors, connected in a series. The inductance is proportional to the square of the number of turns in the respective segment,

and the series resistance of each subinductor is proportional to the number of turns in each segment. The parallel conductance spans both segments.

This parameter indicates the percentage of turns that are assigned to the subinductor that is in contact with the - port of the faulted winding. The remaining turns are assigned to the other subinductor. The default value is 50, which means that the overall inductance of the faulted winding is divided into two equal, coupled subinductors.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

#### Reporting when a fault occurs — Choose whether to issue an assertion when a fault occurs

None (default) | Warn | Error

Choose whether to issue an assertion when a fault occurs:

- None — The block does not issue an assertion.
- Warn — The block issues a warning.
- Error — Simulation stops with an error.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

#### Short-circuit turns — Select whether fault results in one of the segments being short-circuited

No (default) | To negative terminal | To positive terminal

Select whether the fault results in one of the subinductor segments being short-circuited:

- No — The fault does not produce a short circuit.
- To negative terminal — The fault short-circuits the subinductor that is in contact with the - port of the block.
- To positive terminal — The fault short-circuits the subinductor that is in contact with the + port of the block.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

#### Open-circuit at fault node — Select whether to apply an open-circuit fault between the segments

No (default) | Yes

Select whether to apply an open-circuit fault between the two subinductor segments. Even with an open-circuit fault, the characteristics of the subinductors are still related because they are magnetically coupled even in the faulted state.

#### Dependencies

To enable this parameter, set **Enable faults** to On.

**Ground fault — Select whether fault results in one of the segments being short-circuited**

No (default) | Negative terminal side of fault node | Positive terminal side of fault node

Select whether, in case of fault, there is a path for current to flow towards the ground node:

- **No** — The fault does not result in a connection to ground.
- **Negative terminal side of fault node** — The side that is in contact with the - port of the block is connected to ground.
- **Positive terminal side of fault node** — The side that is in contact with the + port of the block is connected to ground.

If the **Open-circuit at fault node** parameter is set to Yes, you must specify which side (negative or positive) is connected to ground. If there is no open circuit, the two options behave similarly. Physically, this corresponds to a breakdown in the insulation between the windings and the grounded core or chassis.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Conductance of faulted ground path — Mutual coupling between the two subinductors**

0 1/0hm (default)

Conductance of the current path to ground. For example, if the path to ground is through the core material, then specify a small conductance value depending on the core material being used. For highly conductive core material or for chassis-shorts, specify a higher conductance value.

**Dependencies**

To enable this parameter, set **Ground fault** to Negative terminal side of fault node or Positive terminal side of fault node.

**Fault transition time constant — Time constant for the transition to faulted state**

1e-3 s (default)

Time constant associated with the transition to the faulted state, as described in “Faults” on page 1-2596.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Enable temporal fault trigger — Select On to enable time-based fault triggering**

No (default) | Yes

Whether to enable time-based fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Simulation time for fault event — Time before entering faulted state**

1 s (default)

Set the simulation time at which you want the block to start entering the fault state.

**Dependencies**

To enable this parameter, set **Enable temporal fault trigger** to On.

**Enable behavioral fault trigger – Select On to enable behavioral fault triggering**

No (default) | Yes

Whether to enable behavioral fault triggering. You can enable the temporal and behavioral trigger mechanisms separately or use them together.

**Dependencies**

To enable this parameter, set **Enable faults** to On.

**Maximum permissible voltage – Voltage threshold to fault transition**

100 V (default)

Voltage threshold to a fault transition. If the voltage value exceeds this threshold a certain number of times, specified by the **Number of events to fail when exceeding voltage** parameter value, then the block starts entering the fault state.

**Dependencies**

To enable this parameter, set **Enable behavioral fault trigger** to On.

**Number of events to fail when exceeding voltage – Maximum number of times the voltage exceeds the threshold**

1 (default)

Number of voltage overshoots that the inductor can withstand before the fault transition begins. This block does not check the time spent in the overvoltage condition, only the number of transitions.

**Dependencies**

To enable this parameter, set **Enable behavioral fault trigger** to On.

**Maximum permissible current – Current threshold to fault transition**

1 A (default)

Current threshold to a fault transition. If the current value exceeds this threshold for longer than the **Time to fail when exceeding current** parameter value, then the block starts entering the fault state.

**Dependencies**

To enable this parameter, set **Enable behavioral fault trigger** to On.

**Time to fail when exceeding current – Maximum length of time the current exceeds the threshold**

1 s (default)

Maximum length of time that the current can exceed the maximum permissible value without triggering the fault.

**Dependencies**

To enable this parameter, set **Enable behavioral fault trigger** to On.

## **Version History**

**Introduced in R2018a**

## **Extended Capabilities**

### **C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

### **See Also**

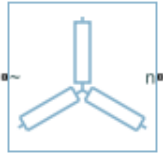
Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase) | Zigzag-Delta-Wye Transformer



# Wye-Connected Load

Three-phase load wired in wye configuration

**Library:** Simscape / Electrical / Passive / RLC Assemblies



## Description

The Wye-Connected Load block models a three-phase load wired in a wye configuration. Each limb of the load can include any combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

You can specify values for the R, L, and C components directly in terms of resistance, inductance, and capacitance, or by rated powers at a rated voltage and frequency.

- If you parameterize the block directly in terms of R, L, and C values, then for initialization, provide a three-element row vector of initial voltages for a capacitor, and a three-element row vector of initial currents for an inductor.
- If you parameterize the block in terms of rated powers, then specify initial conditions in terms of an initial voltage, initial voltage phase, and initial frequency. For example, if the load is connected directly to a three-phase voltage source, then the initial conditions are identical to the source values for RMS line voltage, frequency, and phase shift. To specify zero initial-voltage magnitude, set the initial voltage to 0.

For certain combinations of R, L, and C, for some circuit topologies, specify parasitic resistance or conductance values that help the simulation to converge numerically. These parasitic terms ensure that an inductor has a small parallel resistive path and that a capacitor has a small series resistance. When you parameterize the block in terms of rated powers, the rated power values do not account for these small parasitic terms. The rated powers represent only the R, L, and C values of the load itself.

## Block parameterization

The following two tables list the block parameters for each **Component structure**, based on the selected **Parameterization** option:

- Specify by rated power
- Specify component values directly

**Specify by Rated Power**

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
R	Rated voltage Real power	None	None
L	Rated voltage Rated electrical frequency Inductive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
C	Rated voltage Rated electrical frequency Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency
Series RL	Rated voltage Rated electrical frequency Real power Inductive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
Series RC	Rated voltage Rated electrical frequency Real power Capacitive reactive power	None	Terminal voltage magnitude Terminal voltage angle Frequency
Series LC	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency

Component Structure	Main Parameters	Parasitics Parameters	Initial Conditions Parameters
Series RLC	Rated voltage Rated electrical frequency Real power Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel RL	Rated voltage Rated electrical frequency Real power Inductive reactive power	None	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel RC	Rated voltage Rated electrical frequency Real power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency
Parallel LC	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
Parallel RLC	Rated voltage Rated electrical frequency Real power Inductive reactive power Capacitive reactive power	Parasitic series resistance	Terminal voltage magnitude Terminal voltage angle Frequency

**Specify Component Values Directly**

<b>Component Structure</b>	<b>Main Parameters</b>	<b>Parasitics Parameters</b>	<b>Initial Conditions Parameters</b>
R	Resistance	None	None
L	Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
C	Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Series RL	Resistance Inductance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ]
Series RC	Resistance Capacitance	None	Initial capacitor voltage [ Va Vb Vc ]
Series LC	Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Series RLC	Resistance Inductance Capacitance	Parasitic parallel conductance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RL	Resistance Inductance	None	Initial inductor current [ Ia Ib Ic ]
Parallel RC	Resistance Capacitance	Parasitic series resistance	Initial capacitor voltage [ Va Vb Vc ]
Parallel LC	Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]
Parallel RLC	Resistance Inductance Capacitance	Parasitic series resistance	Initial inductor current [ Ia Ib Ic ] Initial capacitor voltage [ Va Vb Vc ]

**Variables**

Use the **Variables** tab to specify the priority and initial target values for the block variables before simulation. For more information, see “Set Priority and Initial Target for Block Variables”.

To enable the **Variables** tab, set the **Parameterization** parameter to Specify component values directly. The tab is not visible if you set **Component structure** to R.

## Ports

### Conserving

#### **~ — Three-phase port**

electrical

Electrical expandable three-phase port.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

### Main

#### **Parameterization — Parameterization**

Specify by rated power (default) | Specify component values directly

Select one of these values:

- **Specify by rated power** — Specify values for the R, L, and C components by rated powers at a rated voltage and frequency. This is the default.
- **Specify component values directly** — Specify values for the R, L, and C components directly in terms of resistance, inductance, and capacitance.

Switching the **Parameterization** value resets the **Component structure** value. Select the component parameterization option first, and then the component structure. If you later switch the **Parameterization** value, check the **Component structure** value and reselect it, if necessary.

#### **Component structure — Desired combination**

R (default) | L | C | Series RL | Series RC | Series LC | Series RLC | Parallel RL | Parallel RC | Parallel LC | Parallel RLC

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel.

#### **Rated voltage — Rated voltage**

2.4e4 V (default)

Voltage for which load powers are specified. This parameter is visible only when you specify values by rated power.

#### **Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter.

#### **Real power – Total real dissipated power**

1000 W (default) | positive scalar

Total real power dissipated by three-phase load when supplied at the rated voltage. The value must be greater than 0.

#### **Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes a resistor.

#### **Rated electrical frequency – Rated electrical frequency**

60 Hz (default)

Frequency for which reactive load powers are specified.

#### **Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter.

#### **Inductive reactive power – Inductive reactive power**

100 V\*A (default) | positive scalar

Total inductive reactive power taken by the three-phase load when supplied at the rated voltage. The value must be greater than 0.

#### **Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes an inductor.

#### **Capacitive reactive power – Total capacitive reactive power**

-100 V\*A (default) | negative scalar

Total capacitive reactive power taken by the three-phase load when supplied at the rated voltage. The value must be less than 0.

#### **Dependencies**

This parameter is visible only when you select **Specify by rated power** for the **Parameterization** parameter and select a component structure that includes a capacitor.

#### **Resistance – Resistance**

1 Ohm (default)

The resistance of each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes a resistor.

**Inductance — Inductance**

0.001 H (default)

Inductance of each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes an inductor.

**Capacitance — Capacitance**

1e-6 F (default)

Capacitance in each of the load limbs.

**Dependencies**

This parameter is visible only when you select **Specify component values directly** for the **Parameterization** parameter and select a component structure that includes a capacitor.

**Parasitics****Parasitic series resistance — Parasitic series resistance**

0 Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load.

**Parasitic parallel conductance — Parasitic parallel conductance**

0 1/Ohm (default)

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load.

If you set **Component structure** to either **Series RL**, **Series LC**, or **Series RLC**, the parasitic parallel conductance is connected across the full branch of the block.

## Version History

Introduced in R2013b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Delta-Connected Load | RLC (Three-Phase)



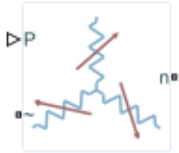
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

## Wye-Connected Variable Load

Three-phase variable load wired in wye configuration

**Library:** Simscape / Electrical / Passive / RLC Assemblies



### Description

The Wye-Connected Variable Load block models a three-phase variable load wired in a wye configuration. Each limb of the load contains a resistor. The block calculates the resistance required to draw the real power of the physical signal input  $P$  at the rated voltage that you specify. Therefore, the block can represent a real load.

To ensure that the resistance is always greater than zero, you specify the minimum real power that the load consumes. The minimum real power must be greater than zero.

### Equations

The resistance is defined by

$$R = \frac{V_{Rated}^2}{P},$$

where:

- $R$  is the per-phase series resistance.
- $V_{Rated}$  is the RMS, rated line-line voltage.
- $P$  is the three-phase real power required.

### Ports

#### Input

##### **P — Real power**

physical signal

Physical signal input port associated with the real power.

#### Conserving

##### **~ — Three-phase port**

electrical

Expandable three-phase port.

**n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

**Parameters****Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Rated voltage — Rated line-line voltage**

24e3 V (default)

RMS, rated line-line voltage for the resistance equation.

**Minimum real power — Minimum real power**

1e3 W (default) | positive scalar

Minimum real power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0.

**Version History**

Introduced in R2014b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Wye-Connected Load | Wye-Connected Variable Load (lagging)

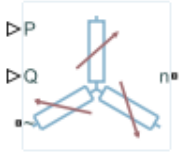
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

## Wye-Connected Variable Load (lagging)

Three-phase variable, lagging load wired in wye configuration

**Library:** Simscape / Electrical / Passive / RLC Assemblies



### Description

The Wye-Connected Variable Load (lagging) block models a three-phase variable, lagging load wired in a wye configuration. Each limb of the load contains a resistor (R) and an inductor (L) connected in series. The block calculates the resistance and inductance required to draw the real and reactive powers of the physical signal inputs P and Q at the rated voltage and rated frequency that you specify. Therefore, the block can represent a real and lagging reactive load.

To ensure that the resistance and inductance are always greater than zero, you specify the minimum real power and the reactive power that the load consumes. The minimum real power and the reactive power must be greater than zero.

### Electrical Defining Equations

The per-phase series resistance and inductance are defined by

$$R = \frac{PV_{Rated}^2}{P^2 + Q^2}$$

and

$$L = \frac{QV_{Rated}^2}{2\pi F_{Rated}(P^2 + Q^2)},$$

where:

- $R$  is the per-phase series resistance.
- $L$  is the per-phase series inductance.
- $V_{Rated}$  is the RMS, rated line-line voltage.
- $F_{Rated}$  is the nominal AC electrical frequency.
- $P$  is the three-phase real power required.
- $Q$  is the three-phase lagging reactive power required.

The inductance is defined as the ratio of the magnetic flux,  $\phi$ , to the steady-state current:

$$L(i) = \frac{\phi(i)}{i}.$$

Therefore the current-voltage relationship for the inductor is:

$$v = \frac{dL}{dt}i + L\frac{di}{dt}.$$

## Variables

To set the priority and initial target values for the block variables prior to simulation, use the **Initial Targets** section in the block dialog box or Property Inspector. For more information, see “Set Priority and Initial Target for Block Variables”.

Nominal values provide a way to specify the expected magnitude of a variable in a model. Using system scaling based on nominal values increases the simulation robustness. Nominal values can come from different sources, one of which is the **Nominal Values** section in the block dialog box or Property Inspector. For more information, see “System Scaling by Nominal Values”.

## Ports

### Input

#### **P — Real power**

physical signal

Physical signal input port associated with the real power.

#### **Q — Reactive power**

physical signal

Physical signal input port associated with the reactive power.

### Conserving

#### **~ — Three-phase port**

electrical

Expandable three-phase electrical conserving port.

#### **n — Neutral phase**

electrical

Electrical conserving port associated with the neutral phase.

## Parameters

### **Modeling option — Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

**Main****Rated voltage — Rated line-line voltage**

24e3 V (default)

RMS, rated line-line voltage for the resistance equation.

**Rated electrical frequency — Nominal AC electrical frequency**

60 Hz (default)

Nominal AC electrical frequency for the inductance equation.

**Minimum real power — Minimum real power**

1e3 W (default) | positive scalar

Minimum real power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0.

**Minimum reactive power — Minimum reactive power**

1e3 V\*A (default) | positive scalar

Minimum reactive power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0.

**Parasitics****Parasitic parallel conductance — Parasitic parallel conductance**

1e-6 1/Ohm (default)

Conductance that the block adds, in parallel, to the series RL.

**Version History**

Introduced in R2014b

**Extended Capabilities****C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Wye-Connected Load | Wye-Connected Variable Load

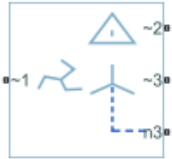
**Topics**

“Expand and Collapse Three-Phase Ports on a Block”

# Zigzag-Delta-Wye Transformer

Linear, non-ideal zigzag-delta-wye transformer with three-limb core with saturation capability

**Library:** Simscape / Electrical / Passive / Transformers



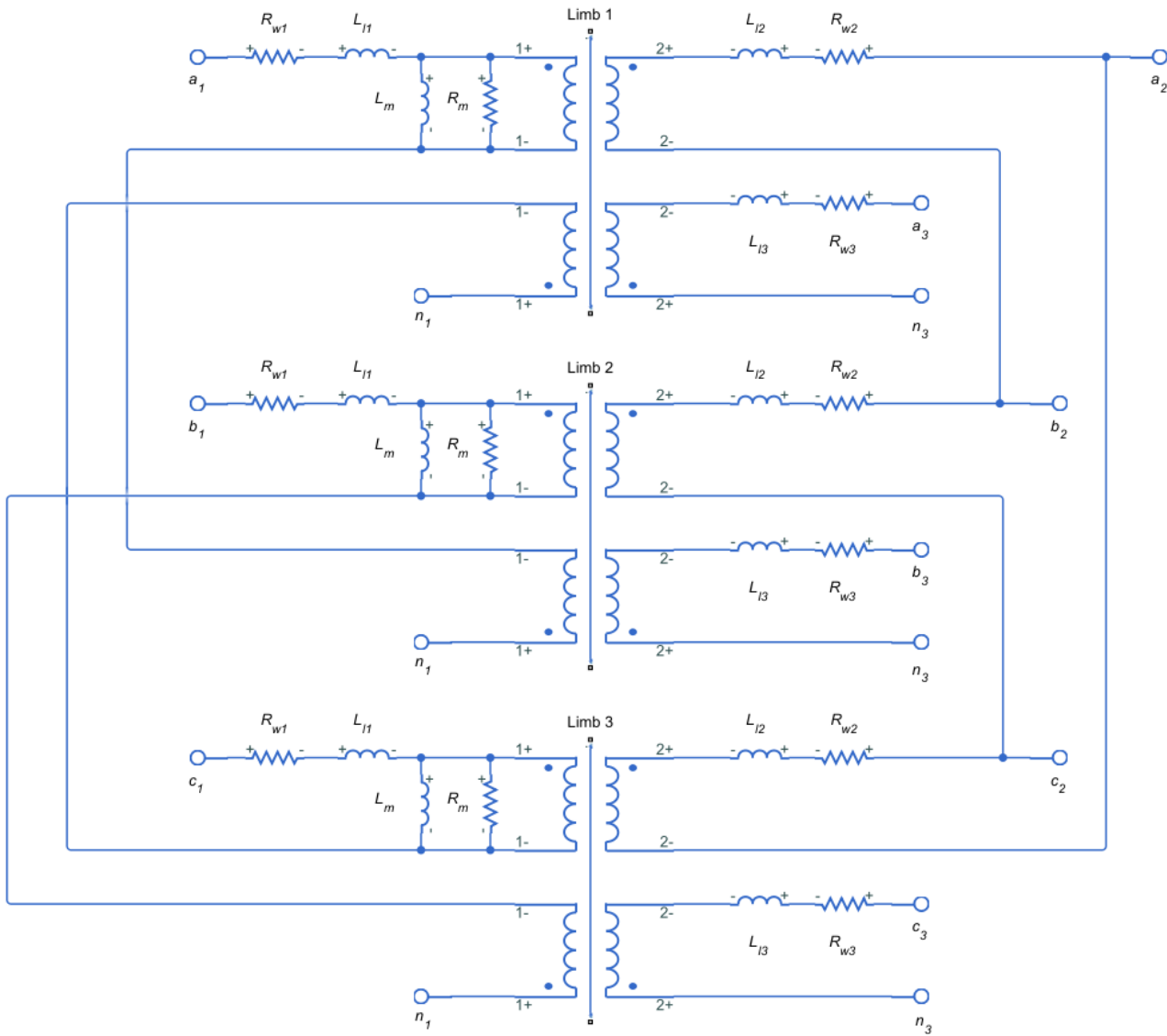
## Description

The Zigzag-Delta-Wye Transformer block models a linear, nonideal transformer with a three-limb core that has primary zigzag windings and secondary delta, wye, or delta and wye windings. You can specify the phase offset between the zigzag and wye windings using the **Wye secondary phase shift** parameter and parameterize the block impedance using per-unit values. The block includes effects for linear winding leakage and linear core magnetization.

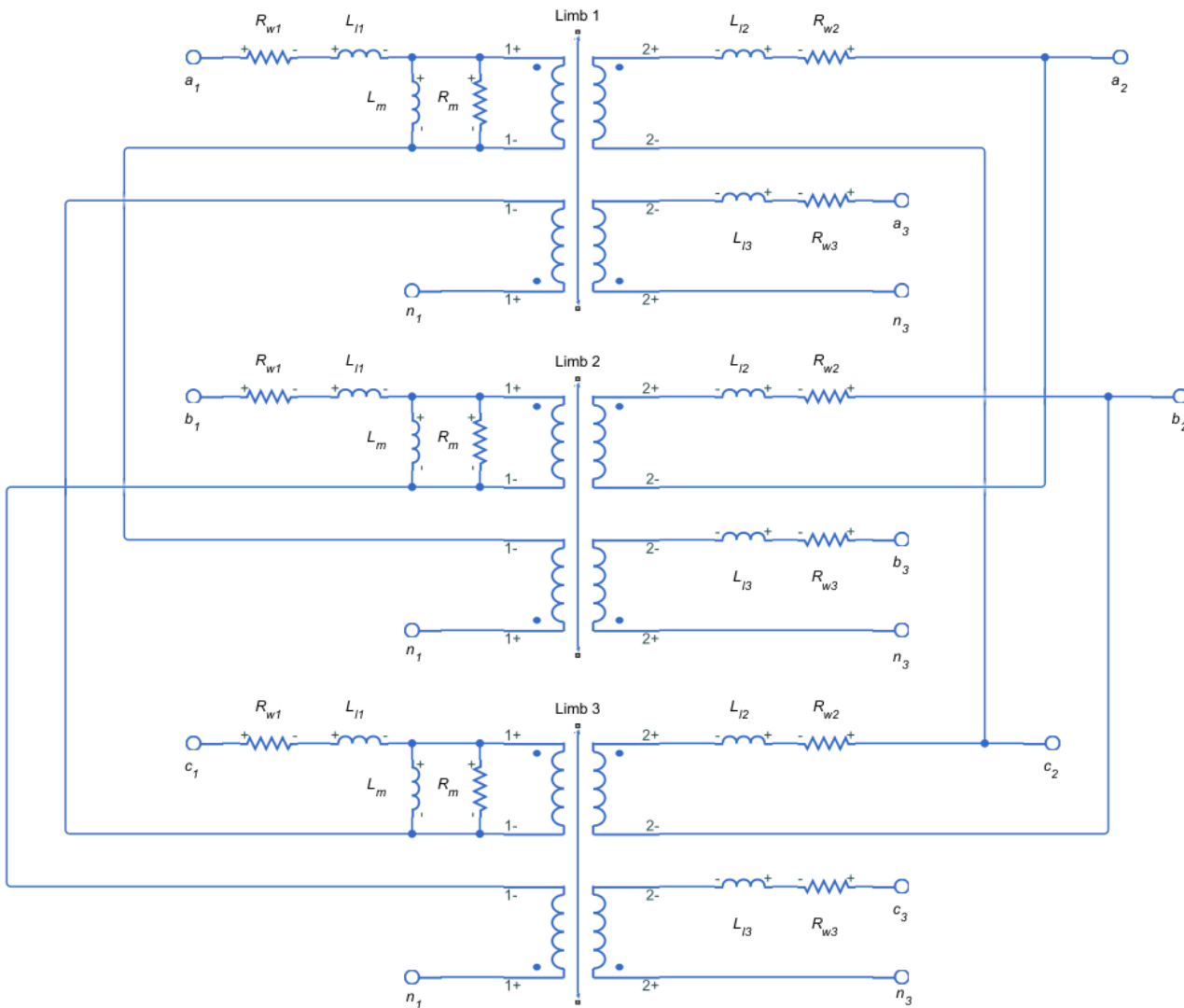
The configuration options for the delta winding types are:

- Delta 1 o'clock — Mesh configuration with a lagging 30-degree phase shift relative to the voltage of a connected wye configuration
- Delta 11 o'clock — Mesh configuration with leading 30-degree phase shift relative to the voltage of a connected wye configuration

These equivalent circuit diagrams show the zigzag-delta-wye transformer in the delta 1 o'clock and delta 11 o'clock configurations, respectively.







In these diagrams:

- $R_{w1}$  is the primary winding resistance.
- $L_{l1}$  is the primary leakage reactance.
- $R_{w2}$  is the delta secondary winding resistance.
- $L_{l2}$  is the delta secondary leakage reactance.
- $R_{w3}$  is the wye secondary winding resistance.
- $L_{l3}$  is the wye secondary leakage reactance.
- $R_m$  is the shunt magnetizing resistance.
- $L_m$  is the shunt magnetizing reactance.

### Variables

Use the **Variables** settings to specify the priority and initial target values for the block variables before simulation. For more information, see "Set Priority and Initial Target for Block Variables".

To enable the **Wye secondary currents** variable, set **Secondary connection** to Wye or Wye and Delta.

To enable the **Delta secondary currents** variable, set **Secondary connection** to Delta or Wye and Delta.

## Ports

### Conserving

#### **~1 – Primary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a1\ b1\ c1]$ , voltage of the primary winding.

#### **~2 – Delta secondary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a2\ b2\ c2]$ , voltage of the first secondary winding.

#### Dependencies

To enable this port, set **Secondary connection** to Delta or Wye and Delta.

#### **~3 – Wye secondary winding voltage**

electrical

Expandable three-phase electrical conserving port associated with the three-phase,  $[a3\ b3\ c3]$ , voltage of the second secondary winding.

#### Dependencies

To enable this port, set **Secondary connection** to Wye or Wye and Delta.

#### **n3 – Second secondary winding neutral point**

electrical

Electrical conserving port associated with the wye secondary winding neutral point.

#### **~1i – Zigzag winding interconnection port**

electrical

Electrical conserving port associated with the interconnection of the zigzag winding.

#### Dependencies

To enable this port, set **Zigzag configuration** to Accessible winding interconnects.

## Parameters

#### **Modeling option – Whether to model composite or expanded three-phase ports**

Composite three-phase ports (default) | Expanded three-phase ports

Whether to model composite or expanded three-phase ports.

Composite three-phase ports represent three individual electrical conserving ports with a single block port. You can use composite three-phase ports to build models that correspond to single-line diagrams of three-phase electrical systems.

Expanded three-phase ports represent the individual phases of a three-phase system using three separate electrical conserving ports.

## Main

### Rated apparent power — Apparent power at rated capacity

100e6 V\*A (default) | positive scalar

Apparent power flowing through the transformer when operating at rated capacity, in V\*A.

### Rated electrical frequency — Connected network electrical frequency

60 Hz (default) | positive scalar

Rated or nominal frequency of the AC network to which the transformer is connected, in Hz. The value must be greater than 0.

### Primary rated voltage — Root mean square line voltage applied to the primary winding

24e3 V (default) | positive scalar

RMS line voltage applied to the primary winding under normal operating conditions, in V. The value must be greater than 0.

### Zigzag configuration — Zigzag configuration type

Internal floating neutral (default) | Accessible winding interconnects

Configuration of zigzag winding.

### Secondary connection — Zigzag configuration type

Delta (default) | Wye | Wye and Delta

Configuration of the secondary connection.

### Delta connection type — Delta connection type

Delta 1 o'clock (default) | Delta 11 o'clock

Type of delta connection.

## Dependencies

To enable this parameter, set **Secondary connection** to Delta or Wye and Delta.

### Delta secondary rated voltage — Root mean square line voltage applied to the delta secondary winding

4160 V (default) | positive scalar

RMS line voltage applied to the delta secondary winding under normal operating conditions, in V. The value must be greater than 0.

## Dependencies

To enable this parameter, set **Secondary connection** to Delta or Wye and Delta.

**Wye secondary rated voltage — Root mean square line voltage applied to the wye secondary winding**

4160 V (default) | positive scalar

RMS line voltage applied to the wye secondary winding under normal operating conditions, in V. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Secondary connection** to Wye or Wye and Delta.

**Secondary phase shift — Phase offset for secondary winding**

-7.5 deg (default)

Phase offset between the zigzag and the secondary windings.

**Impedances****Primary winding resistance (pu) — Primary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the primary winding. The value must be greater than 0.

**Delta secondary winding resistance (pu) — Delta secondary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the delta secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Secondary connection** to Delta or Wye and Delta.

**Wye secondary winding resistance (pu) — Wye secondary winding power loss**

0.01 (default) | positive scalar

Per-unit power loss in the wye secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Secondary connection** to Wye or Wye and Delta.

**Leakage reactance — Leakage reactance modeling**

Exclude (default) | Include

Whether to model magnetic flux losses.

**Primary leakage reactance (pu) — Primary winding magnetic flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the primary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to Include.

**Delta secondary leakage reactance (pu) — Delta secondary winding flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the delta secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to Include and **Secondary connection** to either Delta or Wye and Delta.

**Wye secondary leakage reactance (pu) – Wye secondary winding flux loss**

0.001 (default) | positive scalar

Per-unit magnetic flux loss in the wye secondary winding. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Leakage reactance** to Include and **Secondary connection** to either Wye or Wye and Delta.

**Magnetizing resistance – Magnetizing resistance modeling**

Exclude (default) | Include

Whether to model the transformer core losses.

**Shunt magnetizing resistance (pu) – Transformer core losses**

500 (default) | positive scalar

Per-unit losses in the transformer core. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing resistance** to Include.

**Magnetizing reactance – Magnetizing reactance modeling**

Exclude (default) | Include

Whether to model magnetic effects of the transformer core.

**Magnetic saturation representation – Representation of the magnetic saturation**

None (default) | Lookup table (phi versus i)

Whether to represent magnetic saturation.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include.

**Current vector (pu), i – Per unit vector of currents**

[0, .002, 1, 2] (default)

Per unit vector of currents. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Magnetic flux vector (pu), phi – Per unit vector of magnetic flux**

[0, 1.2, 1.5, 1.51] (default)

Per unit vector of magnetic flux. The first value must be 0. This parameter must be strictly ascending.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to Lookup table (phi versus i).

**Shunt magnetizing reactance (pu) — Transformer core magnetic effects**

500 (default) | positive scalar

Per-unit magnetic effects of the transformer core when operating in its linear region. The value must be greater than 0.

**Dependencies**

To enable this parameter, set **Magnetizing reactance** to Include and **Magnetic saturation representation** to None.

## Version History

Introduced in R2019b

## Extended Capabilities

**C/C++ Code Generation**

Generate C and C++ code using Simulink® Coder™.

**See Also**

Earthing Transformer | Ideal Transformer | Nonlinear Transformer | Tap-Changing Transformer | Three-Winding Transformer (Three-Phase) | Two-Winding Transformer (Three-Phase)

# Functions

---

## ee\_calculateFluxPartialDerivatives

Calculate flux partial derivatives for FEM-Parameterized PMSM block

### Syntax

```
[dFdA,dFdB,dFdC,dFdX] = ee_calculateFluxPartialDerivatives(A,B,C,X,F)
[dFdA,dFdB,dFdC,dFdX,D,Q] = ee_calculateFluxPartialDerivatives(A,B,C,X,F)
```

### Description

[dFdA,dFdB,dFdC,dFdX] = ee\_calculateFluxPartialDerivatives(A,B,C,X,F) calculates the partial derivatives from flux linkage. For improved numerical performance, the FEM-Parameterized PMSM block works with flux linkage partial derivatives, rather than directly with flux linkage. If your finite-element motor design tool does not have an option to output partial derivatives, then you can use this function to calculate the partial derivatives from the flux linkage. The flux linkage  $F$  must be a four-dimensional matrix with the first three dimensions corresponding to the A, B, and C phase currents, and the fourth dimension corresponding to the rotor angle  $X$ . The function returns four-dimensional matrices for the four partial derivatives. Use this syntax in conjunction with the 4-D Data modeling option of the block.

[dFdA,dFdB,dFdC,dFdX,D,Q] = ee\_calculateFluxPartialDerivatives(A,B,C,X,F) returns two additional output arguments corresponding to  $d$ -axis and  $q$ -axis currents, respectively. In this case, the four partial derivatives are three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle. Use this syntax in conjunction with the 3-D Data modeling option of the block.

### Examples

#### Calculate 4-D flux linkage partial derivatives

Suppose  $F$  is a four-dimensional matrix containing flux linkage data, exported by your finite-element motor design tool. The matrix dimensions correspond to the three phase currents and the rotor angle, respectively. The data is cyclical in the fourth dimension, corresponding to the rotor angle.

---

**Tip** If you do not have data from a finite-element motor design tool for your PMSM, to prevent a simulation error, before running the code for this example, first generate the required  $F$  matrix by running the Generate 4-D Flux Linkage Matrix  $F$  example for the ee\_generateIdealPMSMfluxData function.

---

Either directly import or recreate the current vectors. For example, if recreating a current vector with evenly spaced values between -250 and 250 A and 5 A increments, then:

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Import or define the number of pole pairs.



```
N = 6;
```

Import the rotor angle vector or recreate it based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Calculate the flux linkage partial derivatives.

```
[dFdA,dFdB,dFdC,dFdX] = ee_calculateFluxPartialDerivatives(iA,iB,iC,X,F);
```

The function returns four 4-D matrices for the flux linkage partial derivatives. The four matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions also correspond to the three phase currents and the rotor angle.

### Calculate 3-D flux linkage partial derivatives

Suppose  $F$  is a four-dimensional matrix containing flux linkage data, exported by your finite-element motor design tool. The matrix dimensions correspond to the three phase currents and the rotor angle, respectively. The data is cyclical in the fourth dimension, corresponding to the rotor angle.

---

**Tip** If you do not have data from a finite-element motor design tool for your PMSM, to prevent a simulation error, before running the code for this example, first generate the required  $F$  matrix by running the Generate 4-D Flux Linkage Matrix  $F$  example for the `ee_generateIdealPMSMfluxData` function.

---

Either directly import or recreate the current vectors. For example, if recreating a current vector with evenly spaced values between -250 and 250 A and 5 A increments:

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Import or define the number of pole pairs.

```
N = 6;
```

Import the rotor angle vector or recreate it based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Calculate the flux linkage partial derivatives.

```
[dFdA,dFdB,dFdC,dFdX,iD,iQ] = ee_calculateFluxPartialDerivatives(iA,iB,iC,X,F);
```

The function returns four 3-D matrices for the flux linkage partial derivatives and two vectors for the  $d$ -axis and  $q$ -axis current values. The four matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

## Input Arguments

### A — A-phase current, in amperes

vector

A-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: `double`

### **B — B-phase current, in amperes**

vector

B-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: `double`

### **C — C-phase current, in amperes**

vector

C-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: `double`

### **X — Rotor angle, in radians**

vector

The rotor angle, in radians, specified as a vector. The values must be in the range from zero to  $2\pi/N$ , where  $N$  is the number of pole pairs.

Data Types: `double`

### **F — Flux linkage, in weber-turns**

four-dimensional matrix

The flux linkage, in weber-turns, specified as a four-dimensional matrix, with dimensions corresponding to the three phase currents and rotor angle. The data must be cyclical in the fourth (rotor angle) dimension, that is, for all  $i, j$ , and  $k$ ,  $F(i,j,k,0) = F(i,j,k,2\pi/N)$ , where  $N$  is the number of pole pairs.

Data Types: `double`

## **Output Arguments**

### **dFdA — Flux linkage partial derivative with respect to the A-phase current**

matrix

Flux linkage partial derivative with respect to the A-phase current, returned as a matrix. For syntax used with the 4-D Data modeling option of the block, the matrix is four-dimensional. For syntax used with the 3-D Data modeling option of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdB — Flux linkage partial derivative with respect to the B-phase current**

matrix

Flux linkage partial derivative with respect to the B-phase current, returned as a matrix. For syntax used with the 4-D Data modeling option of the block, the matrix is four-dimensional. For syntax used with the 3-D Data modeling option of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdC — Flux linkage partial derivative with respect to the C-phase current**

matrix

Flux linkage partial derivative with respect to the C-phase current, returned as a matrix. For syntax used with the 4-D Data modeling option of the block, the matrix is four-dimensional. For syntax used with the 3-D Data modeling option of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdX — Flux linkage partial derivative with respect to the rotor angle**

matrix

Flux linkage partial derivative with respect to the rotor angle, returned as a matrix. For syntax used with the 4-D Data modeling option of the block, the matrix is four-dimensional. For syntax used with the 3-D Data modeling option of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **D — D-axis current, in amperes**

vector

$D$ -axis current, in amperes, returned as a vector. This is an optional output argument, to be used when you want to generate 3-D flux linkage partial derivatives. The vector defines the  $d$ -axis current values at which the partial derivatives are determined.

### **Q — Q-axis current, in amperes**

vector

$Q$ -axis current, in amperes, returned as a vector. This is an optional output argument, to be used when you want to generate 3-D flux linkage partial derivatives. The vector defines the  $q$ -axis current values at which the partial derivatives are determined.

## **Algorithms**

The function calculates partial derivatives using Akima splines, the same method that is used for smooth interpolation in the Simscape language `tablelookup` function. For more information, see `makima`. Akima splines are suitable for estimating partial derivatives due to their smooth nature and tendency not to introduce local gradient reversals.

## **Version History**

Introduced in R2017a

## **See Also**

`ee_generateIdealPMSMfluxData`

## ee\_calculateThdPercent

Compute the total harmonic distortion (THD) percentage

### Syntax

```
[thdPercent] = ee_calculateThdPercent(harmonicOrder,...
harmonicMagnitude)
```

### Description

[thdPercent] = ee\_calculateThdPercent(harmonicOrder,... harmonicMagnitude) calculates the total harmonic distortion (THD) percentage using these equations:

$$M = \frac{\text{harmonic magnitude}}{\sqrt{2}},$$

and

$$\%THD = 100 \frac{\sqrt{\sum_{i=2}^n M_i^2}}{M_1},$$

where:

- $M_i$  is the root mean square (RMS) value of the harmonic magnitude corresponding to the  $i^{\text{th}}$  harmonic order.
- $M$  is  $V_{RMS}$  or  $I_{RMS}$  as required.

You can use the ee\_getHarmonics function to obtain the vectors of harmonic order and harmonic magnitude for a `simscape.logging.Node`.

### Examples

#### Calculate THD percent

Calculate the THD from harmonic orders [1;5;7;11;13] and harmonic magnitudes [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;0.0127e+03].

```
harmonicOrder = [1;5;7;11;13];
harmonicMagnitude = [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;...
0.0127e+03];
thdPercent = ee_calculateThdPercent( harmonicOrder, harmonicMagnitude )

thdPercent = 4.5480
```

## Input Arguments

### **harmonicOrder** — Harmonic orders

vector

Harmonic orders from 0 up to and including number of harmonics, specified as a vector.

Example: [1;5;7;11;13]

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **harmonicMagnitude** — Harmonic magnitudes

vector

Harmonic magnitudes from the 0<sup>th</sup> harmonic up to and including the number of harmonics included in the analysis, specified as a vector.

Example: [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;0.0127e+03]

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

## Version History

Introduced in R2014a

## See Also

### **Blocks**

Spectrum Analyzer

### **Functions**

ee\_getHarmonics | ee\_plotHarmonics | sscexplore

### **Objects**

simscape.logging.Node

### **Topics**

"Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block"

"Choose a Simscape Electrical Function for an Offline Harmonic Analysis"

"Data Logging"

"Harmonic Analysis of a Three-Phase Rectifier"

## ee\_comtradeDataset

Read and extract values from COMTRADE record

### Syntax

```
dataset = ee_comtradeDataset(comtradeRecordName)
dataset = ee_comtradeDataset(comtradeRecordName, 'HeaderFileType', filetype)
```

### Description

`dataset = ee_comtradeDataset(comtradeRecordName)` reads the `comtradeRecordName` COMTRADE record and outputs a `dataset` structure with the configuration, header, data, and information properties.

To export the data channel signals to a format compatible with the Simulink Signal Editor block, use the `exportMATFile` method to create a `myComtradeRecordName.mat` file:

```
exportMATFile(dataset);
```

`dataset = ee_comtradeDataset(comtradeRecordName, 'HeaderFileType', filetype)` reads the header file of the `comtradeRecordName` COMTRADE record as a file of the extension specified by `filetype`, and outputs a `dataset` structure with configuration, header, data, and information properties.

### Input Arguments

#### **comtradeRecordName** — Name of COMTRADE record

case-sensitive string

File name of the COMTRADE record that you want to read and extract values from, specified as a case-sensitive string without the file extension. Configuration and data files are mandatory for each COMTRADE record. Header and information files are optional. All files must have the same name and different extensions.

These are the accepted extensions for the COMTRADE records:

- `.cfg` for configuration files
- `.dat` for data files
- `.hdr` for header files
- `.inf` for information files

#### **filetype** — Type of header file in COMTRADE record

text | xml

Type of the header file in the COMTRADE record input, specified as 'text' or 'xml'.

## Output Arguments

### **dataset** — Extracted values from COMTRADE record

structure

Extracted values from the COMTRADE record, returned as a structure with these fields:

### **Configuration** — COMTRADE configuration

structure

Data extracted from the configuration file of the COMTRADE record, returned as a structure. This data is necessary to interpret the information contained in the data file.

### **Data** — COMTRADE data

structure

Data extracted from the data file of the COMTRADE record, returned as a structure.

### **Header** — COMTRADE header

structure

Data extracted from the header file of the COMTRADE record, returned as a structure.

### **Information** — COMTRADE information

structure

Data extracted from the information file of the COMTRADE record, returned as a structure.

## Version History

Introduced in R2020b

## See Also

## ee\_convertedMosfetValidation

Generate standard MOSFET characteristics for Simscape and validate conversion results against SPICE simulation tool

### Syntax

```
[outputData] = ee_convertedMosfetValidation(fileName,Name,Value)
```

### Description

[outputData] = ee\_convertedMosfetValidation(fileName,Name,Value) takes a Simscape ssc file name and a set of name-value pairs and returns these characteristics:

- Id versus Vgs
- Id versus Vds
- Gate charge (Qiss)
- Output charge (Qoss)
- Breakdown voltage

If the subcircuit contains the junction terminal, the function generates the Id versus Vgs characteristics for two different combinations of junction and case temperatures.

If you also provide the SPICE netlist files as an input, **outputData** also includes the plots from both Simscape and the specified SPICE simulation tool. When the signals are below the tolerance values, you can observe the differences between the Simscape and SPICE simulation tool results.

This function supports LTspice XVII and SIMetrix.

### Examples

#### Generate and plot simulation results for Simscape

Generate the MOSFET Id versus Vgs characteristics and Id versus Vds characteristics and plot the results.

```
[outputData] = ee_convertedMosfetValidation("ee_BSC019N02KS.ssc",...
"CheckIdVgs",1, "GeneratePlots",1)
```

#### Generate, plot, and compare simulation results against LTspice

Generate the MOSFET standard characteristics and plot the results to compare them with the results obtained in LTspice.

```
[outputData] = ee_convertedMosfetValidation("ee_BSC019N02KS.ssc",...
"SPICEFile","ee_SPICEOptiMOS20V.lib","Subcircuit","BSC019N02KS",...
"SPICEPath","C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe",...
```



```
"SPICETool", "LTspice", "CheckIdVgs", 1, "CheckIdVds", 1, "CheckQiss", 1, ...
"CheckQoss", 1, "CheckBreakdown", 1, "Vt", 1.5, "Vds", 15, "RelTol", 1e-5, ...
"AbsTol", 1e-5, "VnTol", 1e-5, "Ciss", 23e-9, "Coss", 3e-9, "AbsErrTol", 1e-2, ...
"RelErrTol", 0.1, "BreakdownScalingForVds", 3, "GeneratePlots", 1)
```

## Generate, plot, and compare simulation results against SIMetrix

Generate the MOSFET standard characteristics and plot the results to compare them with the results obtained in SIMetrix.

```
[outputData] = ee_convertedMosfetValidation("ee_BSC019N02KS.ssc", ...
"SPICEFile", "ee_SPICE0ptiMOS20V.lib", "Subcircuit", "BSC019N02KS", ...
"SPICETool", "SIMetrix", "SPICEPath", ...
"C:\Program Files\SIMetrix830\bin64\Sim.exe", "CheckQiss", 1, ...
"CheckQoss", 1, "CheckBreakdown", 1, "Vt", 1.5, "Vds", 15, "Ciss", 23e-9, ...
"Coss", 3e-9, "BreakdownScalingForVds", 3, "GeneratePlots", 1)
```

## Input Arguments

### fileName — File name

character vector

Name of the Simscape ssc file you want to generate the characteristics from, specified as a character vector.

### Name-Value Pair Arguments

Specify optional pairs of arguments as Name1=Value1, ..., NameN=ValueN, where Name is the argument name and Value is the corresponding value. Name-value arguments must appear after other arguments, but the order of the pairs does not matter.

*Before R2021a, use commas to separate each name and value, and enclose Name in quotes.*

Example: 'SPICEFile', 'ee\_SPICE0ptiMOS20V.lib', 'Subcircuit', 'BSC019N02KS'

### SPICEFile — SPICE netlist file

character vector

Name of the SPICE netlist file you want to generate the characteristics from, specified as the comma-separated pair consisting of 'SPICEFile' and a character vector.

### Subcircuit — SPICE subcircuit

character vector

Name of the subcircuit from a SPICE netlist file, specified as the comma-separated pair consisting of 'Subcircuit' and a character vector.

### SPICEPath — Path of the SPICE simulation engine executable file

character vector

Path of the SPICE simulation engine executable file, specified as the comma-separated pair consisting of 'SPICEPath' and a character vector.

**SPICETool — Name of the SPICE simulation engine**

character vector

Name of the SPICE simulation engine executable file, specified as the comma-separated pair consisting of 'SPICETool' and either LTSpice or SIMatrix.

**Vt — Threshold voltage**

1.4 (default) | positive scalar

Approximate value of the threshold voltage of a MOSFET, in volts, specified as the comma-separated pair consisting of 'Vt' and a positive scalar. This variable is used to determine the range of the gate-source voltage for plotting. The gate-source voltage sweeps from 0 to 3 \* Vt for Id versus Vgs characteristics.

**Vds — Maximum drain-to-source voltage**

20 (default) | positive scalar

Approximate value of the maximum drain-to-source voltage for a MOSFET, in volts, specified as the comma-separated pair consisting of 'Vds' and a positive scalar. This variable is used to determine the range of the drain-source voltage for plotting. The drain-source voltage sweeps from 0 to Vds for Id versus Vds characteristics.

**CheckIdVgs — Id versus Vgs test flag**

1 (default) | 0

Id versus Vgs test flag, specified as the comma-separated pair consisting of 'CheckIdVgs' and either 0 or 1. To run the Id versus Vgs test, set this value to 1.

**CheckIdVds — Id versus Vds test flag**

1 (default) | 0

Id versus Vds test flag, specified as the comma-separated pair consisting of 'CheckIdVds' and either 0 or 1. To run the Id versus Vds test, set this value to 1.

**CheckQiss — Gate charge test flag**

0 (default) | 1

Gate charge test flag, specified as the comma-separated pair consisting of 'CheckQiss' and either 0 or 1. To run the gate charge test, set this value to 1.

**CheckQoss — Output charge test**

0 (default) | 1

Output charge test flag, specified as the comma-separated pair consisting of 'CheckQoss' and either 0 or 1. To run the output charge test, set this value to 1.

**CheckBreakdown — Breakdown test**

0 (default) | 1

Breakdown test flag, specified as the comma-separated pair consisting of 'CheckBreakdown' and either 0 or 1. To run the breakdown test, set this value to 1.

**Ciss — Input capacitance**

440e-12 (default) | scalar

Approximate value of the input capacitance, in farad, used to determine the driving current for a Qiss characteristic, specified as the comma-separated pair consisting of 'Ciss' and a scalar.

### **Coss — Output capacitance**

30e-12 (default) | scalar

Approximate value of the output capacitance used to determine the driving current for a Qoss characteristic, in farad, specified as the comma-separated pair consisting of 'Coss' and a scalar.

### **RelTol — Relative tolerance**

1e-4 (default) | scalar

Relative tolerance parameter used in SPICE simulations, specified as the comma-separated pair consisting of 'RelTol' and a scalar.

### **AbsTol — Absolute tolerance**

1e-4 (default) | scalar

Absolute current tolerance parameter used in SPICE simulations, specified as the comma-separated pair consisting of 'AbsTol' and a scalar.

### **VnTol — Absolute voltage tolerance**

1e-4 (default) | scalar

Absolute voltage tolerance parameter used in SPICE simulations, specified as the comma-separated pair consisting of 'VnTol' and a scalar.

### **BreakdownScalingForVds — Breakdown test**

2 (default) | scalar

Coefficient of Vds for breakdown test, specified as the comma-separated pair consisting of 'BreakdownScalingForVds' and a scalar. The drain voltage increases up to  $\text{BreakdownScalingForVds} * V_{ds}$ .

### **GeneratePlots — Plot generation**

0 (default) | 1

Whether to plot the results of the simulations, specified as the comma-separated pair consisting of 'GeneratePlots' and either 0 or 1.

### **AbsErrTol — Absolute error tolerance**

1e-3 (default) | scalar

Value of the absolute error tolerated between the SPICE and Simscape results, specified as the comma-separated pair consisting of 'AbsErrTol' and a scalar.

### **RelErrTol — Relative error tolerance**

1e-2 (default) | scalar

Value of the relative error tolerated between the SPICE and Simscape results compared to SPICE, specified as the comma-separated pair consisting of 'RelErrTol' and a scalar. The indices for the differences beyond the error tolerances are stored in the output argument, `outputData`, when the absolute and relative errors between the SPICE and Simscape results are greater than 'AbsErrTol' and 'RelErrTol', respectively.

## Output Arguments

### **outputData** — Simulation results

structure

Simulation results, returned as a structure with these fields:

### **plots\_Simscape/plots\_SPICEToolvsSimscape** — Simulation plot results

structure

Table with the plot results, returned as a structure, that contains:

- The voltage and current values for Simscape (and SPICETool if you provide the SPICE library file) for the Id versus Vgs, Id versus Vds, and Breakdown tests
- The voltage indices for the differences beyond the error tolerances for Id versus Vgs, Id versus Vds, and Breakdown tests (if you provide the SPICE library file)
- The simulation time values and the voltage values for Simscape (and SPICETool if you provide the SPICE library file) for gate charge and output charge tests
- The simulation time indices for the differences beyond the error tolerances for gate charge and output charge tests (if you provide the SPICE library file)

### **SPICE\_library\_file** — SPICE library file name

character vector

Name of the SPICE library file, returned as a character vector.

### **SPICE\_library\_file\_timestamp** — Timestamp of SPICE library file

character vector

Timestamp of the SPICE library file, returned as a character vector.

### **Simscape\_file** — Simscape file name

character vector

Name of the Simscape file, returned as a character vector.

### **Simscape\_file\_timestamp** — Timestamp of Simscape file

character vector

Timestamp of the Simscape file, returned as a character vector

## Version History

Introduced in R2020b

### See Also

subcircuit2ssc

### Topics

“Converting a SPICE Netlist to Simscape Blocks”

# ee\_generateIdealPMSMfluxData

Generate tabulated flux linkage data for ideal PMSM

## Syntax

```
[F,T,dFdA,dFdB,dFdC,dFdX] = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)
F = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)
[F,T,dFdA,dFdB,dFdC,dFdX] = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)
F = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)
```

## Description

`[F,T,dFdA,dFdB,dFdC,dFdX] = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)` generates 4-D flux linkage data, including torque and partial derivatives, for an ideal permanent magnet synchronous motor (PMSM).

Use this function to create test data for the FEM-Parameterized PMSM block, either for validation purposes or to set up a model before the actual flux linkage data is available.

`F = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)` generates 4-D flux linkage matrix  $F$  for an ideal PMSM.

`[F,T,dFdA,dFdB,dFdC,dFdX] = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)` generates 3-D flux linkage data, including torque and partial derivatives, for an ideal PMSM.

`F = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)` generates 3-D flux linkage matrix  $F$  for an ideal PMSM.

## Examples

### Generate 4-D Flux Linkage Data

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the phase current vectors.

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of A-,B-,C-currents and rotor angle

```
[F, T, dFdA, dFdB, dFdC, dFdX] = ee_generateIdealPMSMfluxData(PM, Ld, Lq, L0, iA, iB, iC, X);
```

The function returns a 4-D flux linkage matrix F, a 4-D torque matrix T, and four 4-D matrices for the flux linkage partial derivatives. The four partial derivative matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the three phase currents and the rotor angle.

### Generate 4-D Flux Linkage Matrix F

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the phase current vectors.

```
iA = linspace(-250, 250, 5);
iB = iA;
iC = iA;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0, 360/N, 180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of A-,B-,C-currents and rotor angle

```
F = ee_generateIdealPMSMfluxData(PM, Ld, Lq, L0, iA, iB, iC, X);
```

The function returns a 4-D flux linkage matrix F. The matrix dimensions correspond to the three phase currents and the rotor angle.

### Generate 3-D Flux Linkage Data

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the *d*-axis and *q*-axis current vectors.

```
iD = linspace(-250, 250, 5);
iQ = iD;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0, 360/N, 180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of  $d$ -axis and  $q$ -axis currents and rotor angle.

```
[F,T,dFdA,dFdB,dFdC,dFdX] = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,iD,iQ,X);
```

The function returns a 3-D flux linkage matrix  $F$ , a 3-D torque matrix  $T$ , and four 3-D matrices for the flux linkage partial derivatives. The four partial derivative matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

### Generate 3-D Flux Linkage Matrix $F$

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the  $d$ -axis and  $q$ -axis current vectors.

```
iD = linspace(-250,250,5);
iQ = iD;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of  $d$ -axis and  $q$ -axis currents and rotor angle.

```
F = ee_generateIdealPMSMfluxData(PM,Ld,Lq,L0,iD,iQ,X);
```

The function returns a 3-D flux linkage matrix  $F$ . The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

## Input Arguments

### **PM — Peak permanent magnet flux linkage, in weber-turns**

scalar

Peak permanent magnet flux linkage, in weber-turns, specified as a scalar.

Data Types: double

### **Ld — D-axis inductance, in henries**

scalar

$D$ -axis inductance, in henries, specified as a scalar.

Data Types: double

### **Lq — Q-axis inductance, in henries**

scalar

Q-axis inductance, in henries, specified as a scalar.

Data Types: double

**L0 – Zero-sequence inductance, in henries**

scalar

Zero-sequence inductance, in henries, specified as a scalar.

Data Types: double

**A – A-phase current, in amperes**

vector

A-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**B – B-phase current, in amperes**

vector

B-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**C – C-phase current, in amperes**

vector

C-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**D – D-axis current, in amperes**

vector

D-axis current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 3-D flux linkage data.

Data Types: double

**Q – Q-axis current, in amperes**

vector

Q-axis current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 3-D flux linkage data.

Data Types: double

**X – Rotor angle, in radians**

vector



The rotor angle, in radians, specified as a vector. The values must be in the range from zero to  $2\pi/N$ , where  $N$  is the number of pole pairs.

Data Types: `double`

## Output Arguments

### **F — Flux linkage**

matrix

The flux linkage, in weber-turns, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

### **T — Torque**

matrix

Torque, in  $N\cdot m$ , returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

### **dFdA — Flux linkage partial derivative with respect to the A-phase current**

matrix

Flux linkage partial derivative with respect to the A-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

### **dFdB — Flux linkage partial derivative with respect to the B-phase current**

matrix

Flux linkage partial derivative with respect to the B-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

### **dFdC — Flux linkage partial derivative with respect to the C-phase current**

matrix

Flux linkage partial derivative with respect to the C-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

### **dFdX — Flux linkage partial derivative with respect to the rotor angle**

matrix

Flux linkage partial derivative with respect to the rotor angle, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

## Algorithms

The flux linking each winding has contributions from the permanent magnet plus the three windings. Therefore, the total flux is given by [1] on page 2-20:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix}$$

$$L_{aa} = L_s + L_m \cos(2\theta_r)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_r - 2\pi/3))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_r + 2\pi/3))$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(\theta_r + \pi/6)$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(\theta_r + \pi/6 - 2\pi/3)$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos(\theta_r + \pi/6 + 2\pi/3)$$

$$\psi_{am} = \psi_m \cos \theta_e$$

$$\psi_{bm} = \psi_m \cos(\theta_e - 2\pi/3)$$

$$\psi_{cm} = \psi_m \cos(\theta_e + 2\pi/3)$$

Here,  $\theta_e$  is the electrical angle, which is related to rotor angle  $\theta_r$  by  $\theta_e = N \cdot \theta_r$ . The function assumes that the permanent magnet flux linking the A-phase winding is at the maximum for  $\theta_e = 0$ .

The function output F corresponds to  $\psi_a$  tabulated as a function of A-phase current, B-phase current, C-phase current, and rotor angle.

$L_s$ ,  $L_m$ , and  $M_s$  are related to input arguments  $L_d$ ,  $L_q$ , and  $L_0$  by:

$$L_s = \frac{L_0}{3} + \frac{L_d}{3} + \frac{L_q}{3}$$

$$M_s = \frac{L_d}{6} - \frac{L_0}{3} + \frac{L_q}{6}$$

$$L_m = \frac{L_d}{3} - \frac{L_q}{3}$$

## Version History

Introduced in R2017a

## References

- [1] Anderson, P.M. *Analysis of Faulted Power Systems*. 1st Edition. Wiley-IEEE Press, July 1995, p.187.

## **See Also**

ee\_calculateFluxPartialDerivatives

## **Topics**

“HEV PMSM Drive Test Harness”

“PMSM Iron Losses”

## ee\_getBillOfMaterials

Generate report for current model and all subsystems

### Syntax

```
ee_getBillOfMaterials
ee_getBillOfMaterials(modelName)
reportName = ee_getBillOfMaterials(modelName)
```

### Description

`ee_getBillOfMaterials` generates a report of all subsystems for the top-level SLX model. The report is an HTML page that contains an overview of the model that generated the report, a list of the physical components from the Simscape Foundation Library and Electrical library used inside the model, and the values of the parameters for each component in the model.

`ee_getBillOfMaterials(modelName)` generates a report of all subsystems for the `modelName` model.

`reportName = ee_getBillOfMaterials(modelName)` generates a report of all subsystems for the `modelName` model and returns the filename of the HTML report. To open the report, use the `web` function.

### Examples

#### Generate Bill of Materials Report for the Current Model

Open the model. At the MATLAB command prompt, enter:

```
ee_igbt_losses
```

The `ee_igbt_losses` model is the current model.

Generate the bill of materials report.

```
ee_getBillOfMaterials
```

The `ee_getBillOfMaterials` function generates an HTML report that contains an overview of the `ee_igbt_losses` model, a list of the physical components from the Simscape Foundation Library and Electrical library used inside the model, and the values of the parameters for each component in the model.

#### Generate Bill of Materials Report for a Model That Is Not the Current Model

Open the model. At the MATLAB command prompt, enter:

```
modelName = 'ee_igbt_losses'
open_system(modelName)
```

Generate the bill of materials report.

```
ee_getBillOfMaterials(modelName)
```

The `ee_getBillOfMaterials` function generates an HTML report of the `modelName` model that contains an overview of the model, a list of the physical components from the Simscape Foundation Library and Electrical library used inside the model, and the values of the parameters for each component in the model.

## Input Arguments

### **modelName** — Model name

case-sensitive string

Name of the model for which you want to generate the bill of materials report, specified as a case-sensitive string.

## Output Arguments

### **reportName** — Report name

character vector

File name of the bill of materials report, returned as character vector.

## Limitations

- The report does not include blocks that are commented out in the model.
- If your model contains blocks or subsystems with multiple modeling options, this function only generates a report for the current active modeling option.

## Version History

**Introduced in R2020b**

## See Also

## ee\_getCauerFromFoster

Get equivalent resistance and capacitance vectors for Cauer Thermal Model block from Foster Thermal Model block

### Syntax

`[RthVecCauer,CthVecCauer] = ee_getCauerFromFoster(RthVecFoster,CthVecFoster)`

### Description

`[RthVecCauer,CthVecCauer] = ee_getCauerFromFoster(RthVecFoster,CthVecFoster)` returns the equivalent resistance and capacitance vectors for a Cauer Thermal Model block from the resistance and capacitance vectors of a Foster Thermal Model block.

The thermal models are equivalent because they have the same transfer function between the heat flow rate and the total temperature drop.

The Foster transfer function,  $TF_{foster}(s)$ , defines the relationship between the heat flow rate and the total temperature drop:

$$TF_{Foster}(s) = \sum_{i=1}^N \frac{R_{Foster_i}}{1 + R_{Foster_i}C_{Foster_i}s}$$

where  $R_{Foster}$  is the Foster thermal resistance and  $C_{Foster}$  is the Foster capacitance.

To convert Foster resistances and capacitances to Cauer resistances and capacitances, the function uses the Euclid long division algorithm to express the Foster transfer function in the form of the Cauer transfer function,  $TF_{Cauer,N}(s)$ :

$$TF_{Cauer,N}(s) = \frac{1}{sC_{CauerN} + \frac{1}{R_{CauerN} + \frac{1}{sC_{CauerN-1} + \frac{1}{R_{CauerN-1} + \frac{1}{TF_{Cauer,N-2}(s)}}}}}$$

where  $R_{Cauer}$  is the Cauer thermal resistance and  $C_{Cauer}$  is the Cauer capacitance.

### Input Arguments

#### RthVecFoster — Foster thermal resistance

vector of positive elements

Thermal resistance values of the Foster Thermal Model block, specified as a vector of positive elements. The size of this vector must be equal to the size of the CthVecFoster vector.

#### CthVecFoster — Foster capacitance

vector of positive elements

Capacitance values of the Foster Thermal Model block, specified as a vector of positive elements. The size of this vector must be equal to the size of the RthVecFoster vector.

## Output Arguments

### **RthVecCauer** – Cauer thermal resistance

vector of positive elements

Equivalent thermal resistance values of the Cauer Thermal Model block, returned as a vector of positive elements.

### **CthVecCauer** – Cauer capacitance

vector of positive elements

Equivalent capacitance values of the Cauer Thermal Model block, returned as a vector of positive elements. The size of this vector is equal to the size of the RthVecCauer vector.

## Version History

Introduced in R2022b

## References

- [1] Subhash Joshi, T. G., and Vinod John. *Combined Transient Thermal Impedance Estimation for Pulse-Power Applications*. In 2017 National Power Electronics Conference (NPEC), 42-47. Pune: IEEE, 2017. <https://doi.org/10.1109/NPEC.2017.8310432>.

## See Also

Cauer Thermal Model | Foster Thermal Model

## Topics

“Simulating Thermal Effects in Semiconductors”

## ee\_getEfficiency

Calculate efficiency as function of dissipated power losses

### Syntax

```
efficiency = ee_getEfficiency('loadIdentifier',node)
efficiency = ee_getEfficiency('loadIdentifier',node,...
startTime,endTime)
[efficiency,lossesTable] = ee_getEfficiency('loadIdentifier',node)
```

### Description

`efficiency = ee_getEfficiency('loadIdentifier',node)` returns the efficiency of a circuit based on the data extracted from a Simscape logging node.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all semiconductor blocks in the model. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

Checking efficiency allows you to determine if circuit components are operating within their requirements. All blocks in the Semiconductor Devices library, as well as some other blocks, have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the block. This instantaneous dissipated power includes only the real power (not the reactive or apparent power) that the block dissipates. When you log simulation data, the time-value series for this variable represents the power dissipated by the block over time. You can view and plot this data using the Simscape Results Explorer. The `ee_getPowerLossTimeSeries` function also allows you to access this data.

---

**Note** The `power_dissipated` internal variable does not report dynamic losses incurred from semiconductor switching or magnetic hysteresis.

Three different variables, `lastTurnOnLoss`, `lastTurnOffLoss`, and `lastReverseRecoveryLoss` report the switching losses.

---

Switching losses are losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and also with the energy dissipated during a reverse recovery event. They are frequency dependent.

The `ee_getEfficiency` function calculates the efficiency of the circuit based on the losses for blocks that have a `power_dissipated` variable and that you identify as a load block. The equation for efficiency is

$$Eff = 100 \cdot \frac{P_{load}}{P_{loss} + P_{load}}$$

where:



- $Eff$  is the efficiency of the circuit.
- $P_{load}$  is the output power, that is, the power dissipated by load blocks.
- $P_{loss}$  is the power dissipated by nonload blocks.

This equation assumes that all loss mechanisms are captured by blocks containing at least one `power_dissipated` variable. If the model contains any lossy blocks that do not have this variable, the efficiency calculation gives incorrect results.

Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses to provide the total power loss for the block, averaged over simulation time. The function uses the loss data to calculate the efficiency of the circuit.

`efficiency = ee_getEfficiency('loadIdentifier',node,...  
startTime,endTime)` returns the efficiency of a circuit based on the `power_dissipated` data extracted from a Simscape logging node within a time interval. `startTime` and `endTime` represent the start and end of the time interval for calculating the efficiency. If you omit these two input arguments, the function calculates the efficiency over the whole simulation time.

`[efficiency,lossesTable] = ee_getEfficiency('loadIdentifier',node)` returns the efficiency of a circuit and the power loss contributions of the nonload blocks in a circuit based on the data extracted from a Simscape logging node.

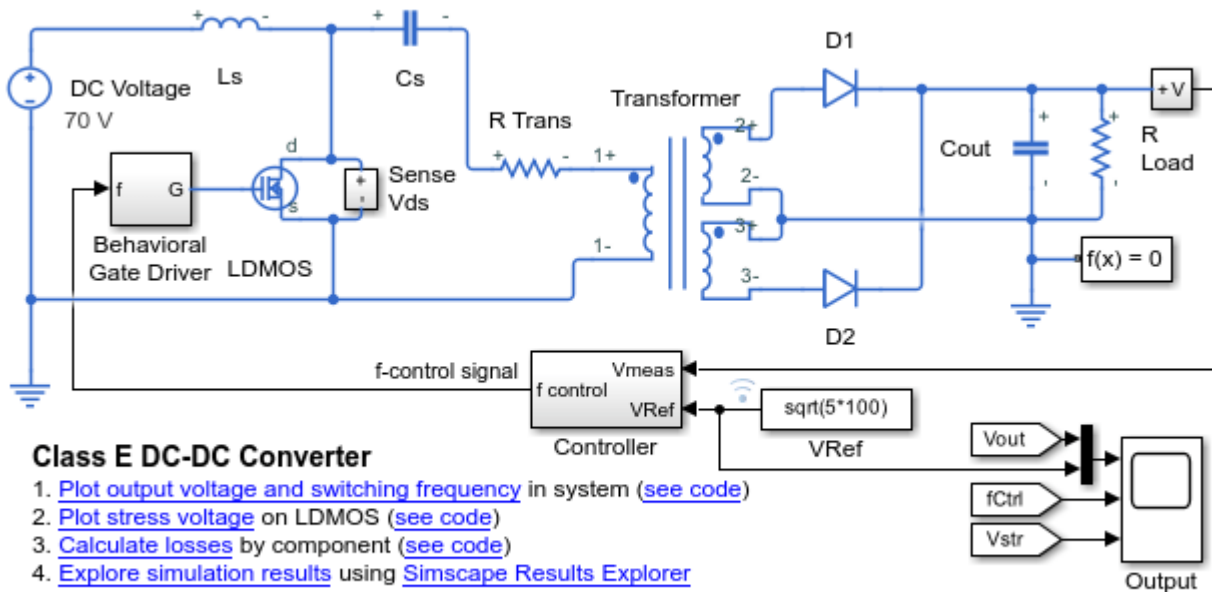
## Examples

### Calculate Efficiency for a Circuit

This example shows how to calculate efficiency based on the power dissipated by blocks in a circuit using the `ee_getEfficiency` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dc_dc_class_e';  
open_system(model)
```



Copyright 2016-2022 The MathWorks, Inc.

The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain Load in their names. Therefore, you can use the string Load as the `loadIdentifier` argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the `loadIdentifier` argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

Calculate efficiency and display the results.

```
efficiency = ee_getEfficiency('Load',simlog_ee_converter_dcdc_class_e)
```

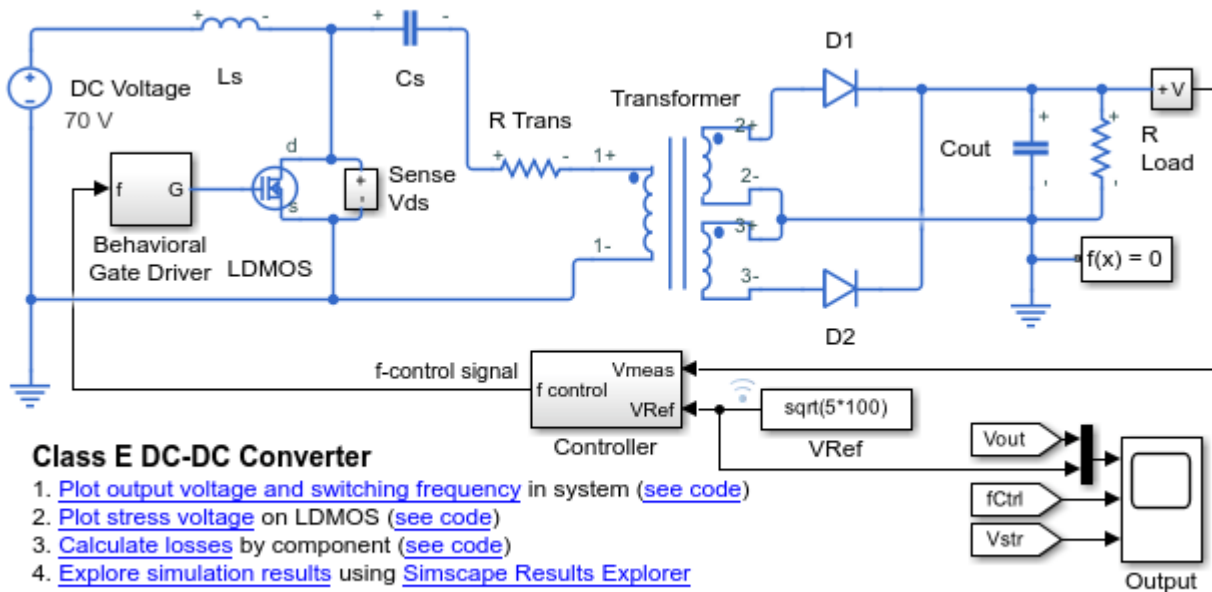
```
efficiency =  
90.0216
```

### Calculate Efficiency of a Circuit for a Specific Time Period

This example shows how to calculate efficiency based on the power dissipated for a specific time period using the `ee_getEfficiency` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dcdc_class_e';
open_system(model)
```



Copyright 2016-2022 The MathWorks, Inc.

The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain Load in their names. Therefore, you can use the string Load as the `loadIdentifier` argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the `loadIdentifier` argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

The model simulation time (`t`) is `1.25e-4` seconds. Calculate efficiency for the interval when `t` is between `1e-4` and `1.25e-4` seconds.

```
efficiency = ee_getEfficiency('Load',simlog_ee_converter_dcdc_class_e,1e-4,1.25e-4)
```

```
efficiency =
```

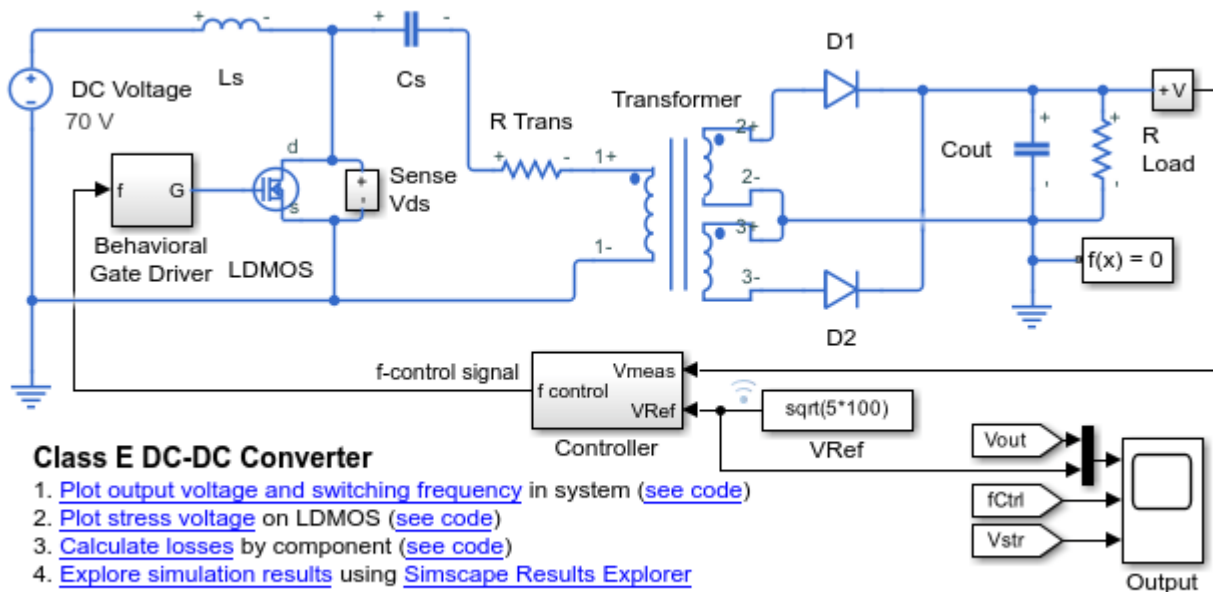
```
90.4782
```

## Calculate Efficiency and Power-Loss Contributions

This example shows how using the `ee_getEfficiency` function allows you to calculate both the efficiency of the circuit and the power-loss contributions of the nonload blocks based on the power that they dissipate.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dcdc_class_e';
open_system(model)
```



Copyright 2016-2022 The MathWorks, Inc.

The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain Load in their names. Therefore, you can use the string Load as the `loadIdentifier` argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the `loadIdentifier` argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

Calculate the efficiency and power-loss contributions due to dissipated power.

```
[efficiency, lossesTable] = ee_getEfficiency('Load', simlog_ee_converter_dcdc_class_e)
```

```
efficiency =
```

```
90.0216
```

```
lossesTable =
```

```
7x2 table
```

LoggingNode	Power
{'ee_converter_dcdc_class_e.LDMOS' }	3.6582
{'ee_converter_dcdc_class_e.R_Trans.Resistor'}	2.911
{'ee_converter_dcdc_class_e.D2' }	1.9516
{'ee_converter_dcdc_class_e.D1' }	1.8436
{'ee_converter_dcdc_class_e.Cs' }	0.27391
{'ee_converter_dcdc_class_e.Ls' }	0.27098
{'ee_converter_dcdc_class_e.Cout' }	0.0004458

## Input Arguments

### 'loadIdentifier' — Identify load blocks in the circuit

case-sensitive string

String that is a complete or partial match for the names of load blocks in the circuit. For example, consider a circuit that contains the four semiconductor blocks shown in the table.

Block Name in the Model	IGBT	IGBT1_Load	Diode	Diode1	
<b>Block Type</b>	N-Channel IGBT	N-Channel IGBT	Diode	Diode	
<b>Block Role in the Model</b>	Source	Load	Load	Load	
<b>'loadIdentifier'</b>	'IGBT'	Yes	Yes	No	No
	'Diode'	No	No	Yes	Yes
	'Load'	No	Yes	No	No
	'1'	No	Yes	No	Yes
	'D'	No	No	Yes	Yes
	'd'	No	Yes	Yes	Yes

The ee\_getEfficiency function returns data just for the three load blocks only when the 'loadIdentifier' is 'd'.

A load-block naming schema that gives you better control over the output of the ee\_getEfficiency function is shown in this table.

Block Name in the Model	IGBT	IGBT1_Load	Diode_Load	Diode1_Load
<b>Block Type</b>	N-Channel IGBT	N-Channel IGBT	Diode	Diode
<b>Block Role in the Model</b>	Source	Load	Load	Load

Block Name in the Model		IGBT	IGBT1_Load	Diode_Load	Diode1_Load
'loadIdentifier'	'IGBT'	Yes	Yes	No	No
	'Diode'	No	No	Yes	Yes
	'Load'	No	Yes	Yes	Yes

Example: 'Load'

Data Types: string

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If **node** is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If **node** is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog.Cell1.MOS1`

### **startTime** — Start of the time interval for calculating the efficiency

0 (default) | real number

Start of the time interval for calculating the efficiency, specified as a real number, in seconds. **startTime** must be greater than or equal to the simulation **Start time** and less than **endTime**.

Data Types: double

### **endTime** — End of the time interval for calculating the efficiency

simulation stop time (default) | real number

End of the time interval for calculating the efficiency, specified as a real number, in seconds. **endTime** must be greater than **startTime** and less than or equal to the simulation **Stop time**.

Data Types: double

## Output Arguments

### **efficiency** — Efficiency of the circuit

percentage

Efficiency of the circuit based on data extracted from a Simscape logging node.

### **lossesTable** — Dissipated power losses and switching losses for each blocks

table

Dissipated power losses and switching losses for each block, returned as a table. The first column lists logging nodes for all blocks that have at least one `power_dissipated` variable. The second

column lists the corresponding losses in watts. The third column lists the switching losses of each block, in watts. Switching losses comprise losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and with the energy dissipated during a reverse recovery event.

## Assumptions

- The output power equals the total power dissipated by blocks that you identify as load blocks.
- The input power equals the output power plus the total power dissipated by blocks that you do not identify as load blocks.

## Version History

Introduced in R2017a

## See Also

[ee\\_getPowerLossSummary](#) | [ee\\_getPowerLossTimeSeries](#) | [sscexplore](#)

## Topics

[“About Simulation Data Logging”](#)

[“About the Simscape Results Explorer”](#)

## ee\_getHarmonics

Return harmonic orders, magnitudes, and fundamental frequency

### Syntax

```
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode,valueIdx)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode,valueIdx,tOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
ee_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest,nHarmonic)
```

### Description

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode) calculates the harmonic orders, magnitudes, and fundamental frequency of a `simscape.logging.Node` of an AC or periodic variable.

The function finds the points in the  $i^{\text{th}}$  signal (`valueIdx`) where the Simscape log crosses a threshold (`offsetOfInterest`). It uses the crossing points to find the required number of periods (`nPeriodOfInterest`) preceding the specified time (`tOfInterest`). Then it inputs the down-selected data to the Goertzel algorithm, which calculates the harmonic magnitudes up to and including the required number of harmonics (`nHarmonic`).

You enter the input arguments in a specific order. The Simscape logging node input argument is required. All other input arguments are optional and have default values. If you are specifying a value for a subsequent optional input argument, enter `[]` to use the default value for an optional input argument.

You can use the `ee_plotHarmonics` function to obtain a bar chart from the same input arguments. You can use the outputs of this function as inputs to the `ee_calculateThdPercent` function to calculate the total harmonic distortion (THD) percentage.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode,valueIdx) uses the index into value data.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode,valueIdx,tOfInterest) uses the simulation time.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest) uses the number of periods of fundamental frequency.



[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...  
 offsetOfInterest) uses the DC offset.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 ee\_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...  
 offsetOfInterest,nHarmonic) uses the number of harmonics.

## Examples

### Analyze Using Default Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the default signal, which is the first, or a-phase, signal at the final simulation time. The function uses the default values of 12 for the number of periods of the signal, 0V for the signal bias, and 30 for the number of harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
[~,harmonicMagnitude,~]= ee_getHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I);
%harmonicMagnitude stores the peak values of the harmonics. To get the RMS values, divide by sqrt(2)
harmonicMagnitude./sqrt(2)
```

ans =

```
1.0e+03 *
Columns 1 through 14
    0.0000    1.3759    0.0000    0.0000    0.0000    0.1548    0.0000    0.0748    0.0000
    0.0000    0.0000    0.0602    0.0000    0.0407
Columns 15 through 28
    0.0000    0.0000    0.0000    0.0357    0.0000    0.0266    0.0000    0.0000
    0.0000    0.0239    0.0000    0.0187    0.0000    0.0000
Columns 29 through 31
    0.0000    0.0170    0.0000
```

### Analyze Using Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the second, or b-phase, signal at a simulation time of 0.5 s. The function uses 10 periods of the signal, assuming a bias of 1V. The function analyzes 15 harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
[~,harmonicMagnitude,~]= ee_getHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I,2,0.5,10,1,15);
%harmonicMagnitude stores the peak values of the harmonics. To get the RMS values, divide by sqrt(2)
harmonicMagnitude./sqrt(2)
```

ans =

```
1.0e+03 *
Columns 1 through 15
    0.0000    1.3761    0.0008    0.0005    0.0006    0.1544    0.0000    0.0748    0.0003
    0.0002    0.0003    0.0599    0.0001    0.0407    0.0004
Column 16
```

```
0.0003
```

### Analyze Using Default and Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the first, or a-phase, signal at a simulation time of 0.5 s. The function uses 12 periods of the signal, assuming a bias of 1V. The function analyzes the default number, 30, of harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
[-,harmonicMagnitude,~]= ee_getHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I,[],0.5,[],1);
%harmonicMagnitude stores the peak values of the harmonics. To get the RMS values, divide by sqrt(2)
harmonicMagnitude./sqrt(2)
```

```
ans =
```

```
1.0e+03 *
Columns 1 through 15
    0.0000    1.3759    0.0000    0.0000    0.0000    0.1548    0.0000    0.0748    0.0000
    0.0000    0.0000    0.0602    0.0000    0.0407    0.0000
Columns 16 through 30
    0.0000    0.0000    0.0357    0.0000    0.0266    0.0000    0.0000    0.0000    0.0239
    0.0000    0.0187    0.0000    0.0000    0.0000    0.0170
Column 31
    0.0000
```

## Input Arguments

### loggingNode — Simscape logging node

1-by-1 `simscape.logging.Node`

Simscape logging node, specified as a 1-by-1 `simscape.logging.Node`. You create a `simscape.logging.Node` by running a simulation with Simscape logging enabled. For information, see “Enable Data Logging for the Whole Model”.

Example: `simlog.Load.V`

The Simscape logging node `simlog.Load.V` contains data from a three-phase voltage.

### valueIdx — Index into value data

1 (default) | scalar

Index into value data, specified as a scalar. Specifies the  $i^{\text{th}}$  variable of interest in the Simscape log.

Example: 2

Specify the b-phase, which is the second signal from a three-phase voltage.

Example: []

Use [] to specify the default value of 1. The a-phase, which is the first signal from a three-phase voltage, is the default signal of interest.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

**tOfInterest – Simulation time**

final time in Simscape log (default) | scalar

Simulation time of interest for harmonic analysis, specified as a scalar.

Example: 0.5

Specify a 0.5 s simulation time.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

**nPeriodOfInterest – Number of periods**

12 (default) | scalar

Number of periods of fundamental frequency to be included in harmonic analysis, specified as a scalar.

Example: 10

Specify 10 periods of the signal.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

**offsetOfInterest – DC offset**

0 (default) | scalar

DC offset in the input signal, specified as a scalar. The function uses this value to find the periods of interest.

Example: 1

Specify a bias of 1 V for the signal.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

**nHarmonic – Number of harmonics**

30 (default) | scalar

Number of harmonics to include in analysis, specified as a scalar.

Example: 15

Specify that the number of harmonics to be analyzed is 15.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

**Output Arguments****harmonicOrder – Harmonic order**

vector

Harmonic orders from 0 up to and including the number of harmonics used in the analysis, returned as a vector.

**harmonicMagnitude – Harmonic magnitude**

vector

Harmonic magnitudes from the 0<sup>th</sup> harmonic up to and including the number of harmonics used in the analysis, returned as a vector.

**fundamentalFrequency — Fundamental frequency**

scalar

Fundamental frequency over the range of the down-selected input data, returned as a scalar.

**Limitations**

- This function requires that you use a fixed-step solver for the Simscape Electrical Power Systems network that you are analyzing. To specify a fixed-step solver for the physical network, use one of the configuration combinations in the table.

Configuration Combination	Global Solver Configuration	Local Solver Configuration
Global variable-step with local fixed-step	Set <b>Type</b> to Variable-step	Enable the options to <b>Use local solver</b> and <b>Use fixed-cost runtime consistency iterations</b>
Global and local fixed-step	Set <b>Type</b> to Fixed-step	Enable the options to <b>Use local solver</b> and <b>Use fixed-cost runtime consistency iterations</b>
Global fixed-step	Set <b>Type</b> to Fixed-step	Clear the option to <b>Use local solver</b>

- This function uses threshold crossing points to determine the fundamental frequency of the data. If your input data is noisy or crosses the threshold more frequently than half of the fundamental period, filter it before you use this function to analyze it.
- This function requires a minimal number of periods. If the minimal number is not met, the function generates a warning message. To increase the number of periods, use one or both of these methods:
  - Increase the simulation time.
  - Increase the switching frequency.

**Version History**

Introduced in R2014a

**See Also****Apps**

Harmonic Analyzer

**Blocks**

Spectrum Analyzer

**Functions**

ee\_calculateThdPercent | ee\_plotHarmonics | sscexplore

**Objects**

simscape.logging.Node

**Topics**

“Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block”

“Choose a Simscape Electrical Function for an Offline Harmonic Analysis”

“Data Logging”

“Harmonic Analysis of a Three-Phase Rectifier”

## ee\_getNodeDvDtSummary

Calculate maximum absolute values of terminal voltage time derivatives (dv/dt) based on logged simulation data

### Syntax

```
summaryTable = ee_getNodeDvDtSummary(node, tau)
summaryTable = ee_getNodeDvDtSummary(node, tau, startTime, endTime)
```

### Description

`summaryTable = ee_getNodeDvDtSummary(node, tau)` calculates the maximum absolute values of rates-of-change of voltage variables for nodes that are based on the `foundation.electrical.electrical` domain, based on logged simulation data. The function returns the data for each terminal in a table. The data in the table appears in descending order according to the maximum magnitude of the rate-of-change of voltage variables with respect to the ground, over the whole simulation time. The table does not contain data for terminals that are held fixed.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the children of that node.

Examining rates-of-change of voltage variables in power electronics circuits is useful for determining the potential for unwanted conducted or radiated emissions. The rate-of-change data also helps you to identify switching devices that might be susceptible to parasitic turn-on. All nodes that are based on the `foundation.electrical.electrical` domain store the potential with respect to electrical ground as the variable `v`. When you log simulation data, the time-value series for this variable represents the trend of the potential over time. You can view and plot this data using the Simscape Results Explorer.

To evaluate the rates-of-change of voltage variables, the `ee_getNodeDvDtSummary` function employs finite difference approximation of the first derivative with respect to time. It performs 1-D data linear interpolation of voltage variables using a uniform grid with the time step, `tau`. The function then applies the central differencing scheme to the interpolated data.

---

**Tip** For small time steps, finite differencing may lead to inaccurate results. The time step `tau` should be small enough to capture waveforms, but not so small that the finite differencing error becomes large. For example, for power transistors with an expected limit of 50 V/ns for their voltage rate-of-change, a reasonable guess for `tau` is 1e-9 s.

---

`summaryTable = ee_getNodeDvDtSummary(node, tau, startTime, endTime)` calculates the maximum absolute values of rates-of-change of voltage variables within a time interval. `startTime` and `endTime` represent the start and end of the time interval for evaluating the maximum values. If

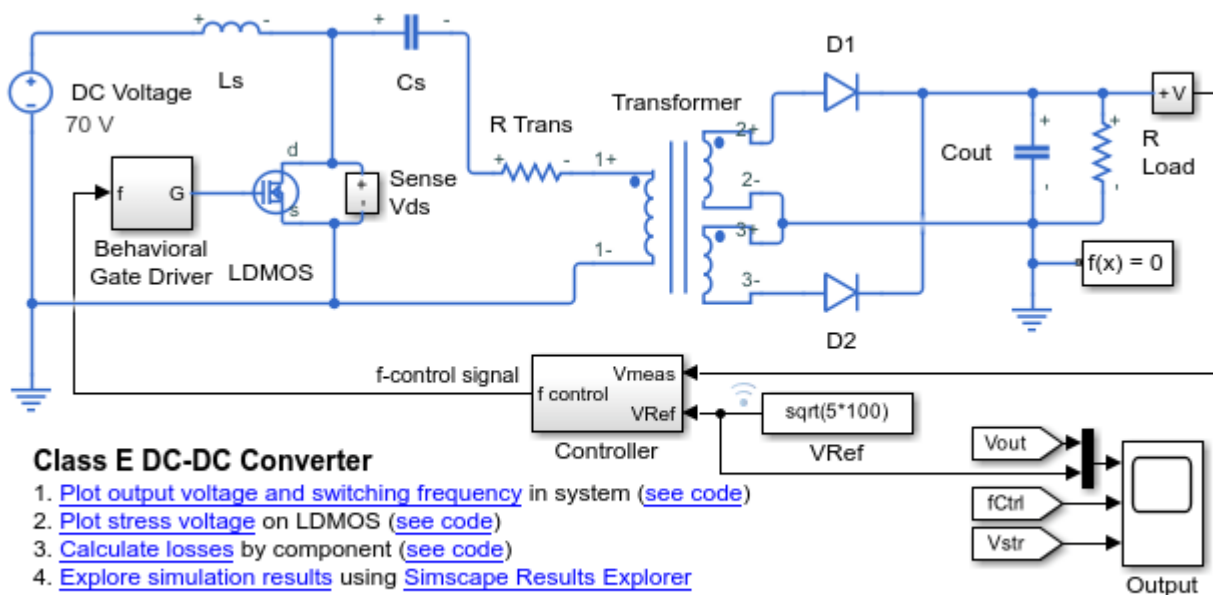
you omit these two input arguments, the function evaluates the maximum absolute values of rates-of-change of voltage variables over the whole simulation time.

## Examples

### Calculate Maximum Voltage Derivatives by Block for the Whole Model

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



Copyright 2016-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

Calculate the maximum absolute values of rates-of-change of voltage variables for the whole model with a time step of  $1e-9$  seconds, and display the results in a table.

```
summaryTable = ee_getNodeDvDtSummary(simlog_ee_converter_dcdc_class_e, 1e-9)
```

```
summaryTable =
```

```
19x3 table
```

```
LoggingNode
```

```
Terminal
```

```
"ee_converter_dcdc_class_e.R_Trans"
```

```
"n"
```

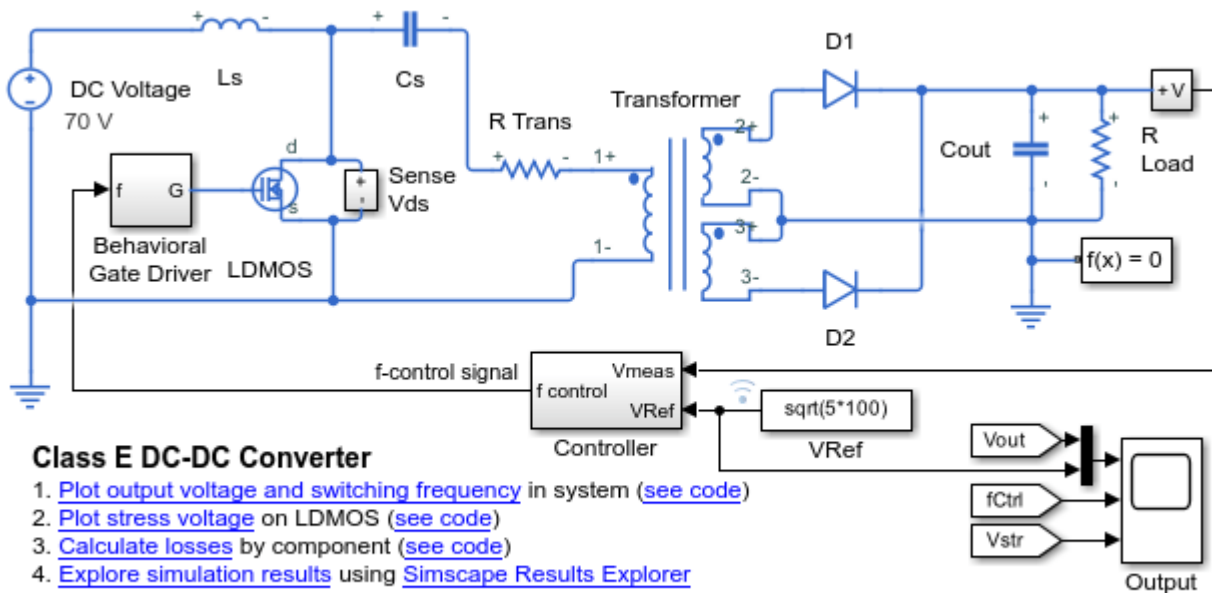
"ee_converter_dcdc_class_e.Transformer"	"p1"
"ee_converter_dcdc_class_e.Cs"	"n"
"ee_converter_dcdc_class_e.R_Trans"	"p"
"ee_converter_dcdc_class_e.Cs"	"p"
"ee_converter_dcdc_class_e.LDMOS"	"D"
"ee_converter_dcdc_class_e.Ls"	"n"
"ee_converter_dcdc_class_e.Sense_Vds.Voltage_Stress_Sensor"	"p"
"ee_converter_dcdc_class_e.D2"	"p"
"ee_converter_dcdc_class_e.Transformer"	"n3"
"ee_converter_dcdc_class_e.D1"	"p"
"ee_converter_dcdc_class_e.Transformer"	"p2"
"ee_converter_dcdc_class_e.Behavioral_Gate_Driver.Controlled_Voltage_Source"	"p"
"ee_converter_dcdc_class_e.LDMOS"	"G"
"ee_converter_dcdc_class_e.Cout"	"p"
"ee_converter_dcdc_class_e.D1"	"n"
"ee_converter_dcdc_class_e.D2"	"n"
"ee_converter_dcdc_class_e.R_Load"	"p"
"ee_converter_dcdc_class_e.Sense_Vout.Voltage_Sensor"	"p"

The table shows the maximum absolute values over the whole simulation time of voltage rates-of-change for all the blocks in the model that have nodes based on the foundation.electrical.electrical domain.

### Calculate Maximum Voltage Derivatives for One Block

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



Copyright 2016-2022 The MathWorks, Inc.



This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

Calculate the maximum absolute values of rates-of-change of voltage variables for the LDMOS block with a time step of  $1e-9$  seconds, and display the results in a table.

```
mosfetTable = ee_getNodeDvDtSummary(simlog_ee_converter_dcdc_class_e.LDMOS,1e-9)
```

```
mosfetTable =
```

```
2x3 table
```

LoggingNode	Terminal	max_abs_dvdt
"LDMOS"	"D"	3.3428e+10
"LDMOS"	"G"	1e+09

The table shows the maximum absolute values over the whole simulation time of voltage rates-of-change for the LDMOS block. The table does not list the S terminal because it is held fixed to the ground.

To explore the voltage data for the LDMOS block further, use the `sscexplore` function.

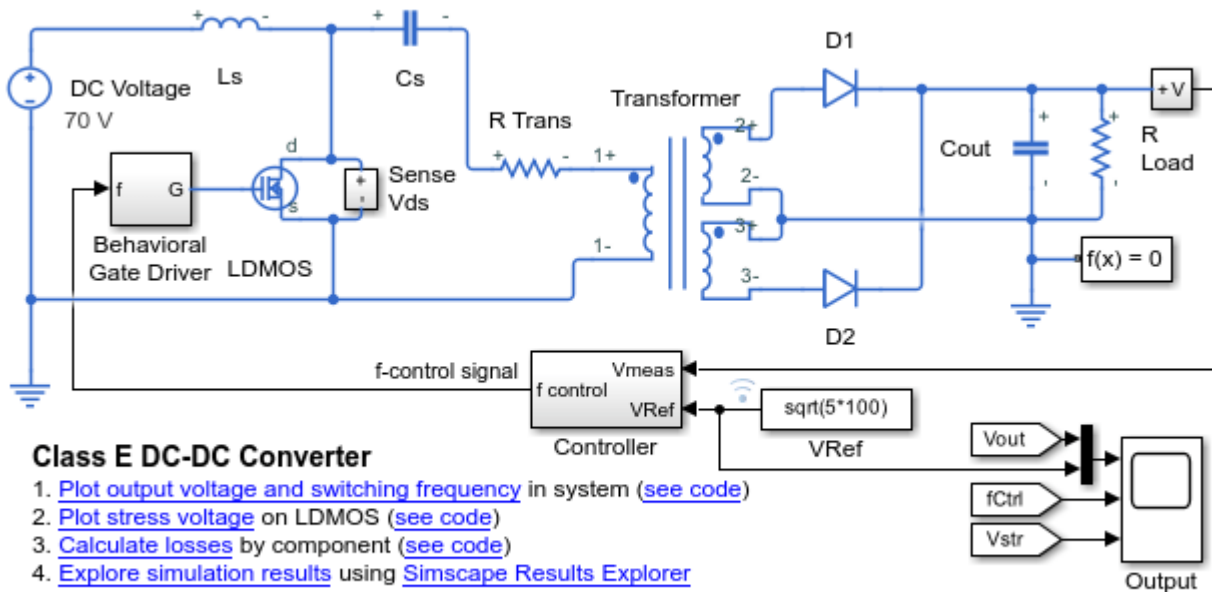
```
sscexplore(simlog_ee_converter_dcdc_class_e.LDMOS,'D.v')
```

The block has a variable, `v`, for each of the D, G, and S terminals.

### Calculate Maximum Voltage Derivatives for a Specific Time Period

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



Copyright 2016-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

The model simulation time is  $1.25e-4$  seconds. Calculate and display the maximum absolute values of rates-of-change of voltage variables for the LDMOS block during the second half of the simulation. Use a time step of  $1e-9$  seconds.

```
mosfetTable1 = ee_getNodeDvDtSummary(simlog_ee_converter_dcdc_class_e.LDMOS,1e-9,0.5*1.25e-4)
```

```
mosfetTable1 =
```

2x3 table

LoggingNode	Terminal	max_abs_dvdt
"LDMOS"	"D"	2.8459e+10
"LDMOS"	"G"	1e+09

The table shows the maximum absolute values of voltage rates-of-change for the LDMOS block during the second half of the simulation. The table does not list the S terminal because it is held fixed to the ground. The magnitude of the D terminal is lower than the magnitude for the whole simulation time because the initial high-magnitude spikes of the stress voltage are disregarded.

To see the voltage time derivative for the D terminal of the LDMOS block over the whole simulation time, use the `ee_getNodeDvDtTimeSeries` function.

## Input Arguments

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog_ee_converter_dcdc_class_e.LDMOS`

### **tau** — Time step for numerical differentiation

real number

Time step for numerical differentiation, specified as a real number, in seconds. `tau` determines the interpolation grid as `startTime:tau:endTime`.

Example: `1e-9`

Data Types: `double`

### **startTime** — Start of the time interval for evaluating the maximum voltage rates-of-change

simulation start time (default) | real number

Start of the time interval for evaluating the maximum absolute values of rates-of-change of voltage variables, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than `endTime`.

Data Types: `double`

### **endTime** — End of the time interval for evaluating the maximum voltage rates-of-change

simulation stop time (default) | real number

End of the time interval for evaluating the maximum absolute values of rates-of-change of voltage variables, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

Data Types: `double`

## Output Arguments

### **summaryTable** — Maximum absolute values of the voltage rates-of-change for each block

table

Maximum absolute values of the voltage rates-of-change for each block, returned as a table. The first column lists all the logging nodes in node that are based on the `foundation.electrical.electrical` domain. The second column lists the terminal names. The third column lists the corresponding maximum absolute values of voltage rates-of-change, in volts per second. The table does not contain data for terminals that are held fixed.

## **Version History**

**Introduced in R2018b**

### **See Also**

`ee_getNodeDvDtTimeSeries` | `sscexplore`

### **Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## ee\_getNodeDvDtTimeSeries

Calculate rates-of-change of voltage variables

### Syntax

```
seriesTable = ee_getNodeDvDtTimeSeries(node,tau)
seriesTable = ee_getNodeDvDtTimeSeries(node,tau,startTime,endTime)
```

### Description

`seriesTable = ee_getNodeDvDtTimeSeries(node,tau)` calculates rates-of-change of voltage variables for nodes that are based on the `foundation.electrical.electrical` domain, based on logged simulation data. The function returns the data for each terminal in a table. The data in the table appears in descending order according to the maximum absolute value of the rate-of-change of voltage variables with respect to the ground, over the whole simulation time. The table does not contain data for terminals that are held fixed.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the children of that node.

Examining rates-of-change of voltage variables in power electronics circuits is useful for determining the potential for unwanted conducted or radiated emissions. The rate-of-change data also helps you to identify unwanted turn-on of switching devices. All nodes that are based on the `foundation.electrical.electrical` domain store the potential with respect to electrical ground as the variable `v`. When you log simulation data, the time-value series for this variable represents the trend of the potential over time. You can view and plot this data using the Simscape Results Explorer.

To evaluate the rates-of-change of voltage variables, the `ee_getNodeDvDtTimeSeries` function employs finite difference approximation of the first derivative with respect to time. It performs 1-D data linear interpolation of voltage variables using a uniform grid with the time step, `tau`. The function then applies the central differencing scheme to the interpolated data.

---

**Tip** For small time steps, finite differencing may lead to inaccurate results. The time step `tau` should be small enough to capture waveforms, but not so small that the finite differencing error becomes large. For example, for power transistors with an expected limit of 50 V/ns for their voltage rate-of-change, a reasonable guess for `tau` is 1e-9 s.

---

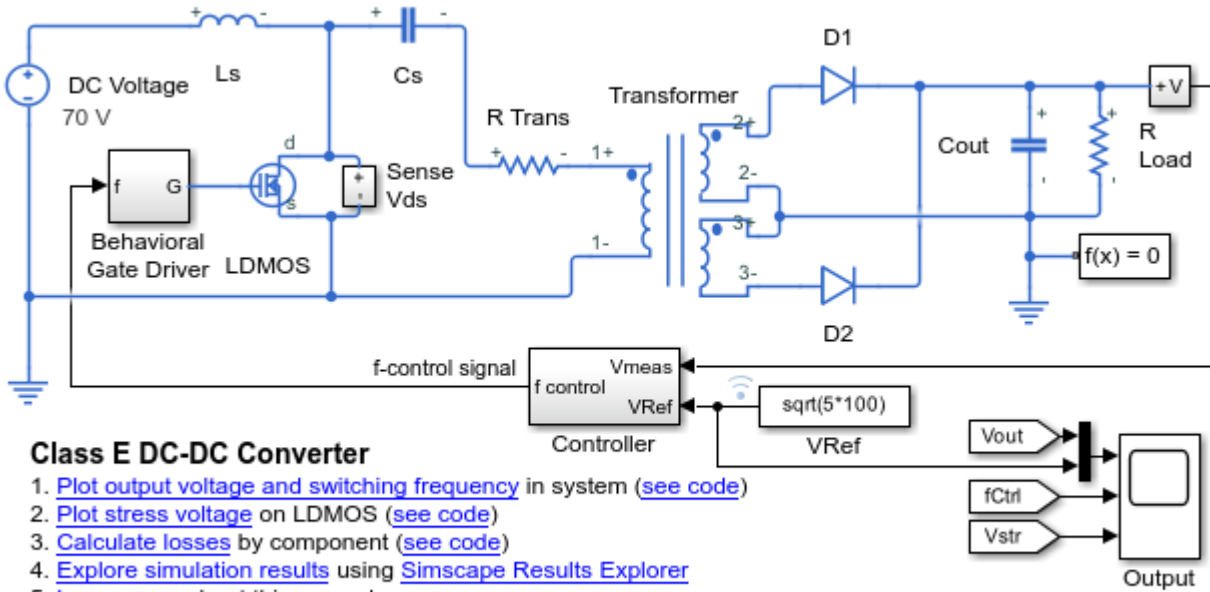
`seriesTable = ee_getNodeDvDtTimeSeries(node,tau,startTime,endTime)` calculates rates-of-change of voltage variables within a time interval. `startTime` and `endTime` represent the start and end of the time interval for evaluating the derivatives of the voltage variables with respect to time. If you omit these two input arguments, the function evaluates rates-of-change of voltage variables over the whole simulation time.

## Examples

### Calculate Voltage Derivatives by Block for the Whole Model

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



#### Class E DC-DC Converter

1. [Plot output voltage and switching frequency](#) in system ([see code](#))
2. [Plot stress voltage](#) on LDMOS ([see code](#))
3. [Calculate losses](#) by component ([see code](#))
4. [Explore simulation results](#) using [Simscape Results Explorer](#)
5. [Learn more](#) about this example

Copyright 2016-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

Calculate rates-of-change of voltage variables for the whole model with a time step of  $1e-9$  seconds, and return the time series data in a table.

```
seriesTable = ee_getNodeDvDtTimeSeries(simlog_ee_converter_dcdc_class_e,1e-9)
```

```
seriesTable =
```

```
19x4 table
```

	LoggingNode	Terminal
	"ee_converter_dcdc_class_e.R_Trans"	"n"
	"ee_converter_dcdc_class_e.Transformer"	"p1"
	"ee_converter_dcdc_class_e.Cs"	"n"
	"ee_converter_dcdc_class_e.R_Trans"	"p"
	"ee_converter_dcdc_class_e.Cs"	"p"

"ee_converter_dcdc_class_e.LDMOS"	"D"
"ee_converter_dcdc_class_e.Ls"	"n"
"ee_converter_dcdc_class_e.Sense_Vds.Voltage_Stress_Sensor"	"p"
"ee_converter_dcdc_class_e.D2"	"p"
"ee_converter_dcdc_class_e.Transformer"	"n3"
"ee_converter_dcdc_class_e.D1"	"p"
"ee_converter_dcdc_class_e.Transformer"	"p2"
"ee_converter_dcdc_class_e.Behavioral_Gate_Driver.Controlled_Voltage_Source"	"p"
"ee_converter_dcdc_class_e.LDMOS"	"G"
"ee_converter_dcdc_class_e.Cout"	"p"
"ee_converter_dcdc_class_e.D1"	"n"
"ee_converter_dcdc_class_e.D2"	"n"
"ee_converter_dcdc_class_e.R_Load"	"p"
"ee_converter_dcdc_class_e.Sense_Vout.Voltage_Sensor"	"p"

The table contains time series data of voltage variables and their first derivatives over the whole simulation time for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain.

View the time series data. From the workspace, open the `seriesTable` table, then open the two `1x125001 double` numeric arrays for the `ee_converter_dcdc_class_e.LDMOS.D`.

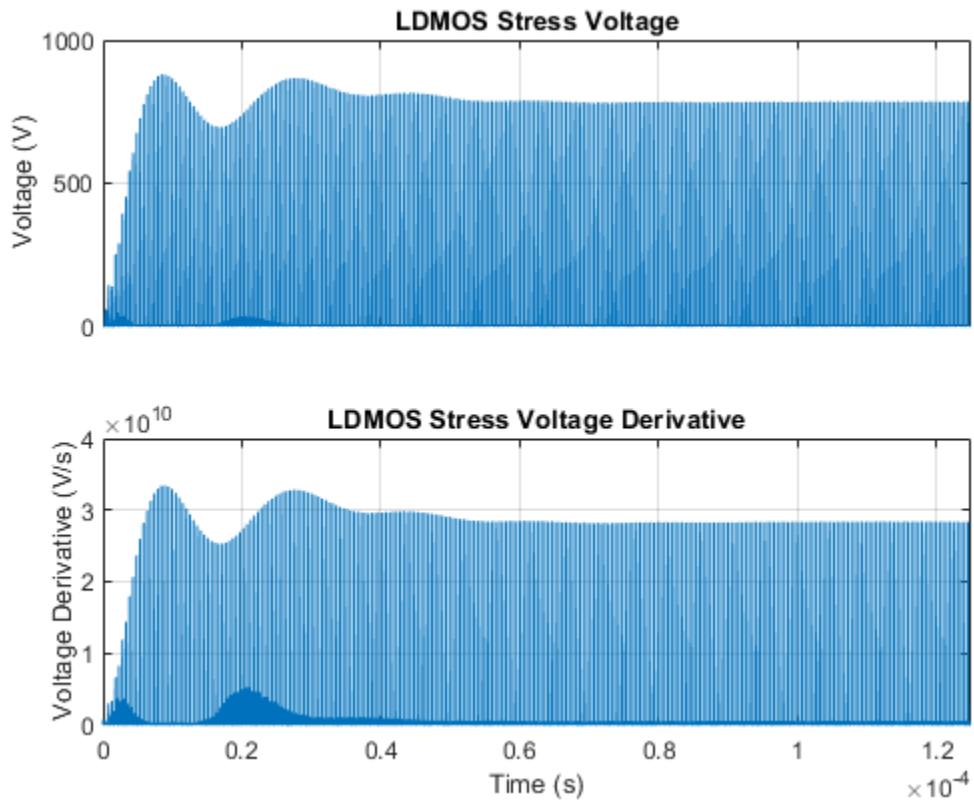
The first array contains the voltage data. The second array contains the voltage derivative data.

Plot the data.

```
time = 0:1e-9:1.25e-4;
vOut = seriesTable.Voltage{6};
dvdtOut = seriesTable.dvdt{6};

ax1 = subplot(2,1,1);
plot(time,vOut),grid;
ylabel('Voltage (V)');
axis([0 1.25e-4 0 1000]);
ax1.XTickLabel = {};
ax1.Title.String = 'LDMOS Stress Voltage';

ax2 = subplot(2,1,2);
plot(time,dvdtOut),grid;
ylabel('Voltage Derivative (V/s)');
xlabel('Time (s)');
axis([0 1.25e-4 0 4e10]);
ax2.Title.String = 'LDMOS Stress Voltage Derivative';
```

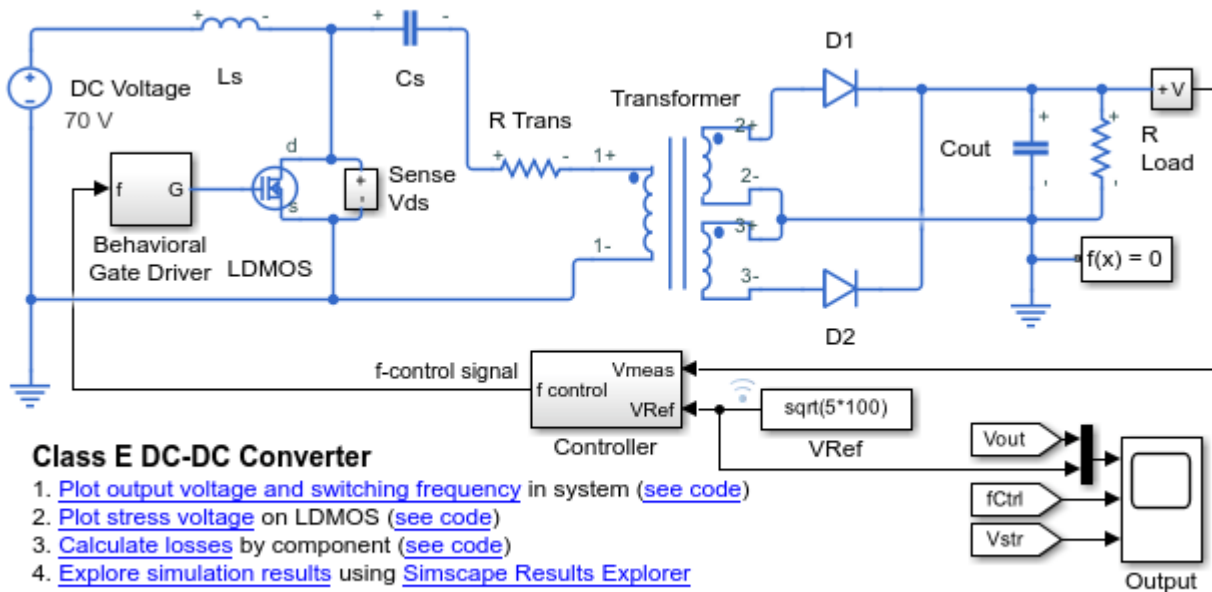


### Calculate Voltage Derivatives for One Block

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```





**Class E DC-DC Converter**

1. [Plot output voltage and switching frequency](#) in system ([see code](#))
2. [Plot stress voltage](#) on LDMOS ([see code](#))
3. [Calculate losses](#) by component ([see code](#))
4. [Explore simulation results](#) using [Simscape Results Explorer](#)
5. [Learn more](#) about this example

Copyright 2016-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

Calculate rates-of-change of voltage variables for the LDMOS block with a time step of  $1e-9$  seconds, and return the time series data in a table.

```
mosfetTable = ee_getNodeDvDtTimeSeries(simlog_ee_converter_dcdc_class_e.LDMOS,1e-9)
```

mosfetTable =

2x4 table

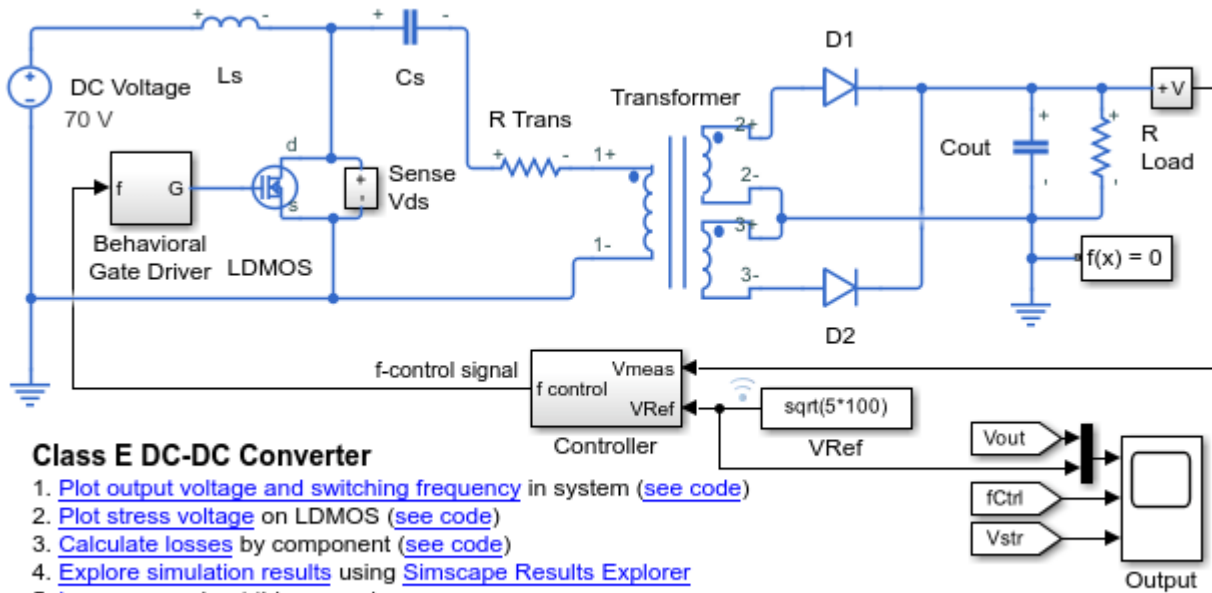
LoggingNode	Terminal	Voltage
"LDMOS"	"D"	{[-0.6661 0.0542 0.0012 3.0486e-04 3.9579e-04 5.0384e-04 6.1195e-04 ...]}
"LDMOS"	"G"	{[10 ...]}

The table contains time series data of voltage variables and their first derivatives over the whole simulation time for the LDMOS block. The table does not list the S terminal because it is held fixed to the ground.

**Calculate Voltage Derivatives for a Specific Time Period**

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



**Class E DC-DC Converter**

1. [Plot output voltage and switching frequency](#) in system (see code)
2. [Plot stress voltage](#) on LDMOS (see code)
3. [Calculate losses](#) by component (see code)
4. [Explore simulation results](#) using [Simscape Results Explorer](#)
5. [Learn more](#) about this example

Copyright 2016-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

The model simulation time is 1.25e-4 seconds. Calculate and display rates-of-change of voltage variables for the Transformer block during the last 0.25e-4 seconds of the simulation. Use a time step of 1e-9 seconds.

```
transformerTable = ee_getNodeDvDtTimeSeries(simlog_ee_converter_dcdc_class_e.Transformer,1e-9,1e-4);
```

```
transformerTable =
```

3x4 table

LoggingNode	Terminal	Voltage
"Transformer"	"p1"	{[-462.7759 -456.2325 -449.6280 -442.9799 -436.2657 -429.49...]}
"Transformer"	"n3"	{[23.1992 23.2005 23.2017 23.2030 23.2042 23.2053 23.2065 23.20...]}
"Transformer"	"p2"	{[22.6972 22.6882 22.6775 22.6642 22.6403 22.5916 22.4859 22.28...]}

The table contains time series data of voltage variables, and their first derivatives, for the Transformer block over the last 0.25e-4 seconds of the simulation. The table does not list terminals that are held fixed to the ground.

View the time series data. From the workspace, open the `transformerTable` table, then open the two 1x25001 double numeric arrays for the Transformer.p1.

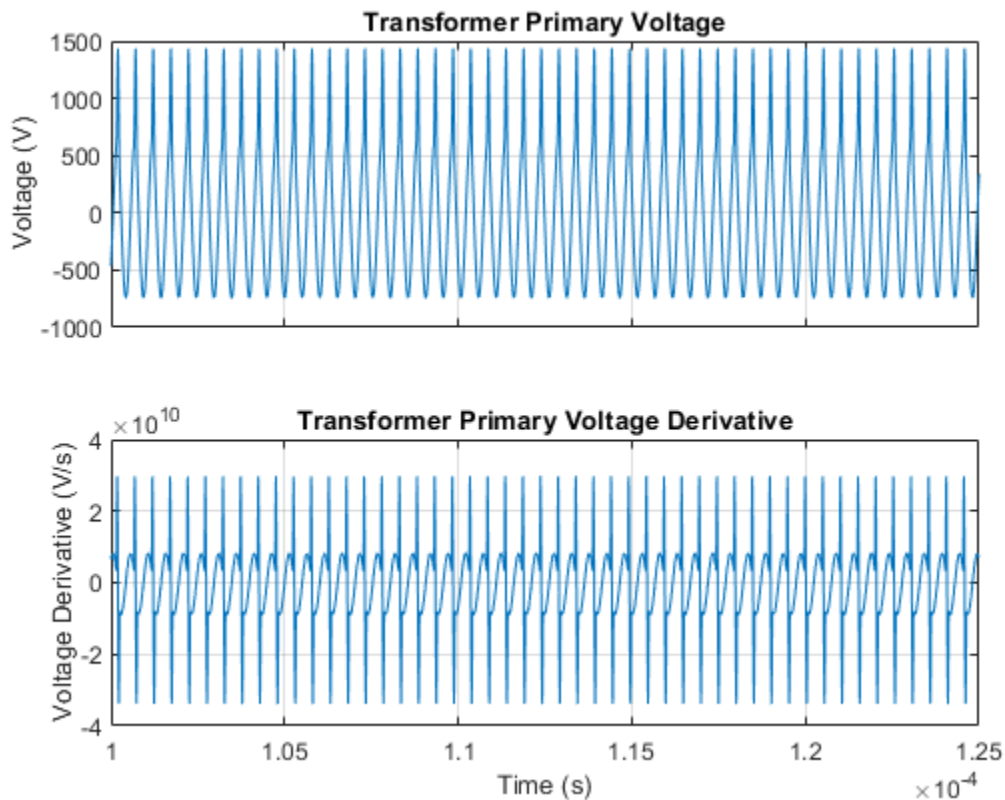
The first array contains the voltage data. The second array contains the voltage derivative data.

Plot the data.

```
time = 1e-4:1e-9:1.25e-4;
vOut = transformerTable.Voltage{1};
dvdtOut = transformerTable.dvdt{1};

ax1 = subplot(2,1,1);
plot(time,vOut),grid;
ylabel('Voltage (V)');
ax1.YLim = [-1000 1500];
ax1.XTickLabel = {};
ax1.Title.String = 'Transformer Primary Voltage';

ax2 = subplot(2,1,2);
plot(time,dvdtOut),grid;
ylabel('Voltage Derivative (V/s)');
xlabel('Time (s)');
ax2.Title.String = 'Transformer Primary Voltage Derivative';
```



## Input Arguments

**node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog_ee_converter_dcdc_class_e.LDMOS`

### **tau — Time step for numerical differentiation**

real number

Time step for numerical differentiation, specified as a real number, in seconds. `tau` determines the interpolation grid as `startTime:tau:endTime`.

Example: `1e-9`

Data Types: `double`

### **startTime — Start of the time interval for evaluating rates-of-change of voltage variables**

simulation start time (default) | real number

Start of the time interval for evaluating rates-of-change of voltage variables, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than `endTime`.

Data Types: `double`

### **endTime — End of the time interval for evaluating rates-of-change of voltage variables**

simulation stop time (default) | real number

End of the time interval for evaluating rates-of-change of voltage variables, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

Data Types: `double`

## **Output Arguments**

### **seriesTable — Time series of the voltage rates-of-change for each block**

table

Time series of the voltage rates-of-change for each block, returned as a table. The first column lists all the logging nodes in node that are based on the `foundation.electrical.electrical` domain. The second column lists the terminal names. The third column lists the corresponding interpolated voltage values, in volts. The fourth column lists the corresponding numerically differentiated values of voltage rates-of-change, in volts per second. The table does not contain data for terminals that are held fixed.

## **Version History**

**Introduced in R2018b**

### **See Also**

`ee_getNodeDvDtSummary` | `sscexplore`

**Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## ee\_getPowerLossSummary

Calculate dissipated power losses and switching losses

### Syntax

```
lossesTable = ee_getPowerLossSummary(node)
lossesTable = ee_getPowerLossSummary(node, startTime, endTime)
```

### Description

`lossesTable = ee_getPowerLossSummary(node)` calculates dissipated power losses and switching losses for semiconductor blocks in a model, based on logged simulation data, and returns the data for each block in a table.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all semiconductor blocks in the model. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

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

---

**Note** The `power_dissipated` internal variable does not report dynamic losses incurred from semiconductor switching or magnetic hysteresis.

Three different variables, `lastTurnOnLoss`, `lastTurnOffLoss`, and `lastReverseRecoveryLoss` report the switching losses.

---

Switching losses are losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and also with the energy dissipated during a reverse recovery event. They are frequency dependent. The `ee_getPowerLossSummary` function averages the switching losses over time and expresses them in watts.

These blocks in the **Semiconductors & Converters** library support the calculation of switching losses and reverse recovery losses, when applicable:

- Diode
- N-Channel IGBT
- GTO
- IGBT (Ideal, Switching)
- MOSFET (Ideal, Switching)

- Thyristor (Piecewise Linear)
- Half-Bridge (Ideal, Switching)

The `ee_getPowerLossSummary` function calculates average losses for each block that has a `power_dissipated` variable. Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses and provides the power loss value for the whole block, averaged over simulation time.

`lossesTable = ee_getPowerLossSummary(node, startTime, endTime)` calculates dissipated power losses within a time interval. `startTime` and `endTime` represent the start and end of the 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.

---

**Note** Depending on the solver and data logging configuration, the node **Start time** and the node **Stop time** can be different from the simulation **Start time** and the simulation **Stop time**.

---

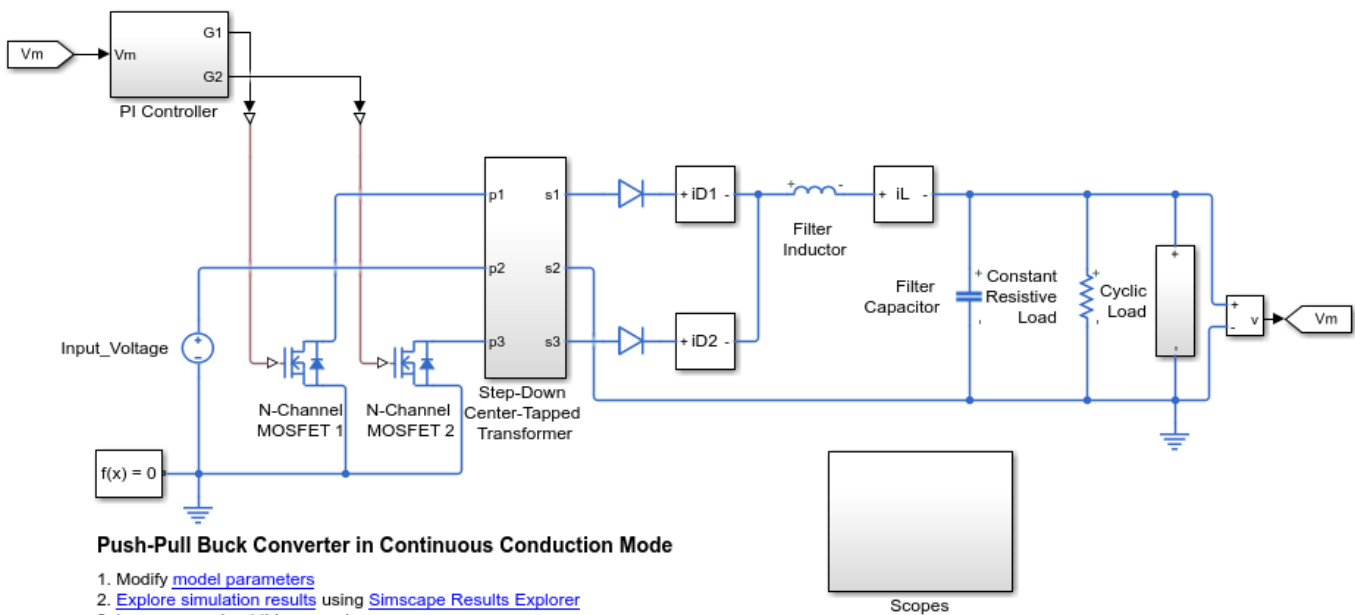
## Examples

### Calculate Average Power Losses for Components of a Block

You can calculate average power losses for the individual components of a block in your model.

1. Open the Push-Pull Buck Converter in Continuous Conduction Mode example model. At the MATLAB® command prompt, enter

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



Copyright 2016-2022 The MathWorks, Inc.

The model has data logging enabled.

2. Add a diode component in the N-Channel MOSFET 1 block using the MATLAB® command prompt:

```
set_param('ee_push_pull_converter_ccm/ N-Channel MOSFET 1', 'diode_param', '2')
```

Alternatively, you can add the component in the Simulink® Editor:

a. Open the Property Inspector pane. In the model window, in the menu bar, click **View > Property Inspector**

b. Click the N-Channel MOSFET1 block to access the block parameters.

c. In the Property Inspector pane, expand the Integral Diode setting and change the value for the **Integral protection** from None to Protection diode with no dynamics.

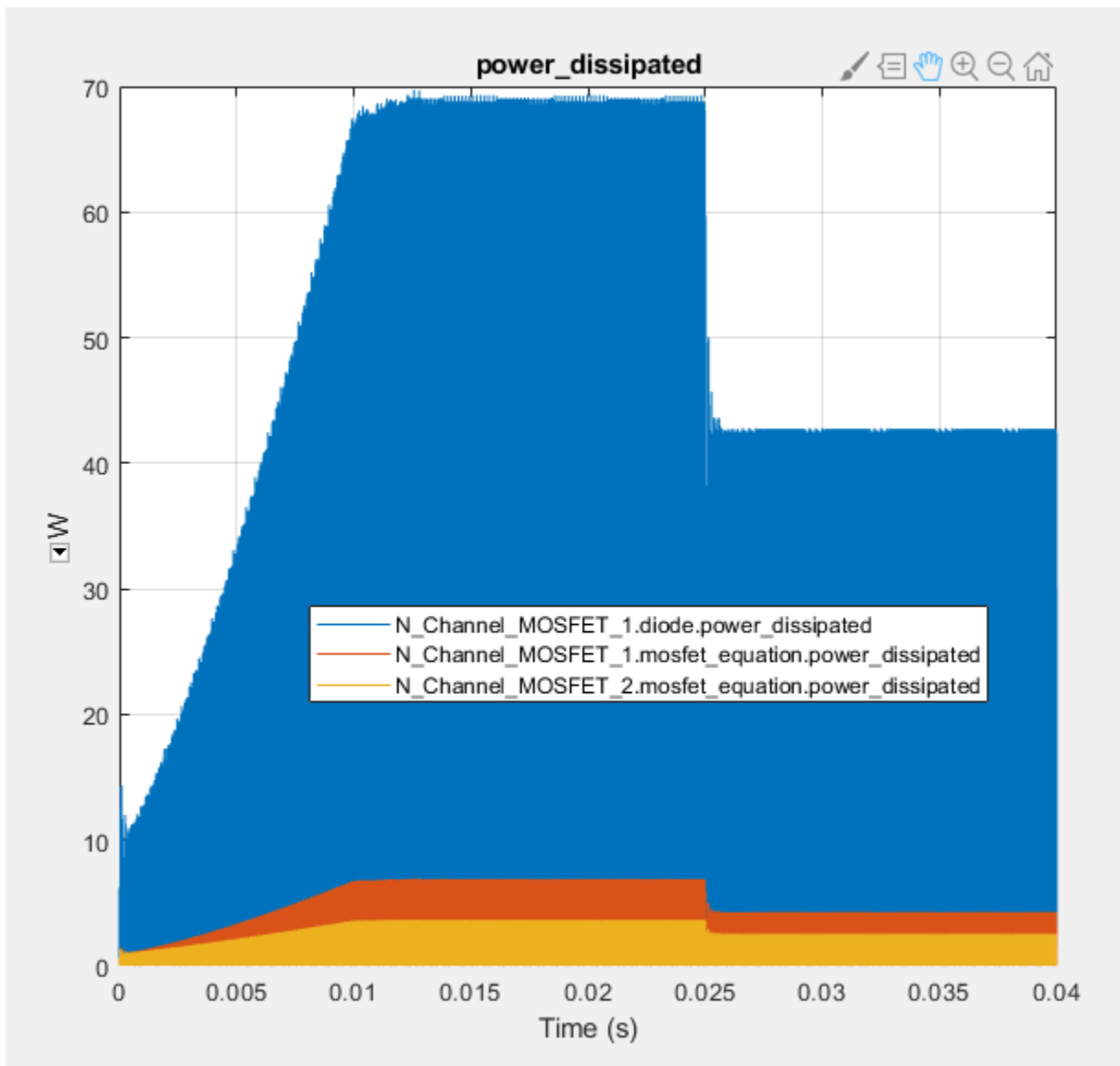
3. Run the simulation, create a simulation log variable, and open the simlog in the Simscape Results Explorer using the `sscexplore` function.

```
sim(model)
sscexplore(simlog_ee_push_pull_converter_ccm)
```

4. View the power loss data for the two N-Channel MOSFET blocks, expand these nodes and CTRL + click the `power_dissipated` nodes:

- N\_Channel\_MOSFET\_1 > diode > power\_dissipated
- N\_Channel\_MOSFET\_1 > mosfet\_equation > power\_dissipated
- N\_Channel\_MOSFET\_2 > mosfet\_equation > power\_dissipated





The N-Channel MOSFET 2 block has only one `power_dissipated` variable. The N-Channel MOSFET 1 block has one `power_dissipated` variable for each of the two components (MOSFET and diode) that the block contains.

5. Calculate power losses for both components of the N-Channel MOSFET 1 block and display the results in a table

```
tabulatedLosses = ee_getPowerLossSummary(simlog_ee_push_pull_converter_ccm.N_Channel_MOSFET_1)
```

```
tabulatedLosses =
```

```
1x2 table
```

LoggingNode	Power
-------------	-------

```
{'N_Channel_MOSFET_1'}    2.6077
```

The table shows the combined dissipated power losses for both the diode and the MOSFET components of the N-Channel MOSFET 1 block, averaged over the total simulation time.

6. Calculate power losses for only the diode component of the NChannel MOSFET 1 block and display the results in a table.

```
tabulatedLosses = ee_getPowerLossSummary(simlog_ee_push_pull_converter_ccm.N_Channel_MOSFET_1.di
```

```
tabulatedLosses =
```

```
1x2 table
```

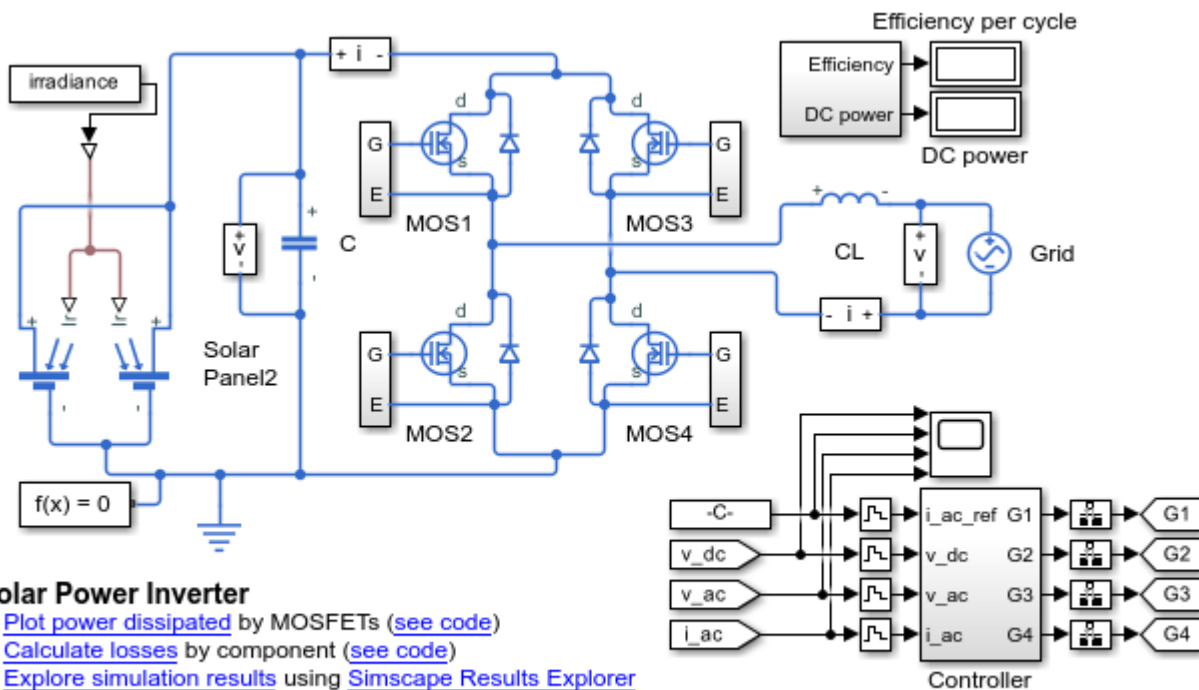
LoggingNode	Power
{'diode'}	2.3671

The table shows dissipated power losses only for the diode component of the block, averaged over the total simulation time.

### **Calculate Power Losses for One Block**

Open the Solar Power Inverter example model.

```
ee_solar_inverter
```



### Solar Power Inverter

1. [Plot power dissipated](#) by MOSFETs ([see code](#))
2. [Calculate losses](#) by component ([see code](#))
3. [Explore simulation results](#) using [Simscape Results Explorer](#)
4. [Learn more](#) about this example

Copyright 2008-2022 The MathWorks, Inc.

This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_solar_inverter` in your current workspace.

```
sim('ee_solar_inverter');
```

Calculate power losses for the MOS1 block.

```
mosfetLosses = ee_getPowerLossSummary(simlog_ee_solar_inverter.MOS1)
```

```
mosfetLosses =
```

```
1x2 table
```

LoggingNode	Power
{'MOS1'}	13.524

The table shows dissipated power losses for the MOS1 block, averaged over the whole simulation time.

Use the `sscexplore` function to further explore the power loss data for the MOSFET block and in the results explorer, expand the **mos > power\_dissipated nodes**.

```
sscexplore(simlog_ee_solar_inverter.MOS1, 'mos.power_dissipated')
```

The block has several `power_dissipated` logging nodes: under `drain_resistor`, under `gate_resistor`, under `mos`, and under `source_resistor`. The **Power** value calculated by the `ee_getPowerLossSummary` function is a sum of all these losses, averaged over the simulation time.

## Input Arguments

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog.Cell1.MOS1`

### **startTime** — Start of the time interval for averaging dissipated power losses

real number

Start of the time interval for averaging dissipated power losses, specified as a real number, in seconds. `startTime` must be greater than or equal to the node **Start time** and less than `endTime`.

Data Types: `double`

### **endTime** — End of the time interval for averaging dissipated power losses

real number

End of the time interval for averaging dissipated power losses, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the node **Stop time**.

Data Types: `double`

## Output Arguments

### **LossesTable** — Dissipated power losses and switching losses for each block

table

Dissipated power losses and switching losses for each block, returned as a table. The first column lists logging nodes for all blocks that have at least one `power_dissipated` variable. The second column lists the corresponding losses in watts. The third column lists the switching losses of each block, in watts. Switching losses comprise losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and with the energy dissipated during a reverse recovery event.

12×3 [table](#)

LoggingNode	Power	SwitchingLosses
{'ee_igbt_losses.Buck_circuit_1.Load' }	7455.90770873158	0
{'ee_igbt_losses.Buck_circuit_2.Load' }	7455.90478815434	0
{'ee_igbt_losses.IGBT_C' }	16.5557040366311	106.740946512331
{'ee_igbt_losses.IGBT_F' }	16.5542585657951	106.824444654678
{'ee_igbt_losses.Buck_circuit_1.Diode' }	11.9943921315891	0
{'ee_igbt_losses.Buck_circuit_2.Diode' }	11.9943893330764	0
{'ee_igbt_losses.Buck_circuit_1.Capacitor' }	0	0
{'ee_igbt_losses.Buck_circuit_1.Inductor' }	0	0
{'ee_igbt_losses.Buck_circuit_2.Capacitor' }	0	0
{'ee_igbt_losses.Buck_circuit_2.Inductor' }	0	0
{'ee_igbt_losses.Rgate_C' }	0	0
{'ee_igbt_losses.Rgate_F' }	0	0

## Version History

Introduced in R2015a

### See Also

[ee\\_getEfficiency](#) | [ee\\_getPowerLossTimeSeries](#) | [sscexplore](#)

### Topics

“Optimize an Inverter Liquid Cooling System”

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## ee\_getPowerLossTimeSeries

Calculate dissipated power losses and switching losses, and return time series data

### Syntax

```
lossesCell = ee_getPowerLossTimeSeries(node)
lossesCell = ee_getPowerLossTimeSeries(node, startTime, endTime)
lossesCell = ee_getPowerLossTimeSeries(node, startTime, ...
endTime, intervalWidth)
[lossesCell, switchingLosses] = ee_getPowerLossTimeSeries(node)
```

### Description

`lossesCell = ee_getPowerLossTimeSeries(node)` calculates dissipated power losses for blocks in a model, based on logged simulation data, and returns the time series data for each block.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file.

The `ee_getPowerLossTimeSeries` function calculates dissipated power losses for each block that has a `power_dissipated` variable. All blocks in the Semiconductor Devices library, as well as some other blocks, have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the block. Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses and provides the power loss value for all of the blocks as functions of time.

---

**Note** The `power_dissipated` internal variable does not report dynamic losses incurred from semiconductor switching or magnetic hysteresis.

Three different variables, `lastTurnOnLoss`, `lastTurnOffLoss`, and `lastReverseRecoveryLoss` report the switching losses.

---

Switching losses are losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and also with the energy dissipated during a reverse recovery event. They are frequency dependent. The `ee_getPowerLossTimeSeries` function returns the switching losses at each switching event and expresses them in joules.

These blocks in the **Semiconductors & Converters** library support the calculation of switching losses and reverse recovery losses, when applicable:

- Diode
- N-Channel IGBT
- GTO
- IGBT (Ideal, Switching)

- MOSFET (Ideal, Switching)
- Thyristor (Piecewise Linear)
- Half-Bridge (Ideal, Switching)

If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that dissipate power (that is, contain at least one `power_dissipated` variable). If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

`lossesCell = ee_getPowerLossTimeSeries(node, startTime, endTime)` calculates dissipated power losses and returns the time series data for time steps from `startTime` to `endTime`. If `startTime` is equal to `endTime`, the interval is effectively zero and the function returns the instantaneous power for the time step that occurs at that moment. If you omit these two input arguments, the function returns data over the whole simulation time.

`lossesCell = ee_getPowerLossTimeSeries(node, startTime, ..., endTime, intervalWidth)` calculates dissipated power losses and returns the time series data for time steps from `startTime` to `endTime`, averaged over the time `intervalWidth`. If you omit the `intervalWidth`, or set it to 0, the function returns the instantaneous data, without averaging. If you omit all three optional arguments, the function returns the instantaneous data over the whole simulation time.

`[lossesCell, switchingLosses] = ee_getPowerLossTimeSeries(node)` calculates dissipated power losses for blocks in a model, based on logged simulation data, and returns the time series data, `lossesCell`, for each block and a cell array, `switchingLosses`, with the switching losses of each device.

If there are no switching losses appear, the `switchingLosses` output is an empty cell array.

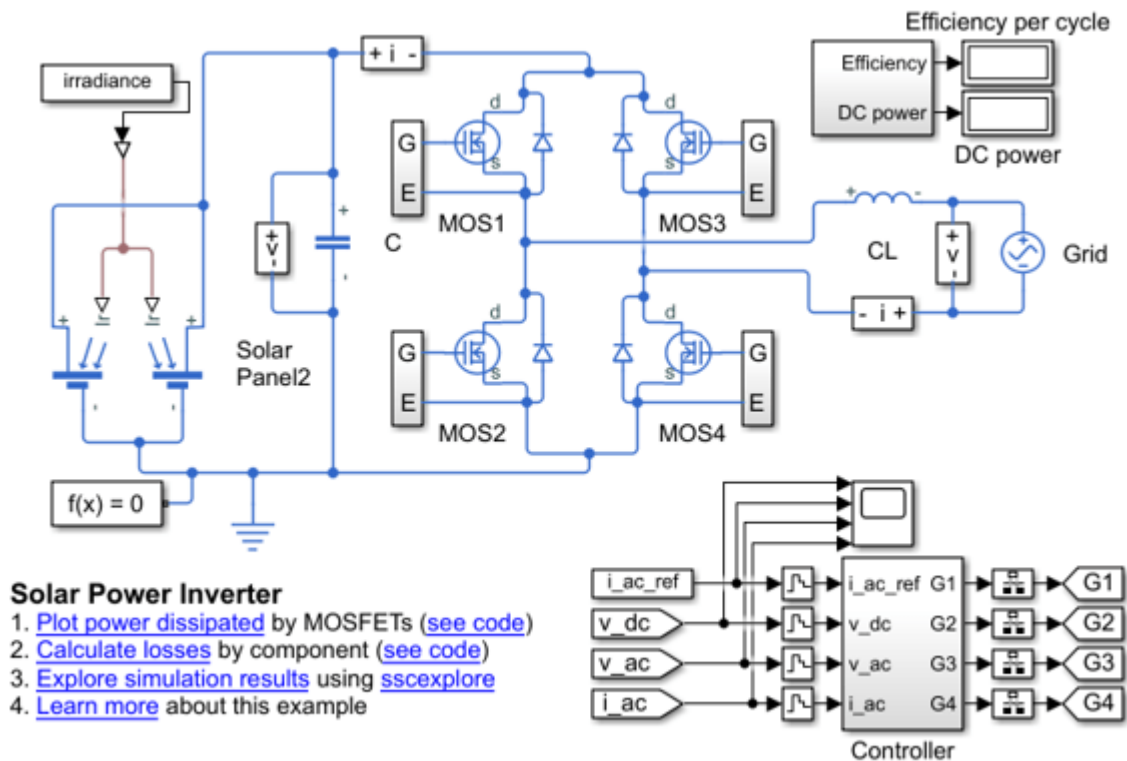
## Examples

### Calculate Dissipated Power Losses for the Entire Simulation Time

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in the entire simulation time using the `ee_getPowerLossTimeSeries` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

Calculate dissipated power losses and return the time series data in cell array.

```
lossesCell = ee_getPowerLossTimeSeries(simlog_ee_solar_inverter)
```

```
lossesCell =
```

8x2 cell array

```
{'ee_solar_inverter.Diode1'} {77180x3 double}
{'ee_solar_inverter.MOS1' } {77180x3 double}
{'ee_solar_inverter.MOS2' } {77180x3 double}
{'ee_solar_inverter.MOS3' } {77180x3 double}
{'ee_solar_inverter.MOS4' } {77180x3 double}
{'ee_solar_inverter.Diode2'} {77180x3 double}
{'ee_solar_inverter.Diode3'} {77180x3 double}
{'ee_solar_inverter.Diode4'} {77180x3 double}
```

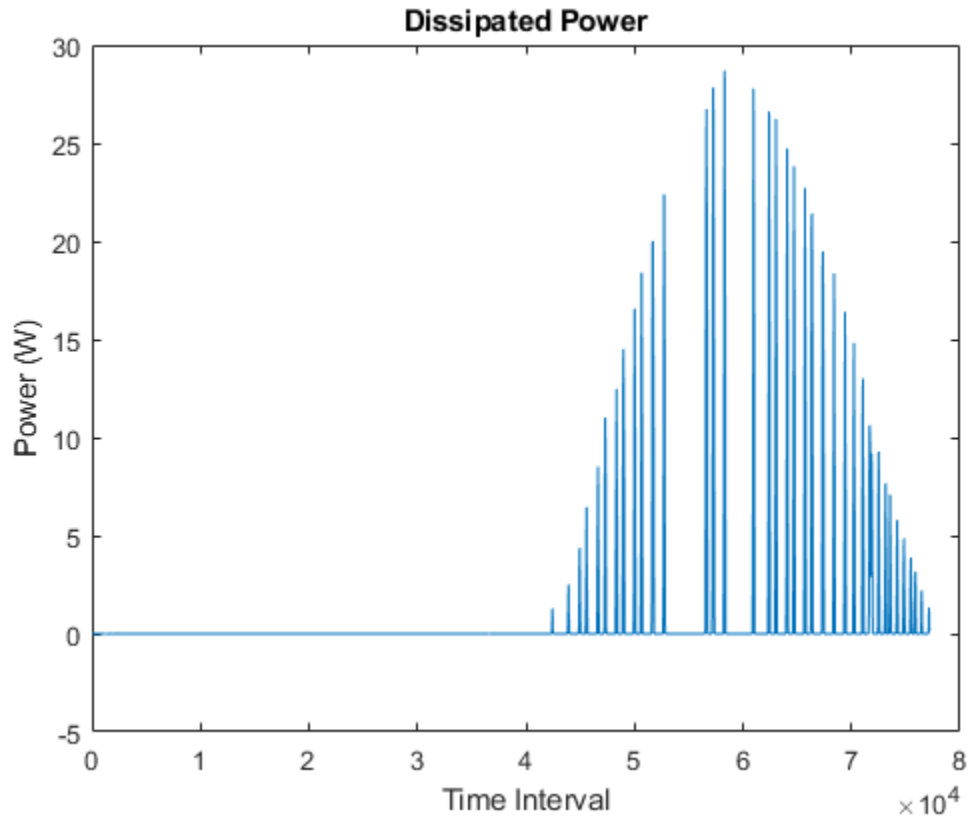
View the time series data. From the workspace, open the `lossesCell` cell array, then open the 77180x3 double numeric array for the `ee_solar_inverter.Diode1`.



The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```

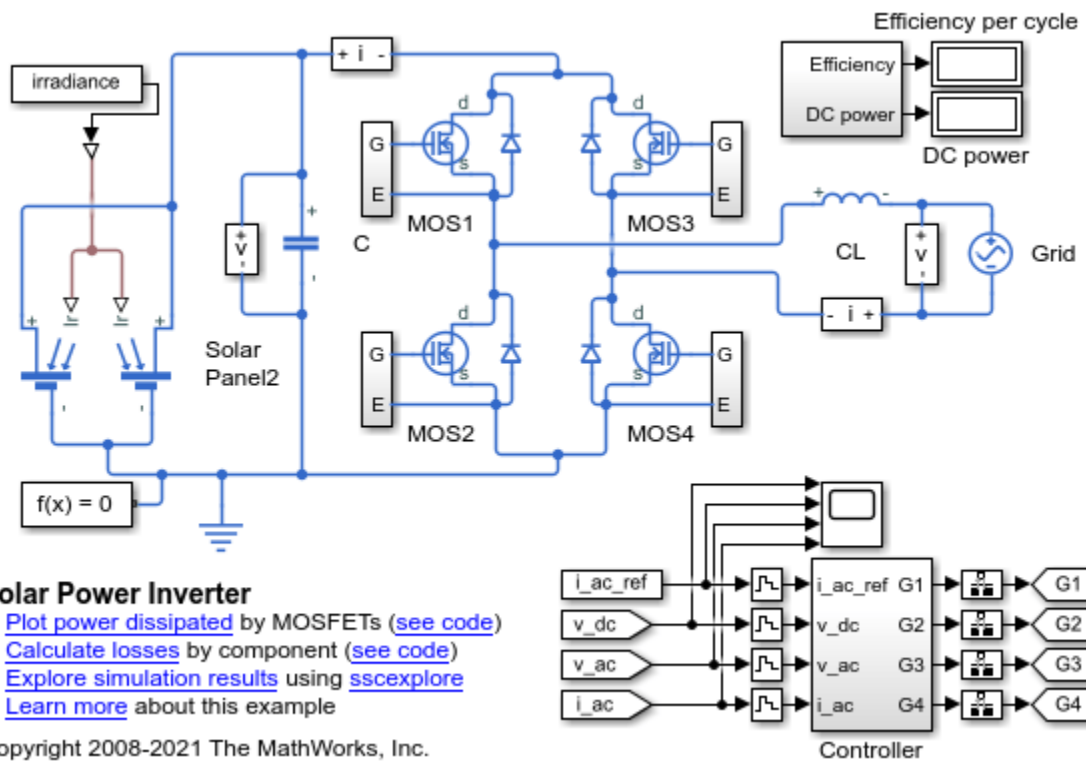


### Calculate Dissipated Power Losses for a Specific Time Period

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in a specific time period using the `ee_getPowerLossTimeSeries` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

The model simulation time (`t`) is 1/60 seconds. Calculate dissipated power losses and return the time series data in cell array for the second half of the simulation cycle, when `t` is between 1/120 and 1/60 seconds.

```
lossesCell = ee_getPowerLossTimeSeries(simlog_ee_solar_inverter,1/120,1/60)
```

```
lossesCell =
```

9×2 cell array

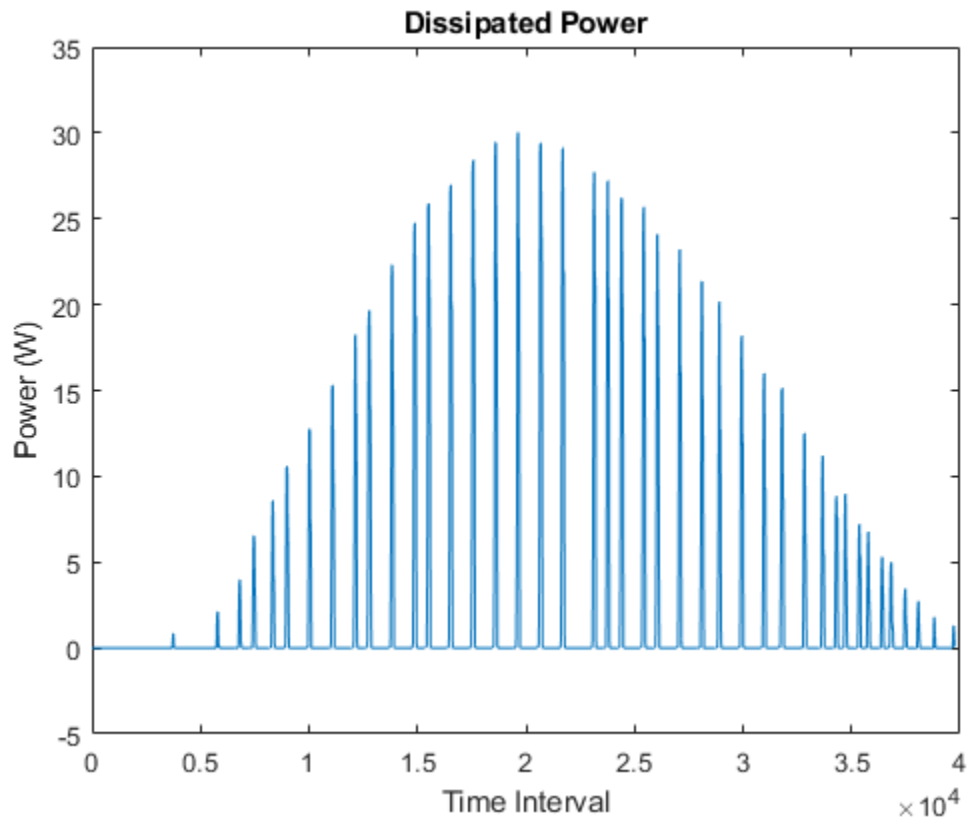
```
{'ee_solar_inverter.Diode1'} {39748×3 double}
{'ee_solar_inverter.MOS1' } {39748×3 double}
{'ee_solar_inverter.CL' } {39748×3 double}
{'ee_solar_inverter.MOS2' } {39748×3 double}
{'ee_solar_inverter.MOS3' } {39748×3 double}
{'ee_solar_inverter.MOS4' } {39748×3 double}
{'ee_solar_inverter.Diode2'} {39748×3 double}
{'ee_solar_inverter.Diode3'} {39748×3 double}
{'ee_solar_inverter.Diode4'} {39748×3 double}
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the 39150x3 double numeric array for the `ee_solar_inverter.Diode1`.

The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```

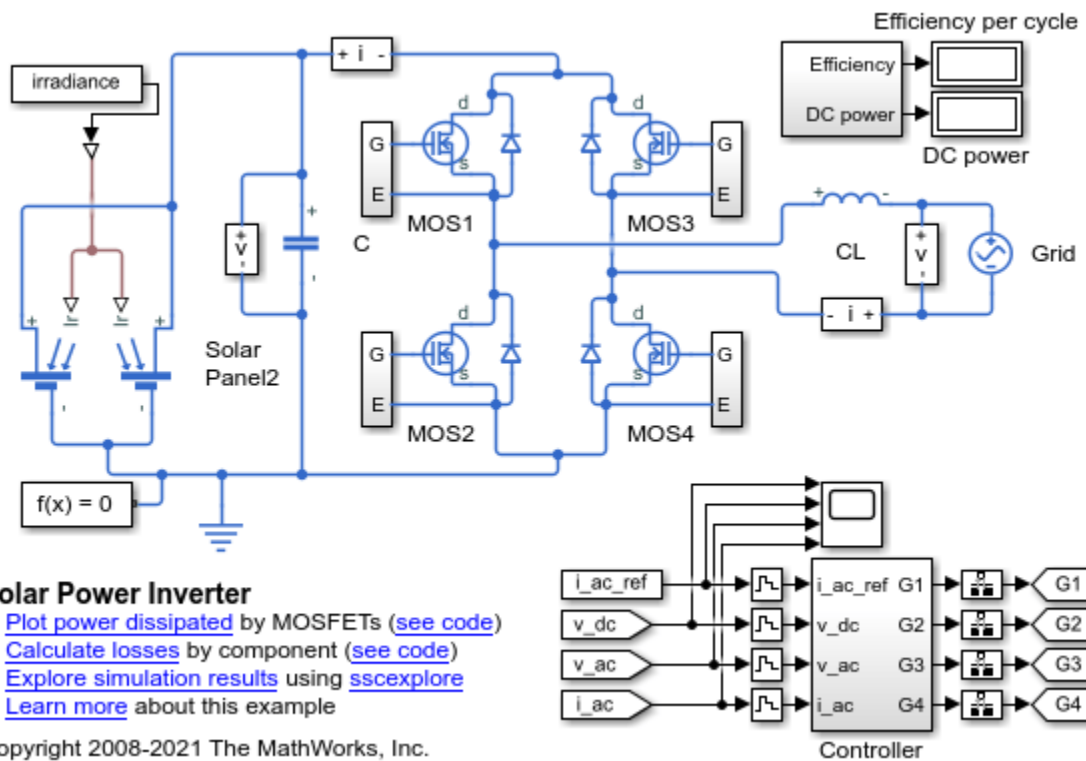


### Calculate Dissipated Power Losses Using Specific Interval Widths

This example shows how to calculate losses based on the power dissipated and return the time series data for a specific time period with averaging applied over intervals of a specified width.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

The model simulation time (`t`) is 1/60 seconds. Calculate the average dissipated power losses for 1.1e-4 s intervals and return the time series data in cell array for the period when simulation time, `t`, is 0.008-0.017 seconds.

```
lossesCell = ee_getPowerLossTimeSeries(simlog_ee_solar_inverter,0.008,0.016,1.1e-4)
```

```
lossesCell =
```

9×2 cell array

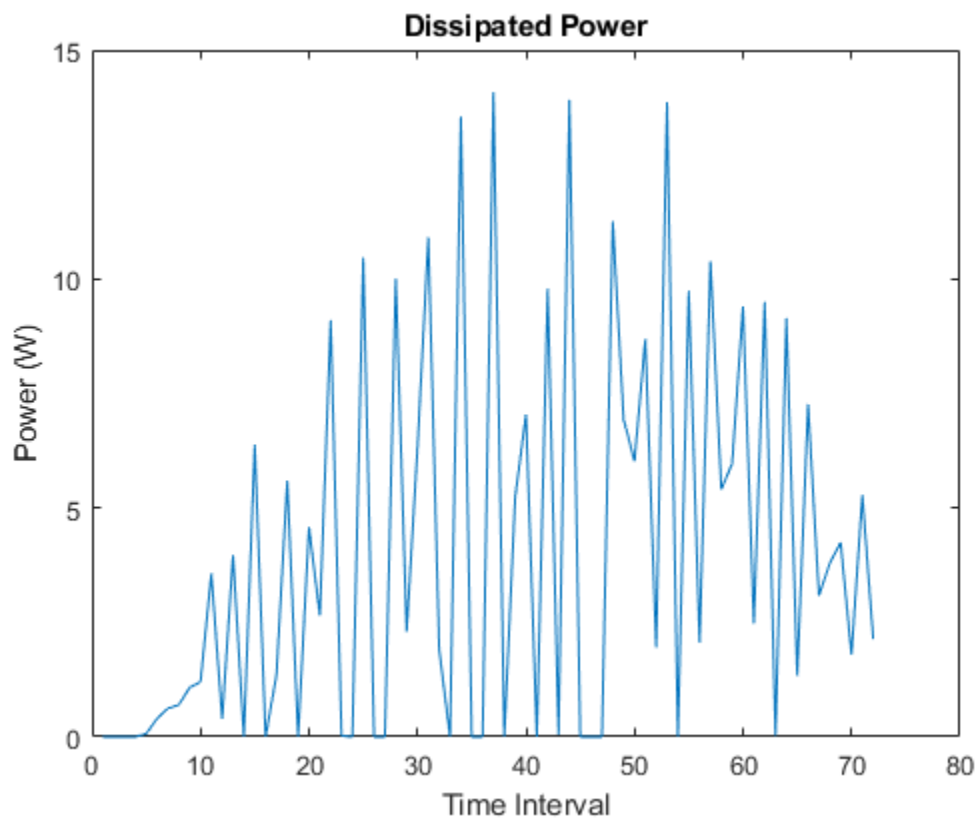
```
{'ee_solar_inverter.Diode1'} {72×3 double}
{'ee_solar_inverter.MOS1' } {72×3 double}
{'ee_solar_inverter.CL' } {72×3 double}
{'ee_solar_inverter.MOS2' } {72×3 double}
{'ee_solar_inverter.MOS3' } {72×3 double}
{'ee_solar_inverter.MOS4' } {72×3 double}
{'ee_solar_inverter.Diode2'} {72×3 double}
{'ee_solar_inverter.Diode3'} {72×3 double}
{'ee_solar_inverter.Diode4'} {72×3 double}
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the double numeric array for the `ee_solar_inverter.Diode1`

The first two columns contain the interval start and end time. The third column contains the power loss data. In this case, to use averaging intervals that are equal in width to  $1.1 \times 10^{-4}$  seconds, the function adjusts the start time for the first interval from the specified value of 0.008 seconds to a value of 0.0081 seconds. There are 72 intervals of  $1.1 \times 10^{-4}$  seconds.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```



## Input Arguments

**node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If `node` is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If `node` is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog.Cell1.MOS1`

**startTime** — Start of the time interval for calculating the data

0 (default) | real number

Start of the time interval for calculating the power loss time series, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than `endTime`.

Data Types: double

**endTime** — End of the time interval for calculating the data

simulation stop time (default) | real number

End of the time interval for calculating the power loss time series, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

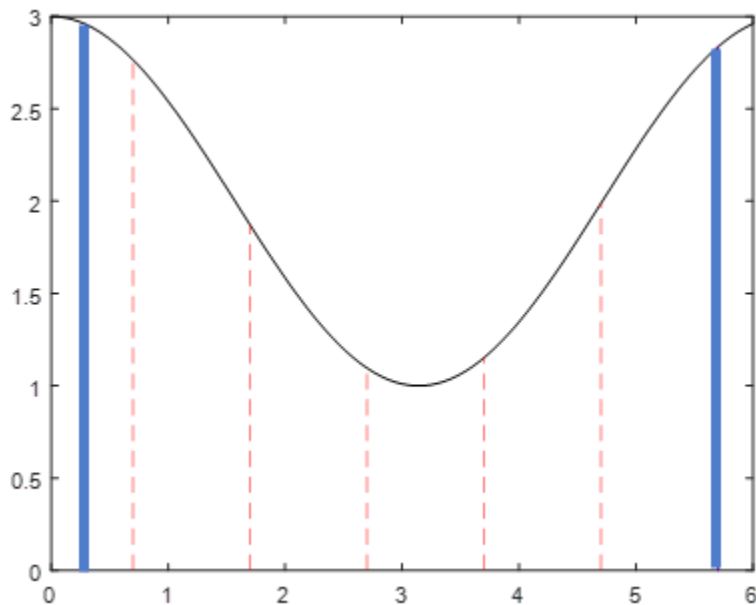
Data Types: double

**intervalWidth** — Size of the time interval for calculating the average power dissipation

0 (default) | real number

Size of the time interval for calculating the average power dissipation, specified as a real number, in seconds. If specified, the function returns data for time steps from `startTime` to `endTime`, averaged over the `intervalWidth`. If you omit the `intervalWidth` argument, or set it to 0, the function returns the instantaneous data, without averaging. If all the optional arguments are omitted, the function returns the instantaneous data over the whole simulation time.

If the time between the specified `startTime` and `endTime` is not an integer multiple of `intervalWidth`, the function adjusts the start time. The figure shows how the function adjusts the start time to ensure that width of each time interval that the dissipated power is averaged over is equal to the specified `intervalWidth`.



The black line is an example of the instantaneous *power\_dissipated* variables summed over all elements in an individual block. The simulation runs for 6 seconds. The `startTime` and `endTime` are indicated by the solid blue lines. The `intervalWidth` is set to 1 second. There are five intervals as indicated by the red dashed lines. The right-most edge of the last interval coincides with `endTime`. The left-most edge of the first interval is always greater than or equal to `startTime`. The edge is equal to `startTime` only if  $(\text{endTime} - \text{startTime}) / \text{intervalWidth}$  is an integer. The output in this case consists of five values for the averaged power dissipation, one point for each time period. The function outputs the actual start and stop times in the tabulated output data.

Example: `1.1e-3`

Data Types: `double`

## Output Arguments

### **lossesCell** — Time series of the dissipated power losses for each block

cell array

Cell array that contains the names of the blocks in the nodes that contain *power\_dissipated* variables and, for each block, a three-column array:

- Column one contains the interval start time.
- Column two contains the interval end time.
- Column three contains the dissipated power for the time interval.

If the interval width is 0 seconds, that is, the start time is equal to the end time, then the dissipated power is the instantaneous power loss. If the interval is greater than 0 seconds, the dissipated power is the average power loss for the time of the interval.

### **switchingLosses** — Time series of the switching power losses for each block

cell array

Cell array that contains the time of each switch in the first column, and the corresponding energy dissipated in that switching event, in Joules, in the second column. Switching losses comprise losses associated with the transition of the semiconductor switch from its on-state to its off-state and viceversa, and with the energy dissipated during a reverse recovery event.

If there are no switching losses appear, the cell array is empty.

## **Version History**

**Introduced in R2017a**

### **See Also**

`ee_getEfficiency` | `ee_getPowerLossSummary` | `sscexplore`

### **Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”



# ee\_importDeviceParameters

Import device parameters from XML file to Simscape Electrical IGBT (Ideal, Switching) and Diode blocks

## Syntax

```
ee_importDeviceParameters(file,format,blockPath)
ee_importDeviceParameters(file,format,blockPath,'Verbose',true)
ee_importDeviceParameters(file,format,blockPath,'Verbose',false)
```

## Description

`ee_importDeviceParameters(file,format,blockPath)` extracts the parameters from an XML file, `file`, and imports them to the Simscape block at the specified `blockPath`, based on the parameterization format of the file.

The XML file must be on the MATLAB path and must use a parameterization format supported by Hitachi. For examples of Hitachi format XML files for IGBT and diode devices, see the Insulated Gate Bipolar Transistor (IGBT) and diode modules with SPT, SPT+, SPT++ and TSPT+ chips page on the Hitachi website.

This function currently only supports IGBT(Ideal, Switching) and Diode blocks.

`ee_importDeviceParameters(file,format,blockPath,'Verbose',true)` also enables warnings about unused fields in the XML file.

`ee_importDeviceParameters(file,format,blockPath,'Verbose',false)` also disables warnings about unused fields in the XML file.

## Examples

### Import Parameters to an IGBT(Ideal, Switching) Block

Edit the `ee_import_igbt_device_parameters_data.xml` file to inspect the content.

```
edit ee_import_igbt_device_parameters_data.xml
```

Choose the block you want to import the parameters into. Open a model that contains an IGBT(Ideal, Switching) block.

```
open_system('ee_import_igbt_device_parameters')
```

Click the IGBT(Ideal, Switching) block, then call the `ee_importDeviceParameters` function.

```
ee_importDeviceParameters('ee_import_igbt_device_parameters_data.xml', 'hitachi', gcb)
```

## Input Arguments

**file** — Name of XML file

case-sensitive string

File name of the XML file that you want to import and extract the parameters from, specified as a case-sensitive string.

**format — Parameterization format**

hitachi

Parameterization format, specified as hitachi.

**blockPath — Simscape block path or handle**

character vector | handle

Path or handle of the IGBT (Ideal, Switching) or Diode block that you want to import the parameters into, specified as a character vector or block handle.

## Version History

**Introduced in R2021b**

### See Also

IGBT (Ideal, Switching) | Diode

### Topics

“Import IGBT Parameters from Hitachi”

“Optimize an Inverter Liquid Cooling System”

# ee\_plotHarmonics

Plot percentage of fundamental magnitude versus harmonic order

## Syntax

```
ee_plotHarmonics(loggingNode)
ee_plotHarmonics(loggingNode,valueIdx)
ee_plotHarmonics(loggingNode,valueIdx,tOfInterest)
ee_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest)
ee_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest)
ee_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest,nHarmonic)
```

## Description

`ee_plotHarmonics(loggingNode)` plots a bar chart of percentage of fundamental magnitude versus harmonic order of the `simscape.logging.Node` of an AC or periodic variable. The title of the bar chart includes the fundamental frequency, fundamental peak value, and total harmonic distortion (THD) percentage.

You enter the input arguments in a specific order. The Simscape logging node input argument is required. All other input arguments are optional and have default values. If you are specifying a value for a subsequent optional input argument, enter `[]` to use the default value for an optional input argument.

The `ee_plotHarmonics` function uses the `ee_getHarmonics` function to:

- Find the points in the  $i^{\text{th}}$  signal (`valueIdx`) where the Simscape log crosses a threshold (`offsetOfInterest`).
- Use the crossing points to find the required number of periods (`nPeriodOfInterest`) preceding the specified time (`tOfInterest`).
- Calculate the harmonic magnitudes, up to and including the required number of harmonics (`nHarmonic`).
- Input the down-selected data to the Goertzel algorithm, which calculates the harmonic magnitudes up to and including the required number of harmonics (`nHarmonic`).

---

**Note** The `ee_getHarmonics` function uses threshold crossing points to determine the fundamental frequency of the data. If your input data is noisy or crosses the threshold more frequently than half of the fundamental period, filter it before you use the `ee_plotHarmonics` function to plot it.

---

The `ee_plotHarmonics` function then inputs the harmonic orders and harmonic magnitudes to the `ee_calculateThdPercent` function to calculate the THD.

`ee_plotHarmonics(loggingNode,valueIdx)` uses the index into value data.

`ee_plotHarmonics(loggingNode,valueIdx,tOfInterest)` uses the simulation time.

`ee_plotHarmonics(loggingNode,valueIdx,t0fInterest,nPeriodOfInterest)` uses the number of periods of fundamental frequency.

`ee_plotHarmonics(loggingNode,valueIdx,t0fInterest,nPeriodOfInterest,...,offsetOfInterest)` uses the DC offset.

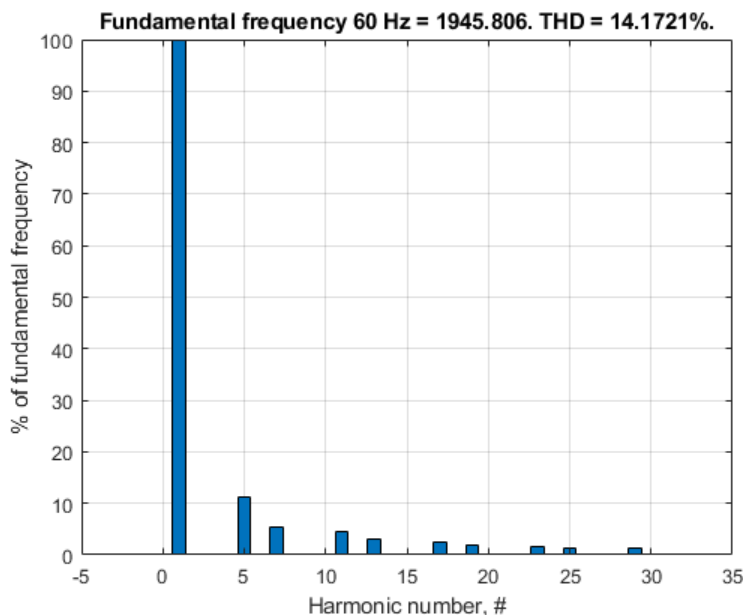
`ee_plotHarmonics(loggingNode,valueIdx,t0fInterest,nPeriodOfInterest,...,offsetOfInterest,nHarmonic)` uses the number of harmonics.

## Examples

### Plot Using Default Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the default signal, which is the first, or a-phase, signal at the final simulation time. The function uses the default values of 12 for the number of periods of the signal, 0V for the signal bias, and 30 for the number of harmonics.

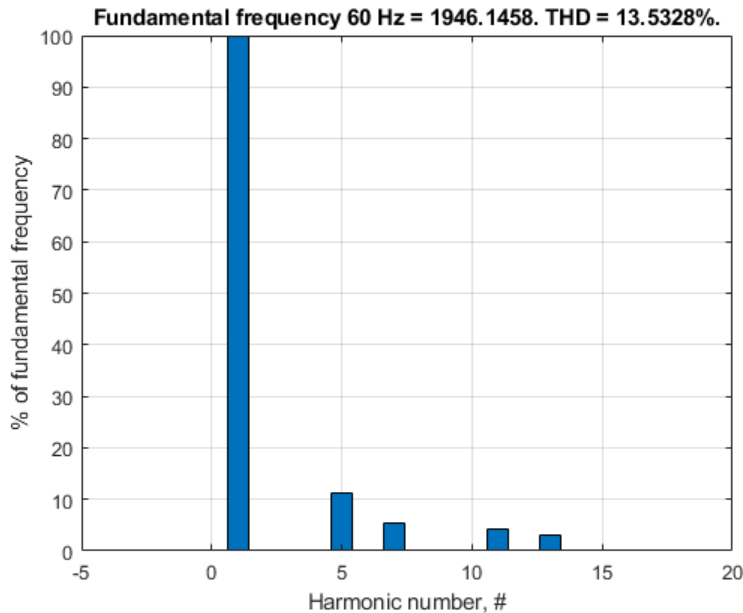
```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
ee_plotHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I)
```



### Plot Using Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the second, or b-phase, signal at a simulation time of 0.5 s. The function uses 10 periods of the signal, assuming a bias of 1V. The function analyzes 15 harmonics.

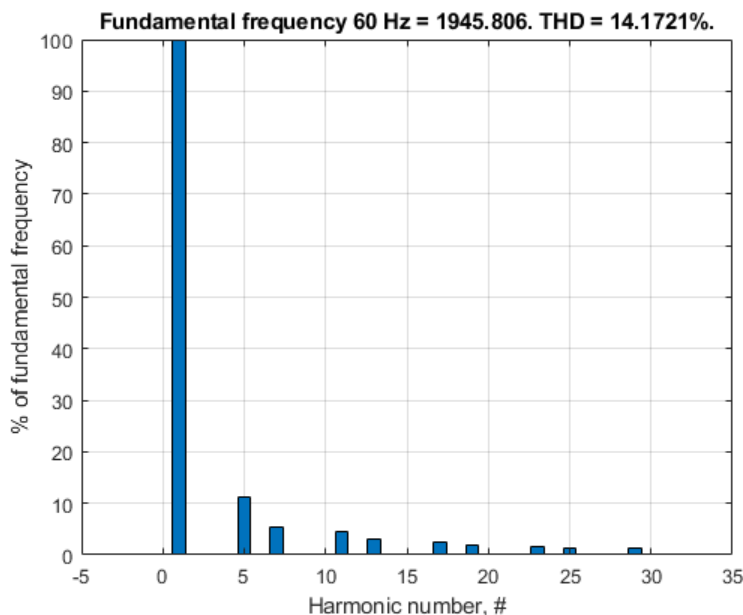
```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
ee_plotHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I,2,0.5,10,1,15)
```



### Plot Using Default and Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I`, which contains data from a three-phase current. The function analyzes the first, or a-phase, signal at a simulation time of 0.5 s. The function uses 12 periods of the signal, assuming a bias of 1V. The function analyzes the default number, 30, of harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
ee_plotHarmonics(simlog_ee_harmonics_rectifier.Sensing_current.Current_Sensor.I,[],0.5,[],1)
```



## Input Arguments

### loggingNode — Simscape logging node

1-by-1 `simscape.logging.Node`

Simscape logging node, specified as a 1-by-1 `simscape.logging.Node`. You create a `simscape.logging.Node` by running a simulation with Simscape logging enabled. To learn how to enable data logging, see “Enable Data Logging for the Whole Model”.

Example: `simlog.Load.V`

The Simscape logging node `simlog.Load.V` contains data from a three-phase voltage.

### valueIdx — Index into value data

1 (default) | scalar

Index into value data, specified as a scalar. Specifies the  $i^{\text{th}}$  variable of interest in the Simscape log.

Example: 2

Specify the b-phase, which is the second signal from a three-phase voltage.

Example: []

Use [] to specify the default value of 1. The a-phase, which is the first signal from a three-phase voltage, is the default signal of interest.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### t0fInterest — Simulation time

final time in Simscape log (default) | scalar

Simulation time of interest for harmonic analysis, specified as a scalar.

Example: 2.3

Specify a 2.3s simulation time.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### nPeriodOfInterest — Number of periods

12 (default) | scalar

Number of periods of fundamental frequency to be included in harmonic analysis, specified as a scalar.

Example: 10

Specify 10 periods of the signal.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### offset0fInterest — DC offset

0 (default) | scalar

DC offset in the input signal, specified as a scalar. The function uses this value to find the periods of interest.

Example: 1

Specify a bias of 1V for the signal.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### **nHarmonic — Number of harmonics**

30 (default) | scalar

Number of harmonics to include in analysis, specified as a scalar.

Example: 15

Specify that the number of harmonics to be analyzed is 15.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

## **Version History**

**Introduced in R2014a**

### **See Also**

#### **Apps**

**Harmonic Analyzer**

#### **Blocks**

**Spectrum Analyzer**

#### **Functions**

`ee_calculateThdPercent` | `ee_getHarmonics`

#### **Objects**

`simscape.logging.Node`

#### **Topics**

“Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block”

“Choose a Simscape Electrical Function for an Offline Harmonic Analysis”

“Data Logging”

“Harmonic Analysis of a Three-Phase Rectifier”

## ee.spice.semiconductorSubcircuit2lookup

Generate lookup table data for semiconductors from SPICE subcircuit

### Syntax

```
lookuptable = ee.spice.semiconductorSubcircuit2lookup(subcircuitFile,
subcircuitName)
lookuptable = ee.spice.semiconductorSubcircuit2lookup(subcircuitFile,
subcircuitName,Name=Value)
```

### Description

lookuptable = ee.spice.semiconductorSubcircuit2lookup(subcircuitFile, subcircuitName) returns the lookup table data for the SPICE subcircuit file at the path subcircuitFile and with the file name subcircuitName. Use this function to create lookup table data that characterizes an N-Channel MOSFET, P-Channel MOSFET, or N-Channel IGBT block.

This function supports SIMetrix 8.4.

lookuptable = ee.spice.semiconductorSubcircuit2lookup(subcircuitFile, subcircuitName,Name=Value) creates a SPICE subcircuit file with the specified name-value arguments.

### Examples

#### Generate Current-Voltage Characteristics from Output Characteristics

Generate the lookup table data for the current-voltage characteristics from the output characteristics of a SPICE subcircuit.

First define the path of the subcircuit, the name of the subcircuit, and the path of the SPICE executable file.

```
subcircuitFile = [matlabroot '\toolbox\physmod\elec\eedemos\applications\devicecharacteristics\I
subcircuitName = "IAUC100N04S6L014";
SPICEPath = "C:\Program Files\SIMetrix850\bin64\Sim.exe";
```

Then generate the lookup table data by calling the semiconductorSubcircuit2lookup function.

```
lookuptable = ...
ee.spice.semiconductorSubcircuit2lookup(subcircuitFile, ...
subcircuitName, 'SPICEPath', SPICEPath, 'terminals', [1,2,3, 0, 5], ...
'flagIdsVgs', 0, 'flagIdsVds', 1, 'T', [27 50 150]);
```

### Input Arguments

#### subcircuitFile — File path

character vector



File path of the SPICE subcircuit file from which to generate the data, specified as a character vector.

### **subcircuitName — File name**

character vector

File path of the SPICE subcircuit file from which to generate the data, specified as a character vector.

### **Name-Value Pair Arguments**

Specify optional pairs of arguments as `Name1=Value1, ..., NameN=ValueN`, where `Name` is the argument name and `Value` is the corresponding value. Name-value arguments must appear after other arguments, but the order of the pairs does not matter.

Example: `'terminals'=[1,2,3,0,5]`

### **SPICETool — Name of SPICE simulation engine**

'SIMatrix' (default)

Name of the SPICE simulation engine executable file, specified as `SIMatrix`.

### **SPICEPath — Path of SPICE simulation engine executable file**

character vector | string

Path of the SPICE simulation engine executable file, specified as a character vector or a string.

### **outputPath — Path of generated outputs**

character vector | string

Path of the generated SPICE netlists and the simulation output files, specified as a character vector or a string. If you do not specify this argument, the function uses a temporary directory to store the files and removes the temporary directory when the function completes.

### **terminals — Vector of terminal orders**

vector of scalars

Vector of terminal orders in the SPICE subcircuit, specified as a vector of scalars. The values of this vector define the ports of the semiconductor device that each node of the SPICE subcircuit connects to:

- 0 — No connection
- 1 — **D** for MOSFET or **C** for IGBT
- 2 — **G** for MOSFET and IGBT
- 3 — **S** for MOSFET or **E** for IGBT
- 4 — **B** for MOSFET and IGBT
- 5 — **Tcase** for MOSFET and IGBT

The size of this vector must match the number of nodes specified in the subcircuit.

For example, if your SPICE subcircuit is `.SUBCKT IAUC100N04S6L014 drain gate source Tj Tcase`, then define this vector as `[1,2,3,0,5]`.

### **flagIdsVgs — Whether to obtain characteristics from transfer characteristics**

true or 1 (default) | false or 0

Whether to obtain current-voltage characteristics from the transfer characteristics, specified as a numeric or logical 1 (true) or 0 (false).

If you also set 'flagIdsVds' to 1, the function obtains the current-voltage characteristics only from transfer characteristics.

**flagIdsVds — Whether to obtain characteristics from output characteristics**

true or 1 (default) | false or 0

Whether to obtain current-voltage characteristics from the output characteristics, specified as a numeric or logical 1 (true) or 0 (false).

If you also set 'flagIdsVgs' to 1, the function obtains the current-voltage characteristics only from transfer characteristics.

**flagCapacitance — Whether to return capacitance characteristics**

true or 1 (default) | false or 0

Whether to return capacitance characteristics, specified as a numeric or logical 1 (true) or 0 (false).

**flagDiodeIv — Whether to return diode characteristics**

true or 1 (default) | false or 0

Whether to return diode characteristics, specified as a numeric or logical 1 (true) or 0 (false).

**flagTailTransient — Whether to return IGBT turn-off tail current transient**

true or 1 (default) | false or 0

Whether to return IGBT turn-off tail current transient, as a numeric or logical 1 (true) or 0 (false).

**VdsRangeIdsVgs — Vgs or Vge transfer characteristics range**

[0, 20] (default) | row vector of two scalars

Range for the transfer characteristics of the MOSFET gate-source voltage, Vgs, or the IGBT gate-emitter voltage, Vge, in volts, specified as a row vector of two scalars.

**VdsStepsIdsVgs — Vds or Vce transfer characteristics steps**

[1:0.2:5] (default) | vector of scalars

Steps for the transfer characteristics of the MOSFET drain-source voltage, Vds, or the IGBT collector-emitter voltage, Vce, in volts, specified as a vector of scalars.

**VdsRangeIdsVds — Vds or Vce output characteristics range**

[0, 5] (default) | row vector of two scalars

Range for the output characteristics of the MOSFET gate-drain voltage, Vds, or the IGBT collector-emitter voltage, Vce, in volts, specified as a row vector of two scalars.

**VgsStepsIdsVds — Vgs or Vge output characteristics steps**

[2:3:20] (default) | vector of scalars

Steps for the output characteristics of the MOSFET gate-source voltage, Vgs, or the IGBT gate-emitter voltage, Vge, in volts, specified as a vector of scalars.

**VgsCapacitance — Vgs or Vge capacitance characteristics steps**

[-5:2:20] (default) | vector of scalars

Steps for the capacitance characteristics of the MOSFET gate-source voltage,  $V_{gs}$ , or the IGBT gate-emitter voltage,  $V_{ge}$ , in volts, specified as a vector of scalars.

#### **VdsCapacitance — Vds or Vce capacitance characteristics steps**

[0:5:30] (default) | vector of scalars

Steps for the capacitance characteristics of the MOSFET drain-source voltage,  $V_{ds}$ , or the IGBT collector-emitter voltage,  $V_{ce}$ , in volts, specified as a vector of scalars.

#### **frequencyCapacitance — AC frequency for capacitance characteristics**

1e6 (default) | positive scalar

AC frequency of the small signal for the capacitance characteristics, in Hz, specified as a positive scalar.

#### **ACVoltageCapacitance — AC amplitude for capacitance characteristics**

0.05 (default) | positive scalar

AC amplitude of the small signal for the capacitance characteristics, in F, specified as a positive scalar.

#### **VdsDiodeIv — Vgs or Vce diode characteristics sweep range**

[0, -3] (default) | row vector of two nonpositive scalars

Sweep range for the diode characteristics of the MOSFET gate-source voltage,  $V_{gs}$ , or the IGBT collector-emitter voltage,  $V_{ce}$ , in volts, specified as a row vector of two nonpositive scalars.

#### **VceTail — Vce voltage for turn-off tail current characteristics**

400 (default) | positive scalar

IGBT collector-emitter voltage,  $V_{ce}$ , in volts, for the turn-off tail current characteristics, specified as a positive scalar.

#### **PulseVgeTail — Gate-emitter pulse amplitude for turn-off tail current characteristics**

15 (default) | positive scalar

Amplitude of the pulse applied between the IGBT gate and emitter for the turn-off tail current characteristics, in volts specified as a positive scalar.

#### **PulsePeriodTail — Gate-emitter pulse period for turn-off tail current characteristics**

5e-6 (default) | positive scalar

Period of the pulse applied between the IGBT gate and emitter for the turn-off tail current characteristics, in seconds, specified as a positive scalar.

#### **T — Device case temperature vectors**

27 (default) | vector of scalars

Device case temperatures for all current-voltage characteristics, specified as a vector of scalars.

#### **IVSimulationTime — Simulation time for current-voltage characteristics**

20 (default) | positive scalar

Simulation time for the current-voltage characteristics, in seconds, specified as a positive scalar.

**IVSimulationStepSize — Simulation step size for current-voltage characteristics**

0.02 (default) | positive scalar

Simulation step size for all current-voltage characteristics, specified as a positive scalar.

**RelTol — Relative tolerance**

1e-3 (default) | scalar

Relative tolerance parameter used in SPICE simulations, specified as a scalar.

**AbsTol — Absolute tolerance**

1e-4 (default) | scalar

Absolute current tolerance parameter used in SPICE simulations, specified as a scalar.

**VnTol — Absolute voltage tolerance**

1e-2 (default) | scalar

Absolute voltage tolerance parameter used in SPICE simulations, specified as a scalar.

**Gmin — Parallel conductance**

1e-6 (default) | scalar

Parallel conductance with all non-linear devices used in SPICE simulations, in 1/Ohm, specified as a scalar.

**CShunt — Shunt resistance**

1e-6 (default) | scalar

Shunt resistance to capacitors used in SPICE simulations, in F, specified as a scalar.

**Output Arguments****lookuptable — Table lookup data**

structure

Generated lookup table data from the SPICE subcircuit file, returned as a structure with these fields:

**channel — Channel data**

structure

Data that characterizes the channel of the SPICE subcircuit, returned as a structure. The function returns the channel structure only when you specify the name-value arguments `flagIdsVgs` or `flagIdsVds` with a value of 1. The structure contains these fields:

**VgsVec — Vgs or Vge voltage**

vector of scalars

MOSFET gate-source voltages, `Vgs`, or IGBT gate-emitter voltages, `Vge`, in volts, returned as a vector of scalars.

**VdsVec — Vds or Vce voltage**

vector of scalars

MOSFET drain-source voltages,  $V_{ds}$ , or IGBT collector-emitter voltages,  $V_{ce}$ , in volts, returned as a vector of scalars.

### **TVec — Temperature**

vector of scalars

Temperatures, in degC, returned as a vector of scalars.

### **IdsMat — Drain-source or collector-emitter currents**

matrix of scalars

MOSFET drain-source currents,  $I_{ds}(V_{gs}, V_{ds}, T)$ , or IGBT collector-emitter currents,  $I_{ce}(V_{ge}, V_{ce}, T)$ , returned as a matrix of scalars.

### **capacitance — Capacitance data**

structure

Data that characterizes the capacitance of the SPICE subcircuit, returned as a structure. The function returns the capacitance structure only when you specify the name-value argument `flagCapacitance` with a value of 1. The structure contains these fields:

#### **VgsVec — Vgs or Vge voltage**

vector of scalars

MOSFET gate-source voltages,  $V_{gs}$ , or IGBT gate-emitter voltages,  $V_{ge}$ , in volts, returned as a vector of scalars.

#### **VdsVec — Vds or Vce voltage**

vector of scalars

MOSFET drain-source voltages,  $V_{ds}$ , or IGBT collector-emitter voltages,  $V_{ce}$ , in volts, returned as a vector of scalars.

#### **CgsMat — Cgs or Cge capacitance**

matrix of scalars

MOSFET gate-source capacitance,  $C_{gs}$ , or IGBT gate-emitter capacitance,  $C_{ge}$ , returned as a matrix of scalars.

#### **CgdMat — Cgd or Cgc capacitance**

matrix of scalars

MOSFET gate-drain capacitance,  $C_{gd}$ , or IGBT gate-collector capacitance,  $C_{gc}$ , returned as a matrix of scalars.

#### **CdsVec — Cds or Cce voltage**

vector of scalars

MOSFET drain-source capacitance,  $C_{ds}$ , or IGBT collector-emitter capacitance,  $C_{ce}$ , returned as a vector of scalars.

### **diode — Diode data**

structure

Data that characterizes the drain-source diode or the collector-emitter diode of the SPICE subcircuit, returned as a structure. The function returns the `diode` structure only when you specify the name-value argument `flagDiodeIv` with a value of 1. The structure contains these fields:

**VVec – Diode voltages**

vector of positive scalars

Diode voltages, in volts, returned as a vector of scalars.

**TVec – Temperatures**

vector of scalars

Temperatures, returned as a vector of scalars.

**IMat – Tabulated currents**

matrix of scalars

Tabulated diode currents,  $I_{diode}(VVec, TVec)$ , or IGBT collector-emitter currents,  $I_{ce}(Vge, Vce, T)$ , in amperes, returned as a matrix of scalars.

**igbtTail – Data of IGBT turn-off tail current transient**

structure

Data that characterizes the IGBT turn-off tail current transient time, returned as a structure. The function returns the `igbtTail` structure only when you specify the name-value argument `flagTailTransient` with a value of 1. The structure contains this field:

**TT – Time constant for IGBT turn-off current tail transient**

positive scalar

Time constant of the IGBT turn-off current tail transient, in seconds, returned as a positive scalar.

## Version History

Introduced in R2022a

### See Also

**Blocks**

N-Channel MOSFET | P-Channel MOSFET | N-Channel IGBT

**Functions**

`subcircuit2ssc`

**Topics**

“SiC MOSFET Parameterization Using Simulation Results from SPICE”

“Converting a SPICE Netlist to Simscape Blocks”

# elec\_calculateFluxPartialDerivatives

Calculate flux partial derivatives for FEM-Parameterized PMSM block

## Syntax

```
[dFdA,dFdB,dFdC,dFdX] = elec_calculateFluxPartialDerivatives(A,B,C,X,F)
[dFdA,dFdB,dFdC,dFdX,D,Q] = elec_calculateFluxPartialDerivatives(A,B,C,X,F)
```

## Description

`[dFdA,dFdB,dFdC,dFdX] = elec_calculateFluxPartialDerivatives(A,B,C,X,F)` calculates the partial derivatives from flux linkage. For improved numerical performance, the FEM-Parameterized PMSM block works with flux linkage partial derivatives, rather than directly with flux linkage. If your finite-element motor design tool does not have an option to output partial derivatives, then you can use this function to calculate the partial derivatives from the flux linkage. The flux linkage  $F$  must be a four-dimensional matrix with the first three dimensions corresponding to the A, B, and C phase currents, and the fourth dimension corresponding to the rotor angle  $X$ . The function returns four-dimensional matrices for the four partial derivatives. Use this syntax in conjunction with the 4-D Data variant of the block.

`[dFdA,dFdB,dFdC,dFdX,D,Q] = elec_calculateFluxPartialDerivatives(A,B,C,X,F)` returns two additional output arguments corresponding to  $d$ -axis and  $q$ -axis currents, respectively. In this case, the four partial derivatives are three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle. Use this syntax in conjunction with the 3-D Data variant of the block.

## Examples

### Calculate 4-D flux linkage partial derivatives

Suppose  $F$  is a four-dimensional matrix containing flux linkage data, exported by your finite-element motor design tool. The matrix dimensions correspond to the three phase currents and the rotor angle, respectively. The data is cyclical in the fourth dimension, corresponding to the rotor angle.

---

**Tip** If you do not have data from a finite-element motor design tool for your PMSM, to prevent a simulation error, before running the code for this example, first generate the required  $F$  matrix by running the Generate 4-D Flux Linkage Matrix  $F$  example for the `ee_generateIdealPMSMfluxData` function.

---

Either directly import or recreate the current vectors. For example, if recreating a current vector with evenly spaced values between -250 and 250 A and 5 A increments, then:

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Import or define the number of pole pairs.

```
N = 6;
```

Import the rotor angle vector or recreate it based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Calculate the flux linkage partial derivatives.

```
[dFdA,dFdB,dFdC,dFdX] = elec_calculateFluxPartialDerivatives(iA,iB,iC,X,F);
```

The function returns four 4-D matrices for the flux linkage partial derivatives. The four matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions also correspond to the three phase currents and the rotor angle.

### Calculate 3-D flux linkage partial derivatives

Suppose  $F$  is a four-dimensional matrix containing flux linkage data, exported by your finite-element motor design tool. The matrix dimensions correspond to the three phase currents and the rotor angle, respectively. The data is cyclical in the fourth dimension, corresponding to the rotor angle.

---

**Tip** If you do not have data from a finite-element motor design tool for your PMSM, to prevent a simulation error, before running the code for this example, first generate the required  $F$  matrix by running the Generate 4-D Flux Linkage Matrix  $F$  example for the `ee_generateIdealPMSMfluxData` function.

---

Either directly import or recreate the current vectors. For example, if recreating a current vector with evenly spaced values between -250 and 250 A and 5 A increments:

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Import or define the number of pole pairs.

```
N = 6;
```

Import the rotor angle vector or recreate it based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Calculate the flux linkage partial derivatives.

```
[dFdA,dFdB,dFdC,dFdX,iD,iQ] = elec_calculateFluxPartialDerivatives(iA,iB,iC,X,F);
```

The function returns four 3-D matrices for the flux linkage partial derivatives and two vectors for the  $d$ -axis and  $q$ -axis current values. The four matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

## Input Arguments

**A** — A-phase current, in amperes

vector



A-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: double

### **B — B-phase current, in amperes**

vector

B-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: double

### **C — C-phase current, in amperes**

vector

C-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points.

Data Types: double

### **X — Rotor angle, in radians**

vector

The rotor angle, in radians, specified as a vector. The values must be in the range from zero to  $2\pi/N$ , where  $N$  is the number of pole pairs.

Data Types: double

### **F — Flux linkage, in weber-turns**

four-dimensional matrix

The flux linkage, in weber-turns, specified as a four-dimensional matrix, with dimensions corresponding to the three phase currents and rotor angle. The data must be cyclical in the fourth (rotor angle) dimension, that is, for all  $i, j$ , and  $k$ ,  $F(i, j, k, 0) = F(i, j, k, 2\pi/N)$ , where  $N$  is the number of pole pairs.

Data Types: double

## **Output Arguments**

### **dFdA — Flux linkage partial derivative with respect to the A-phase current**

matrix

Flux linkage partial derivative with respect to the A-phase current, returned as a matrix. For syntax used with the 4-D Data variant of the block, the matrix is four-dimensional. For syntax used with the 3-D Data variant of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdB — Flux linkage partial derivative with respect to the B-phase current**

matrix

Flux linkage partial derivative with respect to the B-phase current, returned as a matrix. For syntax used with the 4-D Data variant of the block, the matrix is four-dimensional. For syntax used with the

3-D Data variant of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdC — Flux linkage partial derivative with respect to the C-phase current**

matrix

Flux linkage partial derivative with respect to the  $C$ -phase current, returned as a matrix. For syntax used with the 4-D Data variant of the block, the matrix is four-dimensional. For syntax used with the 3-D Data variant of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **dFdX — Flux linkage partial derivative with respect to the rotor angle**

matrix

Flux linkage partial derivative with respect to the rotor angle, returned as a matrix. For syntax used with the 4-D Data variant of the block, the matrix is four-dimensional. For syntax used with the 3-D Data variant of the block, the matrix is three-dimensional, the first two dimensions corresponding to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponding to the rotor angle.

### **D — D-axis current, in amperes**

vector

$D$ -axis current, in amperes, returned as a vector. This is an optional output argument, to be used when you want to generate 3-D flux linkage partial derivatives. The vector defines the  $d$ -axis current values at which the partial derivatives are determined.

### **Q — Q-axis current, in amperes**

vector

$Q$ -axis current, in amperes, returned as a vector. This is an optional output argument, to be used when you want to generate 3-D flux linkage partial derivatives. The vector defines the  $q$ -axis current values at which the partial derivatives are determined.

## **Algorithms**

The function calculates partial derivatives using Akima splines, the same method that is used for smooth interpolation in the Simscape language `tablelookup` function. For more information, see `makima`. Akima splines are suitable for estimating partial derivatives due to their smooth nature and tendency not to introduce local gradient reversals.

## **Version History**

**Introduced in R2017a**

### **`elec_calculateFluxPartialDerivatives` will be removed**

*Not recommended starting in R2019a*

The `elec_calculateFluxPartialDerivatives` function will be removed in a future release. Use the `ee_calculateFluxPartialDerivatives` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

**See Also**

ee\_calculateFluxPartialDerivatives | ee\_generateIdealPMSMfluxData

## elec\_generateIdealPMSMfluxData

Generate tabulated flux linkage data for ideal PMSM

### Syntax

```
[F,T,dFdA,dFdB,dFdC,dFdX] = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)
[F] = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X)
[F,T,dFdA,dFdB,dFdC,dFdX] = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)
[F] = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X)
```

### Description

[F,T,dFdA,dFdB,dFdC,dFdX] = elec\_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X) generates 4-D flux linkage data, including the torque and the partial derivatives, for an ideal permanent magnet synchronous motor (PMSM).

Use this function to create test data for the FEM-Parameterized PMSM block, either for validation purposes or to set up a model before the actual flux linkage data is available.

[F] = elec\_generateIdealPMSMfluxData(PM,Ld,Lq,L0,A,B,C,X) generates 4-D flux linkage matrix  $F$  for an ideal permanent magnet synchronous motor (PMSM).

[F,T,dFdA,dFdB,dFdC,dFdX] = elec\_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X) generates 3-D flux linkage data, including the torque and the partial derivatives, for an ideal PMSM.

[F] = elec\_generateIdealPMSMfluxData(PM,Ld,Lq,L0,D,Q,X) generates 3-D flux linkage matrix  $F$  for an ideal PMSM.

### Examples

#### Generate 4-D Flux Linkage Data

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the phase current vectors.

```
iA = linspace(-250,250,5);
iB = iA;
iC = iA;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of A-,B-,C-currents and rotor angle.

```
[F, T, dFdA, dFdB, dFdC, dFdX] = elec_generateIdealPMSMfluxData(PM, Ld, Lq, L0, iA, iB, iC, X);
```

The function returns a 4-D flux linkage matrix  $F$ , a 4-D torque matrix  $T$ , and four 4-D matrices for the flux linkage partial derivatives. The four partial derivative matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the three phase currents and the rotor angle.

### Generate 4-D Flux Linkage Matrix $F$

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the phase current vectors.

```
iA = linspace(-250, 250, 5);
iB = iA;
iC = iA;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0, 360/N, 180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of A-,B-,C-currents and rotor angle.

```
F = elec_generateIdealPMSMfluxData(PM, Ld, Lq, L0, iA, iB, iC, X);
```

The function returns a 4-D flux linkage matrix  $F$ , a 4-D torque matrix  $T$ , and four 4-D matrices for the flux linkage partial derivatives. The four partial derivative matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the three phase currents and the rotor angle.

### Generate 3-D Flux Linkage Data

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the  $d$ -axis and  $q$ -axis current vectors.

```
iD = linspace(-250, 250, 5);
iQ = iD;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of  $d$ -axis and  $q$ -axis currents and rotor angle.

```
[F,T,dFdA,dFdB,dFdC,dFdX] = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,iD,iQ,X);
```

The function returns a 3-D flux linkage matrix  $F$ , a 3-D torque matrix  $T$ , and four 3-D matrices for the flux linkage partial derivatives. The four partial derivative matrices correspond to the three phase currents and the rotor angle, respectively. The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

### Generate 3-D Flux Linkage Matrix $F$

Specify the motor parameters.

```
PM = 0.1;      % Permanent magnet flux
N = 6;        % Number of pole pairs
Ld = 0.0002;  % D-axis inductance
Lq = 0.0002;  % Q-axis inductance
L0 = 0.00018; % Zero-sequence inductance
Rs = 0.013;   % Stator resistance
```

Define the  $d$ -axis and  $q$ -axis current vectors.

```
iD = linspace(-250,250,5);
iQ = iD;
```

Specify the rotor angle vector based on the number of pole pairs.

```
X = pi/180*linspace(0,360/N,180/N+1);
```

Tabulate flux linkage partial derivatives and torque in terms of  $d$ -axis and  $q$ -axis currents and rotor angle.

```
F = elec_generateIdealPMSMfluxData(PM,Ld,Lq,L0,iD,iQ,X);
```

The function returns a 3-D flux linkage matrix  $F$ . The matrix dimensions correspond to the  $d$ -axis and  $q$ -axis currents and the rotor angle.

## Input Arguments

### **PM** — Peak permanent magnet flux linkage, in weber-turns

scalar

Peak permanent magnet flux linkage, in weber-turns, specified as a scalar.

Data Types: double

### **Ld** — D-axis inductance, in henries

scalar

$D$ -axis inductance, in henries, specified as a scalar.

Data Types: double

**L<sub>q</sub> — Q-axis inductance, in henries**

scalar

Q-axis inductance, in henries, specified as a scalar.

Data Types: double

**L<sub>0</sub> — Zero-sequence inductance, in henries**

scalar

Zero-sequence inductance, in henries, specified as a scalar.

Data Types: double

**A — A-phase current, in amperes**

vector

A-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**B — B-phase current, in amperes**

vector

B-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**C — C-phase current, in amperes**

vector

C-phase current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 4-D flux linkage data.

Data Types: double

**D — D-axis current, in amperes**

vector

D-axis current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 3-D flux linkage data.

Data Types: double

**Q — Q-axis current, in amperes**

vector

Q-axis current, in amperes, specified as a vector. The vector must be monotonically increasing and two-sided (contain both positive and negative values). Best practice is to include zero current as one of the points. Use this input argument to generate 3-D flux linkage data.

Data Types: double

**X — Rotor angle, in radians**

vector

The rotor angle, in radians, specified as a vector. The values must be in the range from zero to  $2\pi/N$ , where  $N$  is the number of pole pairs.

Data Types: double

**Output Arguments****F — Flux linkage**

matrix

The flux linkage, in weber-turns, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

**T — Torque**

matrix

Torque, in N\*m, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

**dFdA — Flux linkage partial derivative with respect to the A-phase current**

matrix

Flux linkage partial derivative with respect to the A-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

**dFdB — Flux linkage partial derivative with respect to the B-phase current**

matrix

Flux linkage partial derivative with respect to the B-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

**dFdC — Flux linkage partial derivative with respect to the C-phase current**

matrix

Flux linkage partial derivative with respect to the C-phase current, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.



## dFdX — Flux linkage partial derivative with respect to the rotor angle

matrix

Flux linkage partial derivative with respect to the rotor angle, returned as a matrix. The matrix can be four-dimensional or three-dimensional, depending on the syntax used to call the function. In a four-dimensional matrix, the first three dimensions correspond to the phase currents, and the fourth dimension corresponds to the rotor angle. In a three-dimensional matrix, the first two dimensions correspond to the  $d$ -axis and  $q$ -axis currents, and the third dimension corresponds to the rotor angle.

## Algorithms

The flux linking each winding has contributions from the permanent magnet plus the three windings. Therefore, the total flux is given by [1] on page 2-20:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix}$$

$$L_{aa} = L_s + L_m \cos(2\theta_r)$$

$$L_{bb} = L_s + L_m \cos(2(\theta_r - 2\pi/3))$$

$$L_{cc} = L_s + L_m \cos(2(\theta_r + 2\pi/3))$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos(\theta_r + \pi/6)$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos(\theta_r + \pi/6 - 2\pi/3)$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos(\theta_r + \pi/6 + 2\pi/3)$$

$$\psi_{am} = \psi_m \cos\theta_e$$

$$\psi_{bm} = \psi_m \cos(\theta_e - 2\pi/3)$$

$$\psi_{cm} = \psi_m \cos(\theta_e + 2\pi/3)$$

Here,  $\theta_e$  is the electrical angle, which is related to rotor angle  $\theta_r$  by  $\theta_e = N \cdot \theta_r$ . The function assumes that the permanent magnet flux linking the A-phase winding is at the maximum for  $\theta_e = 0$ .

The function output F corresponds to  $\psi_a$  tabulated as a function of A-phase current, B-phase current, C-phase current, and rotor angle.

$L_s$ ,  $L_m$ , and  $M_s$  are related to input arguments  $L_d$ ,  $L_q$ , and  $L_0$  by:

$$L_s = \frac{L_0}{3} + \frac{L_d}{3} + \frac{L_q}{3}$$

$$M_s = \frac{L_d}{6} - \frac{L_0}{3} + \frac{L_q}{6}$$

$$L_m = \frac{L_d}{3} - \frac{L_q}{3}$$

## Version History

Introduced in R2017a

### **eIec\_generateIdealPMSMfluxData will be removed**

*Not recommended starting in R2019a*

The `eIec_generateIdealPMSMfluxData` function will be removed in a future release. Use the `ee_generateIdealPMSMfluxData` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

### **References**

[1] Anderson, P.M. *Analysis of Faulted Power Systems*. 1st Edition. Wiley-IEEE Press, July 1995, p.187.

### **See Also**

`ee_calculateFluxPartialDerivatives` | `ee_generateIdealPMSMfluxData`

### **Topics**

“HEV PMSM Drive Test Harness”  
“PMSM Iron Losses”

# elec\_getEfficiency

Calculate efficiency as function of dissipated power losses

## Syntax

```
efficiency = elec_getEfficiency('loadIdentifier',node)
efficiency = elec_getEfficiency('loadIdentifier',...
node,startTime,endTime)
[efficiency,lossesTable] = elec_getEfficiency('loadIdentifier',node)
```

## Description

`efficiency = elec_getEfficiency('loadIdentifier',node)` returns the efficiency of a circuit based on the data extracted from a Simscape logging node.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all semiconductor blocks in the model. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

Checking efficiency allows you to determine if circuit components are operating within their requirements. All blocks in the Semiconductor Devices library, as well as some other blocks, have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the block. This instantaneous dissipated power includes only the real power (not the reactive or apparent power) that the block dissipates. When you log simulation data, the time-value series for this variable represents the power dissipated by the block over time. You can view and plot this data using the Simscape Results Explorer. The `ee_getPowerLossTimeSeries` function also allows you to access this data.

The `elec_getEfficiency` function calculates the efficiency of the circuit based on the losses for blocks that have a `power_dissipated` variable and that you identify as a load block. The equation for efficiency is

$$Eff = 100 \cdot \frac{P_{load}}{P_{loss} + P_{load}}$$

where:

- $Eff$  is the efficiency of the circuit.
- $P_{load}$  is the output power, that is, the power dissipated by load blocks.
- $P_{loss}$  is the power dissipated by nonload blocks.

This equation assumes that all loss mechanisms are captured by blocks containing at least one `power_dissipated` variable. If the model contains any lossy blocks that do not have this variable, the efficiency calculation gives incorrect results.

Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the

MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses to provide the total power loss for the block, averaged over simulation time. The function uses the loss data to calculate the efficiency of the circuit.

`efficiency = elec_getEfficiency('loadIdentifier',... node, startTime, endTime)` returns the efficiency of a circuit based on the `power_dissipated` data extracted from a Simscape logging node within a time interval. `startTime` and `endTime` represent the start and end of the time interval for calculating the efficiency. If you omit these two input arguments, the function calculates the efficiency over the whole simulation time.

`[efficiency, lossesTable] = elec_getEfficiency('loadIdentifier', node)` returns the efficiency of a circuit and the power loss contributions of the nonload blocks in a circuit based on the data extracted from a Simscape logging node.

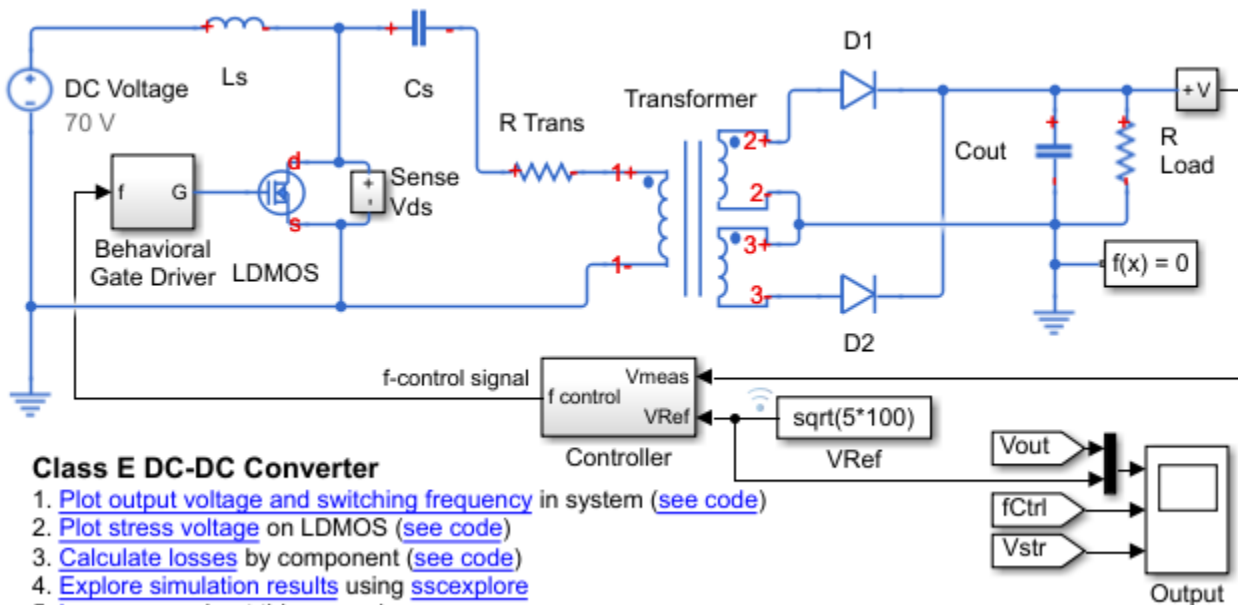
## Examples

### Calculate Efficiency for a Circuit

This example shows how to calculate efficiency based on the power dissipated by blocks in a circuit using the `elec_getEfficiency` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dcdc_class_e';
open_system(model)
```



The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain 'Load' in their names. Therefore, you can use the string 'Load' as the 'loadIdentifier' argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the 'loadIdentifier' argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

Calculate efficiency and display the results.

```
efficiency = elec_getEfficiency('Load',simlog_ee_converter_dcdc_class_e)

efficiency =

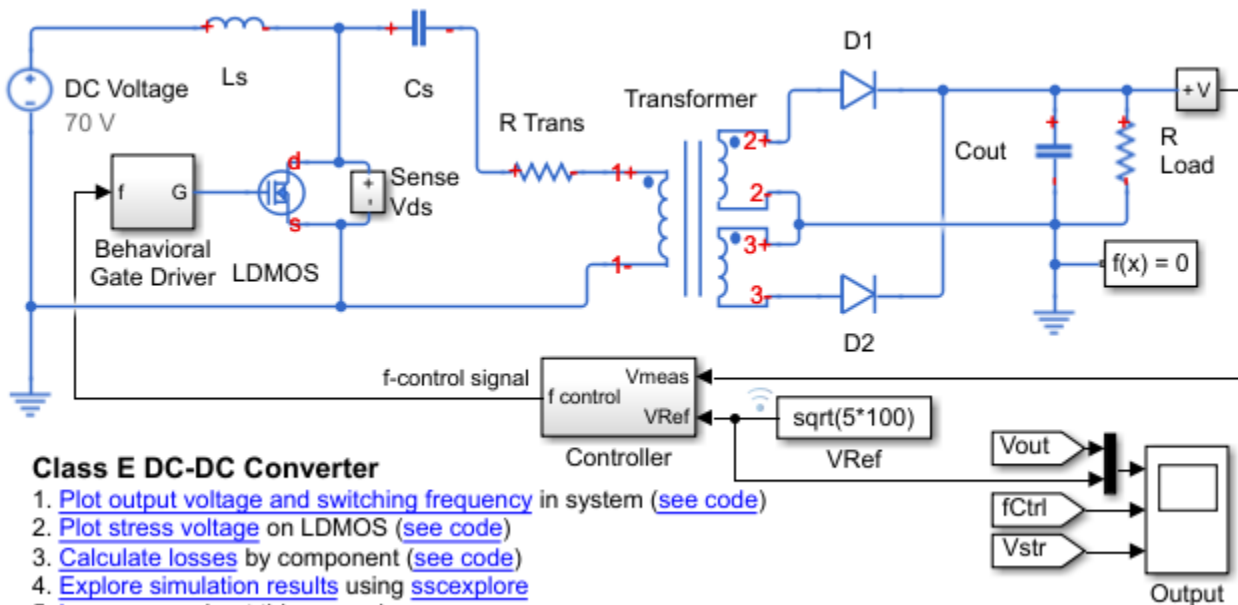
    90.0326
```

### Calculate Efficiency of a Circuit for a Specific Time Period

This example shows how to calculate efficiency based on the power dissipated for a specific time period using the `elec_getEfficiency` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dcdc_class_e';
open_system(model)
```



The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain 'Load' in their names. Therefore, you can use the string 'Load' as the 'loadIdentifier' argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the 'loadIdentifier' argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

The model simulation time ( $t$ ) is  $1.25e-4$  seconds. Calculate efficiency for the interval when  $t$  is between  $1e-4$  and  $1.25e-4$  seconds.

```
efficiency = elec_getEfficiency('Load',simlog_ee_converter_dcdc_class_e,1e-4,1.25e-4)
```

```
efficiency =
```

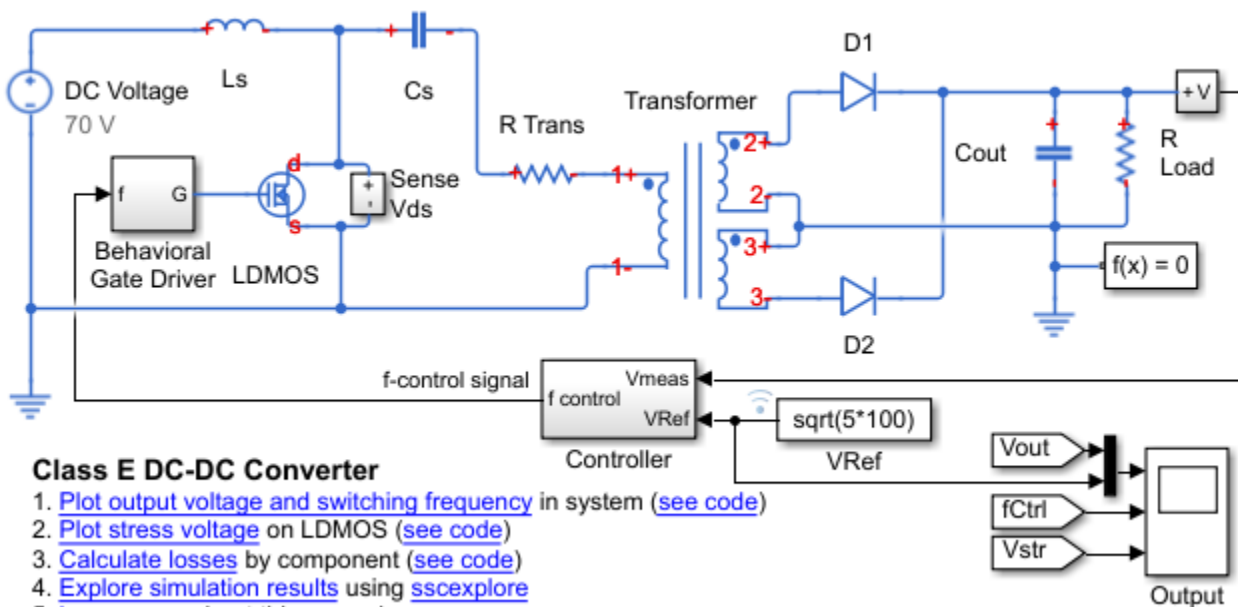
```
90.4899
```

### Calculate Efficiency and Power-Loss Contributions

This example shows how using the `elec_getEfficiency` function allows you to calculate both the efficiency of the circuit and the power-loss contributions of the nonload blocks based on the power that they dissipate.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_converter_dcdc_class_e';
open_system(model)
```



The load in the model is represented by the R Load resistor. No other blocks with `power_dissipated` variables contain 'Load' in their names. Therefore, you can use the string 'Load' as the 'loadIdentifier' argument.

If no string at least partially matches the names of all load blocks in your circuit, rename the load blocks using a schema that satisfies the matching criteria for the 'loadIdentifier' argument.

This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_converter_dcdc_class_e` is saved in your current workspace.

Calculate the efficiency and power-loss contributions due to dissipated power.

```
[efficiency,lossesTable] = elec_getEfficiency('Load',simlog_ee_converter_dcdc_class_e)
```

```
efficiency =
```

```
90.0326
```

```
lossesTable =
```

```
7x2 table array
```

LoggingNode	Power
'ee_converter_dcdc_class_e.LDMOS'	3.6583
'ee_converter_dcdc_class_e.R_Trans.Resistor'	2.911
'ee_converter_dcdc_class_e.D2.diode'	1.9446
'ee_converter_dcdc_class_e.D1.diode'	1.837
'ee_converter_dcdc_class_e.Cs'	0.27391
'ee_converter_dcdc_class_e.Ls'	0.27097
'ee_converter_dcdc_class_e.Cout'	0.00044593

## Input Arguments

### 'loadIdentifier' – Identify load blocks in the circuit

case-sensitive string

String that is a complete or partial match for the names of load blocks in the circuit. For example, consider a circuit that contains the four semiconductor blocks shown in the table.

Block Name in the Model	IGBT	IGBT1_Load	Diode	Diode1	
<b>Block Type</b>	N-Channel IGBT	N-Channel IGBT	Diode	Diode	
<b>Block Role in the Model</b>	Source	Load	Load	Load	
<b>'loadIdentifier'</b>	'IGBT'	Yes	Yes	No	No
	'Diode'	No	No	Yes	Yes

Block Name in the Model	IGBT	IGBT1_Load	Diode	Diode1
'Load'	No	Yes	No	No
'1'	No	Yes	No	Yes
'D'	No	No	Yes	Yes
'd'	No	Yes	Yes	Yes

The `elec_getEfficiency` function returns data just for the three load blocks only when the `'loadIdentifier'` is `'d'`.

A load-block naming schema that gives you better control over the output of the `elec_getEfficiency` function is shown in this table.

Block Name in the Model	IGBT	IGBT1_Load	Diode_Load	Diode1_Load	
<b>Block Type</b>	N-Channel IGBT	N-Channel IGBT	Diode	Diode	
<b>Block Role in the Model</b>	Source	Load	Load	Load	
<b>'loadIdentifier'</b>	'IGBT'	Yes	Yes	No	No
	'Diode'	No	No	Yes	Yes
	'Load'	No	Yes	Yes	Yes

Example: `'Load'`

Data Types: `string`

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If `node` is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If `node` is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog.Cell1.MOS1`

### **startTime** — Start of the time interval for calculating the efficiency

0 (default) | real number

Start of the time interval for calculating the efficiency, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than `endTime`.

Data Types: `double`



**endTime** — End of the time interval for calculating the efficiency

simulation stop time (default) | real number

End of the time interval for calculating the efficiency, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

Data Types: `double`

## Output Arguments

**efficiency** — Efficiency of the circuit

percentage

Efficiency of the circuit based on data extracted from a Simscape logging node.

**lossesTable** — Dissipated power for each nonload blocks

table

Dissipated power losses for each nonload block, returned as a table. The first column lists logging nodes for all blocks that have at least one `power_dissipated` variable. The second column lists the corresponding losses in watts.

## Assumptions

- The output power equals the total power dissipated by blocks that you identify as load blocks.
- The input power equals the output power plus the total power dissipated by blocks that you do not identify as load blocks.
- The `power_dissipated` variables capture all loss contributions.

## Version History

**Introduced in R2017a****elec\_getEfficiency will be removed***Not recommended starting in R2019a*

The `elec_getEfficiency` function will be removed in a future release. Use the `ee_getEfficiency` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## See Also

`ee_getEfficiency` | `ee_getPowerLossSummary` | `ee_getPowerLossTimeSeries` | `sscexplore`

### Topics

"About Simulation Data Logging"

"About the Simscape Results Explorer"

## elec\_getNodeDvDtSummary

Calculate maximum absolute values of terminal voltage time derivatives (dv/dt) based on logged simulation data

### Syntax

```
summaryTable = elec_getNodeDvDtSummary(node,tau)
```

### Description

`summaryTable = elec_getNodeDvDtSummary(node,tau)` calculates the maximum absolute values of rates-of-change of voltage variables for nodes that are based on the `foundation.electrical.electrical` domain, based on logged simulation data. The function returns the data for each terminal in a table. The data in the table appears in descending order according to the maximum magnitude of the rate-of-change of voltage variables with respect to the ground, over the whole simulation time. The table does not contain data for terminals that are held fixed.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the children of that node.

Examining rates-of-change of voltage variables in power electronics circuits is useful for determining the potential for unwanted conducted or radiated emissions. The rate-of-change data also helps you to identify switching devices that might be susceptible to parasitic turn-on. All nodes that are based on the `foundation.electrical.electrical` domain store the potential with respect to electrical ground as the variable `v`. When you log simulation data, the time-value series for this variable represents the trend of the potential over time. You can view and plot this data using the Simscape Results Explorer.

To evaluate the rates-of-change of voltage variables, the `elec_getNodeDvDtSummary` function employs finite difference approximation of the first derivative with respect to time. It performs 1-D data linear interpolation of voltage variables using a uniform grid with the time step, `tau`. The function then applies the central differencing scheme to the interpolated data.

---

**Tip** For small time steps, finite differencing may lead to inaccurate results. The time step `tau` should be small enough to capture waveforms, but not so small that the finite differencing error becomes large. For example, for power transistors with an expected limit of 50 V/ns for their voltage rate-of-change, a reasonable guess for `tau` is 1e-9 s.

---

### Examples

#### Calculate Maximum Voltage Derivatives by Block for the Whole Model

Open the Class E DC-DC Converter example model.



The table shows the maximum absolute values over the whole simulation time of voltage rates-of-change for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain.

## Input Arguments

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog_ee_converter_dcdc_class_e.LDMOS`

### **tau** — Time step for numerical differentiation

real number

Time step for numerical differentiation, specified as a real number, in seconds. `tau` determines the interpolation grid as `startTime:tau:endTime`.

Example: `1e-9`

Data Types: `double`

## Output Arguments

### **summaryTable** — Maximum absolute values of the voltage rates-of-change for each block table

Maximum absolute values of the voltage rates-of-change for each block, returned as a table. The first column lists all the logging nodes in node that are based on the `foundation.electrical.electrical` domain. The second column lists the terminal names. The third column lists the corresponding maximum absolute values of voltage rates-of-change, in volts per second. The table does not contain data for terminals that are held fixed.

## Version History

**Introduced in R2018b**

### **`elec_getNodeDvDtSummary` will be removed**

*Not recommended starting in R2019a*

The `elec_getNodeDvDtSummary` function will be removed in a future release. Use the `ee_getNodeDvDtSummary` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## See Also

`ee_getNodeDvDtSummary` | `ee_getNodeDvDtTimeSeries` | `sscexplore`

**Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## elec\_getNodeDvDtTimeSeries

Calculate rates-of-change of voltage variables

### Syntax

```
seriesTable = elec_getNodeDvDtTimeSeries(node,tau)
```

### Description

`seriesTable = elec_getNodeDvDtTimeSeries(node,tau)` calculates rates-of-change of voltage variables for nodes that are based on the `foundation.electrical.electrical` domain, based on logged simulation data. The function returns the data for each terminal in a table. The data in the table appears in descending order according to the maximum absolute value of the rate-of-change of voltage variables with respect to the ground, over the whole simulation time. The table does not contain data for terminals that are held fixed.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the children of that node.

Examining rates-of-change of voltage variables in power electronics circuits is useful for determining the potential for unwanted conducted or radiated emissions. The rate-of-change data also helps you to identify unwanted turn-on of switching devices. All nodes that are based on the `foundation.electrical.electrical` domain store the potential with respect to electrical ground as the variable `v`. When you log simulation data, the time-value series for this variable represents the trend of the potential over time. You can view and plot this data using the Simscape Results Explorer.

To evaluate the rates-of-change of voltage variables, the `elec_getNodeDvDtTimeSeries` function employs finite difference approximation of the first derivative with respect to time. It performs 1-D data linear interpolation of voltage variables using a uniform grid with the time step, `tau`. The function then applies the central differencing scheme to the interpolated data.

---

**Tip** For small time steps, finite differencing may lead to inaccurate results. The time step `tau` should be small enough to capture waveforms, but not so small that the finite differencing error becomes large. For example, for power transistors with an expected limit of 50 V/ns for their voltage rate-of-change, a reasonable guess for `tau` is 1e-9 s.

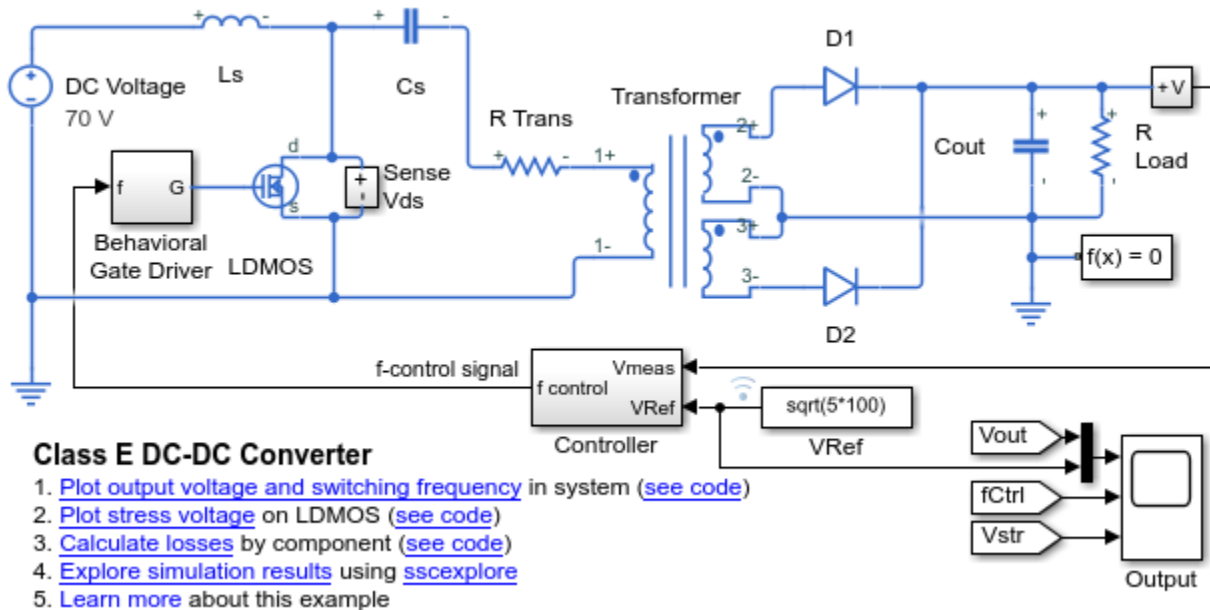
---

### Examples

#### Calculate Voltage Derivatives by Block for the Whole Model

Open the Class E DC-DC Converter example model.

```
open_system('ee_converter_dcdc_class_e')
```



This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_converter_dcdc_class_e` in your current workspace.

```
sim('ee_converter_dcdc_class_e');
```

Calculate rates-of-change of voltage variables for the whole model with a time step of  $1e-9$  seconds, and return the time series data in a table.

```
seriesTable = elec_getNodeDvDtTimeSeries(simlog_ee_converter_dcdc_class_e,1e-9)
```

```
seriesTable =
```

```
19x4 table
```

LoggingNode	Terminal	Voltage	dvdt
"ee_converter_dcdc_class_e.R_Trans"	"n"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Transformer"	"p1"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Cs"	"n"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.R_Trans"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Cs"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.LDMOS"	"D"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Ls"	"n"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Sense_Vds.Voltage_Stress_Sensor"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.D2"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Transformer"	"n3"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.D1"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Transformer"	"p2"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Behavioral_Gate_Driver.Controlled_Voltage_Source"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.LDMOS"	"G"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Cout"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.D1"	"n"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.D2"	"n"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.R_Load"	"p"	[1x125001 double]	[1x125001 double]
"ee_converter_dcdc_class_e.Sense_Vout.Voltage_Sensor"	"p"	[1x125001 double]	[1x125001 double]

The table contains time series data of voltage variables and their first derivatives over the whole simulation time for all the blocks in the model that have nodes based on the `foundation.electrical.electrical` domain.

View the time series data. From the workspace, open the `seriesTable` table, then open the two `1x125001 double numeric arrays` for the `ee_converter_dcdc_class_e.LDMOS.D`.

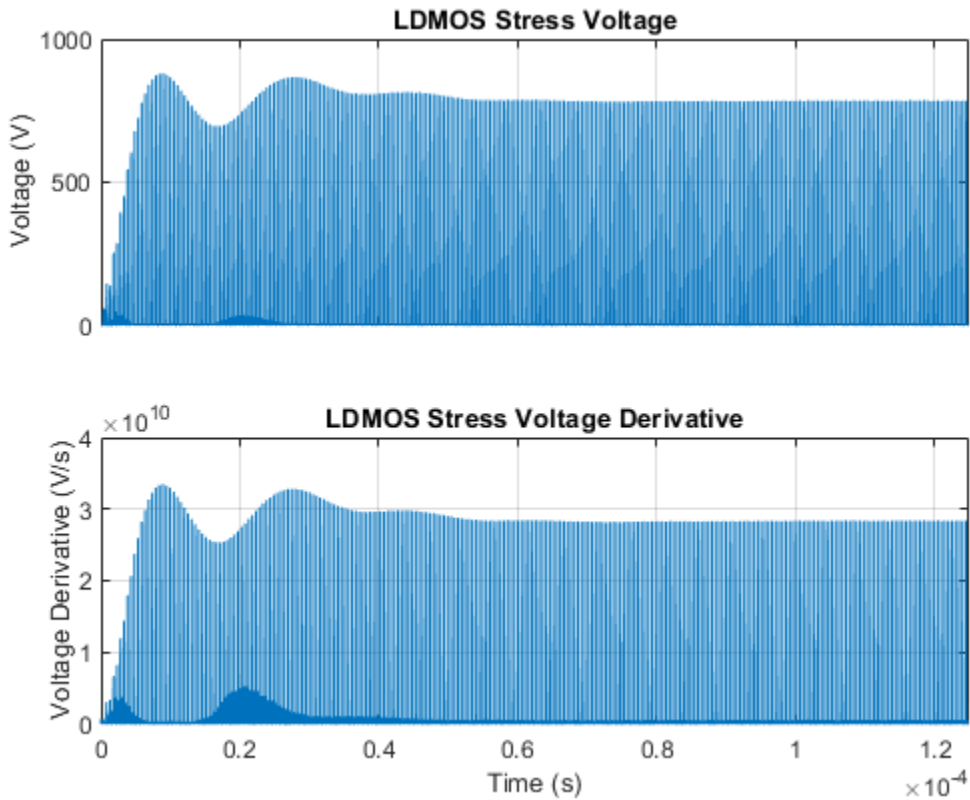
The first array contains the voltage data. The second array contains the voltage derivative data.

Plot the data.

```
time = 0:1e-9:1.25e-4;
v0ut = seriesTable.Voltage{6};
dvdt0ut = seriesTable.dvdt{6};

ax1 = subplot(2,1,1);
plot(time,v0ut),grid;
ylabel('Voltage (V)');
axis([0 1.25e-4 0 1000]);
ax1.XTickLabel = {};
ax1.Title.String = 'LDMOS Stress Voltage';

ax2 = subplot(2,1,2);
plot(time,dvdt0ut),grid;
ylabel('Voltage Derivative (V/s)');
xlabel('Time (s)');
axis([0 1.25e-4 0 4e10]);
ax2.Title.String = 'LDMOS Stress Voltage Derivative';
```





## Input Arguments

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog_ee_converter_dc_dc_class_e.LDMOS`

### **tau** — Time step for numerical differentiation

real number

Time step for numerical differentiation, specified as a real number, in seconds. `tau` determines the interpolation grid as `startTime:tau:endTime`.

Example: `1e-9`

Data Types: `double`

## Output Arguments

### **seriesTable** — Time series of the voltage rates-of-change for each block

table

Time series of the voltage rates-of-change for each block, returned as a table. The first column lists all the logging nodes in node that are based on the `foundation.electrical.electrical` domain. The second column lists the terminal names. The third column lists the corresponding interpolated voltage values, in volts. The fourth column lists the corresponding numerically differentiated values of voltage rates-of-change, in volts per second. The table does not contain data for terminals that are held fixed.

## Version History

**Introduced in R2018b**

### **elec\_getNodeDvDtTimeSeries will be removed**

*Not recommended starting in R2019a*

The `elec_getNodeDvDtTimeSeries` function will be removed in a future release. Use the `ee_getNodeDvDtTimeSeries` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## See Also

`ee_getNodeDvDtSummary` | `ee_getNodeDvDtTimeSeries` | `sscexplore`

### Topics

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## elec\_getPowerLossSummary

Calculate dissipated power losses

### Syntax

```
lossesTable = elec_getPowerLossSummary(node)
```

### Description

`lossesTable = elec_getPowerLossSummary(node)` calculates dissipated power losses for semiconductor blocks in a model, based on logged simulation data, and returns the data for each block in a table.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file. If `node` is the name of the simulation log variable, then the table contains the data for all semiconductor blocks in the model. If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

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

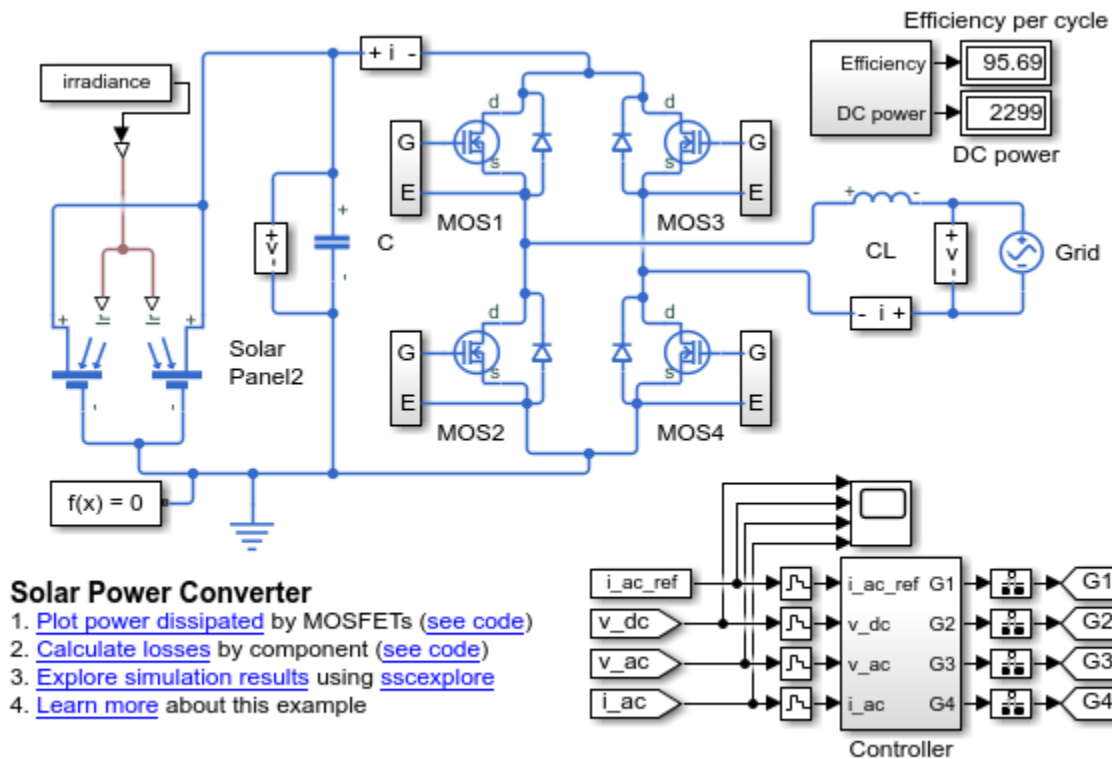
The `elec_getPowerLossSummary` function calculates average losses for each block that has a `power_dissipated` variable. Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses and provides the power loss value for the whole block, averaged over simulation time.

### Examples

#### Calculate Power Losses for One Block

Open the Solar Power Converter example model.

```
ee_solar_inverter
```



This example model has data logging enabled. Run the simulation to create the simulation log variable `simlog_ee_solar_inverter` in your current workspace.

```
sim('ee_solar_inverter');
```

Calculate power losses for the MOS1 block.

```
mosfetLosses = elec_getPowerLossSummary(simlog_ee_solar_inverter.MOS1)
```

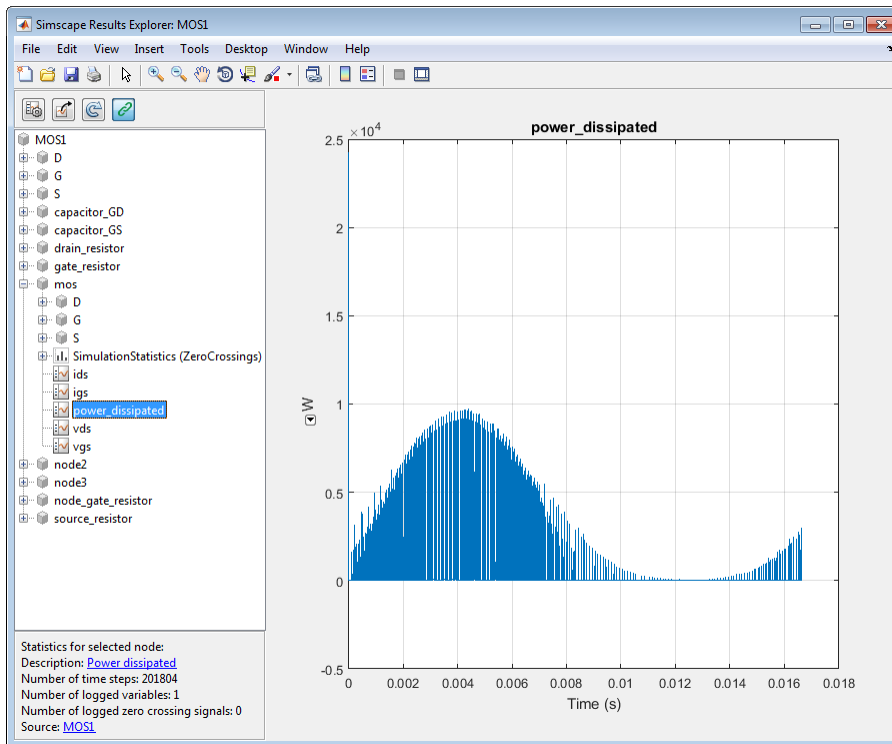
```
mosfetLosses =
```

LoggingNode	Power
'MOS1'	15.316

The table shows dissipated power losses for the MOS1 block, averaged over the whole simulation time.

Use the `sscxplore` function to further explore the power loss data for the MOSFET block.

```
sscxplore(simlog_ee_solar_inverter.MOS1)
```



The block has several `power_dissipated` logging nodes: under `drain_resistor`, under `gate_resistor`, under `mos`, and under `source_resistor`. The 15.316 value calculated by the `elec_getPowerLossSummary` function is a sum of all these losses, averaged over the simulation time.

## Input Arguments

**node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

Example: `simlog.Cell1.MOS1`

## Output Arguments

**lossesTable** — Dissipated power losses for each block

table

Dissipated power losses for each block, returned as a table. The first column lists logging nodes for all blocks that have at least one `power_dissipated` variable. The second column lists the corresponding losses in watts.

## Version History

Introduced in R2015a

### **elec\_getPowerLossSummary will be removed**

*Not recommended starting in R2019a*

The `elec_getPowerLossSummary` function will be removed in a future release. Use the `ee_getPowerLossSummary` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

### **See Also**

`ee_getEfficiency` | `ee_getPowerLossSummary` | `ee_getPowerLossTimeSeries` | `sscexplore`

### **Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## elec\_getPowerLossTimeSeries

Calculate dissipated power losses and return time series data

### Syntax

```
lossesCell = elec_getPowerLossTimeSeries(node)
lossesCell = elec_getPowerLossTimeSeries(node,startTime,endTime)
lossesCell = elec_getPowerLossTimeSeries(node,startTime,...
endTime,intervalWidth)
```

### Description

`lossesCell = elec_getPowerLossTimeSeries(node)` calculates dissipated power losses for blocks in a model, based on logged simulation data, and returns the time series data for each block.

Before you call this function, you must have the simulation log variable in your current workspace. Create the simulation log variable by simulating the model with data logging turned on, or load a previously saved variable from a file.

The `elec_getPowerLossTimeSeries` function calculates dissipated power losses for each block that has a `power_dissipated` variable. All blocks in the Semiconductor Devices library, as well as some other blocks, have an internal variable called `power_dissipated`, which represents the instantaneous power dissipated by the block. Some blocks have more than one `power_dissipated` variable, depending on their configuration. For example, the N-Channel MOSFET block has separate `power_dissipated` logging nodes for the MOSFET, the gate resistor, and for the source and drain resistors if they have nonzero resistance values. The function sums all these losses and provides the power loss value for all of the blocks as functions of time.

If `node` is the name of the simulation log variable, then the table contains the data for all the blocks in the model that dissipate power (that is, contain at least one `power_dissipated` variable). If `node` is the name of a node in the simulation data tree, then the table contains the data only for the blocks within that node.

`lossesCell = elec_getPowerLossTimeSeries(node,startTime,endTime)` calculates dissipated power losses and returns the time series data for time steps from `startTime` to `endTime`. If `startTime` is equal to `endTime`, the interval is effectively zero and the function returns the instantaneous power for the time step that occurs at that moment. If you omit these two input arguments, the function returns data over the whole simulation time.

`lossesCell = elec_getPowerLossTimeSeries(node,startTime,... endTime,intervalWidth)` calculates dissipated power losses and returns the time series data for time steps from `startTime` to `endTime`, averaged over the time `intervalWidth`. If you omit the `intervalWidth`, or set it to 0, the function returns the instantaneous data, without averaging. If you omit all three optional arguments, the function returns the instantaneous data over the whole simulation time.

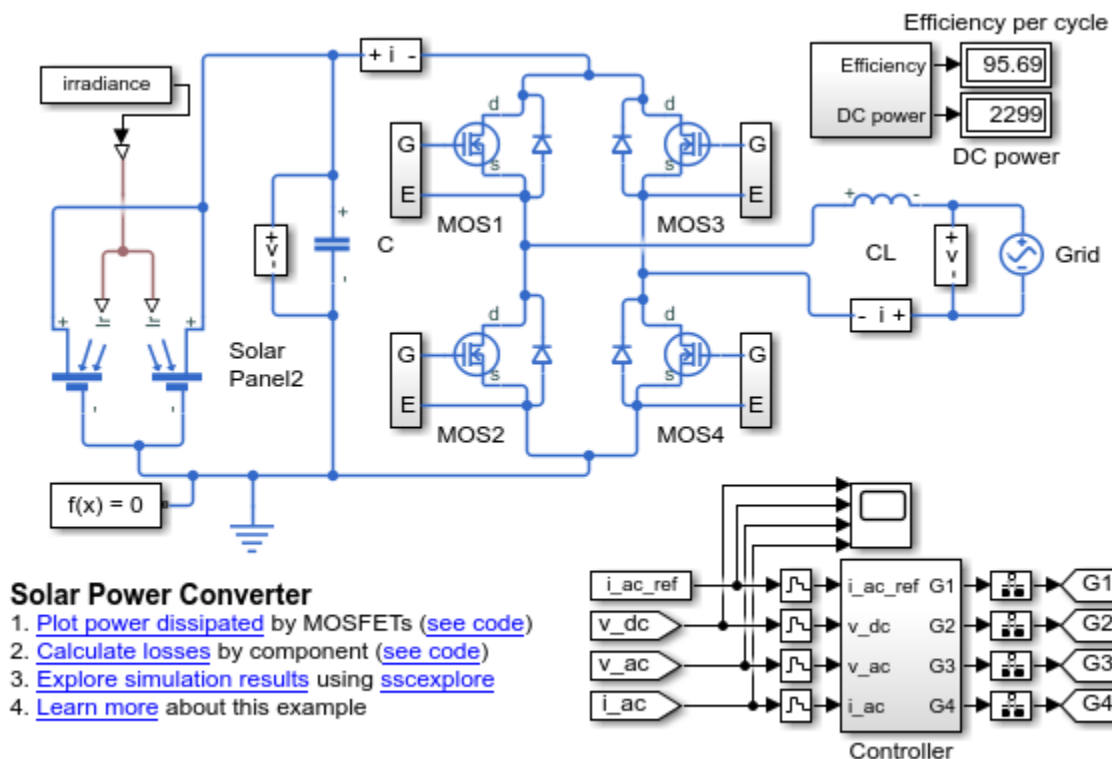
### Examples

## Calculate Dissipated Power Losses for the Entire Simulation Time

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in the entire simulation time using the `elec_getPowerLossTimeSeries` function.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

Calculate dissipated power losses and return the time series data in cell array.

```
lossesCell = elec_getPowerLossTimeSeries(simlog_ee_solar_inverter)
```

```
lossesCell =
```

```
8×2 cell array
```

```
'ee_solar_inverter.Diode1.diode' [201804×3 double]
```



```

'ee_solar_inverter.MOS1'      [201804x3 double]
'ee_solar_inverter.MOS2'      [201804x3 double]
'ee_solar_inverter.MOS3'      [201804x3 double]
'ee_solar_inverter.MOS4'      [201804x3 double]
'ee_solar_inverter.Diode2.diode' [201804x3 double]
'ee_solar_inverter.Diode3.diode' [201804x3 double]
'ee_solar_inverter.Diode4.diode' [201804x3 double]

```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `201804x3 double` numeric array for the `ee_solar_inverter.Diode1.diode`.

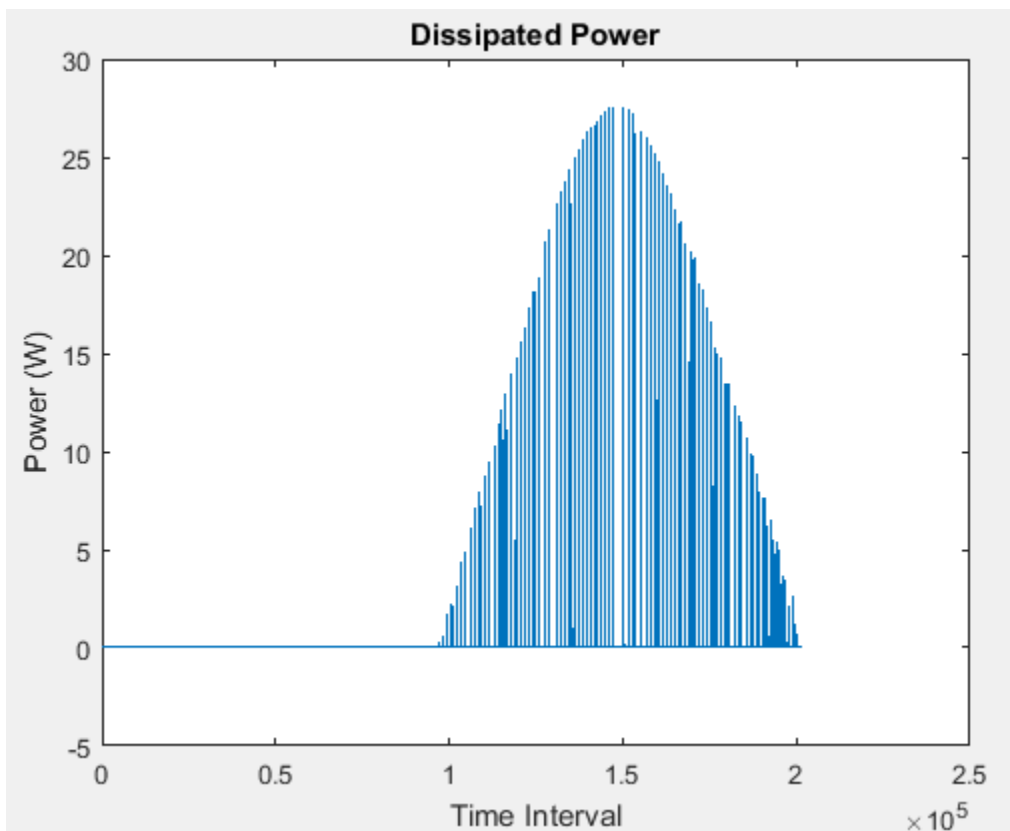
The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```

plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')

```



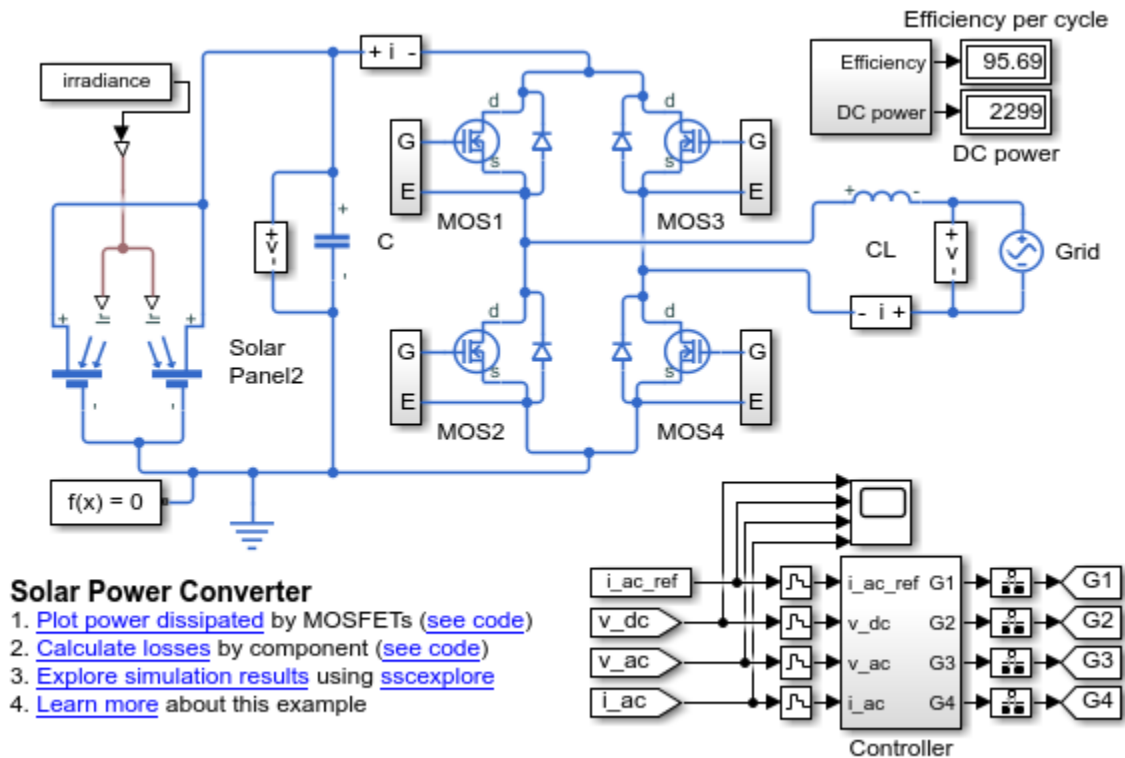
### Calculate Dissipated Power Losses for a Specific Time Period

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in a specific time period using the `elec_getPowerLossTimeSeries` function.



Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

The model simulation time ( $t$ ) is 1/60 seconds. Calculate dissipated power losses and return the time series data in cell array for the second half of the simulation cycle, when  $t$  is between 1/120 and 1/60 seconds..

```
lossesCell = elec_getPowerLossTimeSeries(simlog_ee_solar_inverter,1/120,1/60)
```

```
lossesCell =
```

```
8x2 cell array
```

```
'ee_solar_inverter.Diode1.diode' [105197x3 double]
'ee_solar_inverter.MOS1' [105197x3 double]
'ee_solar_inverter.MOS2' [105197x3 double]
'ee_solar_inverter.MOS3' [105197x3 double]
'ee_solar_inverter.MOS4' [105197x3 double]
'ee_solar_inverter.Diode2.diode' [105197x3 double]
```

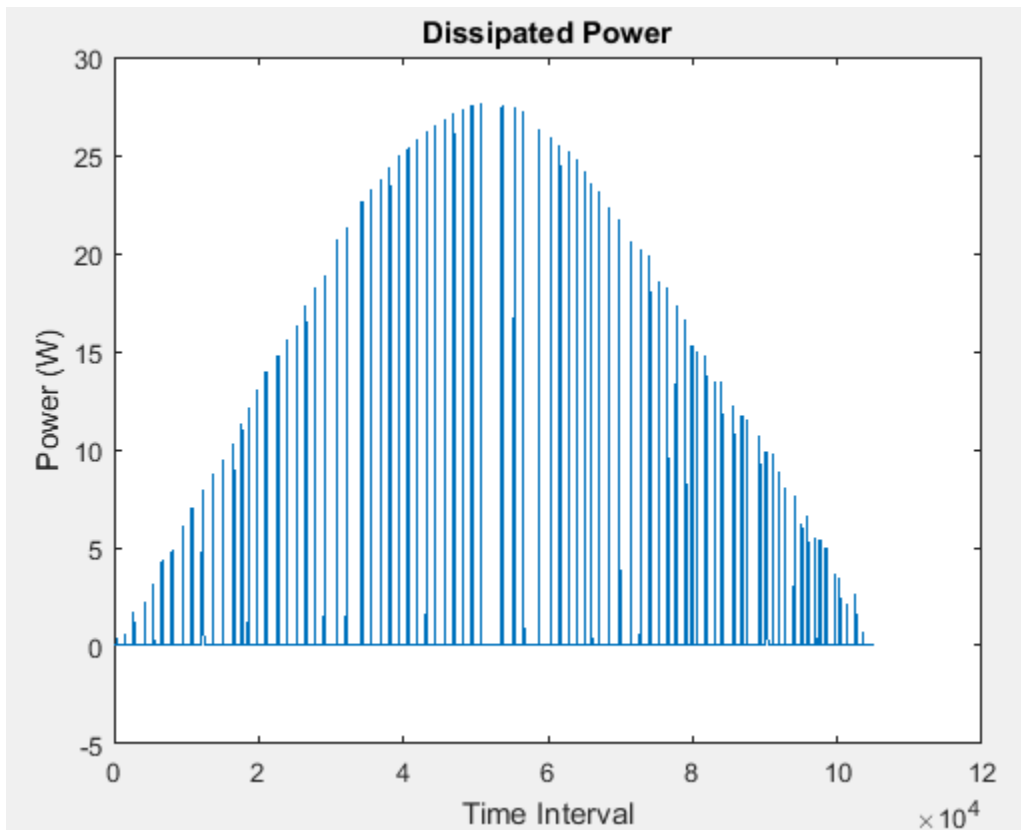
```
'ee_solar_inverter.Diode3.diode' [105197x3 double]
'ee_solar_inverter.Diode4.diode' [105197x3 double]
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `105197x3 double` numeric array for the `ee_solar_inverter.Diode1.diode`.

The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```

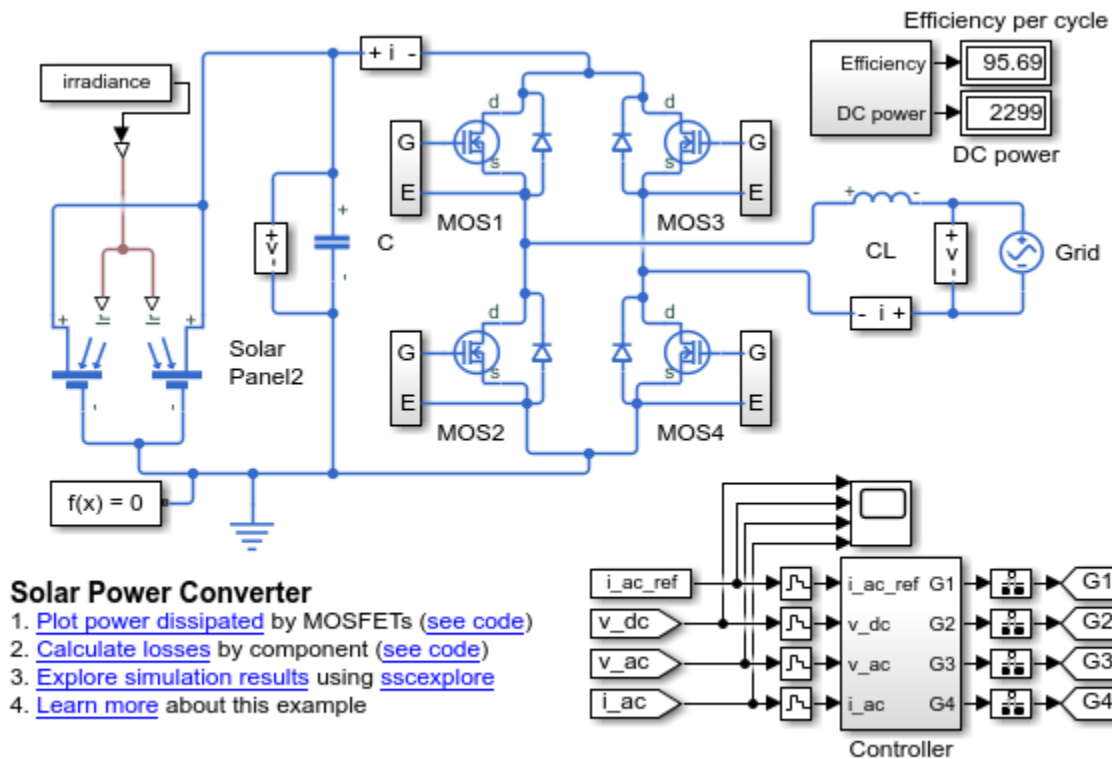


### Calculate Dissipated Power Losses Using Specific Interval Widths

This example shows how to calculate losses based on the power dissipated and return the time series data for a specific time period with averaging applied over intervals of a specified width.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_solar_inverter';
open_system(model)
```



This example model has data logging enabled. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_solar_inverter` is saved in your current workspace.

The model simulation time ( $t$ ) is 1/60 seconds. Calculate the average dissipated power losses for 1.1e-4 s intervals and return the time series data in cell array for the period when simulation time,  $t$ , is 0.008-0.017 seconds.

```
lossesCell = elec_getPowerLossTimeSeries(simlog_ee_solar_inverter,0.008,0.016,1.1e-4)
```

```
lossesCell =
```

```
8×2 cell array
```

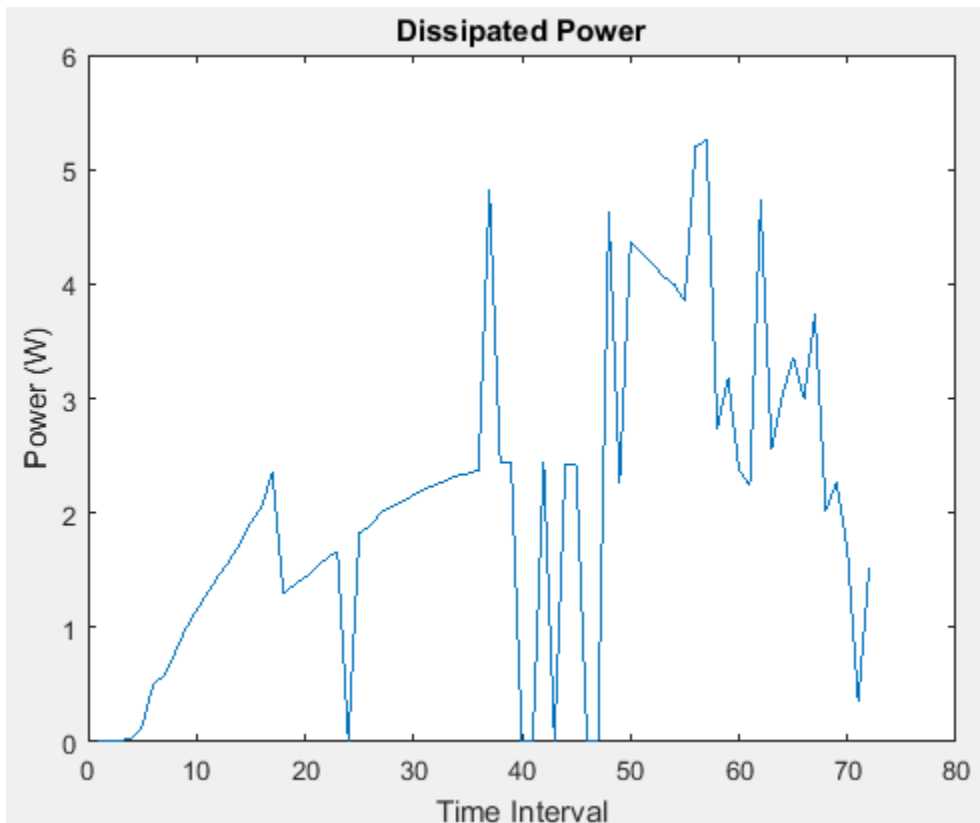
```
'ee_solar_inverter.Diode1.diode' [72×3 double]
'ee_solar_inverter.MOS1' [72×3 double]
'ee_solar_inverter.MOS2' [72×3 double]
'ee_solar_inverter.MOS3' [72×3 double]
'ee_solar_inverter.MOS4' [72×3 double]
'ee_solar_inverter.Diode2.diode' [72×3 double]
'ee_solar_inverter.Diode3.diode' [72×3 double]
'ee_solar_inverter.Diode4.diode' [72×3 double]
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the 105197×3 double numeric array for the `ee_solar_inverter.Diode1.diode`.

The first two columns contain the interval start and end time. The third column contains the power loss data. In this case, to use averaging intervals that are equal in width to  $1.1 \times 10^{-4}$  seconds, the function adjusts the start time for the first interval from the specified value of 0.008 seconds to a value of 0.0081 seconds. There are 72 intervals of  $1.1 \times 10^{-4}$  seconds.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```



## Input Arguments

**node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If **node** is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If **node** is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog.Cell1.MOS1`

**startTime** — Start of the time interval for calculating the data

0 (default) | real number

Start of the time interval for calculating the power loss time series, specified as a real number, in seconds. **startTime** must be greater than or equal to the simulation **Start time** and less than **endTime**.

Data Types: double

**endTime** — End of the time interval for calculating the data

simulation stop time (default) | real number

End of the time interval for calculating the power loss time series, specified as a real number, in seconds. **endTime** must be greater than **startTime** and less than or equal to the simulation **Stop time**.

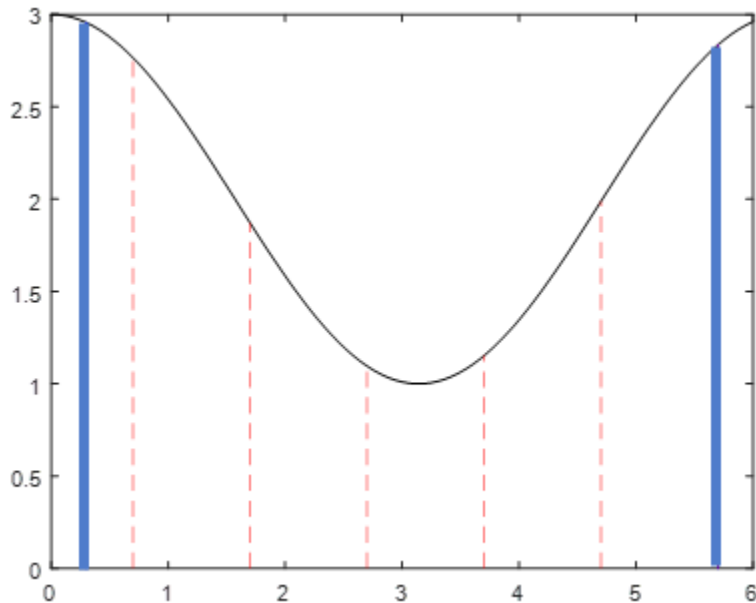
Data Types: double

**intervalWidth** — Size of the time interval for calculating the average power dissipation

0 (default) | real number

Size of the time interval for calculating the average power dissipation, specified as a real number, in seconds. If specified, the function returns data for time steps from **startTime** to **endTime**, averaged over the **intervalWidth**. If you omit the **intervalWidth** argument, or set it to 0, the function returns the instantaneous data, without averaging. If all the optional arguments are omitted, the function returns the instantaneous data over the whole simulation time.

If the time between the specified **startTime** and **endTime** is not an integer multiple of **intervalWidth**, the function adjusts the start time. The figure shows how the function adjusts the start time to ensure that width of each time interval that the dissipated power is averaged over is equal to the specified **intervalWidth**.



The black line is an example of the instantaneous *power\_dissipated* variables summed over all elements in an individual block. The simulation runs for 6 seconds. The `startTime` and `endTime` are indicated by the solid blue lines. The `intervalWidth` is set to 1 second. There are five intervals as indicated by the red dashed lines. The right-most edge of the last interval coincides with `endTime`. The left-most edge of the first interval is always greater than or equal to `startTime`. The edge is equal to `startTime` only if  $(\text{endTime} - \text{startTime}) / \text{intervalWidth}$  is an integer. The output in this case consists of five values for the averaged power dissipation, one point for each time period. The function outputs the actual start and stop times in the tabulated output data.

Example:  $1.1\text{e-}3$

Data Types: `double`

## Output Arguments

### **lossesCell** — Time series of the dissipated power losses for each block

cell array

Cell array that contains the names of the blocks in the nodes that contain *power\_dissipated* variables and, for each block, a three-column array:

- Column one contains the interval start time.
- Column two contains the interval end time.
- Column three contains the dissipated power for the time interval.

If the interval width is 0 seconds, that is, the start time is equal to the end time, then the dissipated power is the instantaneous power loss. If the interval is greater than 0 seconds, the dissipated power is the average power loss for the time of the interval.

## Version History

Introduced in R2017a

### **elec\_getPowerLossTimeSeries will be removed**

*Not recommended starting in R2019a*

The `elec_getPowerLossTimeSeries` function will be removed in a future release. Use the `ee_getPowerLossTimeSeries` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

### **See Also**

`ee_getEfficiency` | `ee_getPowerLossSummary` | `ee_getPowerLossTimeSeries` | `sscexplore`

### **Topics**

“About Simulation Data Logging”

“About the Simscape Results Explorer”

## pe\_calculateThdPercent

Compute the total harmonic distortion (THD) percentage

### Syntax

```
[thdPercent] = pe_calculateThdPercent(harmonicOrder,...
harmonicMagnitude)
```

### Description

[thdPercent] = pe\_calculateThdPercent(harmonicOrder,... harmonicMagnitude) calculates the total harmonic distortion (THD) percentage using these equations:

$$M = \frac{\text{harmonic magnitude}}{\sqrt{2}},$$

and

$$\%THD = 100 \frac{\sqrt{\sum_{i=2}^n M_i^2}}{M_1},$$

where:

- $M_i$  is the root mean square (RMS) value of the harmonic magnitude corresponding to the  $i^{\text{th}}$  harmonic order.
- $M$  is  $V_{RMS}$  or  $I_{RMS}$  as required.

You can use the ee\_getHarmonics function to obtain the vectors of harmonic order and harmonic magnitude for a simscape\_logging.Node.

### Examples

#### Calculate THD percent

Calculate the THD from harmonic orders [1;5;7;11;13] and harmonic magnitudes [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;0.0127e+03].

```
harmonicOrder = [1;5;7;11;13];
harmonicMagnitude = [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;...
0.0127e+03];
thdPercent = pe_calculateThdPercent( harmonicOrder, harmonicMagnitude )
thdPercent = 4.5480
```

### Input Arguments

**harmonicOrder** — Harmonic orders

vector



Harmonic orders from 0 up to and including number of harmonics, specified as a vector.

Example: [1;5;7;11;13]

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **harmonicMagnitude — Harmonic magnitudes**

vector

Harmonic magnitudes from the 0<sup>th</sup> harmonic up to and including the number of harmonics included in the analysis, specified as a vector.

Example: [1.1756e+03;0.0437e+03;0.0221e+03;0.0173e+03;0.0127e+03]

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

## **Version History**

**Introduced in R2014a**

### **pe\_calculateThdPercent will be removed**

*Not recommended starting in R2019a*

The `pe_calculateThdPercent` function will be removed in a future release. Use the `ee_calculateThdPercent` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## **See Also**

### **Blocks**

Spectrum Analyzer

### **Functions**

`ee_calculateThdPercent` | `ee_getHarmonics` | `ee_plotHarmonics` | `sscexplore`

### **Objects**

`simscape.logging.Node`

### **Topics**

“Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block”

“Choose a Simscape Electrical Function for an Offline Harmonic Analysis”

“Data Logging”

“Harmonic Analysis of a Three-Phase Rectifier”

## pe\_getEfficiency

Calculate efficiency as a function of dissipated power losses

### Syntax

```
efficiency = pe_getEfficiency('loadIdentifier',node)
efficiency = pe_getEfficiency('loadIdentifier',node,...
startTime,endTime)
[efficiency,lossesTable] = pe_getEfficiency('loadIdentifier',node)
```

### Description

`efficiency = pe_getEfficiency('loadIdentifier',node)` returns the efficiency of a circuit based on the data extracted from a Simscape logging node.

Before you call this function, generate or load the simulation log variable to your workspace. To generate the variable, simulate the model with simulation data logging enabled. For more information, see “About Simulation Data Logging”. To load a previously saved variable from a file, right-click on the file and select **Load**.

Checking efficiency allows you to determine if circuit components are operating within their requirements. Blocks in the Semiconductor > Fundamental Components library and the Delta-Connected Load, Wye-Connected Load, and RLC (Three-Phase) blocks have an internal block variable called *power\_dissipated*. This variable represents the instantaneous dissipated power, which includes only the real power (not the reactive or apparent power) that the block dissipates. When you log simulation data, the time-value series for this variable represents the power dissipated by the block over time. You can view and plot this data using the Simscape Results Explorer. The `ee_getPowerLossTimeSeries` function also allows you to access this data.

The `pe_getEfficiency` function calculates the efficiency of the circuit based on the losses for blocks that have a *power\_dissipated* variable and that you identify as a load block. The equation for efficiency is

$$Eff = 100 * \frac{P_{load}}{P_{loss} + P_{load}}$$

where:

- *Eff* is the efficiency of the circuit.
- *P<sub>load</sub>* is the output power, that is, the power dissipated by load blocks.
- *P<sub>loss</sub>* is the power dissipated by nonload blocks.

This equation assumes that all loss mechanisms are captured by blocks containing at least one *power\_dissipated* variable. If the model contains any lossy blocks that do not have this variable, the efficiency calculation gives incorrect results.

Some blocks have more than one *power\_dissipated* variable, depending on their configuration. For example, for the MOSFET (Ideal, Switching) block, both the `diode` node and the `ideal_switch` node have a *power\_dissipated* logging node. The function sums the power losses for both nodes to

provide the total power loss for the block, averaged over simulation time. The function uses the loss data to calculate the efficiency of the circuit.

The nonideal semiconductor blocks also have thermal modeling options. Thermal modeling options have thermal ports that allow you to model the heat that is generated due to switching events and conduction losses. If you use a thermal modeling option, the function calculates power losses and efficiencies based on the thermal parameters that you specify. Essentially, the power dissipated is equal to the heat generated.

If you use a modeling option without a thermal port, the function calculates power losses and efficiencies based on the electrical parameters that you specify, such as on-state resistance and off-state conductance.

`efficiency = pe_getEfficiency('loadIdentifier',node,...  
startTime,endTime)` returns the efficiency of a circuit based on the *power\_dissipated* data extracted from a Simscape logging node within a time interval. `startTime` and `endTime` represent the start and end of the time interval for calculating the efficiency. If you omit these two input arguments, the function calculates the efficiency over the whole simulation time.

`[efficiency,lossesTable] = pe_getEfficiency('loadIdentifier',node)` returns the efficiency of a circuit and the power loss contributions of the nonload blocks in a circuit based on the data extracted from a Simscape logging node.

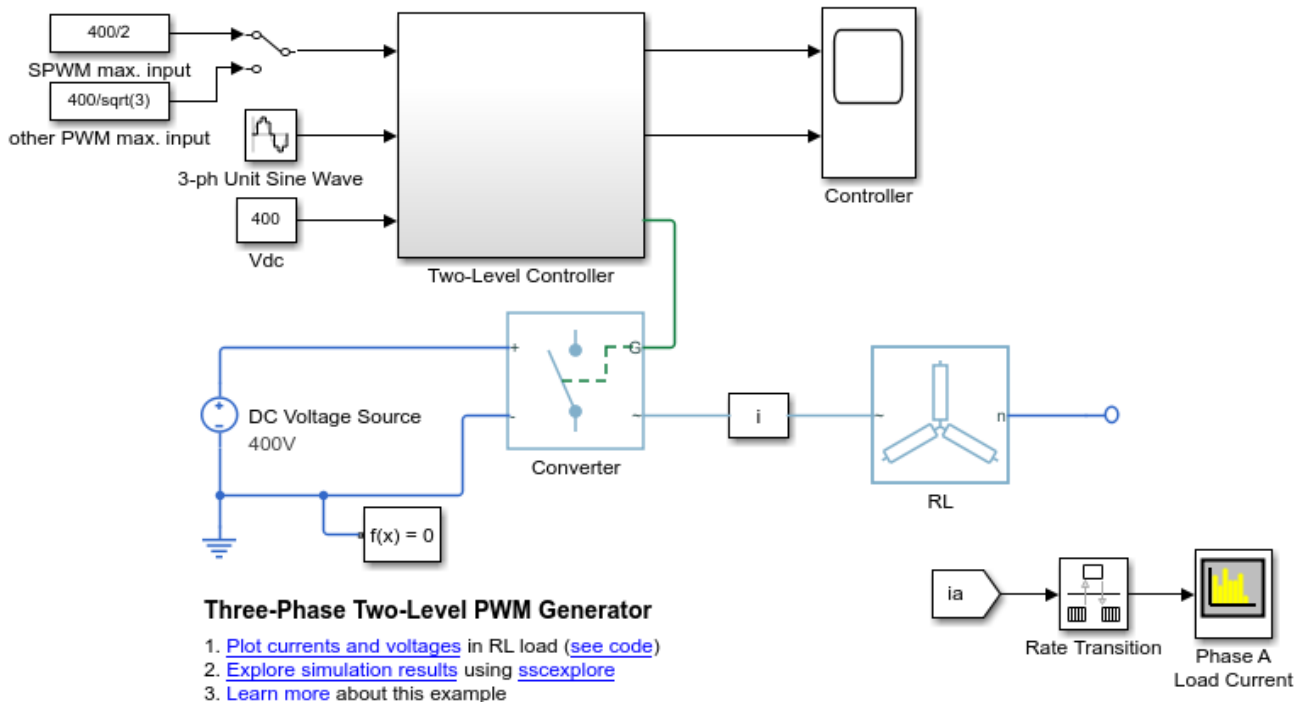
## Examples

### Calculate Efficiency for a Circuit

This example shows how to calculate efficiency based on the power dissipated by blocks in a circuit using the `pe_getEfficiency` function. Data logging is enabled locally, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_pwm_two_level';  
open_system(model)
```



Ensure that all blocks that have *power\_dissipated* variables are considered in the efficiency calculation. Enable data logging for the whole model.

```
set_param(model, 'SimscapeLogType', 'all')
```

Designate the load. Rename the Wye-Connected Load block from RL to RL\_Load.

```
set_param([model, '/RL'], 'Name', 'RL_Load')
```

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_pwm_two_level` is saved in the workspace.

Calculate the efficiency percentage.

```
efficiency = pe_getEfficiency('Load', simlog_ee_pwm_two_level)
```

```
efficiency =
```

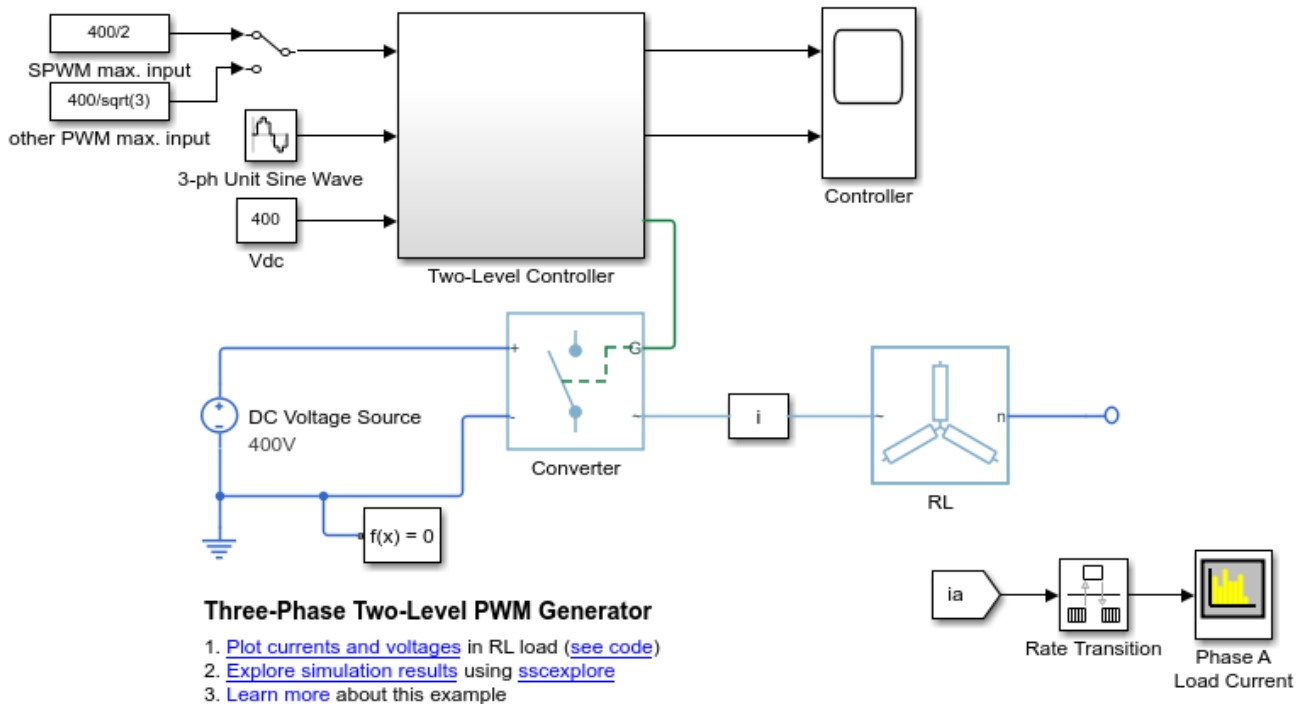
99.1940

### Calculate Efficiency of a Circuit for a Specific Time Period

This example shows how to calculate efficiency based on the power dissipated for a specific time period using the `pe_getEfficiency` function. Data logging is enabled locally, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_pwm_two_level';
open_system(model)
```



Ensure that all blocks that have *power\_dissipated* variables are considered in the efficiency calculation. Enable data logging for the whole model.

```
set_param(model, 'SimscapeLogType', 'all')
```

Designate the load. Rename the Wye-Connected Load block from RL to RL\_Load.

```
set_param([model, '/RL'], 'Name', 'RL_Load')
```

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable *simlog\_ee\_pwm\_two\_level* is saved in the workspace.

The model simulation stop time is 0.2 seconds. Calculate efficiency for the interval when the simulation time, *t*, is between 0.00 and 0.005 seconds.

```
efficiency = pe_getEfficiency('Load', simlog_ee_pwm_two_level, 0.000, 0.005)
```

```
efficiency =
```

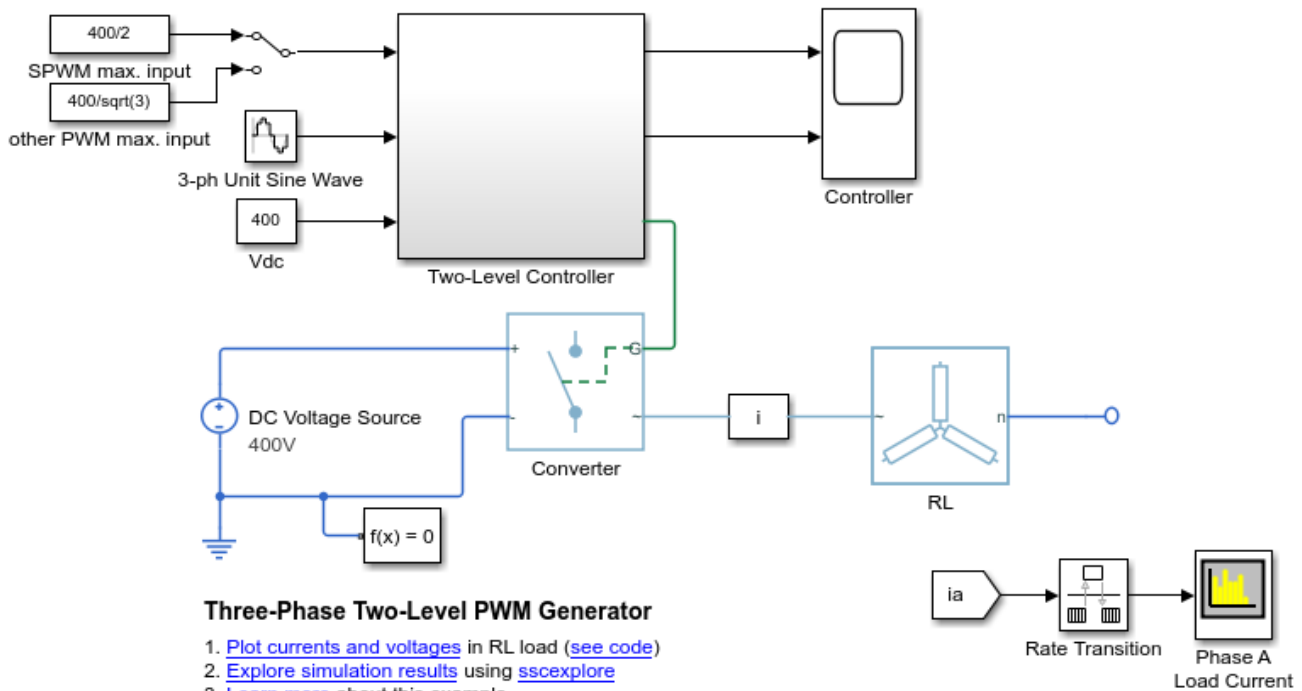
```
99.1093
```

## Calculate Efficiency and Power-Loss Contributions

This example shows how using the `pe_getEfficiency` function allows you to calculate both the efficiency of the circuit and the power-loss contributions of the nonload blocks based on the power that they dissipate. Data logging is enabled locally, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_pwm_two_level';
open_system(model)
```



Ensure that all blocks that have `power_dissipated` variables are considered in the efficiency calculation. Enable data logging for the whole model.

```
set_param(model, 'SimscapeLogType', 'all')
```

Designate the load. Rename the Wye-Connected Load block from RL to RL\_Load.

```
set_param([model, '/RL'], 'Name', 'RL_Load')
```

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_pwm_two_level` is saved in the workspace.

Calculate the efficiency and power-loss contributions due to dissipated power.

```
[efficiency, lossesTable] = pe_getEfficiency('Load', simlog_ee_pwm_two_level)
```

```
efficiency =
```

99.1940

lossesTable =

1×2 table

LoggingNode	Power
'ee_pwm_two_level.Converter.converter_Xabc'	268.73

## Input Arguments

### 'loadIdentifier' — Identify load blocks in the circuit

case-sensitive string

String that is a complete or partial match for the names of load blocks in the circuit. For example, consider a circuit that contains the blocks shown in the table.

Block Name in the Model		DC Impedance	AC Impedance	Y-Ld
<b>Block Type</b>		RLC (Three-Phase)	RLC (Three-Phase)	Wye-Connected Load
<b>Block Role in the Model</b>		Source Impedance	Load Impedance	Load
<b>'loadIdentifier'</b>	'd'	Yes	Yes	Yes
	'Load'	No	No	No
	'D'	Yes	No	No

The pe\_getEfficiency function does not return the correct data for any of these 'loadIdentifier' values.

A load-block naming schema that gives you better control over the output of the pe\_getEfficiency function is shown in this table.

Block Name in the Model		DC Impedance	AC Impedance_Load_1	Y-Load_2
<b>Block Type</b>		RLC (Three-Phase)	RLC (Three-Phase)	Wye-Connected Load
<b>Block Role in the Model</b>		Source Impedance	Load Impedance	Load
<b>'loadIdentifier'</b>	'1'	No	Yes	No
	'2'	No	No	Yes
	'Load'	No	Yes	Yes

Example: 'Load'

Data Types: string

**node — Simulation log variable, or a specific node within the simulation log variable**

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If **node** is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain *power\_dissipated* variables. If **node** is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog_ee_pwm_two_level`

**startTime — Start of the time interval for calculating the efficiency**

0 (default) | real number

Start of the time interval for calculating the efficiency, specified as a real number, in seconds. **startTime** must be greater than or equal to the simulation **Start time** and less than **endTime**.

Data Types: `double`

**endTime — End of the time interval for calculating the efficiency**

simulation stop time (default) | real number

End of the time interval for calculating the efficiency, specified as a real number, in seconds. **endTime** must be greater than **startTime** and less than or equal to the simulation **Stop time**.

Data Types: `double`

**Output Arguments****efficiency — Efficiency of the circuit**

percentage

Efficiency of the circuit based on data extracted from a Simscape logging node.

**lossesTable — Dissipated power for each nonload blocks**

table

Dissipated power losses for each nonload block, returned as a table. The first column lists logging nodes for all blocks that have at least one *power\_dissipated* variable. The second column lists the corresponding losses in watts.

**Assumptions**

- The output power equals the total power dissipated by blocks that you identify as load blocks.
- The input power equals the output power plus the total power dissipated by blocks that you do not identify as load blocks.



- The *power\_dissipated* variables capture all loss contributions.

## Version History

Introduced in R2017a

### **pe\_getEfficiency will be removed**

*Not recommended starting in R2019a*

The `pe_getEfficiency` function will be removed in a future release. Use the `ee_getEfficiency` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

### **See Also**

`ee_getEfficiency` | `ee_getPowerLossSummary` | `ee_getPowerLossTimeSeries` | `sscexplore`

### **Topics**

“Perform a Power-Loss Analysis”

“Data Logging”

“About the Simscape Results Explorer”

## pe\_getHarmonics

Return harmonic orders, magnitudes, and fundamental frequency

### Syntax

```
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest)
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest,nHarmonic)
```

### Description

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 pe\_getHarmonics(loggingNode) calculates the harmonic orders, magnitudes, and fundamental frequency of a `simscape.logging.Node` of an AC or periodic variable.

The function finds the points in the  $i^{\text{th}}$  signal (`valueIdx`) where the Simscape log crosses a threshold (`offsetOfInterest`). It uses the crossing points to find the required number of periods (`nPeriodOfInterest`) preceding the specified time (`tOfInterest`). Then it inputs the down-selected data to the Goertzel algorithm, which calculates the harmonic magnitudes up to and including the required number of harmonics (`nHarmonic`).

You enter the input arguments in a specific order. The Simscape logging node input argument is required. All other input arguments are optional and have default values. If you are specifying a value for a subsequent optional input argument, enter `[]` to use the default value for an optional input argument.

You can use the `pe_plotHarmonics` function to obtain a bar chart from the same input arguments. You can use the outputs of this function as inputs to the `pe_calculateThdPercent` function to calculate the total harmonic distortion (THD) percentage.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 pe\_getHarmonics(loggingNode,valueIdx) uses the index into value data.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 pe\_getHarmonics(loggingNode,valueIdx,tOfInterest) uses the simulation time.

[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...  
 pe\_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest) uses the number of periods of fundamental frequency.

```
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest) uses the DC offset.
```

```
[harmonicOrder,harmonicMagnitude,fundamentalFrequency] =...
pe_getHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest,nHarmonic) uses the number of harmonics.
```

## Examples

### Analyze Using Default Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.AC.N.V`, which contains data from a three-phase voltage. The function analyzes the default signal, which is the first, or a-phase, signal at the final simulation time. The function uses the default values of 12 for the number of periods of the signal, 0V for the signal bias, and 30 for the number of harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
pe_getHarmonics(simlog_ee_harmonics_rectifier.AC.N.V)
```

ans =

Columns 1 through 9

0 1 2 3 4 5 6 7 8

Columns 10 through 18

9 10 11 12 13 14 15 16 17

Columns 19 through 27

18 19 20 21 22 23 24 25 26

Columns 28 through 31

27 28 29 30

### Analyze Using Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.AC.N.V`, which contains data from a three-phase voltage. The function analyzes the second, or b-phase, signal at a simulation time of 0.5 s. The function uses 10 periods of the signal, which has a bias of 1 V. The function analyzes 15 harmonics.

```
open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
pe_getHarmonics(simlog_ee_harmonics_rectifier.AC.N.V,2,0.5,10,1,15)
```

ans =

Columns 1 through 9

0 1 2 3 4 5 6 7 8

Columns 10 through 16

```

9      10      11      12      13      14      15

```

### Analyze Using Default and Specified Values

This set of function arguments uses the Simscape logging node `simlog_ee_harmonics_rectifier.AC.N.V`, which contains data from a three-phase voltage. The function analyzes the first, or a-phase, signal at a simulation time of 0.5 s. The function uses 12 periods of the signal, which has a bias of 1 V. The function analyzes the default number, 30, of harmonics.

```

open_system('ee_harmonics_rectifier')
sim('ee_harmonics_rectifier')
pe_getHarmonics(simlog_ee_harmonics_rectifier.AC.N.V,[],0.5,[],1)

```

```
ans =
```

```
Columns 1 through 9
```

```

0      1      2      3      4      5      6      7      8

```

```
Columns 10 through 18
```

```

9      10     11     12     13     14     15     16     17

```

```
Columns 19 through 27
```

```

18     19     20     21     22     23     24     25     26

```

```
Columns 28 through 31
```

```

27     28     29     30

```

## Input Arguments

### loggingNode — Simscape logging node

1-by-1 `simscape.logging.Node`

Simscape logging node, specified as a 1-by-1 `simscape.logging.Node`. You create a `simscape.logging.Node` by running a simulation with Simscape logging enabled. For information, see “Enable Data Logging for the Whole Model”.

Example: `simlog.Load.V`

The Simscape logging node `simlog.Load.V` contains data from a three-phase voltage.

### valueIdx — Index into value data

1 (default) | scalar

Index into value data, specified as a scalar. Specifies the  $i^{\text{th}}$  variable of interest in the Simscape log.

Example: 2

Specify the b-phase, which is the second signal from a three-phase voltage.

Example: `[]`

Use [] to specify the default value of 1. The a-phase, which is the first signal from a three-phase voltage, is the default signal of interest.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **tOfInterest – Simulation time**

final time in Simscape log (default) | scalar

Simulation time of interest for harmonic analysis, specified as a scalar.

Example: 0.5

Specify a 0.5 s simulation time.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **nPeriodOfInterest – Number of periods**

12 (default) | scalar

Number of periods of fundamental frequency to be included in harmonic analysis, specified as a scalar.

Example: 10

Specify 10 periods of the signal.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **offsetOfInterest – DC offset**

0 (default) | scalar

DC offset in the input signal, specified as a scalar. The function uses this value to find the periods of interest.

Example: 1

Specify a bias of 1 V for the signal.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

### **nHarmonic – Number of harmonics**

30 (default) | scalar

Number of harmonics to include in analysis, specified as a scalar.

Example: 15

Specify that the number of harmonics to be analyzed is 15.

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

## **Output Arguments**

### **harmonicOrder – Harmonic order**

vector

Harmonic orders from 0 up to and including the number of harmonics used in the analysis, returned as a vector.

**harmonicMagnitude — Harmonic magnitude**

vector

Harmonic magnitudes from the 0<sup>th</sup> harmonic up to and including the number of harmonics used in the analysis, returned as a vector.

**fundamentalFrequency — Fundamental frequency**

scalar

Fundamental frequency over the range of the down-selected input data, returned as a scalar.

**Limitations**

- This function requires that you use a fixed-step solver for the Simscape Electrical Power Systems network that you are analyzing. To specify a fixed-step solver for the physical network, use one of the configuration combinations in the table.

Configuration Combination	Global Solver Configuration	Local Solver Configuration
Global variable-step with local fixed-step	Set <b>Type</b> to Variable-step	Enable the options to <b>Use local solver</b> and <b>Use fixed-cost runtime consistency iterations</b>
Global and local fixed-step	Set <b>Type</b> to Fixed-step	Enable the options to <b>Use local solver</b> and <b>Use fixed-cost runtime consistency iterations</b>
Global fixed-step	Set <b>Type</b> to Fixed-step	Clear the option to <b>Use local solver</b>

- This function uses threshold crossing points to determine the fundamental frequency of the data. If your input data is noisy or crosses the threshold more frequently than half of the fundamental period, filter it before you use this function to analyze it.
- This function requires a minimal number of periods. If the minimal number is not met, the function generates a warning message. To increase the number of periods, use one or both of these methods:
  - Increase the simulation time.
  - Increase the switching frequency.

**Version History**

Introduced in R2014a

**pe\_getHarmonics will be removed***Not recommended starting in R2019a*

The `pe_getHarmonics` function will be removed in a future release. Use the `ee_getHarmonics` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## See Also

### Blocks

Spectrum Analyzer

### Functions

ee\_calculateThdPercent | ee\_getHarmonics | ee\_plotHarmonics | ssexplore

### Objects

simscape.logging.Node

### Topics

“Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block”

“Choose a Simscape Electrical Function for an Offline Harmonic Analysis”

“Data Logging”

“Harmonic Analysis of a Three-Phase Rectifier”

## pe\_getPowerLossSummary

Calculate dissipated power losses

### Syntax

```
lossesTable = pe_getPowerLossSummary(node)  
lossesTable = pe_getPowerLossSummary(node, startTime, endTime)
```

### Description

`lossesTable = pe_getPowerLossSummary(node)` calculates dissipated power losses for semiconductor blocks in a model, based on logged simulation data, and returns the data for each block in a table.

Before you call this function, generate or load the simulation log variable into your workspace. To generate the variable, simulate the model with simulation data logging enabled. For more information, see “About Simulation Data Logging”. To load a previously saved variable from a file, right-click on the file and select **Load**.

Checking dissipated power allows you to determine if circuit components are operating within their efficiency requirements. Blocks in the **Semiconductor > Fundamental Components** library have an internal variable called *power\_dissipated*. This variable represents the instantaneous dissipated power, which includes only the real power (not the reactive or apparent power) that the block dissipates. When you log simulation data, the time-value series for this variable represents the power dissipated by the block over time. You can view and plot this data using the Simscape Results Explorer. The `ee_getPowerLossTimeSeries` function also allows you to access this data from a cell array.

The `pe_getPowerLossSummary` function calculates average losses for each block that has a *power\_dissipated* variable. Some blocks have more than one *power\_dissipated* variable, depending on their configuration. For example, for the MOSFET (Ideal, Switching) block, both the `diode` node and the `ideal_switch` node have a *power\_dissipated* logging node. The function sums the power losses for both nodes to provide the total power loss for the block, averaged over simulation time.

The nonideal semiconductor blocks also have thermal modeling options. Thermal modeling options have thermal ports that allow you to model the heat that is generated due to switching events and conduction losses. If you use a thermal modeling option, the function calculates power losses based on the thermal parameters that you specify. Essentially, the power dissipated is equal to the heat generated.

If you use a modeling option without a thermal port, the function calculates power losses based on the electrical parameters that you specify, such as on-state resistance and off-state conductance.

`lossesTable = pe_getPowerLossSummary(node, startTime, endTime)` calculates dissipated power losses within a time interval. `startTime` and `endTime` represent the start and end of the time interval for averaging the power losses. If you omit these two input arguments, the function averages the power losses over the total simulation time.



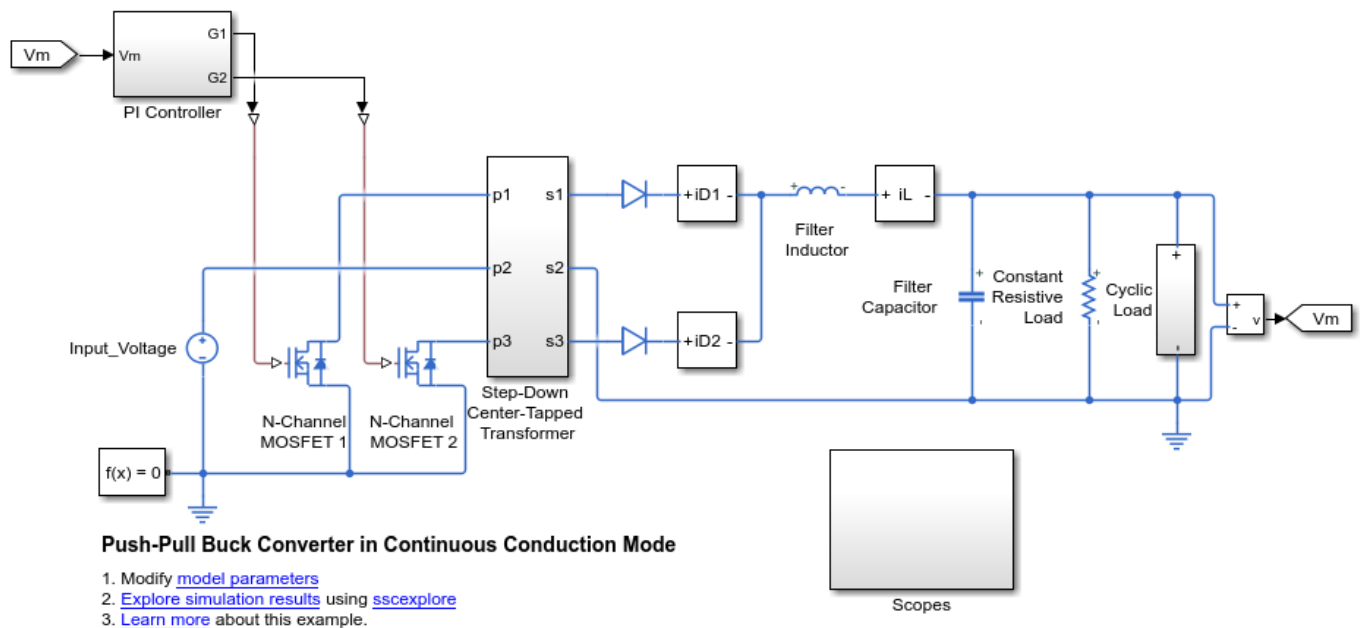
## Examples

### Calculate Average Power Losses by Block for the Whole Model

This example uses the Push-Pull Buck Converter in Continuous Conduction Mode model. Data logging is enabled for the whole model and the option to limit data points is off.

- 1 Open the example model. At the MATLAB command prompt, enter

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



- 2 Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_push_pull_converter_ccm` is saved in your workspace.

- 3 Calculate average power losses for each semiconductor in the model and display the results in a table.

```
tabulatedLosses = pe_getPowerLossSummary(simlog_ee_push_pull_converter_ccm)
```

```
tabulatedLosses =
```

```
5x2 table
```

LoggingNode	Power
'ee_push_pull_converter_ccm.Diode1'	0.30937
'ee_push_pull_converter_ccm.Diode'	0.3092
'ee_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation'	0.24497
'ee_push_pull_converter_ccm.N_Channel_MOSFET_2.mosfet_equation'	0.2449
'ee_push_pull_converter_ccm.Step_Down_Center_Tapped_Transformer.Eddy_Current'	0.070198

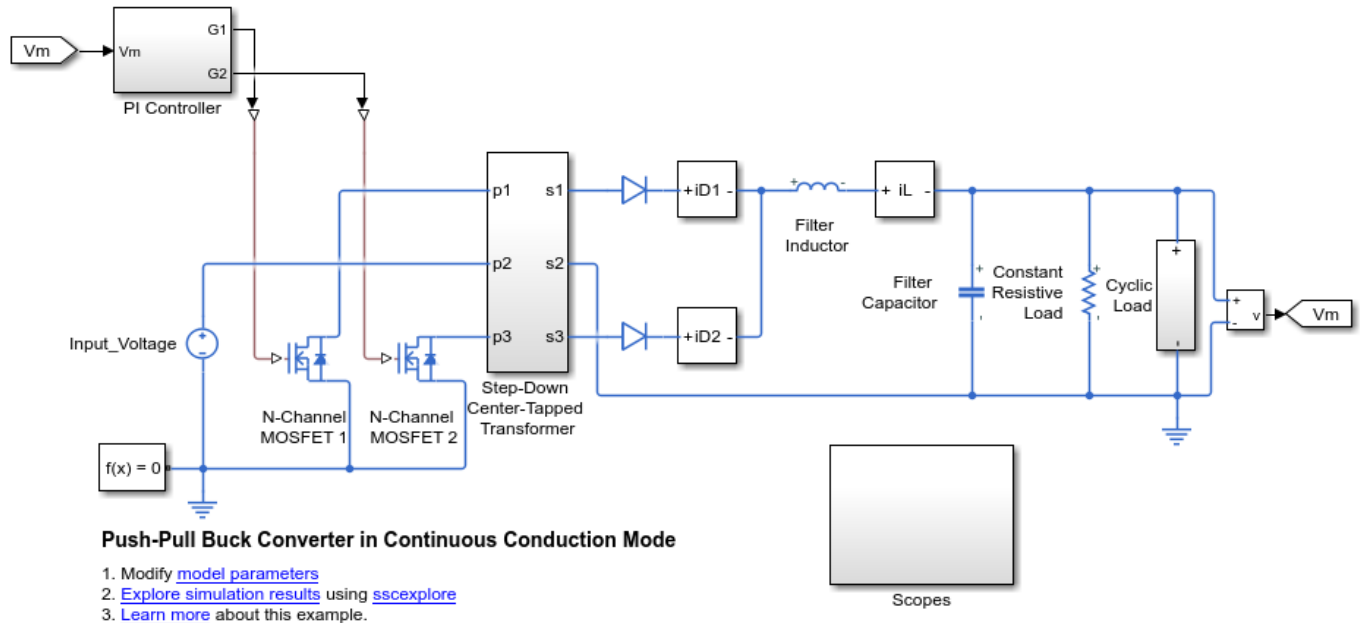
The table shows dissipated power losses for each of the N-Channel MOSFET and Diode blocks, averaged over the entire simulation time.

## Calculate Average Power Losses for a Specific Time Period

This example uses the Push-Pull Buck Converter in Continuous Conduction Mode model. Data logging is enabled for the whole model and the option to limit data points is off.

- 1 Open the example model. At the MATLAB command prompt, enter

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



- 2 Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_push_pull_converter_ccm` is saved in your workspace.

- 3 The model simulation time ( $t$ ) is 0.04 seconds. Calculate average power losses for the interval when  $t$  is 0.010–0.025 seconds

```
tabulatedLosses = pe_getPowerLossSummary(simlog_ee_push_pull_converter_ccm,0.010,0.025)
```

```
tabulatedLosses =
```

```
5×2 table
```

LoggingNode	Power
'ee_push_pull_converter_ccm.Diode'	0.3703
'ee_push_pull_converter_ccm.Diode1'	0.37022
'ee_push_pull_converter_ccm.N_Channel_MOSFET_2.mosfet_equation'	0.26598
'ee_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation'	0.26595
'ee_push_pull_converter_ccm.Step_Down_Center_Tapped_Transformer.Eddy_Current'	0.08443

The table shows dissipated power losses for each of the Diode and MOSFET blocks, averaged over the specified portion of simulation time.

## Input Arguments

### **node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If `node` is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If `node` is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog.Cell1.MOS1`

### **startTime** — Start of the time interval for averaging dissipated power losses

real number

Start of the time interval for averaging dissipated power losses, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than **endTime**.

Data Types: double

### **endTime** — End of the time interval for averaging dissipated power losses

real number

End of the time interval for averaging dissipated power losses, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

Data Types: double

## Output Arguments

### **lossesTable** — Dissipated power losses for each block

table

Dissipated power losses for each block, returned as a table. The first column lists logging nodes for all blocks that have at least one `power_dissipated` variable. The second column lists the corresponding losses in watts.

## Version History

**Introduced in R2017a**

### **pe\_getPowerLossSummary will be removed**

*Not recommended starting in R2019a*

The `pe_getPowerLossSummary` function will be removed in a future release. Use the `ee_getPowerLossSummary` function instead. The only difference between these functions is the

function name. To prevent your code from generating an error when the function is removed, update to the new function name.

### **See Also**

[ee\\_getEfficiency](#) | [ee\\_getPowerLossSummary](#) | [ee\\_getPowerLossTimeSeries](#) | [sscexplore](#)

### **Topics**

[“Perform a Power-Loss Analysis”](#)

[“Data Logging”](#)

[“About the Simscape Results Explorer”](#)

## pe\_getPowerLossTimeSeries

Calculate dissipated power losses and return the time series data

### Syntax

```
lossesCell = pe_getPowerLossTimeSeries(node)
lossesCell = pe_getPowerLossTimeSeries(node,startTime,endTime)
lossesCell = pe_getPowerLossTimeSeries(node,startTime,...
endTime,intervalWidth)
```

### Description

`lossesCell = pe_getPowerLossTimeSeries(node)` calculates dissipated power losses for blocks, based on logged Simscape simulation data, and returns the time series data for each block.

Before you call this function, generate or load the simulation log variable into your workspace. To generate the variable, simulate the model with simulation data logging enabled. For more information, see “About Simulation Data Logging”. To load a previously saved variable from a file, right-click on the file and select **Load**.

Checking dissipated power allows you to determine if circuit components are operating within their efficiency requirements. Blocks in the **Semiconductor > Fundamental Components** library have an internal variable called *power\_dissipated*. This variable represents the instantaneous dissipated power, which includes only the real power (not the reactive or apparent power) that the block dissipates. When you log simulation data, the time-value series for this variable represents the power dissipated by the block over time. You can view and plot this data using the Simscape Results Explorer. The `pe_getPowerLossTimeSeries` function also allows you to access this data from a cell array.

The `pe_getPowerLossTimeSeries` function calculates losses for each block that has a *power\_dissipated* variable. Some blocks have more than one *power\_dissipated* variable, depending on their configuration. For example, for the MOSFET (Ideal, Switching) block, both the `diode` node and the `ideal_switch` node have a *power\_dissipated* logging node. The function sums the power losses for both nodes to provide the total power loss for the block.

The nonideal semiconductor blocks also have thermal modeling options. Thermal modeling options have thermal ports that allow you to model the heat that is generated due to switching events and conduction losses. If you use a thermal modeling option, the function calculates power losses based on the thermal parameters that you specify. Essentially, the power dissipated is equal to the heat generated.

If you use a modeling option without a thermal port, the function calculates power losses based on the electrical parameters that you specify, such as on-state resistance and off-state conductance.

`lossesCell = pe_getPowerLossTimeSeries(node,startTime,endTime)` calculates dissipated power losses for blocks in a model, based on logged Simscape simulation data, and returns the time series data for each block for time steps from `startTime` to `endTime`. If `startTime` is equal to `endTime`, the interval is effectively zero and the function returns the instantaneous power for the time step that occurs at that moment.

`lossesCell = pe_getPowerLossTimeSeries(node, startTime, ...`  
`endTime, intervalWidth)` calculates dissipated power losses for blocks in a model, based on logged Simscape simulation data, and returns the time series data for each block for time steps from `startTime` to `endTime`, with averaging applied over intervals equal to `intervalWidth`. If `intervalWidth` is 0, the function returns the instantaneous power dissipation.

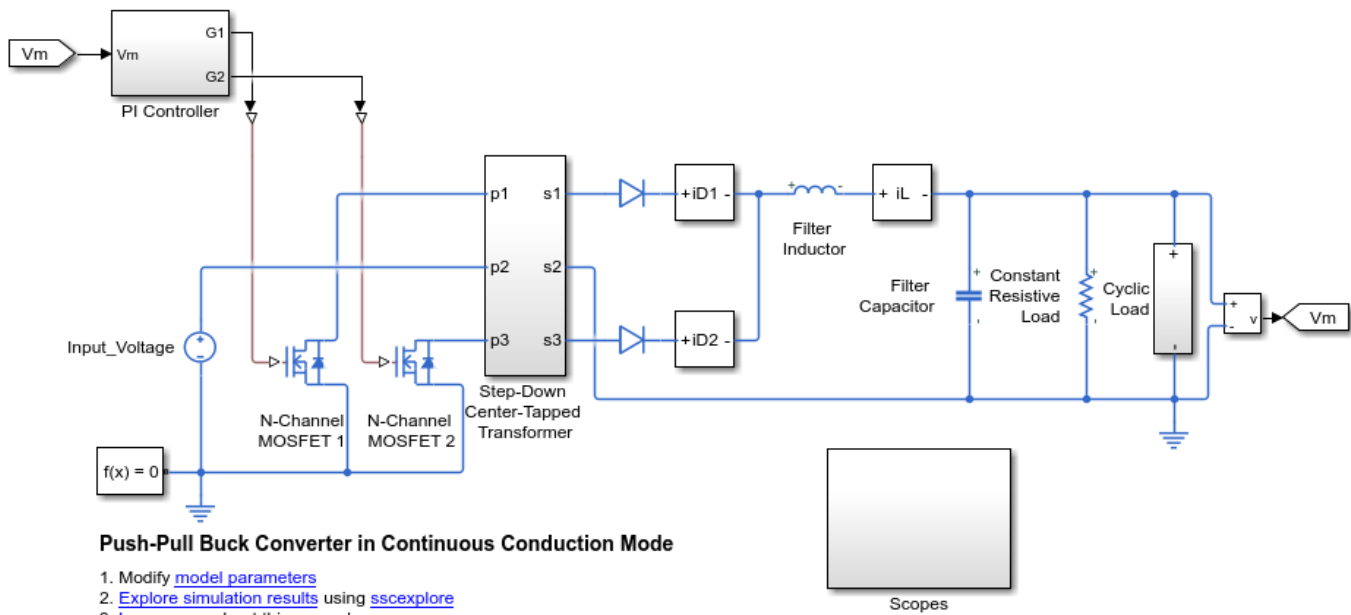
## Examples

### Calculate Dissipated Power Losses for the Entire Simulation Time

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in the entire simulation time using the `pe_getPowerLossTimeSeries` function. Data logging is enabled for the whole example model, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_push_pull_converter_ccm` is saved in your workspace.

Calculate dissipated power losses and return the time series data in a cell array.

```
lossesCell = pe_getPowerLossTimeSeries(simlog_ee_push_pull_converter_ccm)
lossesCell =
```

5×2 cell array

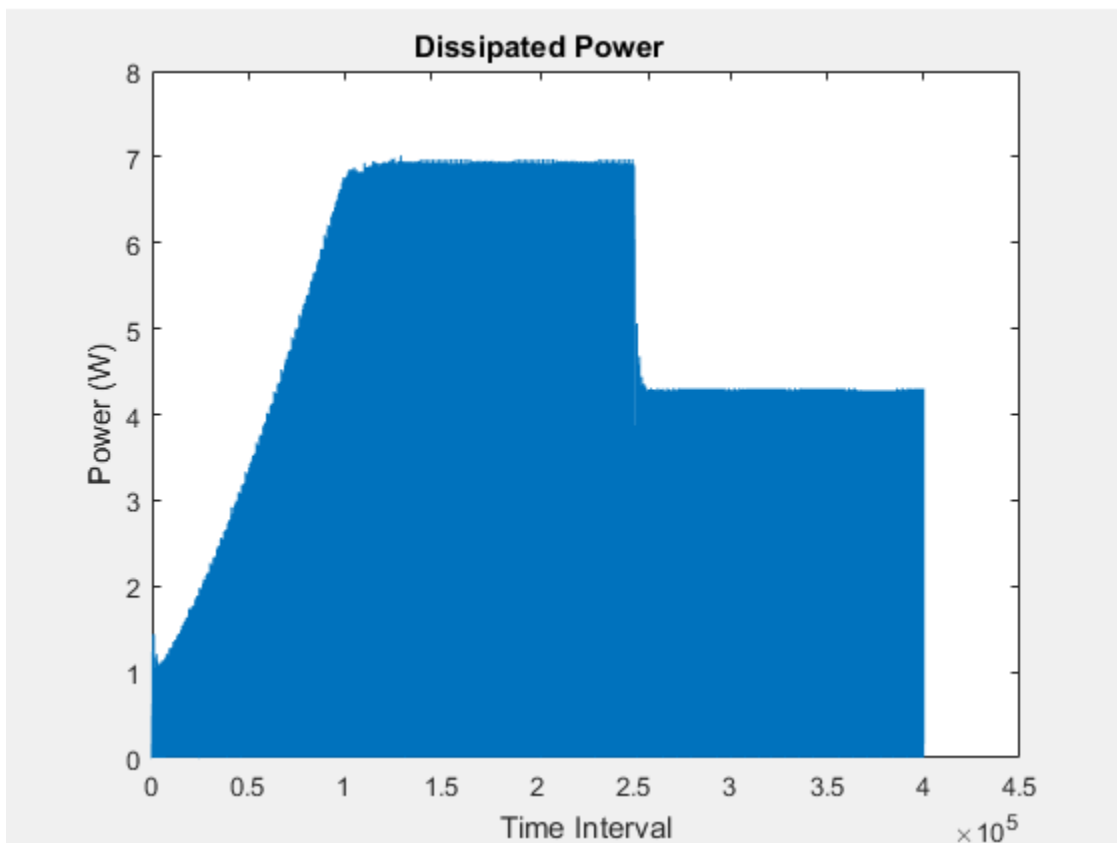
```
{'ee_push_pull_converter_ccm...'} {400001×3 double}
{'ee_push_pull_converter_ccm...'} {400001×3 double}
{'ee_push_pull_converter_ccm...'} {400001×3 double}
{'ee_push_pull_converter_ccm...'} {400001×3 double}
{'ee_push_pull_converter_ccm...'} {400001×3 double}
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `400001×3 double` numeric array for the `ee_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation`.

The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```



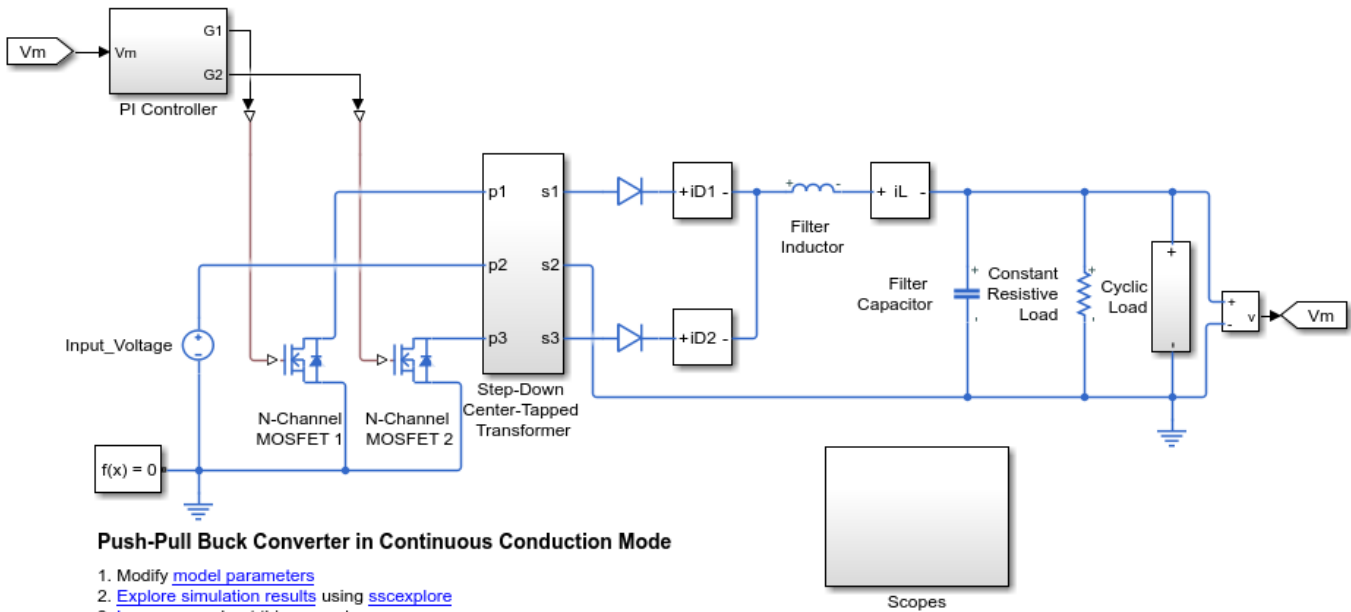
### Calculate Dissipated Power Losses for a Specific Time Period

This example shows how to calculate instantaneous losses based on the power dissipated and return the time series data for all time steps in a specific time period using the

`pe_getPowerLossTimeSeries` function. Data logging is enabled for the whole example model, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_push_pull_converter_ccm` is saved in your workspace.

The model simulation time ( $t$ ) is 0.04 seconds. Calculate dissipated power losses and return the time series data in a cell array for the interval when  $t$  is 0.010–0.025 seconds.

```
lossesCell = pe_getPowerLossTimeSeries(simlog_ee_push_pull_converter_ccm,0.010,0.025)
```

```
lossesCell =
```

5×2 cell array

```
{'ee_push_pull_converter_ccm...'} {150002×3 double}
{'ee_push_pull_converter_ccm...'} {150002×3 double}
{'ee_push_pull_converter_ccm...'} {150002×3 double}
{'ee_push_pull_converter_ccm...'} {150002×3 double}
{'ee_push_pull_converter_ccm...'} {150002×3 double}
```

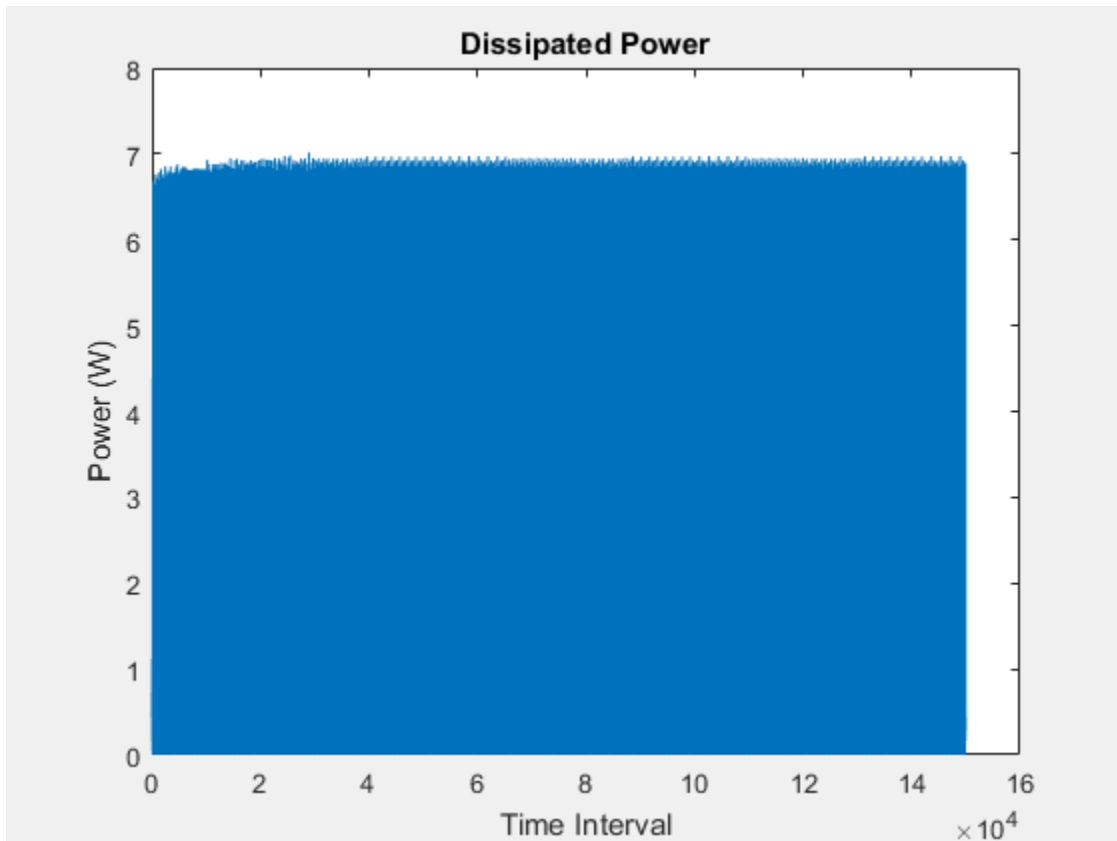
View the time series data. From the workspace, open the `lossesCell` cell array, then open the `150002×3 double` numeric array for the `ee_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation`.



The first two columns contain the interval start and end time. The third column contains the power loss data.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
xlabel('Time Interval')
ylabel('Power (W)')
```

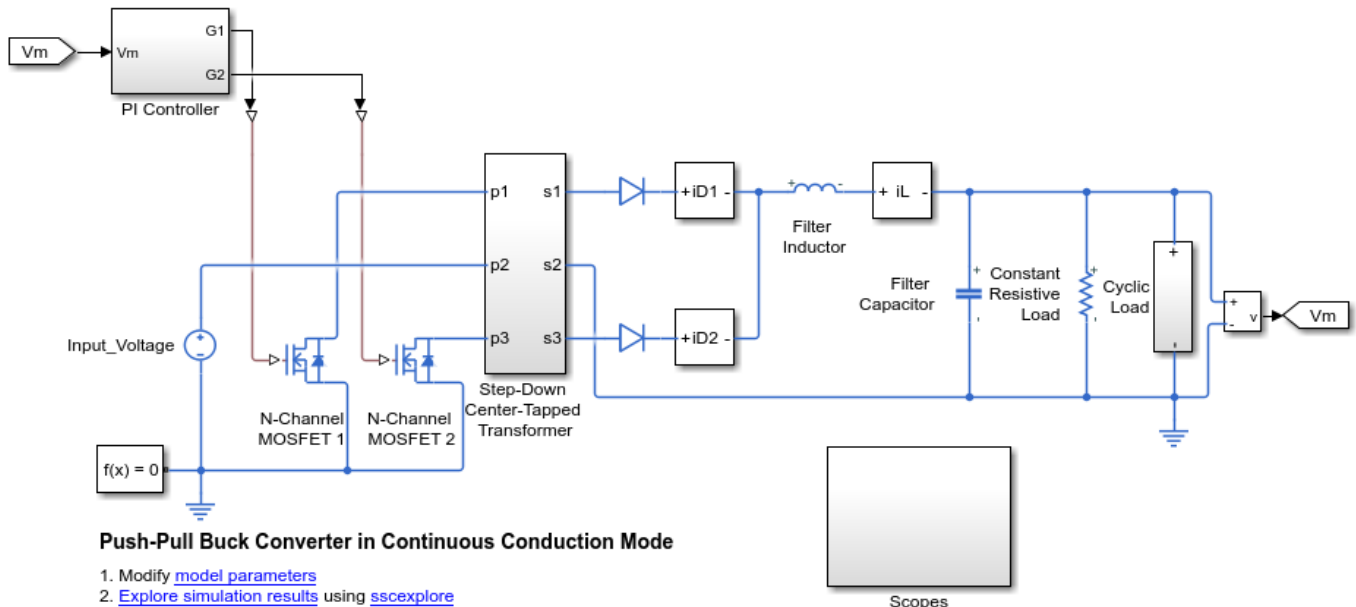


### Calculate Dissipated Power Losses Using Specific Interval Widths

This example shows how to calculate losses based on the power dissipated and return the time series data for a specific time period with averaging applied over intervals of a specified width. Data logging is enabled for the whole example model, and the option to limit data points is off.

Open the model. At the MATLAB® command prompt, enter:

```
model = 'ee_push_pull_converter_ccm';
open_system(model)
```



Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_ee_push_pull_converter_ccm` is saved in your workspace.

The model simulation time,  $t$ , is 0.04 seconds. Calculate the average dissipated power losses for 1.1e-4 s intervals and return the time series data in a cell array for the period when simulation time,  $t$ , is 0.010-0.025 seconds.

```
lossesCell = pe_getPowerLossTimeSeries(simlog_ee_push_pull_converter_ccm,0.010,0.025,1.1e-4)
```

```
lossesCell =
```

5x2 cell array

```
{'ee_push_pull_converter_ccm.N_...' } {136x3 double}
{'ee_push_pull_converter_ccm.N_...' } {136x3 double}
{'ee_push_pull_converter_ccm.Di...' } {136x3 double}
{'ee_push_pull_converter_ccm.Di...' } {136x3 double}
{'ee_push_pull_converter_ccm.St...' } {136x3 double}
```

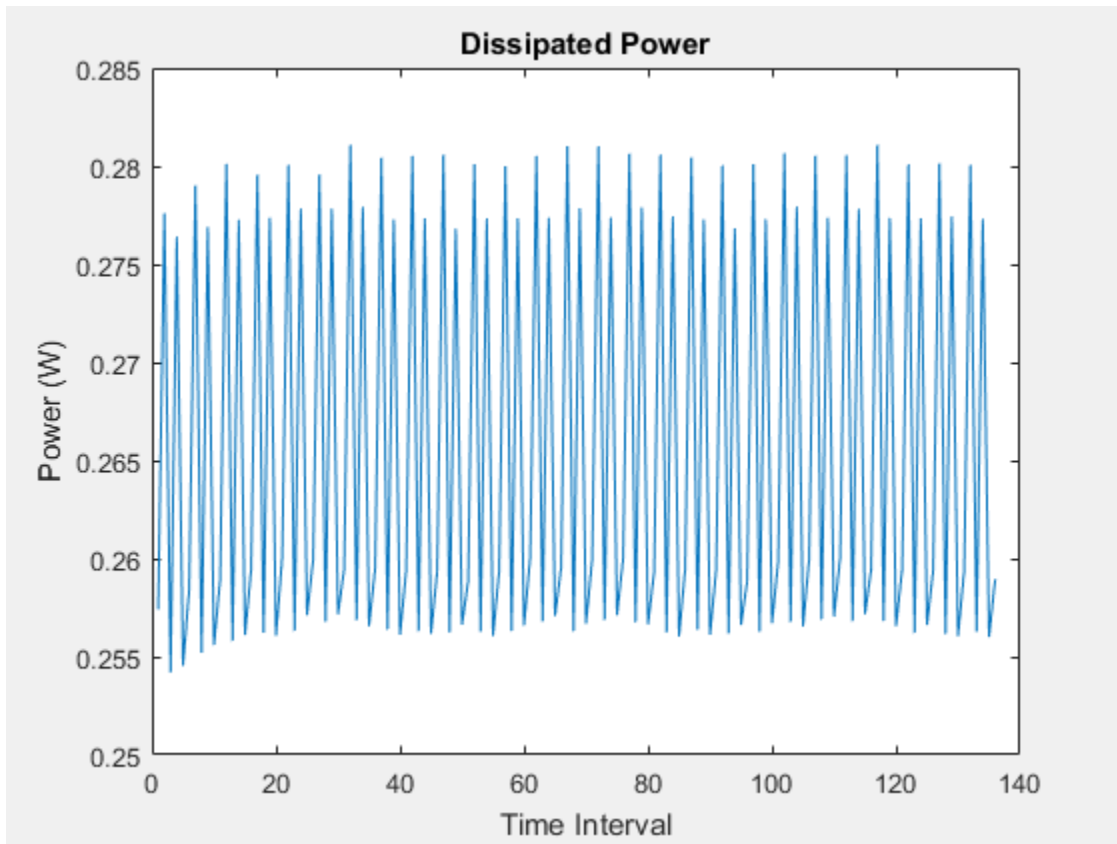
View the time series data. From the workspace, open the `lossesCell` cell array, then open the 136x3 double numeric array for the `ee_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation`.

The first two columns contain the interval start and end time. The third column contains the power loss data. In this case, to use averaging intervals that are equal in width to 1.1e-4 seconds, the function adjusts the start time for the first interval from the specified value of 0.010 seconds to a value of 0.01004 seconds. There are 136 intervals of 1.1e-4 seconds.

Plot the data.

```
plot(lossesCell{1, 2}(:,end))
title('Dissipated Power')
```

```
xlabel('Time Interval')
ylabel('Power (W)')
```



## Input Arguments

**node** — Simulation log variable, or a specific node within the simulation log variable

Node object

Simulation log workspace variable, or a node within this variable, that contains the logged model simulation data, specified as a Node object. You specify the name of the simulation log variable by using the **Workspace variable name** parameter on the **Simscape** pane of the Configuration Parameters dialog box. To specify a node within the simulation log variable, provide the complete path to that node through the simulation data tree, starting with the top-level variable name.

If `node` is the name of the simulation log variable, then the table contains the data for all blocks in the model that contain `power_dissipated` variables. If `node` is the name of a node in the simulation data tree, then the table contains the data only for:

- Blocks or variables within that node
- Blocks or variables within subnodes at all levels of the hierarchy beneath that node

Example: `simlog_ee_push_pull_converter_ccm`

**startTime** — Start of the time interval for calculating the data

0 (default) | real number

Start of the time interval for calculating the power loss time series, specified as a real number, in seconds. `startTime` must be greater than or equal to the simulation **Start time** and less than `endTime`.

Data Types: double

**endTime** — End of the time interval for calculating the data

simulation stop time (default) | real number

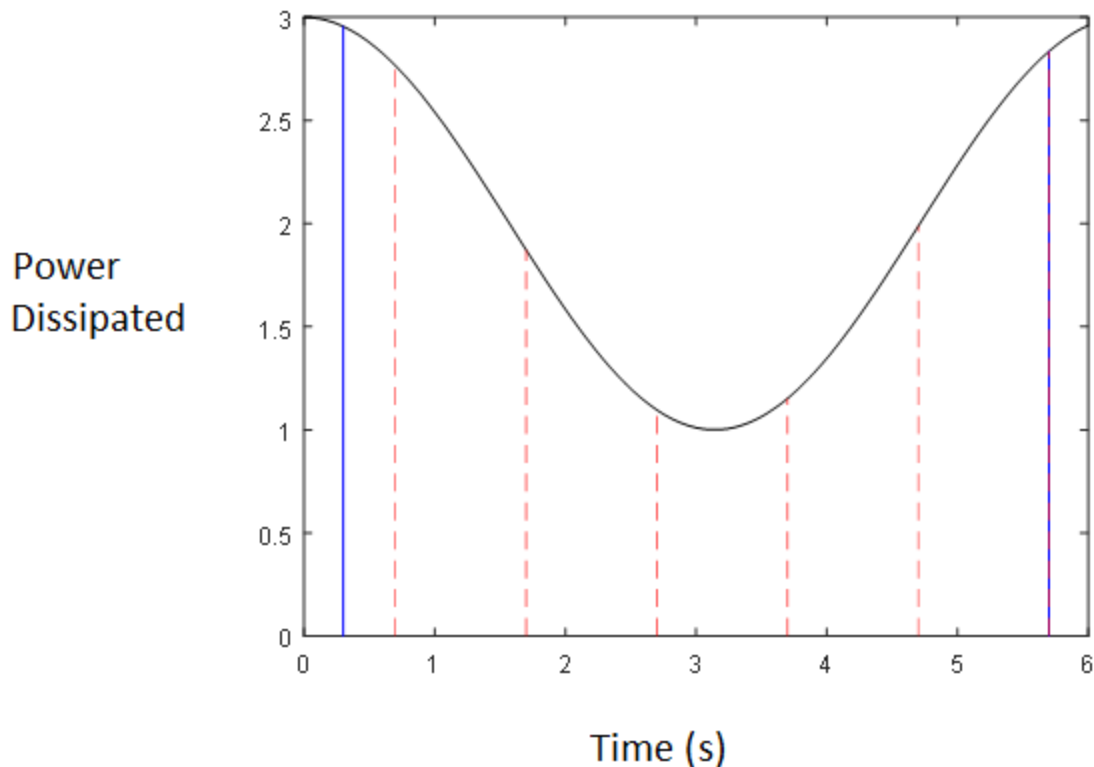
End of the time interval for calculating the power loss time series, specified as a real number, in seconds. `endTime` must be greater than `startTime` and less than or equal to the simulation **Stop time**.

Data Types: double

**intervalWidth** — size of the interval in time for calculating the average power dissipation

0 (default)

If the time between the specified `startTime` and `endTime` is not an integer multiple of `intervalWidth`, the function adjusts the start time. The figure shows how the function adjusts the start time to ensure that width of each time interval that the dissipated power is averaged over is equal to the specified `intervalWidth`.



The black line is an example of the instantaneous `power_dissipated` variables summed over all elements in an individual block. The simulation runs for 6 seconds. The `startTime` and `endTime` are indicated by the solid blue lines. The `intervalWidth` is set to 1 second. There are five intervals as indicated by the red dashed lines. The right-most edge of the last interval coincides with `endTime`.

The left-most edge of the first interval is always greater than or equal to `startTime`. The edge is equal to `startTime` only if  $(endTime - startTime) / intervalWidth$  is an integer. The output in this case consists of five values for the averaged power dissipation, one point for each time period. The function outputs the actual start and stop times in the tabulated output data.

Example: `1.1e-3`

Data Types: `double`

## Output Arguments

### **lossesCell** – Time series of the dissipated power losses for each block

cell array

Cell array that contains the names of the blocks in the nodes that contain *power\_dissipated* variables and, for each block, a three-column array:

- Column one contains the interval start time.
- Column two contains the interval end time.
- Column three contains the dissipated power for the time interval.

If the interval width is 0 seconds, that is, the start time is equal to the end time, then the dissipated power is the instantaneous power loss. If the interval is greater than 0 seconds, the dissipated power is the average power loss for the time of the interval.

## Version History

Introduced in R2017a

### **pe\_getPowerLossTimeSeries will be removed**

*Not recommended starting in R2019a*

The `pe_getPowerLossTimeSeries` function will be removed in a future release. Use the `ee_getPowerLossTimeSeries` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

## See Also

`ee_getEfficiency` | `ee_getPowerLossSummary` | `ee_getPowerLossTimeSeries` | `sscxplore`

### Topics

“Perform a Power-Loss Analysis”

“Data Logging”

“About the Simscape Results Explorer”

## pe\_plotHarmonics

Plot percentage of fundamental magnitude versus harmonic order

### Syntax

```
pe_plotHarmonics(loggingNode)
pe_plotHarmonics(loggingNode,valueIdx)
pe_plotHarmonics(loggingNode,valueIdx,tOfInterest)
pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest)
pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest)
pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...
offsetOfInterest,nHarmonic)
```

### Description

`pe_plotHarmonics(loggingNode)` plots a bar chart of percentage of fundamental magnitude versus harmonic order of the `simscape.logging.Node` of an AC or periodic variable. The title of the bar chart includes the fundamental frequency, fundamental peak value, and total harmonic distortion (THD) percentage.

You enter the input arguments in a specific order. The Simscape logging node input argument is required. All other input arguments are optional and have default values. If you are specifying a value for a subsequent optional input argument, enter `[]` to use the default value for an optional input argument.

The `pe_plotHarmonics` function uses the `pe_getHarmonics` function to:

- Find the points in the  $i^{\text{th}}$  signal (`valueIdx`) where the Simscape log crosses a threshold (`offsetOfInterest`).
- Use the crossing points to find the required number of periods (`nPeriodOfInterest`) preceding the specified time (`tOfInterest`).
- Calculate the harmonic magnitudes, up to and including the required number of harmonics (`nHarmonic`).
- Input the down-selected data to the Goertzel algorithm, which calculates the harmonic magnitudes up to and including the required number of harmonics (`nHarmonic`).

---

**Note** The `pe_getHarmonics` function uses threshold crossing points to determine the fundamental frequency of the data. If your input data is noisy or crosses the threshold more frequently than half of the fundamental period, filter it before you use the `pe_plotHarmonics` function to plot it.

---

The `pe_plotHarmonics` function then inputs the harmonic orders and harmonic magnitudes to the `pe_calculateThdPercent` function to calculate the THD.

`pe_plotHarmonics(loggingNode,valueIdx)` uses the index into value data.

`pe_plotHarmonics(loggingNode,valueIdx,tOfInterest)` uses the simulation time.

`pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest)` uses the number of periods of fundamental frequency.

`pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...,offsetOfInterest)` uses the DC offset.

`pe_plotHarmonics(loggingNode,valueIdx,tOfInterest,nPeriodOfInterest,...,offsetOfInterest,nHarmonic)` uses the number of harmonics.

## Examples

### Plot Using Default Values

This set of function arguments uses the Simscape logging node `simlog.Load.V`, which contains data from a three-phase voltage. The function analyzes the default signal, which is the first, or a-phase, signal at the final simulation time. The function uses the default values of 12 for the number of periods of the signal, 0V for the signal bias, and 30 for the number of harmonics.

```
pe_plotHarmonics(simlog.Load.V)
```

### Plot Using Specified Values

This set of function arguments uses the Simscape logging node `simlog.Load.V`, which contains data from a three-phase voltage. The function analyzes the second, or b-phase, signal at a simulation time of 2.3 s. The function uses 10 periods of the signal, which has a bias of 1 V. The function analyzes 15 harmonics.

```
pe_plotHarmonics(simlog.Load.V,2,2.3,10,1,15)
```

### Plot Using Default and Specified Values

This set of function arguments uses the Simscape logging node `simlog.Load.V`, which contains data from a three-phase voltage. The function analyzes the first, or a-phase, signal at a simulation time of 2.3 s. The function uses the default number (12) of periods of the signal, which has a bias of 1 V. The function analyzes the default number (30) of harmonics.

```
pe_plotHarmonics(simlog.Load.V,[],2.3,[],1)
```

## Input Arguments

### loggingNode — Simscape logging node

1-by-1 `simscape.logging.Node`

Simscape logging node, specified as a 1-by-1 `simscape.logging.Node`. You create a `simscape.logging.Node` by running a simulation with Simscape logging enabled. To learn how to enable data logging, see “Enable Data Logging for the Whole Model”.

Example: `simlog.Load.V`

The Simscape logging node `simlog.Load.V` contains data from a three-phase voltage.

### valueIdx — Index into value data

1 (default) | scalar

Index into value data, specified as a scalar. Specifies the  $i^{\text{th}}$  variable of interest in the Simscape log.

Example: 2

Specify the b-phase, which is the second signal from a three-phase voltage.

Example: []

Use [] to specify the default value of 1. The a-phase, which is the first signal from a three-phase voltage, is the default signal of interest.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### **tOfInterest** – Simulation time

final time in Simscape log (default) | scalar

Simulation time of interest for harmonic analysis, specified as a scalar.

Example: 2.3

Specify a 2.3s simulation time.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### **nPeriodOfInterest** – Number of periods

12 (default) | scalar

Number of periods of fundamental frequency to be included in harmonic analysis, specified as a scalar.

Example: 10

Specify 10 periods of the signal.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### **offsetOfInterest** – DC offset

0 (default) | scalar

DC offset in the input signal, specified as a scalar. The function uses this value to find the periods of interest.

Example: 1

Specify a bias of 1V for the signal.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

### **nHarmonic** – Number of harmonics

30 (default) | scalar

Number of harmonics to include in analysis, specified as a scalar.

Example: 15

Specify that the number of harmonics to be analyzed is 15.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

## **Version History**

**Introduced in R2014a**



**pe\_plotHarmonics will be removed**

*Not recommended starting in R2019a*

The `pe_plotHarmonics` function will be removed in a future release. Use the `ee_plotHarmonics` function instead. The only difference between these functions is the function name. To prevent your code from generating an error when the function is removed, update to the new function name.

**See Also****Blocks**

Spectrum Analyzer

**Functions**

`ee_calculateThdPercent` | `ee_getHarmonics` | `ee_plotHarmonics` | `sscexplore`

**Objects**

`simscape.logging.Node`

**Topics**

“Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block”

“Choose a Simscape Electrical Function for an Offline Harmonic Analysis”

“Data Logging”

“Harmonic Analysis of a Three-Phase Rectifier”

## subcircuit2ssc

Convert SPICE subcircuit to custom Simscape components

### Syntax

```
subcircuit2ssc(filename,target)
subcircuit2ssc( ___,subcircuit1,...,subcircuitN)
subcircuitArray = subcircuit2ssc( ___ )
[subcircuitArray,unsupportedCommands] = subcircuit2ssc( ___ )
```

### Description

`subcircuit2ssc(filename,target)` reads the SPICE netlist specified by `filename` and converts every subcircuit into one or more Simscape files in the folder specified by `target`.

The function lists SPICE commands not supported by the conversion process in the comments of the corresponding Simscape files. After conversion, review the generated Simscape files and make manual edits for any unsupported items. You can also obtain a list of unsupported commands by using an optional output argument, described below.

For a detailed explanation of supported conversions, see “Converting a SPICE Netlist to Simscape Blocks”.

`subcircuit2ssc( ___,subcircuit1,...,subcircuitN)` converts only the subcircuits with the specified names.

`subcircuitArray = subcircuit2ssc( ___ )` returns an array of objects containing the subcircuit information.

`[subcircuitArray,unsupportedCommands] = subcircuit2ssc( ___ )` returns an array of objects containing the subcircuit information and a `struct` array containing the subcircuit names and SPICE commands found in the converted subcircuits that are not supported by the conversion process.

### Examples

#### Create Custom Simscape Blocks from a SPICE Netlist

Create a SPICE netlist named `rcsubcircuit.cir` that contains a simple RC subcircuit.

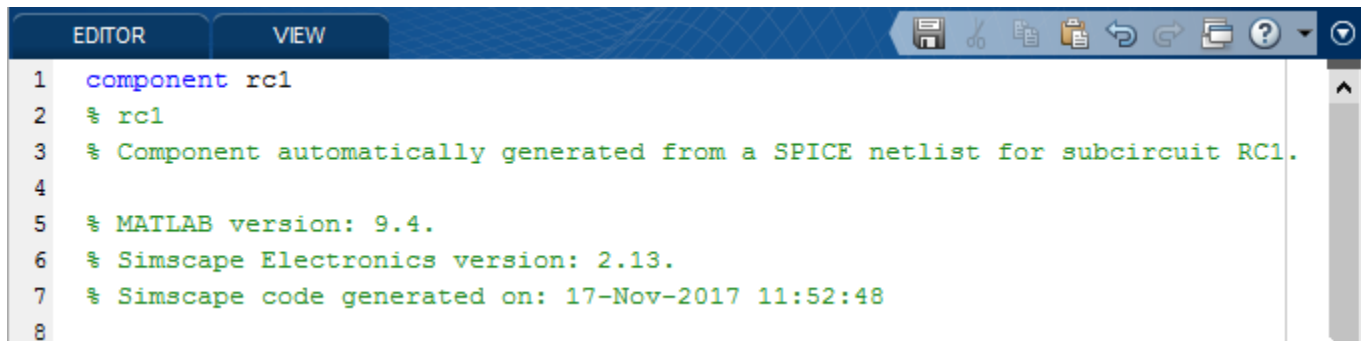
```
.SUBCKT RC1      1
*
R1      1      2      1k
C1      2      0      0.32mF
*
.ENDS
```

Convert all SPICE subcircuits in `rcsubcircuit.cir` to equivalent Simscape files and place them in a package directory called `mylibrary`.

```
subcircuit2ssc('rcsubcircuit.cir','+mylibrary');
```

Netlist converted. Review files and make manual edits for any unsupported items before building the Simscape library located at: +mylibrary.

Check the comments at the beginning of the generated component file rc1.ssc in the mylibrary package to verify that no manual conversion is required.



```

1 component rc1
2 % rc1
3 % Component automatically generated from a SPICE netlist for subcircuit RC1.
4
5 % MATLAB version: 9.4.
6 % Simscape Electronics version: 2.13.
7 % Simscape code generated on: 17-Nov-2017 11:52:48
8

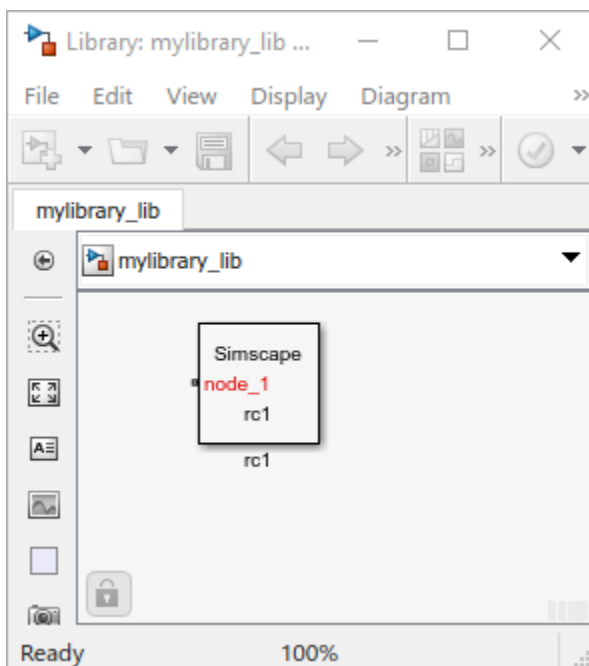
```

Generate the Simscape library using `ssc_build`.

```
ssc_build mylibrary;
```

Generating Simulink library 'mylibrary\_lib' in the current directory

Open the generated library `mylibrary_lib.slx` to access the RC component as a Simscape block.



## Make Manual Changes to a Converted Simscape Component File

Create a SPICE netlist named `temperatureresistor.cir` that contains a resistor with temperature dependence.

```
TEMPERATURERESISTOR.CIR - TEMPERATURE RESISTOR SUBCIRCUIT
*
.SUBCKT TemperatureResistor      p n
*
R1 p n 1k TC=0.01,-0.002
*
.ENDS
```

Convert all SPICE subcircuits in `temperatureresistor.cir` to equivalent Simscape files and place them in a package directory called `mylibrary`.

```
subcircuit2ssc('temperatureresistor.cir','+mylibrary');
```

Netlist converted. Review files and make manual edits for any unsupported items before building the Simscape library located at: `+mylibrary`.

Check the comments at the beginning of the generated component file `temperatureresistor.ssc` in the `mylibrary` package to identify required manual conversions.

```
component temperatureresistor
% temperatureresistor
% Component automatically generated from a SPICE netlist for subcircuit TEMPERATURERESISTOR.
%   MATLAB version: 9.7.
%   Simscape Electrical version: 7.1.
%   Simscape code generated on: 11-Dec-2018 09:45:20
%
% Users should manually implement the following SPICE commands in order to
% achieve a complete implementation:
%   R1: tc 0.01 -0.002
```

The comments suggest that you must manually convert the temperature coefficients `TC`.

In the `components` section of the component file, replace the resistor with a SPICE resistor, which models temperature dependence:

```
components(ExternalAccess=observe)
    R1 = ee.additional.spice_passives.res(...
        R={1*1000}, '0hm'},...
        TC1={0.01}, '1/K'},...
        TC2={(-0.002)}, '1/K^2'});
end
```

Generate the Simscape library using `ssc_build`.

```
ssc_build mylibrary;
```

Generating Simulink library 'mylibrary\_lib' in the current directory

Open the generated library `mylibrary_lib.slx` to access the resistor with temperature dependence component as a Simscape block.

## Input Arguments

**filename** — SPICE network file

character vector (default) | string scalar

Name of the SPICE network file to read. This file must be on the path.

Example: 'SpiceSubcircuits.cir'

Data Types: char | string

### **target — Location of generated Simscape component files**

character vector (default) | string scalar

Name of the folder where the Simscape language files are generated. To allow the building of custom block libraries, specify a package directory with the '+' precursor. If the specified folder does not exist, the function creates it in the current folder.

Example: '+SimscapeSubcircuits'

Data Types: char | string

### **subcircuit1, ..., subcircuitN — SPICE subcircuit names**

character vectors (default) | string scalars

Names of the SPICE subcircuits to convert to Simscape language files.

Example: 'Subcircuit1','Subcircuit2'

Data Types: char | string

## **Output Arguments**

### **subcircuitArray — Array of objects containing the subcircuit information**

array

Array of objects containing the subcircuit information.

### **unsupportedCommands — List of unsupported commands**

struct array

A struct array containing the subcircuit names and SPICE commands found in the converted subcircuits that are not supported by the conversion process.

## **Limitations**

- The netlist must be written in Cadence® PSpice format and be syntactically correct. The conversion assistant does not check for proper PSpice syntax.
- Only a subset of the PSpice netlist language is supported. However, unsupported PSpice commands are identified at the top of the corresponding Simscape component file to facilitate manual conversion.
- To build generated Simscape components into Simscape blocks, parameter values must conform to Simscape constraints. For example, capacitance of a fundamental capacitor and inductance of a fundamental inductor must be nonzero.
- The `subcircuit2ssc` function is useful only to convert circuits with a small number of R, L, C, and transistor devices. You should not use this function to simulate integrated circuits with hundreds of transistors. Large networks of semiconductor devices are more challenging to simulate and they only simulate reliably by using the SPICE simulation engine for which they were originally written. If you need to model application-specific integrated circuits (ASICs), create a

behavioral representation of the device function instead. This results in a more useful model that simulates faster.

- If you are unable to start the simulation due to numerical issues, use the local solver option with a step size that is about ten times smaller than the fastest time constant of the circuit. The original subcircuit might consist of discontinuities in values or in the first or second derivatives where custom functions have been used. These discontinuities might cause poor numerical behavior.

## **Version History**

**Introduced in R2018b**

### **See Also**

#### **Topics**

“Converting a SPICE Netlist to Simscape Blocks”

“Building Custom Block Libraries”

“Composite Components”

# Tools and Apps

---

## Circuit Simulation Onramp

Free, self-paced, interactive Simscape Electrical circuit simulation course

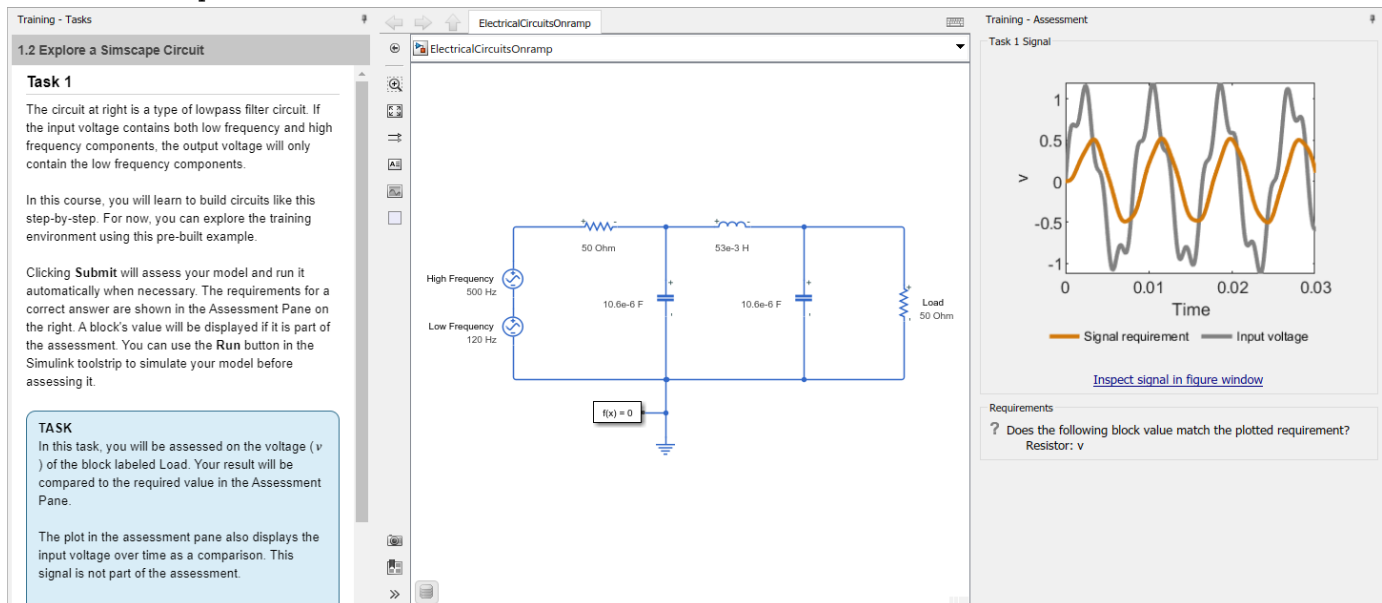
### Description

Circuit Simulation Onramp is a free, self-paced, interactive course that helps you get started with Simscape Electrical.

Circuit Simulation Onramp teaches you to:

- Use RC and RLC circuits, where R is a resistor, C is a capacitor, and L is an inductor.
- Implement nonlinear elements to simulate a power supply circuit, including fault protection.
- Model circuits that contain operational amplifiers (op amps).
- Simulate electrical filters and analyzes their performance.

Circuit Simulation Onramp uses tasks to teach concepts incrementally, such as through a real-life example with a Wein Bridge oscillator. You receive automated assessments and feedback after submitting tasks. Your progress is saved when you exit the course, so you can complete the course in multiple sessions.



### Open the Circuit Simulation Onramp

- Simulink Start Page: On the **Learn** tab, click the **Launch** button that appears when you pause on **Circuit Simulation Onramp**.
- MATLAB Command Window: Enter `learning.simulink.launchOnramp("circuits")`.

**Note** If you do not have a Simscape Electrical license, you can take the course at Self-Paced Online Courses.



## **Version History**

Introduced in R2021b

### **See Also**

#### **Tools**

**Simulink Onramp | Simscape Onramp | Power Electronics Simulation Onramp**

#### **Functions**

`learning.simulink.launchOnramp`

#### **Topics**

“Build and Simulate a Simple Circuit”

“DC Motor Model”

“Build and Simulate Composite and Expanded Three-Phase Models”

“Build and Simulate a Simple Circuit Using Specialized Power Systems”

“Analyze a Simple Circuit”

“Essential Electrical Modeling Techniques”

## Load-Flow Analyzer

Determine the steady-state voltage magnitudes and angles and real and reactive power flows for a three-phase AC, DC, or mixed AC/DC network

### Description

The **Load-Flow Analyzer** app performs a load-flow, or power-flow, analysis for a three-phase AC or DC electrical power transmission system that you model and configure for analysis using Simscape Electrical. The app generates three tables. The AC Nodes table contains data for the AC network nodes, as represented by AC busbar, load flow source, synchronous machine, induction machine, and three-phase load blocks. The AC & DC Busbars table contains data for all busbars. The AC Connections table contains data for the network connections, as represented by transmission line and transformer blocks.

The **Load-Flow Analyzer** app allows you to:

- Run a load-flow analysis.
- Highlight and update load-flow input block parameter values for busbar, load flow source, synchronous machine, induction machine, and three-phase load blocks.
- Change the bus type of load flow source, synchronous machine, and induction machine blocks.
- Select and highlight node and connection blocks in the model.
- Sort columns in the tables by increasing or decreasing values.
- Export the data to a spreadsheet, a MAT-file, or comma-separated variable (CSV) files.

This table shows the available Simscape Electrical blocks that you can use to model the network nodes and connections:

Network Nodes	Network Connections
Busbar	Transmission Line (Three-Phase)
Busbar (DC)	Two-Winding Transformer (Three-Phase)
Delta-Connected Load	
Induction Machine Squirrel Cage	
Induction Machine Wound Rotor	
Load Flow Source	
Synchronous Machine Model 2.1	
Synchronous Machine Round Rotor	
Synchronous Machine Salient Pole	
Wye-Connected Load	

The screenshot displays the Simscape Electrical Load-Flow Analyzer interface. The top toolbar includes buttons for Highlight Table, Highlight Model, Refresh, Settings, Run, and Export. Below the toolbar, there are two tabs: 'AC & DC Busbars' and 'AC Nodes'. The 'AC Nodes' tab is active, showing a table with columns: Block Type, Bus Type, Rated Voltage, kV, Specified Voltage Magnitude, pu, Actual Voltage Magnitude, pu, Voltage Angle, deg, and Specified Real Generation. The 'AC Connections' tab is also visible, showing a table with columns: Block Type, From Busbar, To Busbar, Rated Voltage, kV, Voltage V1, pu, Voltage V2, pu, Voltage Angle12, deg, Real Power Flow P12, MW, and Reactive Power.

	Block Type	Bus Type	Rated Voltage, kV	Specified Voltage Magnitude, pu	Actual Voltage Magnitude, pu	Voltage Angle, deg	Specified Real Generation
G1C//Swing 1.04pu	Load Flow Source	Swing	16.5000	1.0400	NaN	NaN	
G2C//PV 163MW 1.025pu	Load Flow Source	PV	18.0000	1.0250	NaN	NaN	
G3C//PV 85MW 1.025pu	Load Flow Source	PV	13.8000	1.0250	NaN	NaN	
Bus1 16.5kV	Busbar	None	16.5000	NaN	NaN	NaN	
Bus2 18kV	Busbar	None	18.0000	NaN	NaN	NaN	
Bus3 13.8kV	Busbar	None	13.8000	NaN	NaN	NaN	
Bus4 230kV	Busbar	None	230.0000	NaN	NaN	NaN	
Bus5 230kV	Busbar	None	230.0000	NaN	NaN	NaN	
Bus6 230kV	Busbar	None	230.0000	NaN	NaN	NaN	

	Block Type	From Busbar	To Busbar	Rated Voltage, kV	Voltage V1, pu	Voltage V2, pu	Voltage Angle12, deg	Real Power Flow P12, MW	Reactive Power
B4 to B5 50 km	Transmission Line	Bus4 230kV	Bus5 230kV	230/230	NaN	NaN	NaN	NaN	NaN
B4 to B6 50 km	Transmission Line	Bus4 230kV	Bus6 230kV	230/230	NaN	NaN	NaN	NaN	NaN
B5 to B7 100 km	Transmission Line	Bus5 230kV	Bus7 230kV	230/230	NaN	NaN	NaN	NaN	NaN
B6 to B9 100 km	Transmission Line	Bus6 230kV	Bus9 230kV	230/230	NaN	NaN	NaN	NaN	NaN
B7 to B8 50 km	Transmission Line	Bus7 230kV	Bus8 230kV	230/230	NaN	NaN	NaN	NaN	NaN
B8 to B9 100 km	Transmission Line	Bus8 230kV	Bus9 230kV	230/230	NaN	NaN	NaN	NaN	NaN
TF 4-1	Transformer	Bus4 230kV	Bus1 16.5kV	230/16.5	NaN	NaN	NaN	NaN	NaN
TF 7-2	Transformer	Bus7 230kV	Bus2 18kV	230/18	NaN	NaN	NaN	NaN	NaN
TF 9-3	Transformer	Bus9 230kV	Bus3 13.8kV	230/13.8	NaN	NaN	NaN	NaN	NaN

## Open the Load-Flow Analyzer App

Simulink Toolstrip: On the **Apps** tab, under **Simscape**, click the **Load-Flow Analyzer** icon.

MATLAB Command Window: To prepare to perform a power-flow analysis on the current model, which is the open model that you most recently interacted with, enter `ee_loadFlowApp`. To prepare for an analysis on an open model that is not the current model, you must pass the model name as an input argument to the command.

When you open the app, the tables are preloaded with node and connection blocks and with specified parameter values for the blocks in the current or specified model that are inputs to the load-flow analysis. If the simulation has not yet been executed to generate the corresponding Simscape Simulation Data object in the MATLAB base workspace, then the fields for load-flow analysis output data contain NaN.

After you run the load-flow analysis, the tables also display the load-flow analysis output values, including the steady-state voltage magnitudes, voltage phase angles, active power, and reactive power for the node and connection blocks.

## Examples

### Preload Block Parameters for the Current Model

Open the model. At the MATLAB command prompt, enter:

```
ee_loadflow_ieee9bus
```

The `ee_loadflow_ieee9bus` model is the current model.

Open the **Load-Flow Analyzer** app.

```
ee_loadFlowApp
```

The **Load-Flow Analyzer** app opens and is preloaded with load-flow input parameter values for applicable node and connection blocks in the `ee_loadflow_ieee9bus` model.

### Preload Block Parameters for a Model That Is Not the Current Model

Open the model. At the MATLAB command prompt, enter:

```
modelName = 'ee_loadflow_ieee9bus'  
open_system(modelName)
```

Open the **Load-Flow Analyzer** app by passing the name of the model as an input argument.

```
ee_loadFlowApp(modelName)
```

The **Load-Flow Analyzer** app opens and is preloaded with load-flow input parameter values for applicable node and connection blocks in the specified model.

### Highlight Inputs

To identify load-flow analysis inputs that you can tune by updating values in the **Load-Flow Analyzer** app or block parameter settings, select **Highlight Table**.

### Update the Bus Types

To change the **Bus Type** for Load-Flow Source, wye-connected load, synchronous machine, or induction machine blocks, double-click the field for the bus type, and select an applicable type. Options are:

Load-Flow Source	Synchronous Machine	Induction Machine	W y e - C o n n e c t e d L o a d Z
Time	Swing	PV	
Swing	PV		
PV	PQ	PQ	
PQ			

### Tune the Input Parameters

To change a load-flow analysis input, update the applicable values in the **Load-Flow Analyzer** app or in the block parameter settings.

---

**Note** If you update input values in the table, the values displayed in the block settings update immediately. However, if you update input values in the block settings or rewire the diagram, the values displayed in the table reflect the change only if you click the **Refresh** button in the **Load-Flow Analyzer** app.

---

If a parameter value is not defined, such as when a model references a variable in the base workspace that you have already cleared, then you might not be able to set a value for that specific parameter, or you haven't run a simulation on that specific model yet.

### Set the Solver Configuration

To set the solver configuration type for the model, click the **Settings** button and change the value of the **Simscape Solver Configuration** parameter in the **Load-Flow Analyzer** app. Options are:

- Frequency and time (phasor) — Sets the **Equation formulation** parameter of all Solver Configuration blocks in the model to Frequency and time.
- Time and steady state — Sets the **Equation formulation** parameter of all Solver Configuration blocks in the model to Time and selects the **Start simulation from steady state** check box.

- **Use local settings** — Allows for different Solver Configuration settings on multiple networks inside the same model.

For more information on solver configurations, see Solver Configuration.

### Set the Simulation Configuration

To set the simulation configuration, in the **Load-Flow Analyzer** app, click the **Settings** button and change the value of the **Simulation Configuration** parameter. Options are:

- **Static** — The model performs a static load-flow analysis from  $t = 0$ .
- **Dynamic** — The model performs a dynamic simulation from the simulation start time, specified in the **Start time** parameter, to the simulation stop time, specified in the **Stop time** parameter in the model. To select the simulation time of interest for static load-flow results after the simulation, specify the **Time** settings.

### Run a Load-Flow Analysis

To run a load-flow analysis, click the **Run** button in the **Load-Flow Analyzer** app or in the model.

### Sort the Data

Sorting the data allows you to check that your model is configured correctly.

To sort the data for block parameter settings or load-flow components in ascending or descending order, click the arrow icon in the column heading of the data you want to sort.

### Highlight Blocks in the Model

To identify individual blocks or groups of blocks in the model, select **Highlight model**. Then you can sort the column data to:

- Discover over-voltage or under-voltage magnitudes.
- Discover large voltage angle differences.
- Review differences between specified and actual values.
- Discover which connections are consuming the most real power and sourcing or sinking the most reactive power.

### Export the Data

To export the tables' data to a spreadsheet, a MAT-file, or three comma-separated variable (CSV) files, click the **Export** button.

- 2-Bus Loadflow

- IEEE 9-Bus Loadflow
- Induction Motor Initialization with Loadflow
- Synchronous Machine Initialization with Loadflow
- “Mixed AC/DC System Loadflow”

## Programmatic Use

`ee_loadFlowApp`

Open the **Load-Flow Analyzer** app preloaded with load-flow input parameter values for applicable node and connection blocks in the current model.

`ee_loadFlowApp(modelName)`

Open the **Load-Flow Analyzer** app preloaded with load-flow input parameter values for applicable node and connection blocks in the specified model.

## Limitations

- You cannot perform a load-flow analysis on a model if the model is not configured for such an analysis. For more information, see “Perform a Load-Flow Analysis Using Simscape Electrical”.
- You cannot perform a load-flow analysis using the **Load-Flow Analyzer** app if the three-phase ports on relevant node or connection blocks in the model are expanded into separate phases. For more information, see “Comparison of Three-Phase Port Types”.
- To obtain all the load-flow analysis results, you must put a Busbar or a Busbar (DC) block in the connection lines between each block.
- You cannot alter the RLC structure of a constant impedance load from the app. To alter the structure, change the dropdown on the block mask.
- If you parameterize your load-flow model using variables in the MATLAB base or model workspaces, the **Load Flow Analyzer** app evaluates these variables and provides the associated numeric values. If you then update the values of these variables inside the app, the numeric values replace the respective variable names in the block mask.

## Version History

**Introduced in R2020a**

### See Also

Busbar | Busbar (DC) | Load Flow Source

### Topics

2-Bus Loadflow

IEEE 9-Bus Loadflow

Induction Motor Initialization with Loadflow

Synchronous Machine Initialization with Loadflow

“Mixed AC/DC System Loadflow”

“Perform a Load-Flow Analysis Using Simscape Electrical”

## Harmonic Analyzer

Analyze and plot harmonic spectrum of Simscape logging nodes

### Description

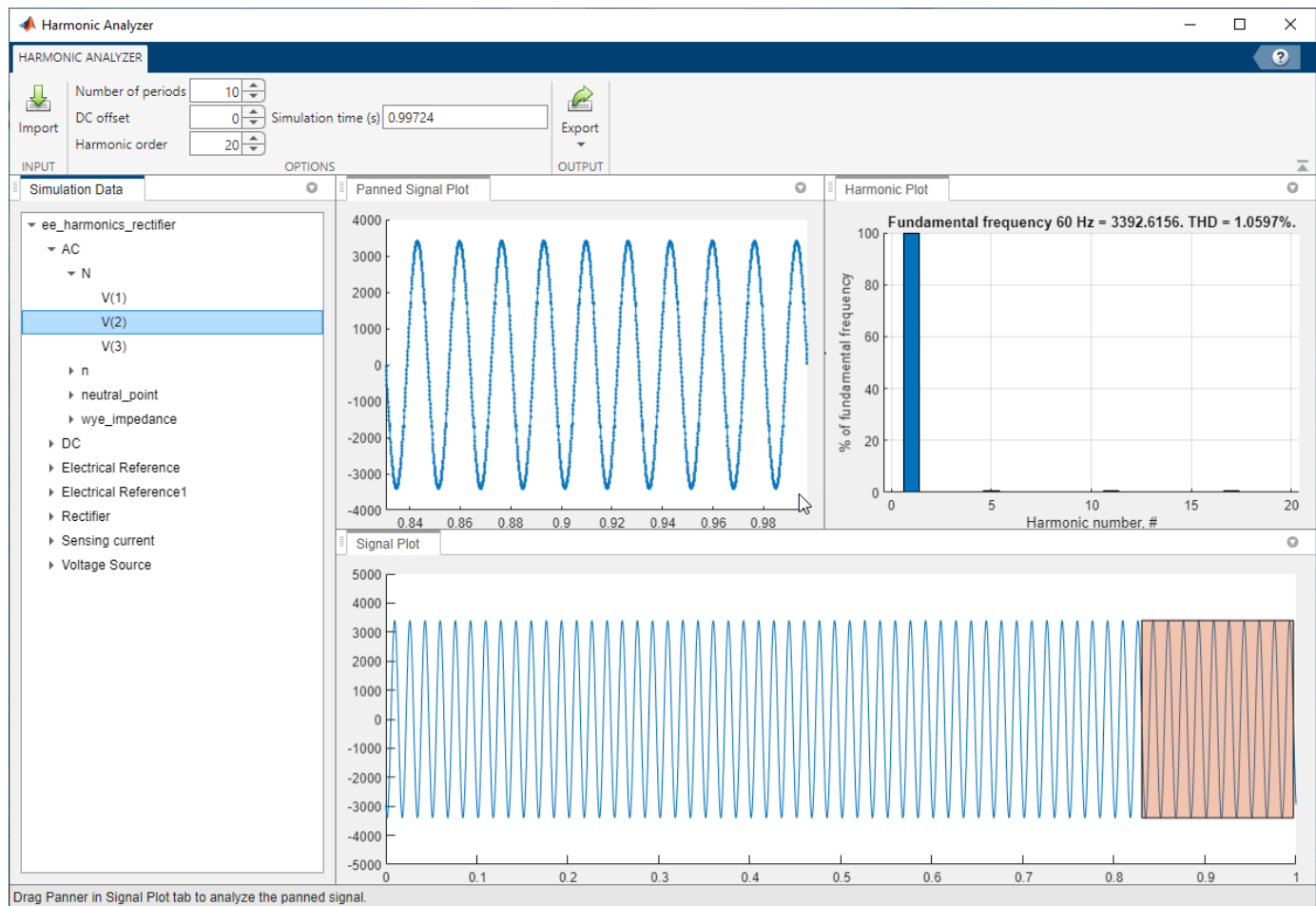
The **Harmonic Analyzer** app calculates the harmonic orders, magnitudes, and fundamental frequency of a `simscape.logging.Node` of an AC or periodic variable and plots a bar chart of the percentage of fundamental magnitude versus harmonic order.

The **Harmonic Analyzer** app allows you to:

- Import the simulation data of a model stored in a `simscape.logging.Node` variable.
- Analyze and plot the harmonic spectrum of the selected `simscape.logging.Node` object.
- Export the data in MATLAB as a script or a function.

The app finds the points in the selected leaf node, or a node at the very end of the tree navigation, where the Simscape log crosses a threshold specified by the **DC offset** parameter. It then uses the crossing points to find the required number of periods, specified by the **Number of periods** parameter, preceding the value specified by the **Simulation time (s)** parameter. Finally it inputs the selected data to the Goertzel algorithm, which calculates the harmonic magnitudes up to the value in the **Harmonic order** parameter.





## Open the Harmonic Analyzer App

To open the app, at the MATLAB Command Window, enter: `ee_HarmonicAnalyzer`.

## Examples

### Load Simulation Data Variable for the Current Model

Open the model. At the MATLAB command prompt, enter:

```
ee_harmonics_rectifier
```

Run the model. In the **Simulation** tab, click the **Run** button.

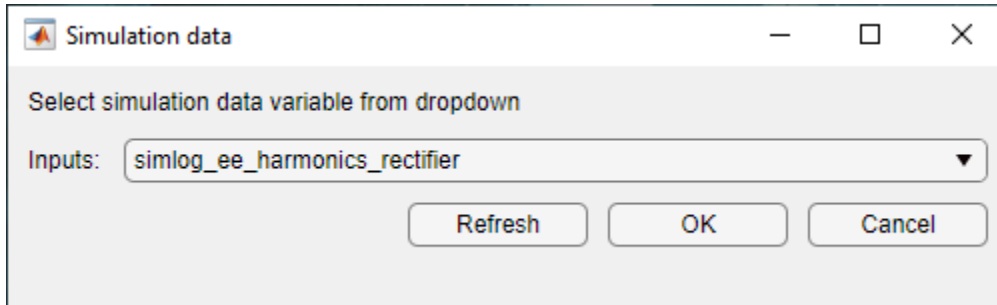
The base workspace now contains the `simlog_ee_harmonics_rectifier` Node object.

Open the **Harmonic Analyzer** app by entering:

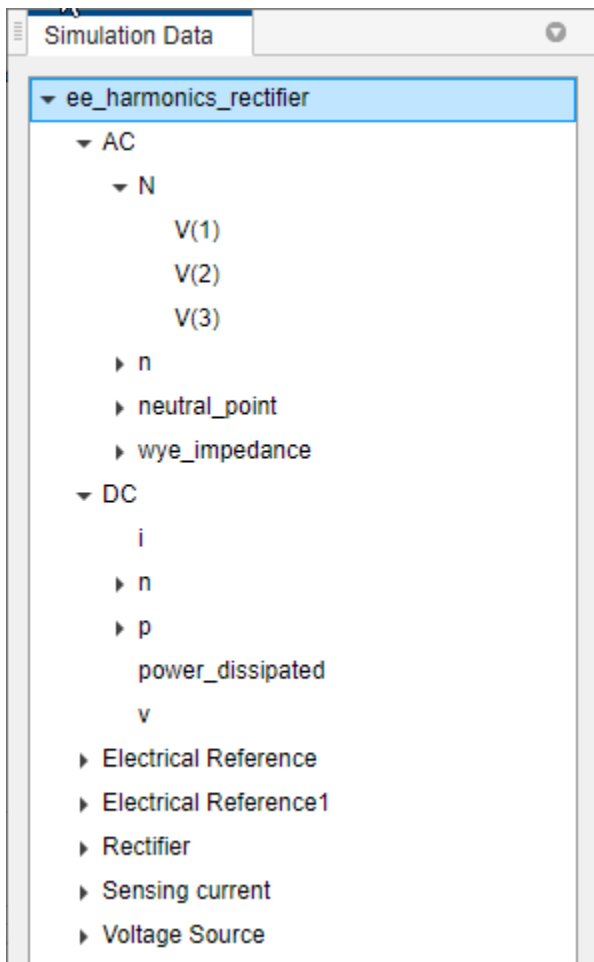
```
ee_HarmonicAnalyzer
```

To load the simulation data variable of the `ee_harmonics_rectifier` model, in the app, click **Import**.

In the Simulation data dialog, set **Inputs** to the `simlog_ee_harmonics_rectifier` Node object to import from the base workspace.



The app imports all Simscape logging nodes contained in the `simlog_ee_harmonics_rectifier` Node object.



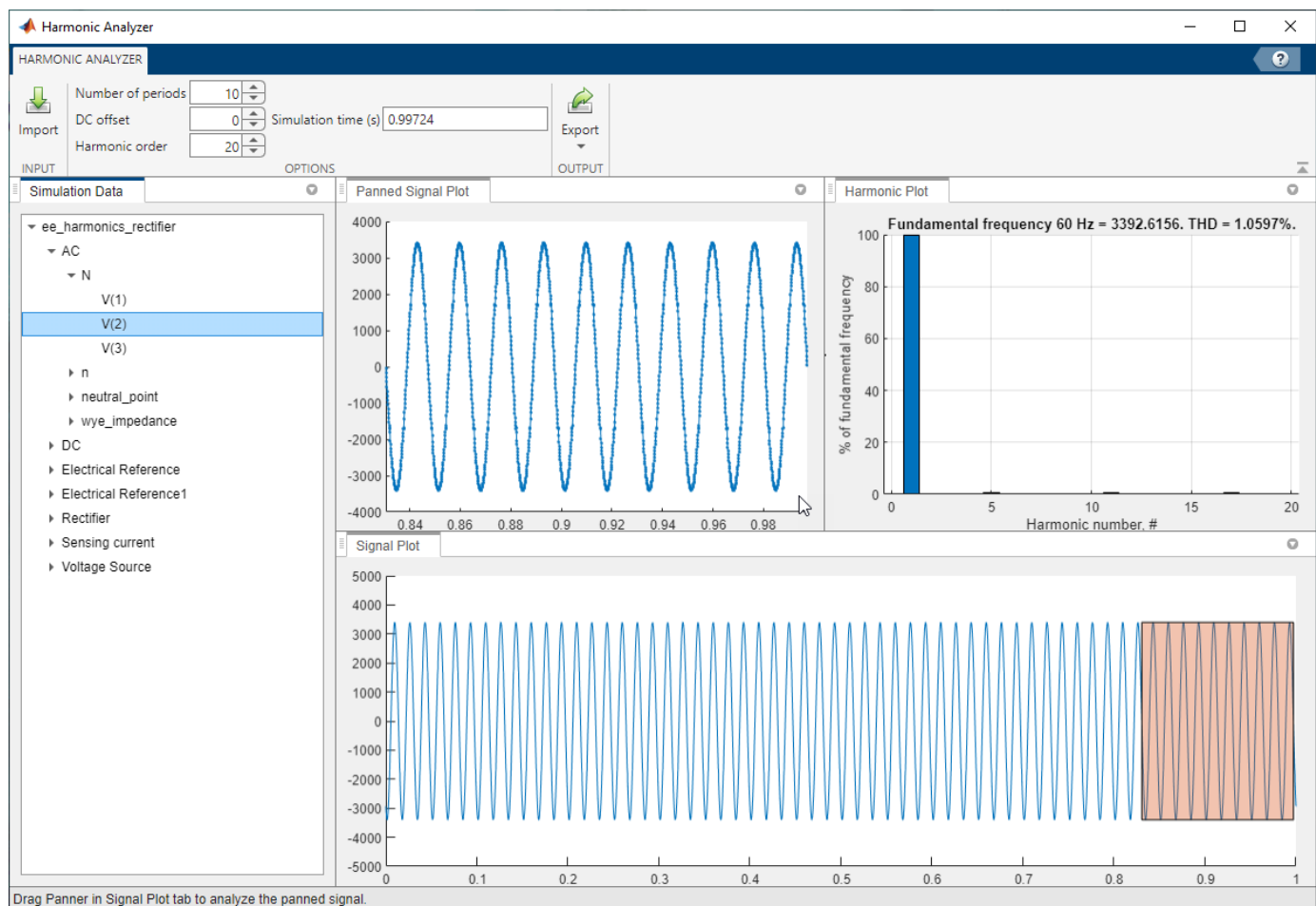
For more information about data logging, see “Enable Data Logging for the Whole Model”.

### Analyze and Plot the Signal and the Harmonic Spectrum for the Selected Logging Node

To analyze and plot the signal and the harmonic spectrum of a logging node, open the **Harmonic Analyzer** app, click the **Simulation Data** tab, then select a node. You can only plot the harmonic spectrum of a leaf node.

Follow the steps in “Load Simulation Data Variable for the Current Model” on page 3-11 to open the model, open the app, and load the Node object for `simlog_ee_harmonics_rectifier`.

Then select the **V(2)** node to plot the b-phase of the **AC** node in the **Signal Plot** tab, as well as its harmonic spectrum in the **Harmonic Plot** tab.



The **Panned Signal Plot** tab plots the panned signal. Drag the panner in the **Signal Plot** tab to interactively focus on a specific part of the signal. To increase or decrease the size of the panner, specify the **Number of periods** parameter.

## Export the Data

To export the MATLAB commands required to plot the harmonic spectrum of the selected node outside of the **Harmonic Analyzer** app, click the **Export** button and then **Script** or **Function** to generate a script or a function, respectively.

- “Harmonic Analysis of a Three-Phase Rectifier”

## Parameters

### Number of periods — Number of periods

10 (default) | positive scalar

Number of periods of fundamental frequency in the harmonic analysis. The value of this parameter also affects the size of the panner box in the **Signal Plot** tab.

### DC offset — Input DC offset

0 (default) | positive scalar

DC offset in the input signal. The app uses this value to find the periods of interest.

### Harmonic order — Number of harmonics

20 (default) | positive scalar

Number of harmonics to include in the analysis.

### Simulation time (s) — Simulation time of interest

positive double

Time of the last period panned to compute the harmonic spectrum. The final permissible simulation time is equal to the simulation time specified in the model.

## Limitations

The Harmonic Analyzer app does not support `simscape.logging.Node` objects generated by variable step size models.

## Version History

Introduced in R2022a

## See Also

### Functions

`ee_getHarmonics` | `ee_plotHarmonics`

### Topics

“Harmonic Analysis of a Three-Phase Rectifier”

# Power Electronics Simulation Onramp

Free, self-paced, interactive Simscape Electrical power electronics simulation course

## Description

Power Electronics Simulation Onramp is a free, self-paced, interactive course that helps you get started with Simscape Electrical.

Power Electronics Simulation Onramp teaches you to:

- Model and simulate a buck power converter circuit.
- Choose the appropriate level of detail, or model fidelity, to model your circuits.
- Compare accuracy and simulation speed of models with these buck converter representations:
  - Behavioral — Ideal DC-DC converter
  - Averaged switch — Simplified buck converter with an averaged switching device
  - Piecewise linear — Simplified buck converter with an ideal semiconductor switching device
  - Nonlinear — Individual circuit components of a buck converter
- Apply closed-loop feedback control for a voltage-controlled buck converter.

To teach concepts incrementally, Power Electronics Simulation Onramp uses tasks. You receive automated assessments and feedback after submitting tasks. Your progress is saved when you exit the training, so you can complete the training in multiple sessions.

The screenshot displays the Power Electronics Simulation Onramp interface, which is divided into three main sections:

- Task 1 (Left Panel):**
  - Course Overview:** Power Electronics Simulation (Voltage Control)
  - Task 1:** By the end of this course, you will build this voltage-controlled buck converter model.
  - Description:** The input voltage is 48 V and it uses feedback control to maintain the desired output voltage of 12 V, despite a step-change in the load resistance. The PS Step block in the model has been configured to provide an output beginning at 5 Ohm and then stepping to 2.5 Ohm halfway through the simulation.
  - Instructions:** Throughout this course, you will complete tasks and receive feedback in the pane at right. If the assessment is related to a signal or variable in your model, a plot will automatically be displayed in this area. This interaction assesses both a block connection and the voltage of the Variable Resistor (named Variable Load).
  - TASK:** Connect the PS Step block to the physical signal port, R, of the block labeled Variable Load.
  - Diagram:** A small schematic shows a variable resistor labeled 'Variable Load' with a signal port 'R' and a 'DC' input.
  - Buttons:** Hint | See Solution | Reset | Submit
- Simulation Environment (Center Panel):**
  - Block Diagram:** Shows a closed-loop control system. A 'Reference Voltage' block (12V) feeds into a 'Discrete PID Controller', which outputs to a 'PWM Generator'. The PWM Generator drives a 'Buck Converter' (consisting of a MOSFET, diode, inductor, and capacitor). The output of the buck converter is connected to a 'Variable Load' and a 'Voltage Sensor'. A 'PS Step' block is connected to the 'Variable Load'.
  - DC Input Voltage:** 48 V
  - Control Signal:**  $f(x) = 0$
- Training - Assessment (Right Panel):**
  - Requirements:** Are the following blocks connected correctly?
  - PS Step**
  - Variable Resistor: (R)**

## Open the Power Electronics Simulation Onramp

- Simulink Start Page: On the **Learn** tab, click the **Launch** button that appears when you pause on **Power Electronics Simulation Onramp**.
- MATLAB Command Window: Enter  
`learning.simulink.launchOnramp("powerelectronics").`

---

**Note** If you do not have a Simscape Electrical license, you can take the course at Self-Paced Online Courses.

---

## Version History

Introduced in R2022b

### See Also

#### Tools

**Simulink Onramp** | **Simscape Onramp** | **Circuit Simulation Onramp**

#### Functions

`learning.simulink.launchOnramp`

#### Simscape Blocks

Buck Converter

#### Topics

“Build and Simulate a Simple Circuit”

“Essential Electrical Modeling Techniques”

“Buck Converter”

“Buck Converter Voltage Control”

# Parameter Dependencies

## Parameter Dependencies

A Simscape block parameter is considered *visible* when it appears in the Simulink Property Inspector, in a block dialog box, or as a block choice on the Simscape context menu. A block parameter is considered *configurable* or *enabled* when you can configure it by selecting an option, entering a value, or selecting or clearing a check box. Parameters that are visible but are not enabled are typically dimmed.

The visibility and configurability of some parameters depend on the options that you select for other parameters. Parameter dependencies are typically listed in the parameter description in the documentation for the block. For some blocks, the documentation also includes a parameter dependency table.

### Parameter Dependency Tables

Parameter dependency tables show how the visibility and configurability of some parameters depend on the options that you select for other parameters.

Parameter dependency tables indicate which parameter options affect the *visibility* of other parameters. For example, the next table shows the dependencies for the **Effects and Initial Conditions** parameters for the Pipe (2P) block from the **Simscape > Foundation Library > Two-Phase Fluid > Elements** library.

The row numbers in this table are for reference only. The first column of a parameter dependency table typically includes all parameters that are visible by default.

Parameters that affect the visibility or configurability of other parameters are shown in bold text. The options for the parameters that affect other parameters are shown in italic text in the following row of the table. Parameters that do not affect other parameters are shown in plain text.

#### Effects and Initial Conditions Parameter Dependencies

Row	Parameters and Options					
1	<b>Fluid Inertia</b>					
2	<i>Off</i>			<i>On</i>		
3	<b>Initial fluid regime</b>			<b>Initial fluid regime</b>		
4	<i>Subcooled liquid</i>	<i>Two-phase mixture</i>	<i>Superheated vapor</i>	<i>Subcooled liquid</i>	<i>Two-phase mixture</i>	<i>Superheated vapor</i>
5	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure
6	Initial fluid temperature	Initial fluid vapor quality	Initial fluid temperature	Initial fluid temperature	Initial fluid vapor quality	Initial fluid temperature
7				Initial mass flow rate from port A to port B	Initial mass flow rate from port A to port B	Initial mass flow rate from port A to port B
8	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant



The figure shows the **Effects and Initial Conditions** parameters that are visible by default on the block dialog box for the Pipe (2P) block.

Settings

Geometry Friction and Heat Transfer **Effects and Initial Conditions**

Fluid inertia:	Off	
Initial fluid regime:	Subcooled liquid	
Initial fluid pressure:	0.101325	MPa
Initial fluid temperature:	293.15	K
Phase change time constant:	0.1	s

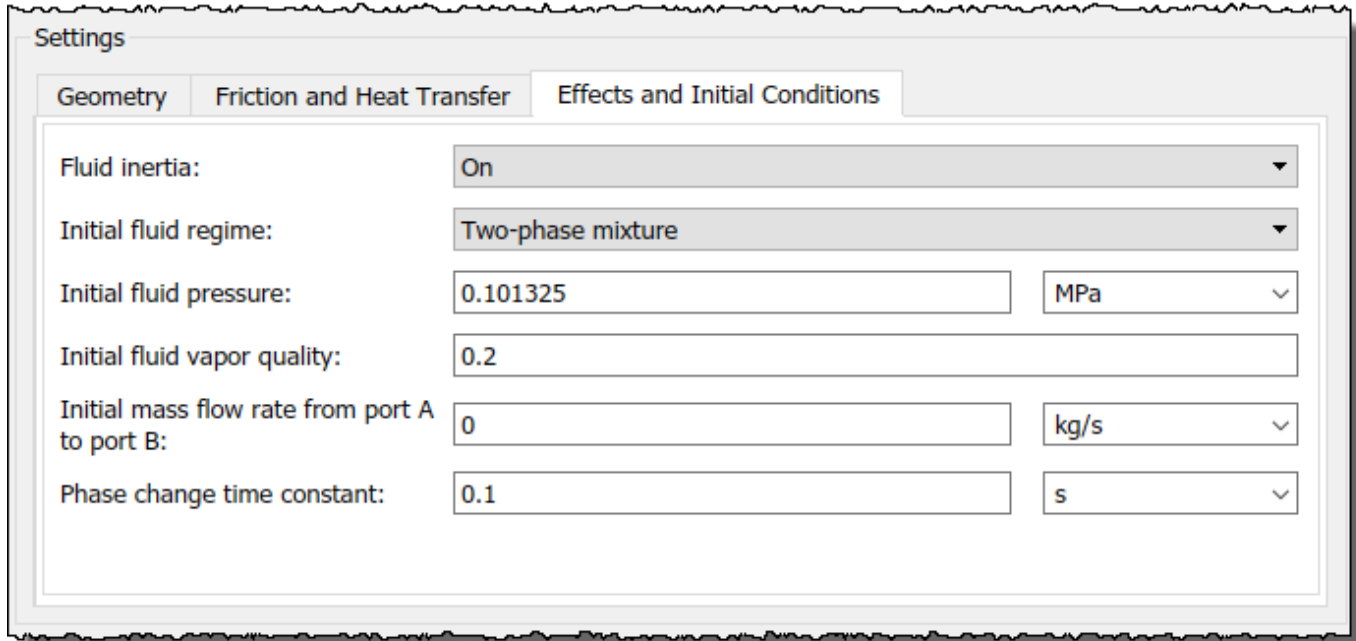
If you change the **Fluid Inertia** parameter from Off to On, the **Initial mass flow rate from port A to port B** parameter becomes visible. The table shows this dependency in rows 1, 2, and 7.

Settings

Geometry Friction and Heat Transfer **Effects and Initial Conditions**

Fluid inertia:	On	
Initial fluid regime:	Subcooled liquid	
Initial fluid pressure:	0.101325	MPa
Initial fluid temperature:	293.15	K
Initial mass flow rate from port A to port B:	0	kg/s
Phase change time constant:	0.1	s

If you then change the **Initial fluid regime** parameter from Subcooled liquid to Two-phase mixture, the **Initial fluid temperature** parameter is hidden and **Initial fluid vapor quality** becomes visible. The table shows these dependencies in rows 3, 4, and 6.



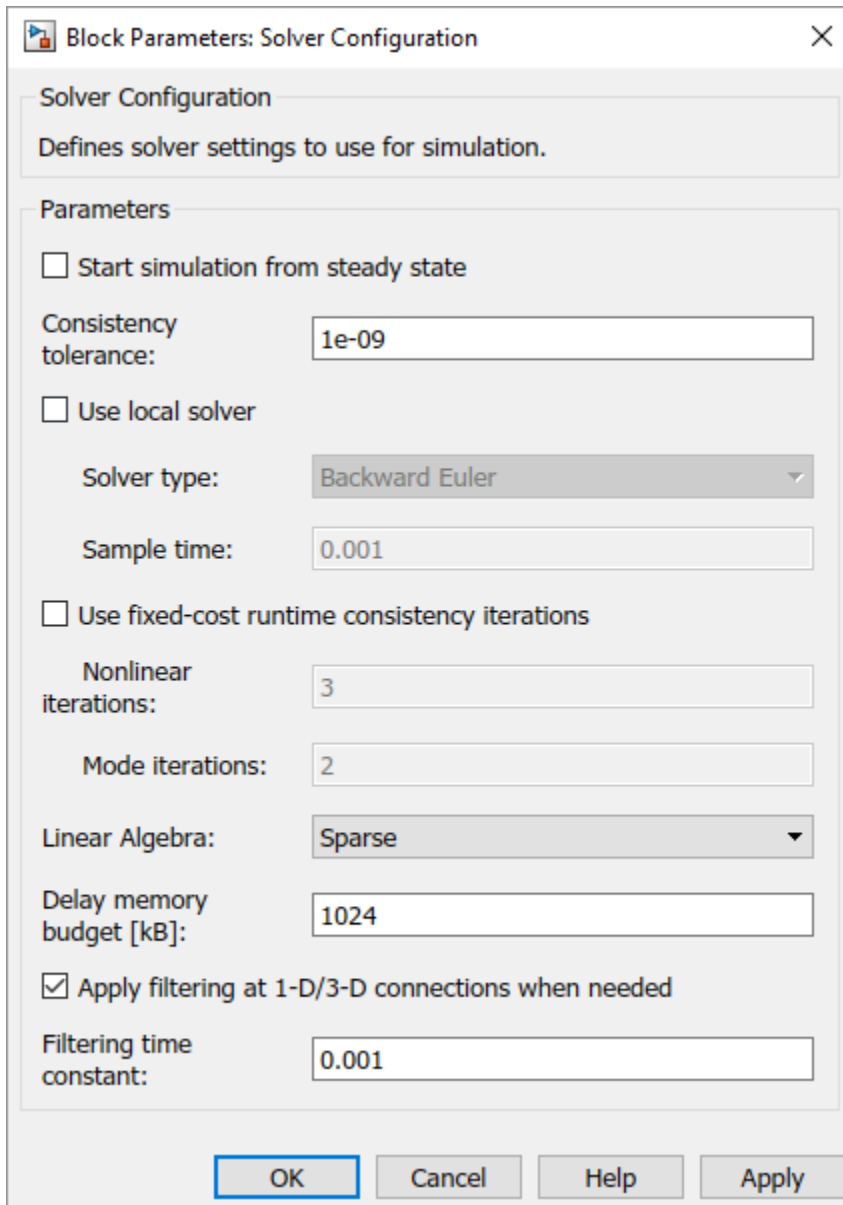
Parameter dependency tables also indicate which parameter options *enable* other parameters. For example, the next table shows the dependencies for the parameters for the Solver Configuration block from the **Simscape > Foundation Library > Utilities** library. The first column in the table contains row numbers, which are for reference only. The second column includes all parameters that are visible by default.

Parameters that affect the visibility and configurability of other parameters are shown in bold text. The options for the parameters that affect other parameters are shown as selected and cleared check boxes in the following row of the table. Parameters that are shown in plain text are enabled but do not affect other parameters. Parameters that are not enabled are not shown.

### Solver Configuration Block Parameter Dependencies

Row	Parameters and Options			
1	Start simulation from steady state			
2	Consistency tolerance			
3	<b>Use local Solver</b>			
4	<input type="checkbox"/>		<input checked="" type="checkbox"/>	
5			Solver type	
6			Sample time	
7	<b>Use fixed-cost runtime consistency iterations</b>		<b>Use fixed-cost runtime consistency iterations</b>	
8	<input type="checkbox"/>		<input checked="" type="checkbox"/>	
9			Nonlinear iterations	
10			Mode iterations	
11	Linear Algebra			
12	Equation formulation			
13	Delay memory budget [kB]			
14	<b>Apply filtering at 1-D/3-D connections when needed</b>			
15	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
16	Filtering time constant			

The figure shows the parameters that are visible by default on the block dialog box for the Solver Configuration block. The dimmed parameters, for example, **Solver type**, are not enabled.

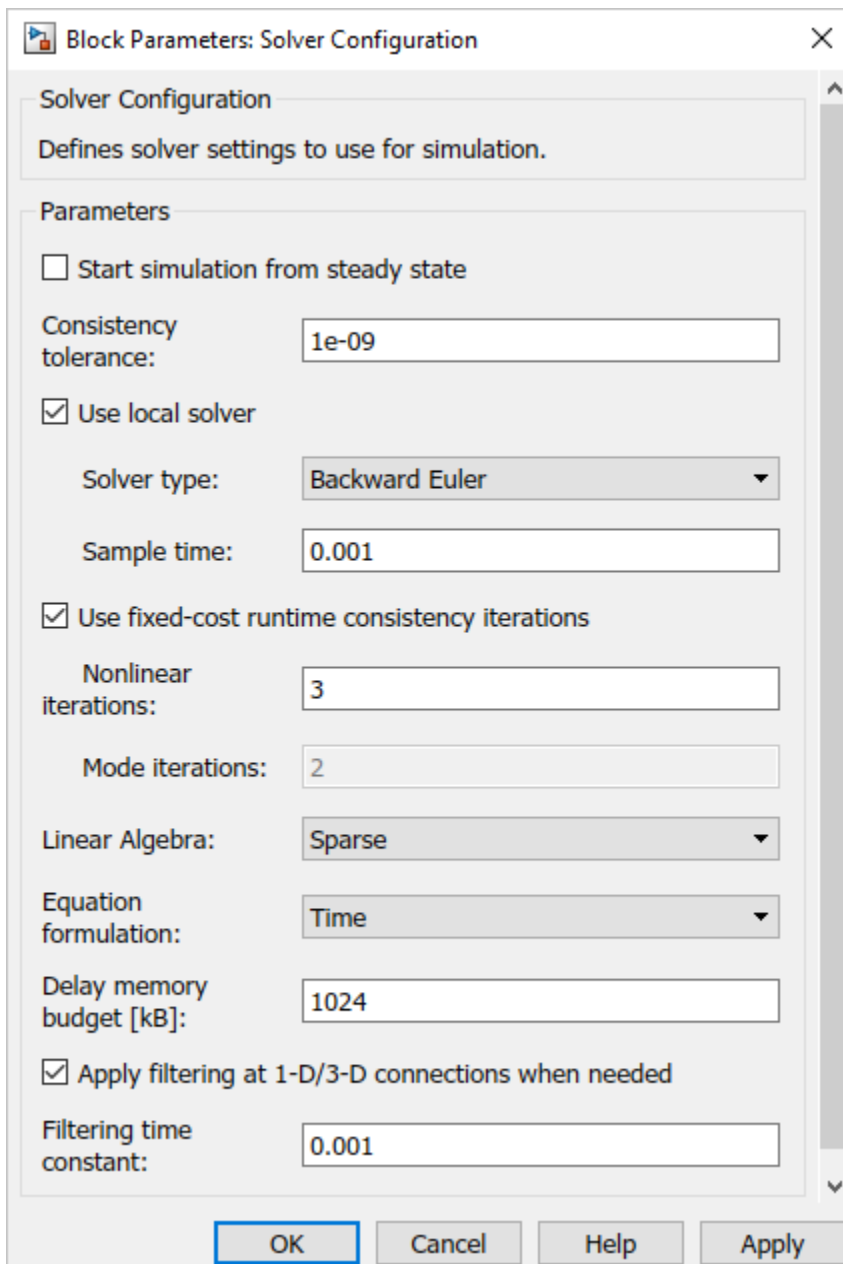


If you select the **Use local solver** check box, the **Use fixed-cost runtime consistency** check box becomes selected and these parameters become enabled.

- **Solver type**
- **Nonlinear iterations**
- **Sample time**

Selecting **Use local solver** does not enable the **Mode iterations** parameter. The **Mode iterations** parameter is only enabled when the **Use local solver** check box is cleared while **Use fixed-cost runtime consistency** check box is selected.

The table shows these dependencies in rows 3-10.



Block Parameters: Solver Configuration

Solver Configuration  
Defines solver settings to use for simulation.

Parameters

Start simulation from steady state

Consistency tolerance:

Use local solver

Solver type:

Sample time:

Use fixed-cost runtime consistency iterations

Nonlinear iterations:

Mode iterations:

Linear Algebra:

Equation formulation:

Delay memory budget [kB]:

Apply filtering at 1-D/3-D connections when needed

Filtering time constant:

OK Cancel Help Apply

