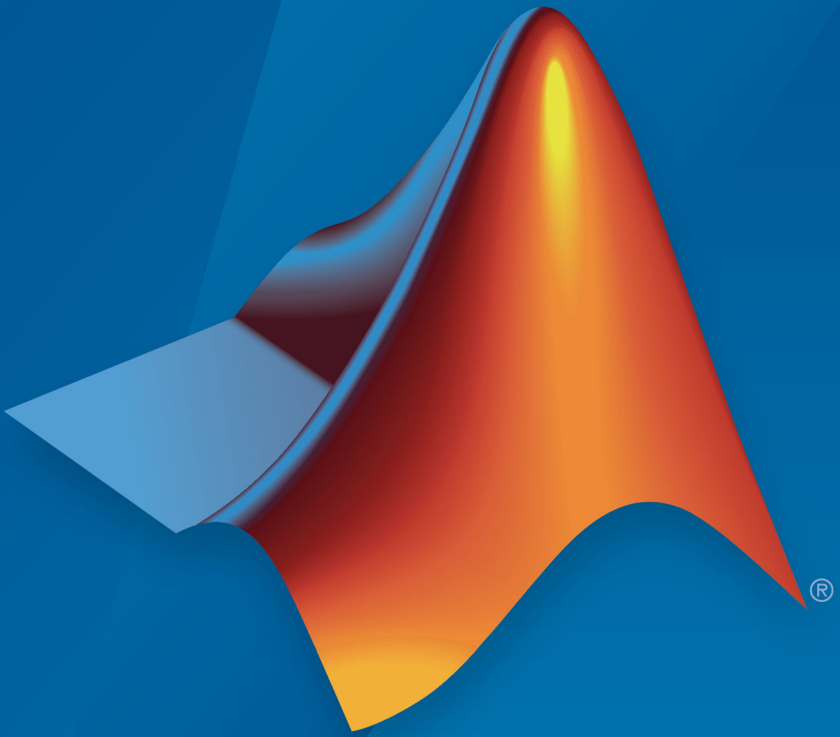
Simscape™ Power Systems™

Reference (Simscape™ Components)



MATLAB® & SIMULINK®

R2017b

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

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

Simscape™ Power Systems™ Reference (Simscape™ Components)

© COPYRIGHT 2013–2017 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

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 *SimPowerSystems™ Reference (Third Generation)*)
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 *SimPowerSystems™ Reference (Simscape™ Components)*)
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)

Contents

1 Blocks — Alphabetical List

2 Functions — Alphabetical List

Abbreviations and Naming Conventions in Simscape Components Libraries

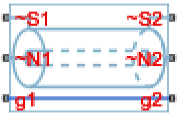
1

Blocks — Alphabetical List

AC Cable

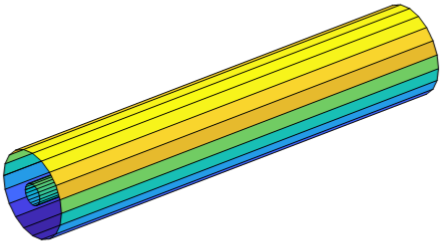
Model three-phase AC power cabling

Library: Simscape / Power Systems / Simscape Components / Passive Devices



Description

The AC Cable 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.



The block has two variants:

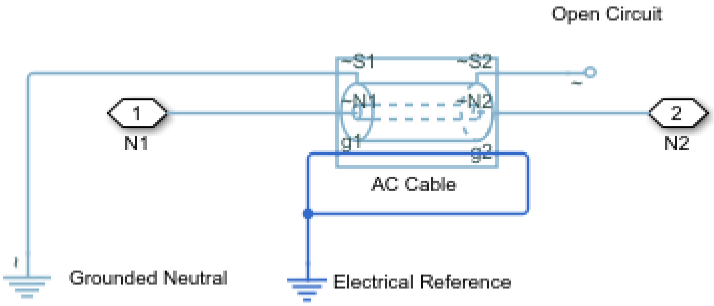
- Composite three-phase variant (default) --- Contains three-phase connection ports for the sheaths and phases and a single-phase connection port for each electrical reference node.
- Expanded three-phase variant --- Contains single-phase connection ports for each sheath, phase, and electrical reference node.

The AC Cable 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 block to a source block, include source impedance using one of these methods:

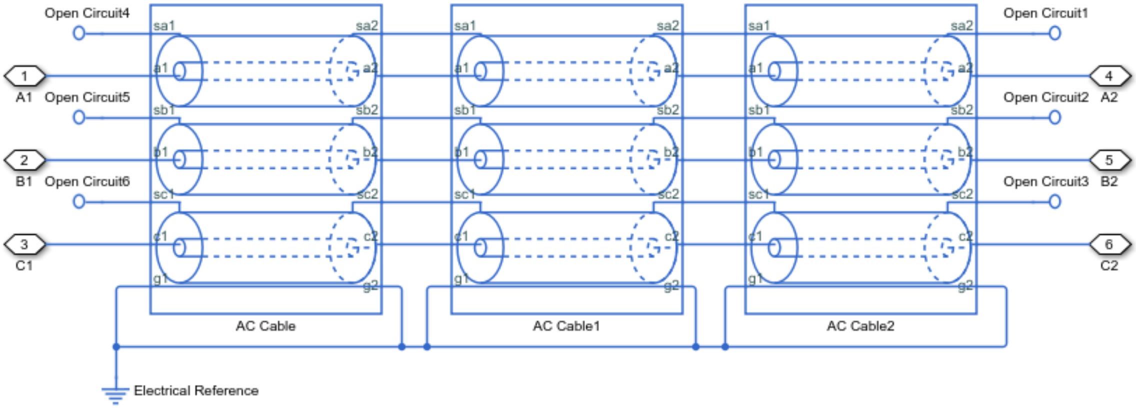
- Configure the source block to include impedance.
- Insert a block that models impedance between the source block and the AC Cable block.

To model unbonded sheaths, connect the unbonded sheaths to an Open Circuit block. The figure shows a model of single-point bonding using the composite three-phase variant of the block.

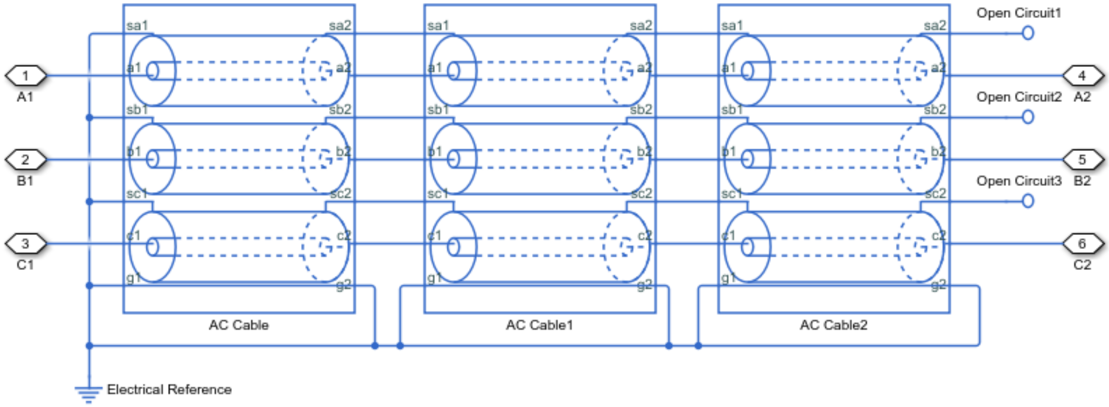


For high performance modeling, in terms of simulation speed, use a single AC Cable block. To improve model fidelity in terms of frequency behavior, connect several AC Cable blocks in series. For series-connected blocks, the sheaths and main conductors act as coupled transmission lines with perfect transposition of the phases. The number of AC Cable 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:

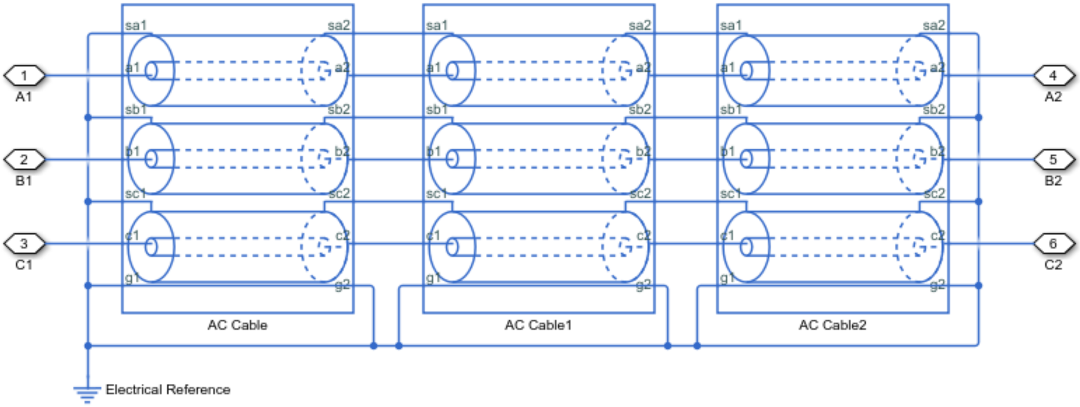
- Unbonded continuous cables



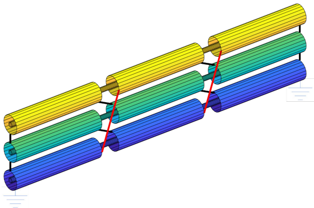
- Single-point bonded continuous cables



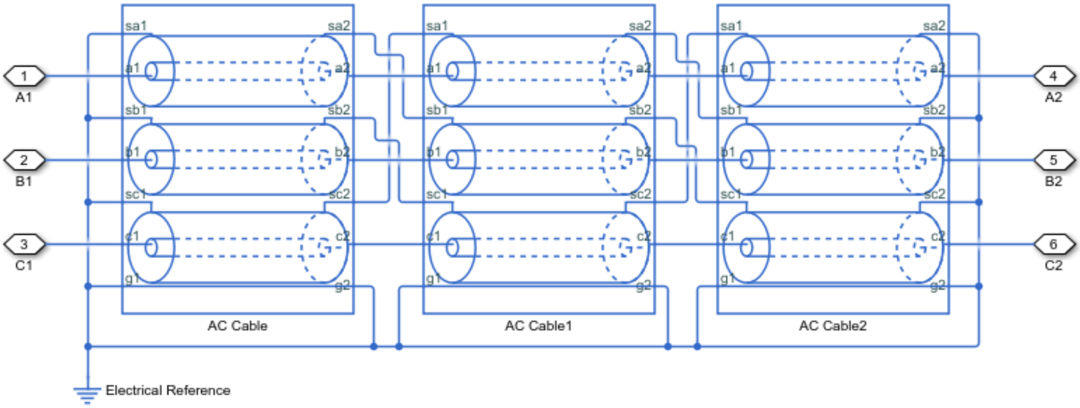
- Double-point bonded continuous cables



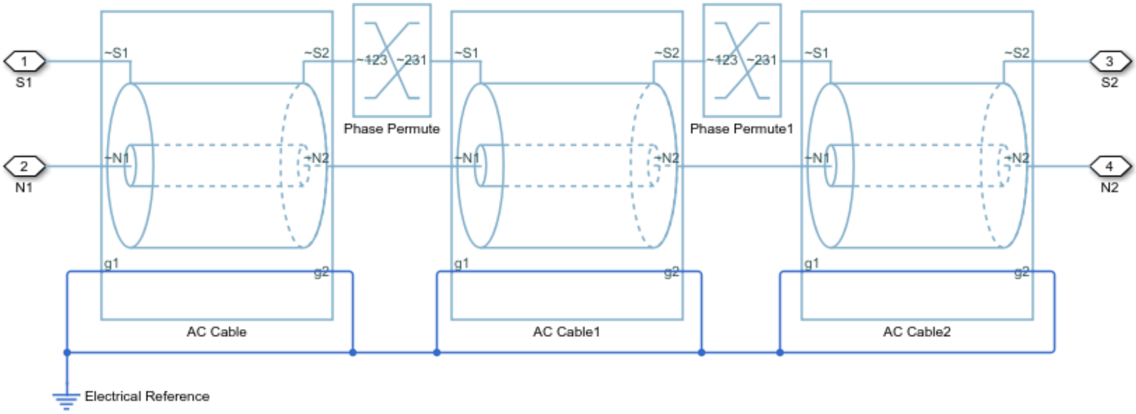
You can also model cross-bonded cables using the AC Cable block.



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”.

AC Cable Model

The AC Cable 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

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

$$R_a = R_a^{'} l$$

$$R_g = R_{\mathrm{return}}^{'} l,$$

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

- For a return parameterization based on distance and resistance $R_{\mathrm{return}}^{'} = R_g^{'}$.
- For a return parameterization based on frequency and Earth resistivity $R_{\mathrm{return}}^{'} = \pi^2 10^{-7} f$

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'_{return} is the resistance per unit length of the return. The value of R'_{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

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

$$C_{as_a} = \frac{2\pi\varepsilon_r\varepsilon_0 l}{\ln\left(\frac{r_s}{r_a}\right)}$$

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

$$C_{s_a g} = \frac{\pi \varepsilon_{\text{env}} \varepsilon_0 l}{\ln\left(\frac{D_{\text{return}}}{r_s}\right)},$$

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

- For a return parameterization based on distance and resistance $D_{\text{return}} = D_{\text{e}}$.
- For a return parameterization based on frequency and Earth resistivity

  $$D_{\text{return}} = 1650\sqrt{\frac{\rho}{2\pi f}}.$$

where:

- C is the capacitance matrix.
- C_{as_a} is the capacitance between each phase and the sheath of that phase.
- $C_{s_a g}$ is the capacitance between each sheath and return.
- ϵ_r is the permittivity of the dielectric.
- ϵ_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.
- GMR is the geometric mean radius of the conductor. For a single-strand conductor, $GMR = r_{\text{strand}} e^{-\frac{1}{4}}$, where r_{strand} is the radius of the strand.
- ϵ_{env} is the permittivity of the material between the sheathed lines and the return path.
- 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.
- ρ is the effective Earth resistivity for an Earth-return.
- f is the frequency that is used to determine the return path properties.

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:

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

$$D_a = L_a + L_g - 2M_{ag}$$

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

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

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

$$D_s = L_s + L_g - 2M_{sg}$$

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

$$\delta = L_g + M_{as_a} - M_{ag} - M_{sg}$$

$$\alpha = L_g + M_{as_b} - M_{ag} - M_{sg}$$

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

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 $d_{ab} = D_{ab}\sqrt[3]{2}$.

$$A = L_g + M_{ab} - 2M_{ag}$$

$$S = L_g + M_{s_a s_b} - 2M_{sg},$$

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.
- L_g is the partial self-inductance of the Earth- or neutral-return.
- 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, 2×10^{-7} is equal to $\mu_0 / 2\pi$, because permeability of free space, μ_0, is equal to 1.257×10^{-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.
- δ is the effective mutual inductance between a phase and the sheath of that phase.
- α 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_{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

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

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:

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

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

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

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 port associated with sheath 1.

`~N1` — Neutral
electrical

Expandable three-phase port associated with neutral 1.

`g1` — Ground
electrical

Electrical conserving port associated with ground 1.

`~S2` — Sheath
electrical

Expandable three-phase port associated with sheath 2.

`~N2` — Neutral
electrical

Expandable three-phase port associated with neutral 2.

`g2` — Ground
electrical

Electrical conserving port associated with ground 2.

Parameters

`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 $GMR * e^{\frac{1}{4}}$.

`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.

Model Examples

See Also

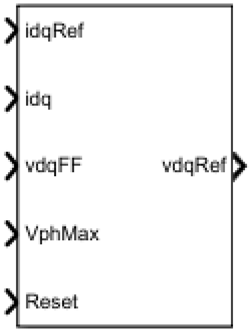
Phase Permute

Introduced in R2017b

ASM Current Controller

Discrete-time PI-based asynchronous machine current control

Library: Simscape / Power Systems / Simscape Components / Control / ASM Control



Description

The ASM Current Controller implements a discrete-time proportional-integral (PI) based asynchronous machine (ASM) current controller in the rotor *d-q* reference frame. You typically use the ASM 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 ASM 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:

$$v_d^{ref} = \left(K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) \left(i_d^{ref} - i_d \right) + v_{d\_FF},$$

and

$$v_q^{ref} = \left(K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) \left(i_q^{ref} - i_q \right) + v_{q\_FF},$$

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:

$$v_1^{sat} = \min\left(\max\left(v_1^{unsat}, -V_{ph\_max}\right), V_{ph\_max}\right)$$

and

$$v_2^{sat} = \min\left(\max\left(v_2^{unsat}, -V_{2\_max}\right), V_{2\_max}\right),$$

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 $v_{2\_max} = \sqrt{\left(V_{ph\_max}\right)^2 - \left(v_1^{sat}\right)^2}$.

In the case of d-q equivalence, the direct and quadrature axes have the same priority, and the constrained voltages are:

$$v_d^{sat} = \min\left(\max\left(v_d^{unsat}, -V_{d\_max}\right), V_{d\_max}\right)$$

and

$$v_q^{sat} = \min\left(\max\left(v_q^{unsat}, -V_{q\_max}\right), V_{q\_max}\right),$$

where:

$$V_{d\_max} = \frac{V_{ph\_max}\left|v_d^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}\left|v_q^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

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:

$$K_{i\_id} + K_{aw\_id}\left(v_d^{sat} - v_d^{unsat}\right)$$

and

$$K_{i\_iq} + K_{aw\_iq}\left(v_q^{sat} - v_q^{unsat}\right),$$

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 asynchronous motor, in A.

Data Types: `single` | `double`

`idq` — Measured currents

vector

Actual d- and q-axis currents of the controlled asynchronous motor, 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 asynchronous motor, in V.

Data Types: `single` | `double`

Parameters

Control Parameters

`D-axis current proportional gain` — ***d*-axis proportional gain**
`1` (default)

Proportional gain of the PI controller used for direct-axis current control.

`D-axis current integral gain` — ***d*-axis integral gain**
`100` (default)

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)

Anti-windup gain of the PI controller used for direct-axis current control.

`Q-axis current proportional gain` — ***q*-axis proportional gain**
`1` (default)

Proportional gain of the PI controller used for quadrature-axis current control.

`Q-axis current integral gain` — ***q*-axis integral gain**
`100` (default)

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)

Anti-windup gain of the PI controller used for quadrature-axis current control.

`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?” (Simulink) and “Specify Sample Time” (Simulink).

`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.

See Also

Blocks

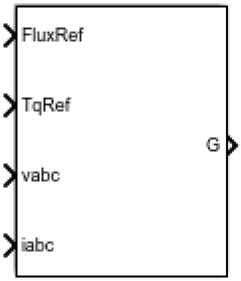
ASM Current Controller | ASM Direct Torque Control | ASM Field-Oriented Control | ASM Flux Observer | ASM Scalar Control

Introduced in R2017b

ASM Direct Torque Control

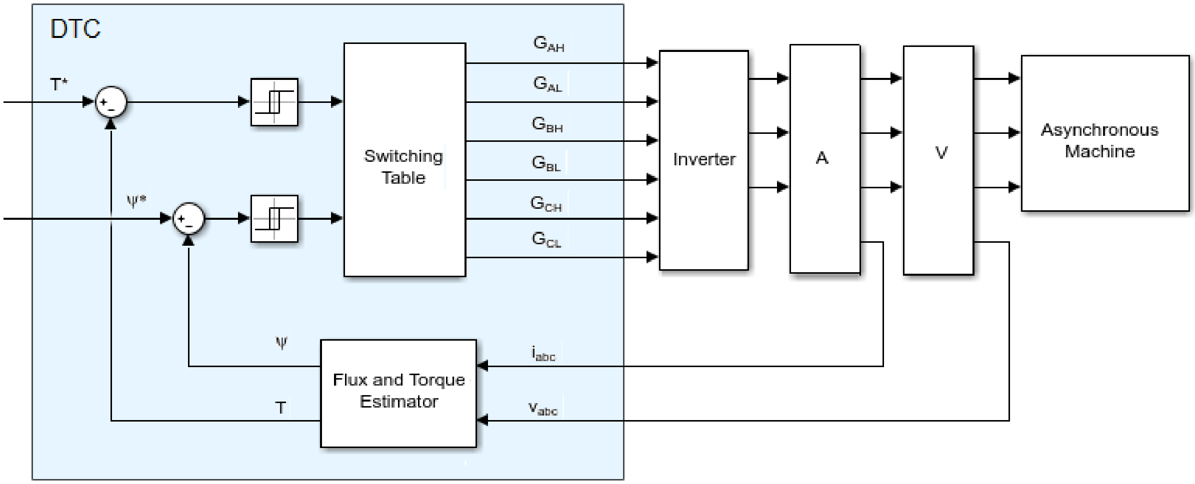
Asynchronous machine direct torque control

Library: Simscape / Power Systems / Simscape Components / Control / ASM Control



Description

The ASM Direct Torque Control implements an asynchronous machine (ASM) direct torque control (DTC) structure. The figure shows the equivalent circuit for the ASM DTC.



Equations

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:

$$\psi_\alpha = (v_\alpha - i_\alpha R_s)\frac{T_s z}{z-1}$$

and

$$\psi_\beta = \left(v_\beta - i_\beta R_s\right)\frac{T_s z}{z-1}$$

where:

- v_α is α-axis voltage.
- i_α is α-axis current.
- R_s is the stator resistance.
- Ψ_α is the α-axis stator flux.
- v_β is β-axis voltage.
- i_β is β-axis current.
- Ψ_β is the β-axis stator flux.

The block calculates the torque and flux as:

$$T = \frac{3p}{2}\left(\psi_\alpha i_\beta - \psi_\beta i_\alpha\right)$$

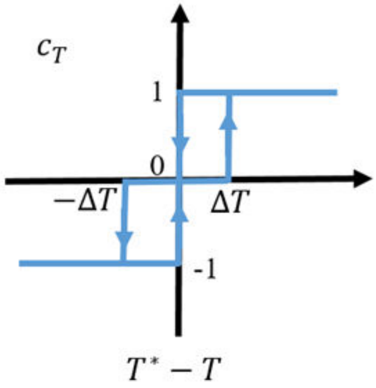
and

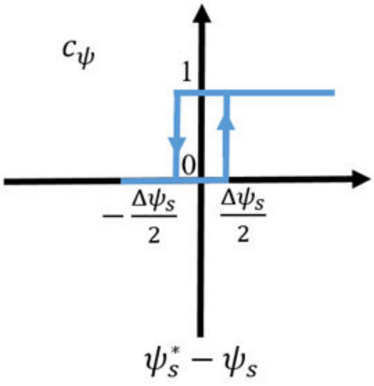
$$\psi_s = \sqrt{\psi_\alpha^2 + \psi_\beta^2}$$

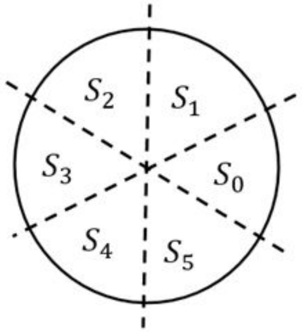
where:

- p is the number of pole pairs.
- Ψ_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*ψ, *c*$_T$*S*(*θ*)		*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
	$c_T = -1$	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 (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.

`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)` — **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?” (Simulink) and “Specify Sample Time” (Simulink).

Model Examples

Asynchronous Machine Direct Torque Control

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.

See Also

Blocks

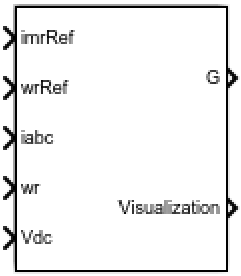
ASM Current Controller | ASM Field-Oriented Control | ASM Flux Observer | ASM Scalar Control

Introduced in R2017b

ASM Field-Oriented Control

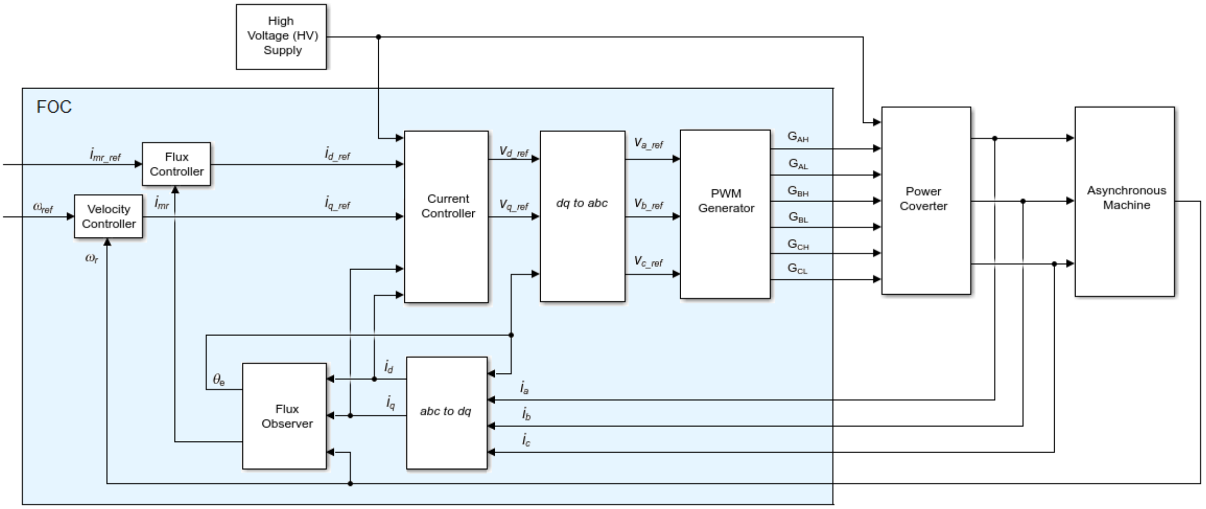
Asynchronous machine field-oriented control

Library: Simscape / Power Systems / Simscape Components / Control / ASM Control



Description

The ASM Field-Oriented Controller block implements an asynchronous machine (ASM) 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:

- ω_r is the measured angular velocity.
- ω_{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.
- θ_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.

Model Examples

Electric Engine Dyno

See Also

Blocks

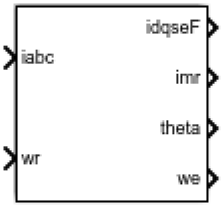
ASM Current Controller | ASM Direct Torque Control | ASM Flux Observer | ASM Scalar Control

Introduced in R2017b

ASM Flux Observer

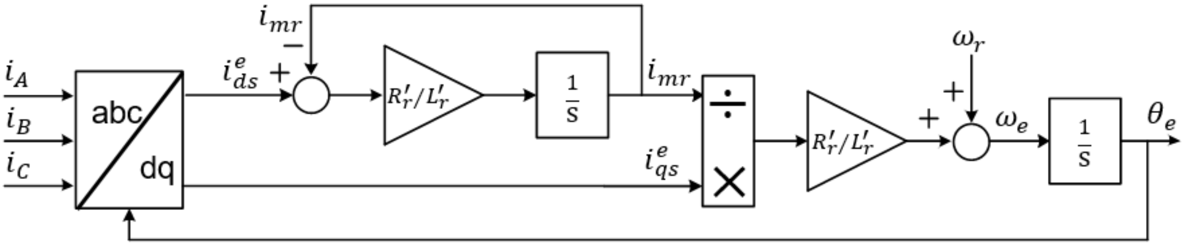
Asynchronous machine flux observer

Library: Simscape / Power Systems / Simscape Components / Control / Observers



Description

The ASM Flux Observer block obtains the synchronous speed, ω_e, and electrical angle, θ_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 ASM Flux Observer block uses these relationships:

$$\lambda'^{e}_{dr} = L'_r i'^{e}_{dr} + L_M i^{e}_{ds} \triangleq L_M i_{mr},$$

$$0 = R'_r i'^{e}_{dr} - (\omega_e - \omega_r)\lambda'^{e}_{qr} + \frac{d\lambda'^{e}_{dr}}{dt},$$

and

$$i_{qr}^{'e} = -\frac{L_M}{L_r^{'}} i_{qs}^{e},$$

in these combined forms:

$$i_{dr}^{'e} = \frac{L_M}{L_r^{'}}\left(i_{mr} - i_{ds}^{e}\right)$$

$$\frac{di_{mr}}{dt} = \frac{R_r^{'}}{L_r^{'}}\left(i_{ds}^{e} - i_{mr}\right)$$

and

$$\omega_e = \omega_r + \frac{R_r^{'}}{L_r^{'}}\frac{i_{qs}^{e}}{i_{mr}}$$

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.
- ω_e is the synchronous speed.
- ω_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)` — 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?” (Simulink) and “Specify Sample Time” (Simulink).

Model Examples

Three-Phase Asynchronous Drive with Sensor Control

References

[1] Vas, P. *Electrical Machines and Drives: A Space-vector Theory Approach*. New York: Oxford University Press, 1992.

See Also

Blocks

ASM Current Controller | ASM Direct Torque Control | ASM Field-Oriented Control | ASM Scalar Control

Introduced in R2017b

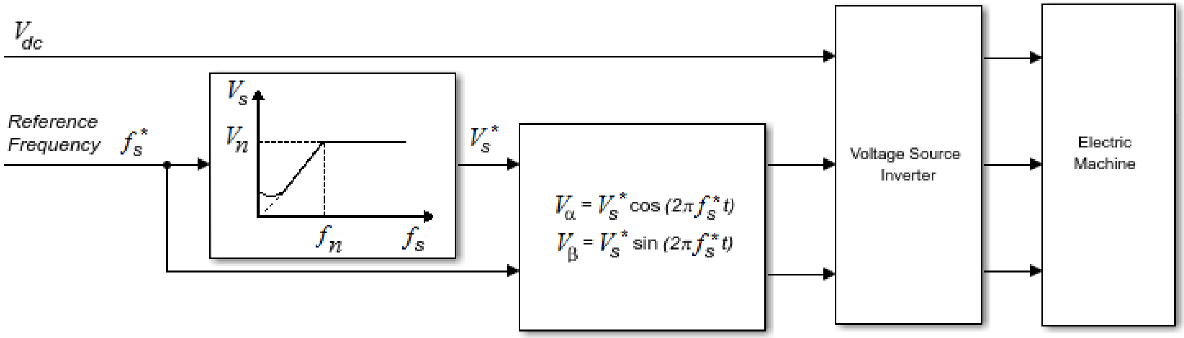
ASM Scalar Control

Asynchronous machine V/f control

Library: Simscape / Power Systems / Simscape Components / Control / ASM Control

Description

The ASM Scalar Control block implements an asynchronous machine (ASM) 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 ASM 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_α and V_β 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)` — 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?” (Simulink) and “Specify Sample Time” (Simulink).

Model Examples

Asynchronous Machine Scalar Control

See Also

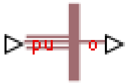
Blocks

ASM Current Controller | ASM Direct Torque Control | ASM Field-Oriented Control | ASM Flux Observer

Introduced in R2017b

Asynchronous Machine Measurement

Per-unit measurement from asynchronous machine



Library

Machines

Description

The Asynchronous Machine Measurement block outputs a per-unit measurement associated with a connected Asynchronous Machine Squirrel Cage or Asynchronous Machine Wound Rotor block. The input of the Asynchronous Machine Measurement block connects to the `pu` output port of the asynchronous machine block.

You set the **Output** parameter to a per-unit measurement associated with the asynchronous machine. Based on the value you select, the Asynchronous 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 Asynchronous Machine Measurement block outputs a per-unit measurement from the asynchronous 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	$\sqrt{pu\_vds^2 + pu\_vqs^2}$
Terminal current	$\sqrt{pu\_ids^2 + pu\_iqs^2}$
Power factor angle (rad)	power_factor_angle = atan2(pu_Qt, pu_Pt)
Power factor	cos(power_factor_angle)

Parameters

Output

Per-unit measurement from asynchronous machine. The default value is `Electrical torque`.

Ports

The block has the following ports:

`pu`

Physical signal vector port associated with per-unit measurements from a connected asynchronous machine. The vector elements are:

- pu_torque
- pu_velocity
- pu_vds
- pu_vqs
- pu_v0s
- pu_ids
- pu_iqs
- pu_i0s

`o`

Per-unit measurement output port.

See Also

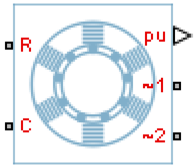
Asynchronous Machine Squirrel Cage (fundamental)

Asynchronous Machine Wound Rotor (fundamental)

Introduced in R2013b

Asynchronous Machine Squirrel Cage (fundamental)

Squirrel-cage-rotor asynchronous machine with fundamental parameterization



Library

Machines / Asynchronous Machine (Squirrel Cage)

Description

The Asynchronous Machine Squirrel Cage (fundamental) block models a squirrel-cage-rotor asynchronous machine using fundamental parameters. A squirrel-cage-rotor asynchronous 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 Asynchronous Machine Wound Rotor (fundamental) or the Asynchronous Machine Wound Rotor (fundamental, SI) 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.

Electrical Defining Equations

The asynchronous machine equations are expressed with respect to a synchronous reference frame, defined by

$$\theta_e(t) = \int_0^t 2\pi f_{rated}\, dt,$$

wheref_{rated} is the value of the **Rated electrical frequency** parameter.

Park's transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. Park's 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 θ_e is the electrical angle.

Park's 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.
- ω_{base} is the per-unit base electrical speed.

- ψ_{ds}, ψ_{qs}, and ψ_{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.
- ψ_{dr} and ψ_{qr} are the d-axis and q-axis rotor flux linkages.
- ω is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- ω_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

$L_{rrd} = L_{lrd} + L_m$.

Plotting and Display Options

You can perform plotting and display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

All default parameter values are based on a machine delta-winding configuration.

- “Main Tab” on page 1-52

- “Impedances Tab” on page 1-52
- “Initial Conditions Tab” on page 1-53

Main Tab

Rated apparent power

Rated apparent power of the asynchronous machine. The default value is `15e3 V*A`.

Rated voltage

RMS line-line voltage. The default value is `220 V`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Impedances Tab

Stator resistance, Rs (pu)

Stator resistance. The default value is `0.0258`.

Stator leakage inductance, Lls (pu)

Stator leakage inductance. The default value is `0.0930`.

Referred rotor resistance, Rr' (pu)

Rotor resistance referred to the stator. The default value is `0.0145`.

Referred rotor leakage inductance, Llr' (pu)

Rotor leakage inductance referred to the stator. The default value is `0.0424`.

Magnetizing inductance, Lm (pu)

Magnetizing inductance, that is, the peak value of stator-rotor mutual inductance. The default value is `1.7562`.

Stator zero-sequence inductance, L0 (pu)

Stator zero-sequence inductance. The default value is `0.0930`.

Initial Conditions Tab

Initial rotor angle

Initial rotor angle. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Initial stator *d*-axis flux linkage. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Initial stator zero-sequence flux linkage. The default value is `0` pu.

Initial rotor d-axis magnetic flux linkage

Initial rotor *d*-axis flux linkage. The default value is `0` pu.

Initial rotor q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0` pu.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`~1`

Expandable three-phase port associated with the stator positive-end connections.

`~2`

Expandable three-phase port associated with the stator negative-end connections.

`pu`

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

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.

See Also

Asynchronous Machine Measurement | Asynchronous Machine Squirrel Cage (fundamental, SI) | Asynchronous Machine Wound Rotor (fundamental, SI) | Asynchronous Machine Wound Rotor (fundamental)

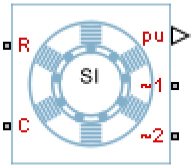
Topics

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

Introduced in R2013b

Asynchronous Machine Squirrel Cage (fundamental, SI)

Squirrel-cage-rotor asynchronous machine with fundamental parameterization in SI units



Library

Machines / Asynchronous Machine (Squirrel Cage)

Description

The Asynchronous Machine Squirrel Cage (fundamental, SI) block models a squirrel-cage-rotor asynchronous machine using fundamental parameters expressed in the International System of Units (SI). A squirrel-cage asynchronous 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 Asynchronous Machine Wound Rotor (fundamental, SI) block or the Asynchronous Machine Wound Rotor (fundamental) 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.

Electrical Defining Equations

The Asynchronous Machine Squirrel Cage (fundamental, SI) 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 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.

Park’s transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. Park’s 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 θ_e is the electrical angle.

Park’s 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.

- ω_{base} is the per-unit base electrical speed.
- ψ_{ds}, ψ_{qs}, and ψ_{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.
- ψ_{dr} and ψ_{qr} are the d-axis and q-axis rotor flux linkages.
- ω is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- ω_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

$$L_{rrd} = L_{lrd} + L_m.$$

Plotting and Display Options

You can perform plotting and display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

All default parameter values are based on a machine delta-winding configuration.

- “Main Tab” on page 1-59
- “Impedances Tab” on page 1-59
- “Initial Conditions Tab” on page 1-60

Main Tab

Rated apparent power

Rated apparent power of the asynchronous machine. The default value is `15e3 V*A`.

Rated voltage

RMS line-line voltage. The default value is `220 V`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Impedances Tab

Stator resistance, Rs

Stator resistance. The default value is `0.25 Ohm`.

Stator leakage reactance, Xls

Stator leakage reactance. The default value is `0.9 Ohm`.

Referred rotor resistance, Rr'

Rotor resistance referred to the stator. The default value is `0.14 Ohm`.

Referred rotor leakage reactance, Xlr'

Rotor leakage reactance referred to the stator. The default value is `0.41 Ohm`.

Magnetizing reactance, Xm

Magnetizing reactance The default value is `17 Ohm`.

Stator zero-sequence reactance, X0

Stator zero-sequence reactance. The default value is `0.9 Ohm`.

Initial Conditions Tab

Initial rotor angle

Initial rotor angle. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Initial stator *d*-axis flux linkage. The default value is `0 Wb`.

Initial stator q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0 Wb`.

Initial stator zero-sequence magnetic flux linkage

Initial stator zero-sequence flux linkage. The default value is `0 Wb`.

Initial rotor d-axis magnetic flux linkage

Initial rotor *d*-axis flux linkage. The default value is `0 Wb`.

Initial rotor q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0 Wb`.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`~1`

Expandable three-phase port associated with the stator positive-end connections.

`~2`

Expandable three-phase port associated with the stator negative-end connections.

`pu`

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

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.

See Also

Asynchronous Machine Measurement | Asynchronous Machine Squirrel Cage (fundamental) | Asynchronous Machine Wound Rotor (fundamental) | Asynchronous Machine Wound Rotor (fundamental, SI)

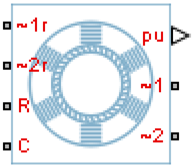
Topics

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

Introduced in R2015a

Asynchronous Machine Wound Rotor (fundamental)

Wound-rotor asynchronous machine with fundamental parameterization



Library

Machines / Asynchronous Machine (Wound Rotor)

Description

The Asynchronous Machine Wound Rotor (fundamental) block models a wound-rotor asynchronous machine using fundamental parameters. 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 Asynchronous Machine Squirrel Cage (fundamental) 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.

Electrical Defining Equations

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.

Park's transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. Park's 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 θ_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 θ_r and number of pole pairs N:

$$P_r = \frac{2}{3}\begin{bmatrix} \cos(\theta_e - N\theta_r) & \cos(\theta_e - N\theta_r - \frac{2\pi}{3}) & \cos(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ -\sin(\theta_e - N\theta_r) & -\sin(\theta_e - N\theta_r - \frac{2\pi}{3}) & -\sin(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's 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.
- ω_{base} is the per-unit base electrical speed.
- ψ_{ds}, ψ_{qs}, and ψ_{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

$$v_{dr} = \frac{1}{\omega_{base}} \frac{d\psi_{dr}}{dt} - (\omega - \omega_r)\psi_{qr} + R_{rd} i_{dr},$$

$$v_{qr} = \frac{1}{\omega_{base}} \frac{d\psi_{qr}}{dt} + (\omega - \omega_r)\psi_{dr} + R_{rd} i_{qr},$$

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 ~1r and ~2r.
- ψ_{dr}, ψ_{qr}, and ψ_{0r} are the d-axis, q-axis, and zero-sequence rotor flux linkages.
- ω is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- ω_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 ~1r to port ~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_{lrd}, and magnetizing inductance L_m are related by

$$L_{rrd} = L_{lrd} + L_m .$$

Plotting and Display Options

You can perform plotting and display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

All default parameter values are based on a machine delta-winding configuration.

- “Main Tab” on page 1-67
- “Impedances Tab” on page 1-67

- "Initial Conditions Tab" on page 1-68

Main Tab

Rated apparent power

Rated apparent power of the asynchronous machine. The default value is `15e3 V*A`.

Rated voltage

RMS line-line voltage. The default value is `220 V`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Impedances Tab

Stator resistance, Rs (pu)

Stator resistance. The default value is `0.0258`.

Stator leakage inductance, Lls (pu)

Stator leakage inductance. The default value is `0.0930`.

Referred rotor resistance, Rr' (pu)

Rotor resistance referred to the stator. The default value is `0.0145`.

Referred rotor leakage inductance, Llr' (pu)

Rotor leakage inductance referred to the stator. The default value is `0.0424`.

Magnetizing inductance, Lm (pu)

Magnetizing inductance, that is, the peak value of stator-rotor mutual inductance. The default value is `1.7562`.

Stator zero-sequence inductance, L0 (pu)

Stator zero-sequence inductance. The default value is `0.0930`.

Initial Conditions Tab

Initial rotor angle

Initial rotor angle. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Initial stator *d*-axis flux linkage. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Initial stator zero-sequence flux linkage. The default value is `0` pu.

Initial rotor d-axis magnetic flux linkage

Initial rotor *d*-axis flux linkage. The default value is `0` pu.

Initial rotor q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0` pu.

Initial rotor zero-sequence magnetic flux linkage

Initial rotor zero-sequence flux linkage. The default value is `0` pu.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`~1`

Expandable three-phase port associated with the stator positive-end connections.

`~2`

Expandable three-phase port associated with the stator negative-end connections.

`~1r`

Expandable three-phase port associated with the rotor positive-end connections.

`~2r`

Expandable three-phase port associated with the rotor negative-end connections.

`pu`

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

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.

See Also

Asynchronous Machine Measurement | Asynchronous Machine Squirrel Cage (fundamental) | Asynchronous Machine Squirrel Cage (fundamental, SI) | Asynchronous Machine Wound Rotor (fundamental, SI)

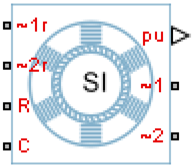
Topics

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

Introduced in R2013b

Asynchronous Machine Wound Rotor (fundamental, SI)

Wound-rotor asynchronous machine with fundamental parameterization in SI units



Library

Machines / Asynchronous Machine (Wound Rotor)

Description

The Asynchronous Machine Wound Rotor (fundamental, SI) block models a wound-rotor asynchronous machine using fundamental parameters expressed 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 Asynchronous Machine Squirrel Cage (fundamental) or Asynchronous Machine Squirrel Cage (fundamental,SI) 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.

Electrical Defining Equations

The Asynchronous Machine Wound Rotor (fundamental, SI) 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 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.

Park's transformation maps stator equations to a reference frame that is stationary with respect to the rated electrical frequency. Park's 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 θ_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 θ_r and number of pole pairs N:

$$P_r = \frac{2}{3}\begin{bmatrix} \cos(\theta_e - N\theta_r) & \cos(\theta_e - N\theta_r - \frac{2\pi}{3}) & \cos(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ -\sin(\theta_e - N\theta_r) & -\sin(\theta_e - N\theta_r - \frac{2\pi}{3}) & -\sin(\theta_e - N\theta_r + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's 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.
- ω_{base} is the per-unit base electrical speed.
- ψ_{ds}, ψ_{qs}, and ψ_{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

$$v_{dr} = \frac{1}{\omega_{base}} \frac{d\psi_{dr}}{dt} - (\omega - \omega_r)\psi_{qr} + R_{rd} i_{dr},$$

$$v_{qr} = \frac{1}{\omega_{base}} \frac{d\psi_{qr}}{dt} + (\omega - \omega_r)\psi_{dr} + R_{rd} i_{qr},$$

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 ~1r and ~2r.
- ψ_{dr}, ψ_{qr}, and ψ_{0r} are the d-axis, q-axis, and zero-sequence rotor flux linkages.
- ω is the per-unit synchronous speed. For a synchronous reference frame, the value is 1.
- ω_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 ~1r to port ~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_{lrd}, and magnetizing inductance L_m are related by

$$L_{rrd} = L_{lrd} + L_m.$$

Plotting and Display Options

You can perform display and plotting actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

All default parameter values are based on a machine delta-winding configuration.

- “Main Tab” on page 1-75
- “Impedances Tab” on page 1-75
- “Initial Conditions Tab” on page 1-76

Main Tab

Rated apparent power

Rated apparent power of the asynchronous machine. The default value is `15e3 V*A`.

Rated voltage

RMS line-line voltage. The default value is `220 V`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Impedances Tab

Stator resistance, Rs

Stator resistance. The default value is `0.25 Ohm`.

Stator leakage reactance, Xls

Stator leakage reactance. The default value is `0.9 Ohm`.

Referred rotor resistance, Rr'

Rotor resistance referred to the stator. The default value is `0.14 Ohm`.

Referred rotor leakage reactance, Xlr'

Rotor leakage reactance referred to the stator. The default value is `0.41 Ohm`.

Magnetizing reactance, Xm

Magnetizing reactance The default value is `17 Ohm`.

Stator zero-sequence reactance, X0

Stator zero-sequence reactance. The default value is `0.9 Ohm`.

Initial Conditions Tab

Initial rotor angle

Initial rotor angle. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Initial stator *d*-axis flux linkage. The default value is `0 Wb`.

Initial stator q-axis magnetic flux linkage

Initial stator *q*-axis flux linkage. The default value is `0 Wb`.

Initial stator zero-sequence magnetic flux linkage

Initial stator zero-sequence flux linkage. The default value is `0 Wb`.

Initial rotor d-axis magnetic flux linkage

Initial rotor *d*-axis flux linkage. The default value is `0 Wb`.

Initial rotor q-axis magnetic flux linkage

Initial rotor *q*-axis flux linkage. The default value is `0 Wb`.

Initial rotor zero-sequence magnetic flux linkage

Initial rotor zero-sequence flux linkage. The default value is `0 Wb`.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`~1`

Expandable three-phase port associated with the stator positive-end connections.

`~2`

Expandable three-phase port associated with the stator negative-end connections.

`~1r`

Expandable three-phase port associated with the rotor positive-end connections.

`~2r`

Expandable three-phase port associated with the rotor negative-end connections.

`pu`

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

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.

See Also

Asynchronous Machine Measurement | Asynchronous Machine Squirrel Cage (fundamental) | Asynchronous Machine Squirrel Cage (fundamental, SI) | Asynchronous Machine Wound Rotor (fundamental)

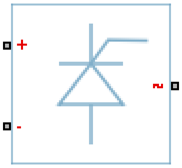
Topics

"Expand and Collapse Three-Phase Ports on a Block"
Three-Phase Asynchronous Machine Starting

Introduced in R2015a

Average-Value Inverter

Convert DC voltage to three-phase AC voltage with fixed power loss



Library

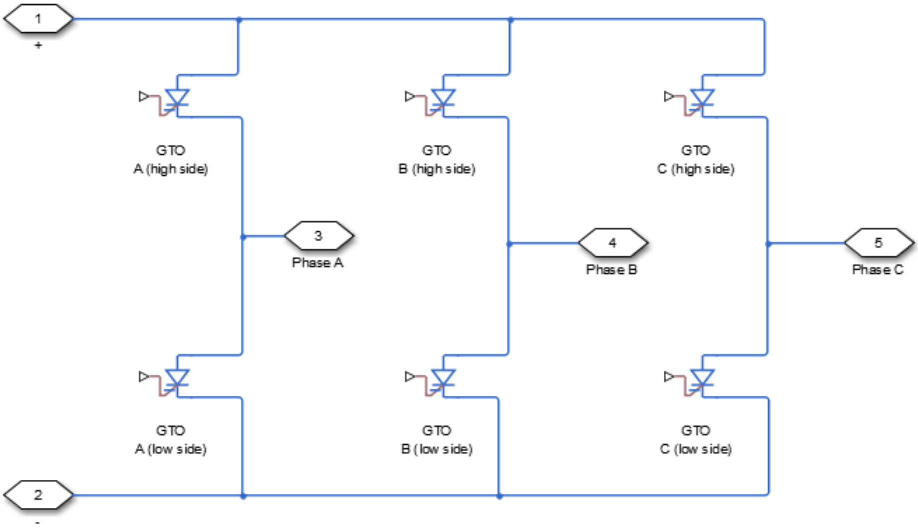
Semiconductors

Description

The Average-Value Inverter 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 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 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 1-82 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.
- φ 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.

Parameters

Rated AC frequency

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. The default value is `60 Hz`.

Phase shift

Phase shift in angular units. The default value is `0 deg`.

Ratio of rated AC voltage to rated DC voltage

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

Control Method	V_{RMS} *(line-line)*	Ratio of V_{RMS} *(line-line)* to v_{DC}
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

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

DC voltage for turn on

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

DC voltage for turn off

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

Ports

The block has the following ports:

\+

Electrical conserving port associated with the positive terminal

–

Electrical conserving port associated with the negative terminal

~

Expandable three-phase port

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.

See Also

Average-Value Rectifier | Converter | Rectifier | Three-Level Converter

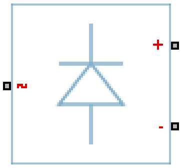
Topics

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

Introduced in R2015a

Average-Value Rectifier

Convert three-phase AC voltage to DC voltage with fixed power loss



Library

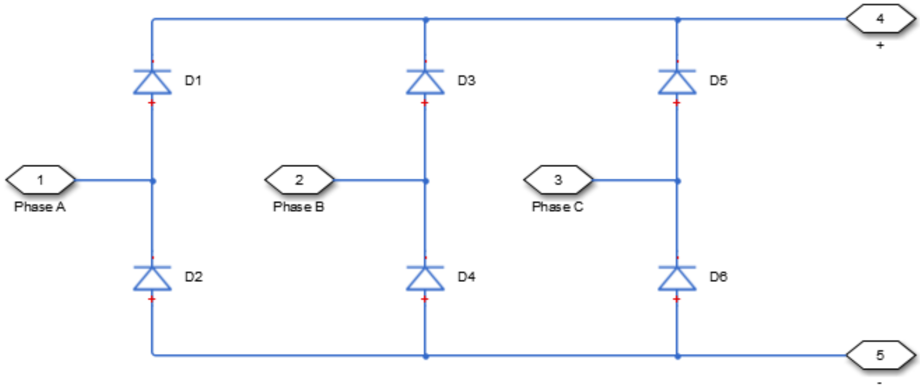
Semiconductors

Description

The Average-Value Rectifier 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 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 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:

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

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

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

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

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

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

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

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

and

$$\begin{bmatrix} i_a & i_b & i_c \end{bmatrix} = \frac{\begin{bmatrix} v_a & v_b & v_c \end{bmatrix} - v_{ref}}{R_{AC}},$$

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.

Parameters

Rated AC voltage

Rated voltage of the AC system. The default value is `4160 V`.

Fixed power loss

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. The default value is `1e3 W`.

Ports

The block has the following ports:

~

Expandable three-phase port

+

Electrical conserving port associated with the positive terminal

–

Electrical conserving port associated with the negative terminal

See Also

Average-Value Inverter | Converter | Rectifier | Three-Level Converter

Topics

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

Introduced in R2014b

Battery

Simple battery model

Library: Simscape / Power Systems / Simscape Components / Sources

Description

The Battery block represents a simple battery model. For information on how you can create a detailed battery model, see the “Lead-Acid Battery” (Simscape).

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

- **Uninstrumented | No thermal port** (default) — 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 variants have an extra physical signal port that outputs the internal state of charge. Use this functionality to change load behavior as a function of state of charge, without the complexity of building a charge state estimator.

The thermal port variants expose a thermal port, which represents the battery thermal mass. When you select this option, provide additional parameters to define battery behavior at a second temperature. For more information, see “Modeling Thermal Effects” on page 1-92.

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 whose voltage as a function of charge has the following reciprocal relationship:

$$V = V_0\left[1 - \frac{\alpha(1-x)}{1-\beta(1-x)}\right]$$

where:

- x is the ratio of current charge to rated battery capacity in ampere-hours (AH).
- V_0 is the voltage when the battery is fully charged, as defined by the **Nominal voltage** parameter.
- The block calculates the constants α and β to satisfy these battery conditions:
  - The battery voltage is zero when the charge is zero, that is, when $x = 0$.
  - When $x = AH1/AH$, that is, when the charge is the value of the **Charge AH1 when no-load volts are V1** parameter, the battery voltage is *V1* (the **Voltage V1 < Vnom when charge is AH1** parameter value).

The equation defines a reciprocal relationship between voltage and remaining charge. It approximates a real battery, but it does replicate the increasing rate of voltage drop at low charge values. It also ensures that the battery voltage becomes zero when the charge level is zero. This simple model has the advantage of requiring few parameters, and these parameters 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, which is sometimes referred to as battery fade. To enable battery fade, set the **Model battery fade?** parameter to `Include`. 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 λ_{AH}, λ_{R1}, and λ_{V1} on the **Ampere-hour rating**, **Internal**

resistance, and **Voltage V1 < Vnom when charge is AH1** parameter values, respectively. These multipliers, in turn, depend on the number of discharge cycles:

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

$$\lambda_{R1} = 1 + k_2 N^{0.5}$$

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

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

where:

- λ_{AH} is the multiplier for battery nominal capacity.
- λ_{R1} is the multiplier for battery series resistance.
- λ_{V1} is the multiplier for voltage *V1* to scale for a number of discharge cycles when the charge is *AH1*.
- 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 AH.
- *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 **Fade** 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 1-93.

Modeling Thermal Effects

For thermal variants of the block, 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:

$$V = V_{0T}\left[1-\left(\frac{\alpha_T(1-x)}{1-\beta_T(1-x)}\right)\right]$$

$$\alpha_T = \alpha\left(1+\lambda_\alpha\left(T-T_1\right)\right)$$

$$\beta_T = \beta\left(1+\lambda_\beta\left(T-T_1\right)\right)$$

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

where:

- T is the battery temperature.
- T_1 is the temperature at which nominal values for α and β are provided.
- λ_α, λ_β, and λ_V are the parameter temperature dependence coefficients for α, β, and V_0, respectively.

The internal series resistance (R_1) and self-discharge resistance (R_2) also become functions of temperature:

$$R_{1T} = R_1\left(1+\lambda_{R1}\left(T-T_1\right)\right)$$

$$R_{2T} = R_2\left(1+\lambda_{R2}\left(T-T_1\right)\right)$$

where λ_{R1} and λ_{R2} are the parameter temperature dependence coefficients. All the temperature dependence coefficients are determined from the corresponding values you provide at the nominal and second measurement temperatures.

The battery temperature is determined from:

$$M_{th}\dot{T} = i^2 R_{1T} + V^2/R_{2T}$$

where:

- M_{th} is the battery thermal mass.

- *i* is the battery output current.

Variables

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

When you model battery fade, the **Discharge cycles** variable lets you specify the number of charge-discharge cycles completed before the start of simulation. For more information, see “Modeling Battery Fade” on page 1-90. If you omit 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 variant of the battery, you provide fade data only for the reference temperature operation. The block applies the same derived λ_{AH}, λ_{R1}, and λ_{V1} multipliers to parameter values corresponding to the second temperature.
- When using the thermal block variants, use caution when operating at temperatures outside of the temperature range bounded by the **Measurement temperature** and **Second temperature measurement** values. The block uses linear interpolation for the derived equation coefficients, and simulation results might become nonphysical outside of the specified range. The block checks that the internal series resistance, self-discharge resistance, and nominal voltage always remain positive, and issues error messages if there is a violation.

Ports

Output

`q` — Battery charge level, C

physical signal

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

Dependencies

Enabled for the instrumented variants of the block: **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 representing 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 1-92.

Dependencies

Enabled for the thermal variants of the block: **Uninstrumented | Show thermal port** and **Instrumented | Show thermal port**.

Parameters

Main

`Nominal voltage` **— Output voltage when battery is fully charged**
12 V (default)

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

`Internal resistance` **— Battery internal resistance**
2 Ω (default)

Internal connection resistance of the battery.

`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 the charge drawn from the battery.
- `Finite` — The battery voltage decreases as the charge decreases.

`Ampere-hour rating` **— Nominal battery capacity when fully charged**
50 hr*A (default)

This parameter is the maximum (nominal) battery charge in AH. 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 1-93.

Dependencies

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

`Voltage V1 < Vnom when charge is AH1` **— Output voltage at charge level AH1**
11.5 V (default)

Battery output voltage when the charge level is AH1, as specified by the **Charge AH1 when no-load volts are V1** parameter.

Dependencies

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

`Charge AH1 when no-load volts are V1` — Charge level when the no-load output voltage is V1

25 hr*A (default)

Battery charge level, in AH, corresponding to the no-load output voltage specified by the **Voltage V1 < Vnom when charge is AH1** parameter.

Dependencies

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

`Model self-discharge resistance?` — Select whether to model the self-discharge resistance of the battery

`Omit` (default) | `Include`

Select whether to model the self-discharge resistance of the battery:

- `Omit` — Do not include resistance across the battery output terminals in the model.
- `Include` — Include resistance R2 across the battery output terminals in the model.

Dependencies

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

`Self-discharge resistance` — Resistance that represents battery self-discharge

2000 Ω (default)

Resistance across the battery output terminals that represents battery self-discharge.

Dependencies

Enabled when the **Model self-discharge resistance?** parameter is set to `Include`.

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

298.15 K (default)

Temperature T_1, at which the block parameters in the **Main** section are measured. For more information, see “Modeling Thermal Effects” on page 1-92.

Dependencies

Enabled for blocks with exposed thermal port.

Fade

`Model battery fade?` — Select whether to model battery performance deterioration with aging

`Omit` (default) | `Include`

Select whether to include battery fade modeling:

- `Omit` — The battery performance is not age-dependent.
- `Include` — 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 specified 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 1-90.

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)

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 **Model battery fade?** parameter is set to `Include`.

`Ampere-hour rating after N discharge cycles` — Maximum battery capacity after N discharge cycles

45 hr*A (default)

Maximum battery charge, in AH, after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

Dependencies

Enabled when the **Model battery fade?** parameter is set to `Include`.

`Internal resistance after N discharge cycles` — Battery internal resistance after N discharge cycles
2.02 Ω (default)

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

Dependencies

Enabled when the **Model battery fade?** parameter is set to `Include`.

`Voltage V1 at charge AH1 after N discharge cycles` — Output voltage at charge level AH1 after N discharge cycles
10.35 V (default)

Battery output voltage, at charge level in AH is *AH1*, after the number of discharge cycles specified by the **Number of discharge cycles, N** parameter.

Dependencies

Enabled when the **Model battery fade?** parameter is set to `Include`.

Temperature Dependence

This section appears only for blocks with exposed thermal port. For more information, see “Modeling Thermal Effects” on page 1-92.

`Nominal voltage at second measurement temperature` — Output voltage when battery is fully charged
12 V (default)

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

`Internal resistance at second measurement temperature` — Battery internal resistance
2.2 Ω (default)

Internal connection resistance of the battery.

`Voltage V1 at second measurement temperature` — Output voltage at charge level AH1
11.4 V (default)

Battery output voltage when the charge level is AH1, as specified by the **Charge AH1 when no-load volts are 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 Ω (default)

Resistance across the battery output terminals that represents battery self-discharge.

Dependencies

Enabled when the **Model self-discharge resistance?** parameter in the **Main** section is set to `Include`.

`Second measurement temperature` — Temperature at which the block parameters in this section are measured
273.15 K (default)

Temperature T_2, at which the block parameters in the **Temperature Dependence** section are measured. For more information, see “Modeling Thermal Effects” on page 1-92.

To specify the initial temperature at the start of simulation, use the high-priority **Temperature** variable. For more information, see “Variables” on page 1-93.

Thermal Port

This section appears only for blocks with exposed thermal port. For more information, see “Modeling Thermal Effects” on page 1-92.

`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.

Model Examples

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.

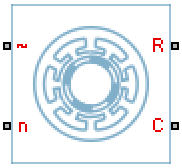
See Also

Current Source | Voltage Source

Introduced in R2013b

Brushless DC Motor

Three-winding brushless DC motor with trapezoidal flux distribution



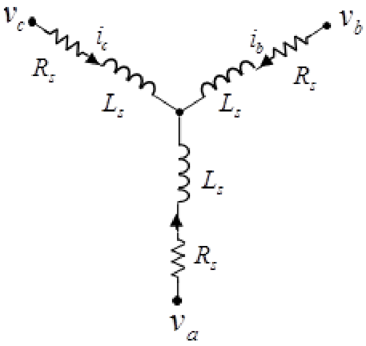
Library

Machines / Permanent Magnet Rotor

Description

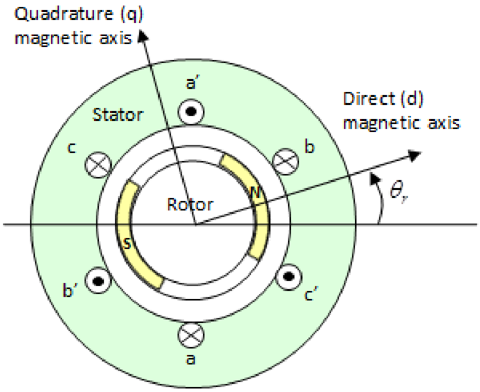
The Brushless DC Motor block models a permanent magnet synchronous machine with a three-phase wye-wound stator. The block has four options for defining the permanent magnet flux distribution as a function of rotor angle. Two options allow for simple parameterization by assuming a perfect trapezoid for 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

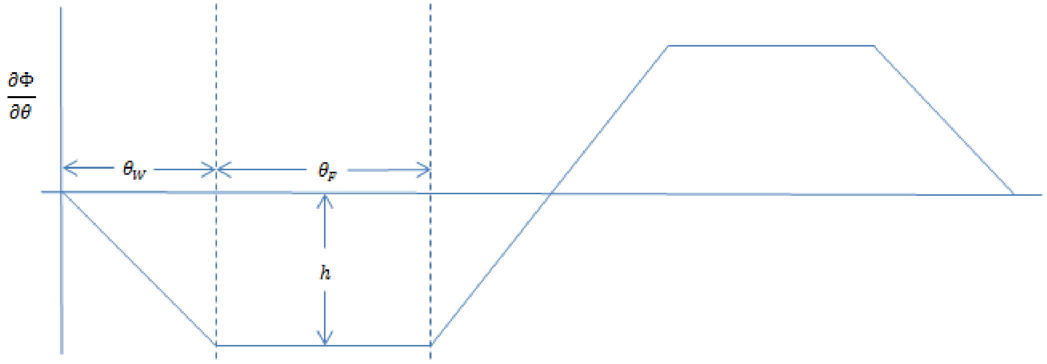
This figure shows the motor construction with a single pole-pair on the rotor.



For the axes convention in the preceding figure, the a-phase and permanent magnet fluxes are aligned when rotor angle θ_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:

- Φ is the permanent magnet flux linkage.
- θ is the rotor angle.
- ω 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 π/2,

$$\Phi_{max}=\frac{h}{2}(\theta_F+\theta_W),$$

where:

- Φ_{max} is the permanent magnet flux linkage.
- h is the rate of change of flux profile height.
- θ_F is the rotor angle range over which the back emf that the permanent magnet flux induces in the stator is constant.
- θ_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.
- $$\frac{d\psi_a}{dt}, \frac{d\psi_b}{dt}, \text{ and } \frac{d\psi_c}{dt}$$

  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:

- ψ_a, ψ_b, and ψ_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.
- ψ_{am}, ψ_{bm}, and ψ_{cm} are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of rotor angle, defined by

$$L_{aa} = L_s + L_m \cos(2\theta_r),$$

$$L_{bb} = L_s + L_m \cos(2(\theta_r - 2\pi/3)),$$

$$L_{cc} = L_s + L_m \cos(2(\theta_r + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos\left(2\left(\theta_r + \pi/6\right)\right),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos\left(2\left(\theta_r + \pi/6 - 2\pi/3\right)\right),$$

and

$$L_{ca} = L_{ac} = -M_s - L_m \cos\left(2\left(\theta_r + \pi/6 + 2\pi/3\right)\right),$$

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 amplitude of 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

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P\left(\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} - N\omega \begin{bmatrix} \dfrac{\partial \psi_a}{\partial \theta_r} \\ \dfrac{\partial \psi_b}{\partial \theta_r} \\ \dfrac{\partial \psi_c}{\partial \theta_r} \end{bmatrix}\right),$$

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N\omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega i_d L_d,$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt},$$

and

$$T = \frac{3}{2} N\left(i_q i_d L_d - i_d i_q L_q\right) + \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \begin{bmatrix} \dfrac{\partial \psi_a}{\partial \theta_r} \\ \dfrac{\partial \psi_b}{\partial \theta_r} \\ \dfrac{\partial \psi_c}{\partial \theta_r} \end{bmatrix},$$

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

$$P = 2/3 \begin{bmatrix} \cos\theta_e & \cos\left(\theta_e - 2\pi/3\right) & \cos\left(\theta_e + 2\pi/3\right) \\ -\sin\theta_e & -\sin\left(\theta_e - 2\pi/3\right) & -\sin\left(\theta_e + 2\pi/3\right) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

- N is the number of rotor permanent magnet pole pairs.
- ω is the rotor mechanical rotational speed.
- $$\frac{\partial \psi_a}{\partial \theta_r},\ \frac{\partial \psi_b}{\partial \theta_r},\ \text{and}\ \frac{\partial \psi_c}{\partial \theta_r}$$

  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).

Parameters

Rotor Tab

Back EMF profile

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

Peak permanent magnet flux linkage with any of the stator windings. This parameter is visible only when **Back EMF profile** is set to `Perfect trapezoid - specify maximum flux linkage`. The default value is `0.03` `Wb`.

Rotor angle over which back emf is constant

Rotor angle range over which the permanent magnet flux linking the stator winding is constant. This angle is θ_F in the figure that shows the “Trapezoidal Rate of Change of Flux" on page 1-102. This parameter is visible only when **Back EMF profile** is set to `Perfect trapezoid - specify maximum flux linkage`. The default value is `pi / 12` `rad`.

Maximum rotor-induced back emf

Peak rotor-induced back emf into the stator windings. This parameter is visible only when **Back EMF profile** is set to `Perfect trapezoid - specify maximum rotor-induced back emf`. The default value is `9.6` `V`.

Rotor-induced back emf

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. This parameter is visible only when **Back EMF profile** is set to `Tabulated - specify rotor-induced back emf as a function of rotor angle`. The default value is `[0.0, -9.6, -9.6, 9.6, 9.6, 0.0]` `V`.

Flux linkage partial derivative with respect to rotor angle

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. This parameter is visible only when **Back EMF profile** is set to `Tabulated - specify flux partial derivative with respect to rotor angle`. The default value is `[0.0, -0.1528, -0.1528, 0.1528, 0.1528, 0.0]` `Wb/rad`.

Corresponding rotor angles

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 angel 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. This parameter is visible only when **Back EMF profile** is set to `Tabulated - specify flux partial derivative with respect to rotor angle` or to `Tabulated - specify rotor-induced back emf as a function of rotor angle`. The default value is `[0, 7.5, 22.5, 37.5, 52.5, 60]` `deg`.

Rotor speed used for back emf measurement

Specify the rotor speed used when quotoring the maximum rotor-induced back emf. This parameter is visible only when **Back EMF profile** is set to `Perfect trapezoid - specify maximum rotor-induced back emf` or `Tabulated - specify rotor-induced back emf as a function of rotor angle`. The default value is `600 rpm`.

Number of pole pairs

Number of permanent magnet pole pairs on the rotor. The default value is `6`.

Stator Tab

Stator parameterization

Choose `Specify Ld, Lq, and L0`, the default value, or `Specify Ls, Lm, and Ms`.

Stator d-axis inductance, Ld

D-axis inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00022 H`.

Stator q-axis inductance, Lq

Q-axis inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00022 H`.

Stator zero-sequence inductance, L0

Zero-sequence inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00016 H`.

Stator self-inductance per phase, Ls

Average self-inductance of each of the three stator windings. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `0.00002 H`.

Stator inductance fluctuation, Lm

Amplitude of the fluctuation in self-inductance and mutual inductance of the stator windings with rotor angle. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `0 H`.

Stator mutual inductance, Ms

Average mutual inductance between the stator windings. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `0.00002 H`.

Stator resistance per phase, Rs

Resistance of each of the stator windings. The default value is `0.013 Ohm`.

Initial Conditions Tab

Initial currents, [i_d i_q i_0]

Initial *d*-axis, *q*-axis, and zero-sequence currents. The default value is `[0, 0, 0] A`.

Rotor angle definition

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 1-102 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.

Initial rotor angle

Initial angle of the rotor. The default value is `0 deg`.

Ports

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral phase

`R`

Mechanical rotational conserving port associated with the motor rotor

`C`

Mechanical rotational conserving port associated with the motor case

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.

See Also

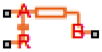
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Cauer Thermal Model Element

Heat transfer through an individual layer of a semiconductor module

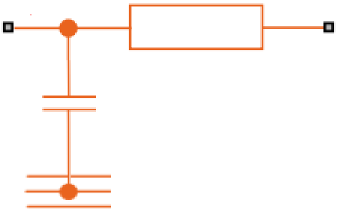


Library

Semiconductors / Fundamental Components / Thermal

Description

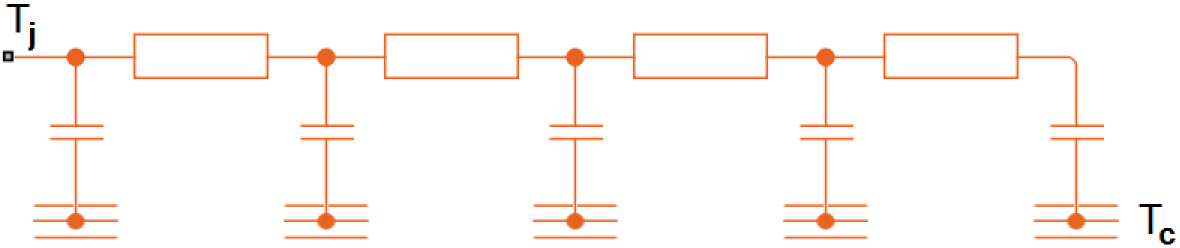
The Cauer Thermal Model Element block represents heat transfer through an individual layer of a semiconductor module. The figure shows an equivalent circuit for a Cauer Thermal Model Element block.



A Cauer thermal model represents the multiple layers that constitute the packaging of a semiconductor. Layers include chip, solder, substrate, solder, and base. Other terms that describe a Cauer thermal model are:

- Continued fraction circuit
- T model
- Ladder network

To create a Cauer thermal model, connect multiple instances of the Cauer Thermal Model Element block in series. In the figure of the Cauer thermal model, T_j is the junction temperature and T_c is the base plate temperature.



The defining equations for the Cauer Thermal Model Element block 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.
- τ 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.

Parameters

- “Parameters Tab” on page 1-114
- “Variables Tab” on page 1-114

Parameters Tab

Thermal resistance

The default value for the thermal resistance, $R_{thermal}$, is `5e-3 K/W`.

Thermal time constant

The default value for the thermal time constant, τ, is `0.1 s`.

Variables Tab

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

`A`

Thermal conserving port associated with the first surface of the individual layer of the semiconductor.

`R`

Thermal conserving port associated with the chosen thermal reference.

`B`

Thermal conserving port associated with the second surface of the individual layer of the semiconductor.

References

[1] Schütze, T. *AN2008-03: Thermal equivalent circuit models.* Application Note. V1.0. Germany: Infineon Technologies AG, 2008.

See Also

Foster Thermal Model | Thermal Resistor

Topics

"Quantifying IGBT Thermal Losses"

Introduced in R2016a

Commutation Diode

Piecewise linear diode with charge dynamics and junction capacitance

Library

Semiconductors / Fundamental Components

Description

The Commutation Diode block augments the Diode block with a model of charge dynamics. For a description of the piecewise linear diode operation that the Commutation Diode block uses, see Diode.

Use the Commutation Diode block in place of the Diode block when you want to specify precisely the charge dynamics of the device as it operates in reverse mode. For example, suppose that your model uses the diode to divert inductive currents from a motor drive or inverter. In this case, precise reverse-mode operation is important and an appropriate time to use the Commutation Diode block.

The Commutation Diode uses a charge model proposed by Lauritzen and Ma [1]. The defining expressions for this charge model are:

$$i = \frac{q_E - q_M}{T_M}$$

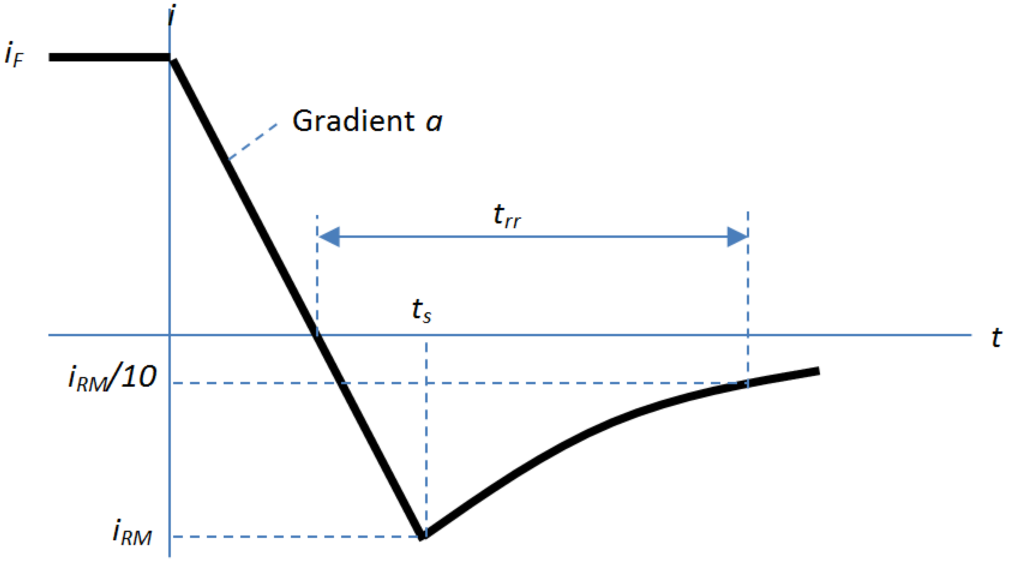
$$\frac{dq_M}{dt} + \frac{q_M}{\tau} - \frac{q_E - q_M}{T_M} = 0$$

$$\begin{aligned} q_E &= (\tau + T_M)(v_D - v_F(1 - RG))/R \quad && \text{if} \quad v_D > v_F \\ q_E &= (\tau + T_M)Gv_D && \text{if} \quad v_D \le v_F \end{aligned}$$

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.
- τ 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.

On the Charge Dynamics tab of the block, you specify characteristics of your diode device. The block uses these values to calculate the diode charge dynamics expressed in equations 1–1, 1–2, and 1–3.

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 τ based on the values you enter on the Charge Dynamics tab of the block dialog box. The block uses T_M and τ to solve the charge dynamics equations 1–1, 1–2, and 1–3.

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.

Using Equation 1–1,

$$i_F + at = \frac{q_E - q_M}{T_M}.$$

Substituting Equation 1–4 into Equation 1–2,

$$\frac{dq_M}{dt} + \frac{q_M}{\tau} = i_F + at.$$

Solving Equation 1–5 for q_M,

$$q_M = i_F\tau - a\tau^2 + \frac{k}{\exp\left(\frac{t}{\tau}\right)} + a\tau t,$$

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 1–6 and solving the equation gives $k = a\tau^2$.

Therefore,

$$q_M = i_F \tau + a\tau^2 \left(\frac{1}{\exp\left(\frac{t}{\tau}\right)} - 1 \right) + a\tau t.$$

At time $t = t_s$, the current is i_{RM} and the junction charge q_E is zero.

Substituting these values into Equation 1–1,

$$i_{RM} = \frac{-q_M}{T_M}.$$

Rearranging Equation 1–8 to solve for q_M and substituting the result into Equation 1–7,

$$-T_M i_{RM} = i_F \tau + a\tau^2 \left(\frac{1}{\exp\left(\frac{t_s}{\tau}\right)} - 1 \right) + a\tau t_s.$$

Expressing time t_s in terms of i_{RM}, i_F, and a,

$$t_s = \frac{i_{RM} - i_F}{a}.$$

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

$$i = i_{RM} \exp[\frac{-(t - t_s)}{\tau_{rr}}],$$

where

$$\frac{1}{\tau_{rr}} = \frac{1}{\tau} + \frac{1}{T_M}.$$

The block now relates the expression in Equation 1–12 to the reverse recovery time t_{rr}.

When $t = \frac{i_{RM}}{a} + t_{rr}$, the current is $\frac{i_{RM}}{10}$.

Therefore,

$$\exp\left(-\frac{t - t_s}{\tau_{rr}}\right) = 0.1$$

and

$$t_{rr} = \tau_{rr} \log(10) + \frac{i_{RM}}{a}.$$

The block uses equations 1–9 and 1–14 to calculate values for T_M and τ. The calculation uses an iterative scheme because of the exponential term in Equation 1–9.

Alternatives to Specifying t_{rr} Directly

In addition to allowing you to specify reverse recovery time t_{rr} directly, the block supports two alternative parameterizations. The block can derive t_{rr} from either of these parameters:

- Reverse recovery time stretch factor λ
- Reverse recovery charge Q_{rr}, when the data sheet specifies this value instead of the reverse recovery time.

The relationship between reverse recovery time stretch factor λ 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 λ is 3. λ 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 the equation

$$Q_s = \frac{1}{2}\left(-i_{RM}\right)\frac{i_{RM}}{a}.$$

Integrating Equation 1–11 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 the equation

$$Q_{rr} = -\frac{i_{RM}^2}{2a} + \tau_{rr} i_{RM}.$$

Rearranging Equation 1–16 to solve for τ_{rr} and substituting the result into Equation 1–14 gives an equation that expresses t_{rr} in terms of Q_{rr}:

$$t_{rr} = \left(\frac{Q_{rr}}{i_{RM}} + \frac{i_{RM}}{2a}\right)\log\left(10\right) + \frac{i_{RM}}{a}.$$

Modeling Variants

The block provides a thermal modeling variant. To select a variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

- **No thermal port** — This variant does not simulate heat generation in the device. This variant is the default.
- **Show thermal port** — This variant 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. The thermal port is hidden by default. When you select a thermal variant of the block, the thermal port appears.

Ports

`+`

Electrical conserving port associated with the diode positive terminal

`-`

Electrical conserving port associated with the diode negative terminal

`H`

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

- “Main Tab” on page 1-122
- “Charge Dynamics Tab” on page 1-123

Main Tab

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Number of series diodes

The number of diodes connected in series between the + and − block ports. Each diode has the **Forward voltage**, **On resistance**, and **Off conductance** that you specify. The default value is `1`.

Number of parallel diodes

The number of parallel diodes, or number of parallel paths formed by series-connected diodes, between the + and − block ports. Each diode has the **Forward**

voltage, **On resistance**, and **Off conductance** that you specify. The default value is `1`.

Charge Dynamics Tab

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 µAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

References

[1] Lauritzen, P.O. & C.L. Ma, “A Simple Diode Model with Reverse Recovery.” *IEEE® Transactions on Power Electronics*. Vol. 6, No. 2, 1991, pp. 188–191.

See Also

Diode | GTO | IGBT | Ideal Semiconductor Switch | MOSFET | Thyristor

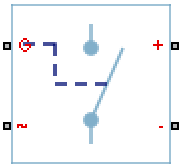
Topics

"Quantifying IGBT Thermal Losses"
"Simulate Thermal Losses in Semiconductors"

Introduced in R2013b

Converter

Connect three-phase AC network to DC network



Library

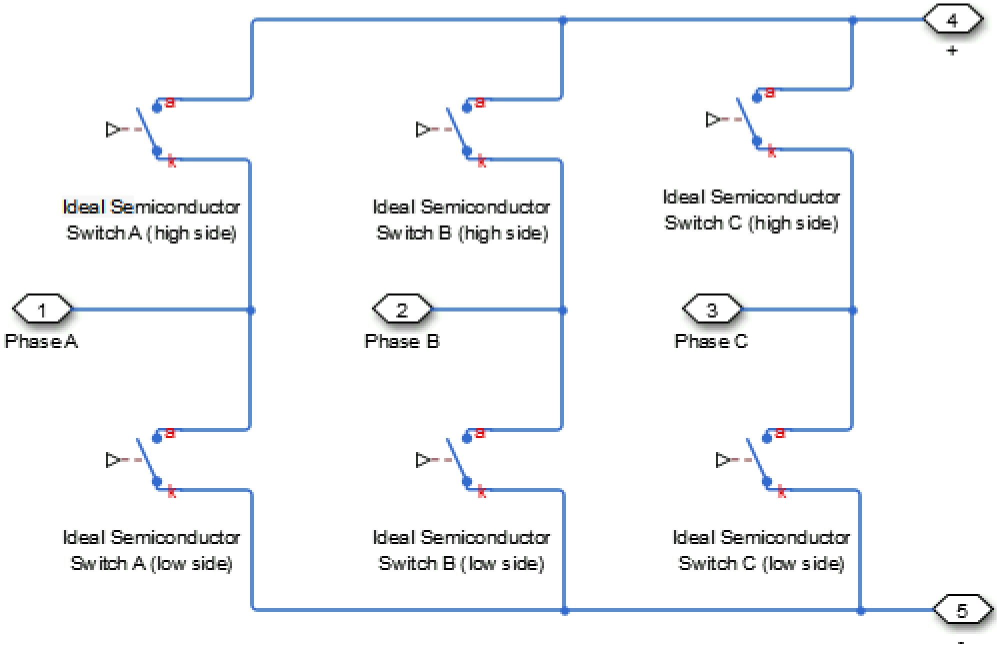
Semiconductors

Description

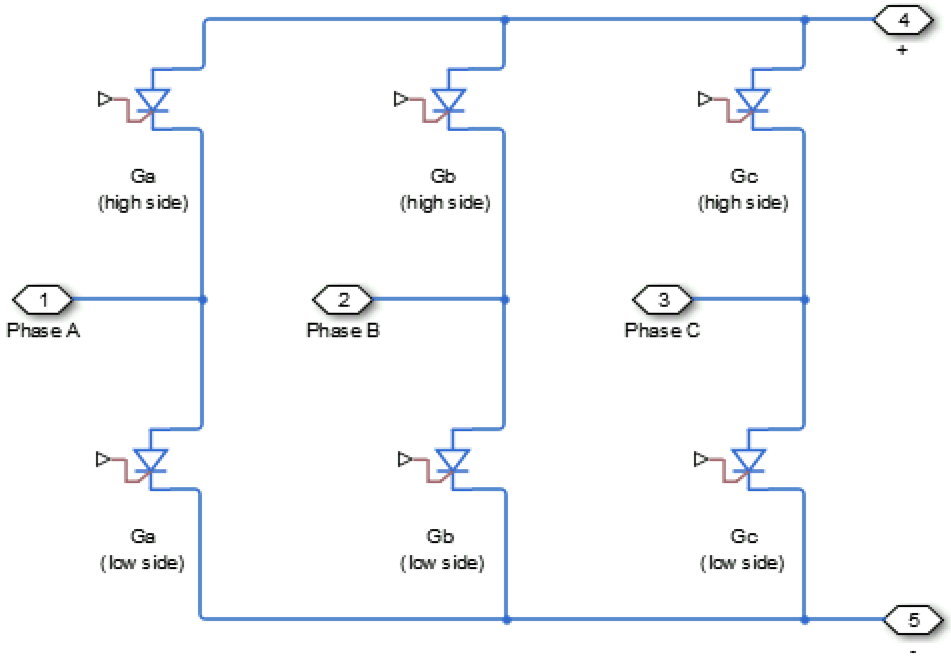
The Converter block models a three-arm converter circuit that connects a three-phase AC network to a DC network.

Each component in the three-arm circuit is the same switching device, which you specify using an option on the Converter block dialog box. The switching devices that you can specify are in the Semiconductors / Fundamental Components library.

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 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 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.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	`Protection diode with no dynamics`	The block includes an integral copy of the Diode block. The block dialog box shows parameters relating to the Diode block.
Precisely specify reverse-mode charge dynamics.	`Protection diode with charge dynamics`	The block includes an integral copy of the Commutation Diode block. The block dialog box shows parameters relating to the Commutation 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.

Parameters

- "Switching Devices Tab" on page 1-129
- "Protection Diodes Tab" on page 1-132
- "Snubbers Tab" on page 1-135

Switching Devices Tab

Switching device

Converter switching device. The default value is `Ideal Semiconductor Switch`.

The switching devices you can select are:

- `GTO`
- `Ideal Semiconductor Switch`
- `IGBT`
- `MOSFET`

- `Thyristor`

When you select `GTO`, parameters for the GTO block appear.

Additional GTO Parameters

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Rate of change of voltage versus current above the forward voltage. The default value is `0.001 Ohm`.

Off-state conductance

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-6 1/Ohm`.

Gate trigger voltage, Vgt

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `1 V`.

Gate turn-off voltage, Vgt_off

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value. The default value is `-1 V`.

Holding current

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. The default value is `1 A`.

For more information, see GTO.

When you select `Ideal Semiconductor Switch`, parameters for the Ideal Semiconductor Switch block appear.

Additional Ideal Semiconductor Switch Parameters

On-state resistance

Anode-cathode resistance when the device is on. The default value is `0.001 1/Ohm`.

Off-state conductance

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-6 1/Ohm`.

Threshold voltage, Vth

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `6 V`.

For more information, see Ideal Semiconductor Switch.

When you select `IGBT`, parameters for the IGBT block appear.

Additional IGBT Parameters

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Collector-emitter resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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**. The default value is `1e-6 1/Ohm`.

Threshold voltage, Vth

Collector-emitter voltage at which the device turns on. The default value is `6 V`.

For more information, see IGBT.

When you select `MOSFET`, parameters for the MOSFET block appear.

Additional MOSFET Parameters

On-state resistance, R_DS(on)

Drain-source resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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**. The default value is `1e-6 1/Ohm`.

Threshold voltage, Vth

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value. The default value is `6 V`.

For more information, see MOSFET.

When you select `Thyristor`, parameters for the Thyristor block appear.

Additional Thyristor Parameters

Forward voltage, Vf

Forward voltage at which the device turns on. The default value is `0.8 V`.

On-state resistance

Anode-cathode resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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-6 1/Ohm`.

Gate trigger voltage, Vgt

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `1 V`.

Holding current

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. The default value is `1 A`.

For more information, see Thyristor.

Protection Diodes Tab

Integral protection diode

Integral protection diode for each switching device. The default value is `None`.

The diodes you can select are:

- `Protection diode with no dynamics`

- `Protection diode with charge dynamics`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection Diode with No Dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection Diode with Charge Dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 µAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Snubbers Tab

Snubber

Snubber for each switching device. The default value is `None`.

Snubber resistance

Snubber resistance. This parameter is visible only if you set **Snubber** to `RC snubber`. The default value is `0.1 Ohm`.

Snubber capacitance

Snubber capacitance. This parameter is visible only if you set **Snubber** to `RC snubber`. The default value is `1e-7 F`.

Ports

The block has the following ports:

`G`

Vector input port associated with the gate terminals of the switching devices. Connect this port to a Six-Pulse Gate Multiplexer block.

`~`

Expandable three-phase port

`+`

Electrical conserving port associated with the DC positive terminal

`-`

Electrical conserving port associated with the DC negative terminal

See Also

Average-Value Inverter | Average-Value Rectifier | Rectifier | Six-Pulse Gate Multiplexer | Three-Level Converter

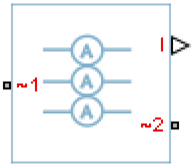
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Current Sensor

Measure phase currents in three-phase system



Library

Sensors

Description

The Current Sensor 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.

Ports

The block has the following ports:

`~1`

Expandable three-phase port.

`~2`

Expandable three-phase port.

`I`

Three-element physical signal vector output port associated with the phase currents.

See Also

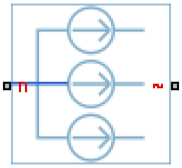
Topics

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

Introduced in R2013b

Current Source

Ideal three-phase current source



Library

Sources

Description

The Current Source 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:

$$I_0 = \sqrt{2} i_{phase\_rms}$$

$$i_a = I_0 \sin(2\pi f t + \varphi)$$

$$i_b = I_0 \sin(2\pi f t + \varphi - 120^\circ)$$

$$i_c = I_0 \sin(2\pi f t + \varphi + 120^\circ),$$

where:

- I_0 is the peak phase current.
- $i_{\mathrm{phase\_rms}}$ is the RMS phase current.
- i_{a}, i_{b}, i_{c} are the respective phase currents.

- *f* is the frequency.
- φ 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. Port ~ is an expandable three-phase port representing the three phases, a, b, and c.

Parameters

Current (phase RMS)

RMS phase current. The default value is `100/sqrt(2)`, or 70.7107, `A`.

Phase shift

Phase shift in angular units. The default value is `0 deg`.

Frequency

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. The default value is `60 Hz`.

Ports

The block has the following ports:

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the center of the wye

See Also

Voltage Source

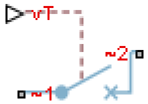
Topics

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

Introduced in R2013b

Circuit Breaker

Three-phase circuit breaker controlled by external signal



Library

Switches & Breakers

Description

The Circuit Breaker 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` controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
`vT` < **Threshold**	The breaker is closed. Each phase in the composite three-phase port `~1` connects to the corresponding phase in the port `~2`.	**Closed Resistance**
`vT` ≥ **Threshold**	When the current in any phase of the composite port `~1` crosses zero, the phase disconnects from the corresponding phase at port `~2`. The breaker is open.	**Open Conductance**

Parameters

Closed resistance

Resistance between ports `~1` and `~2` when the breaker is closed. The default value is `0.001 Ohm`.

Open conductance

Conductance between ports `~1` and `~2` when the breaker is open. The default value is `1e-6 1/Ohm`.

Threshold

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 each phase as its current crosses zero. When the control port voltage falls below the threshold, the breaker closes. The default value is `0 V`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

`~2`

Expandable three-phase port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Single-Phase Circuit Breaker | Single-Phase Circuit Breaker (with arc)

Topics

"Three-Phase Synchronous Machine Control"
"Three-Phase Custom Simplified Synchronous Machine"
"Marine Full Electric Propulsion Power System"

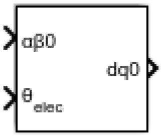
"Switch Between Physical Signal and Electrical Ports"
"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Clarke to Park Angle Transform

Implement *αβ0* to *dq0* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

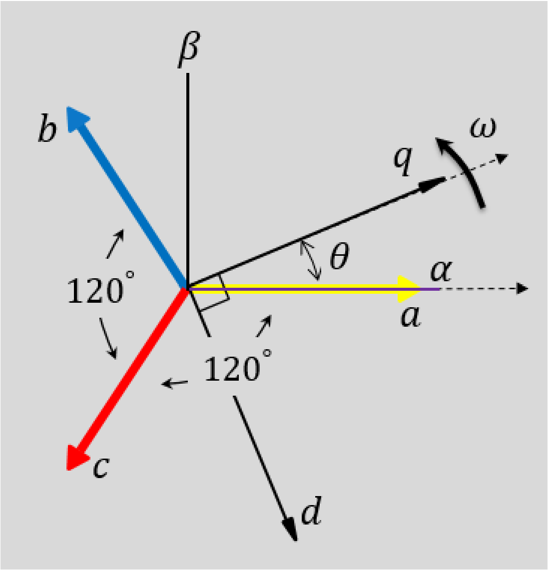


Description

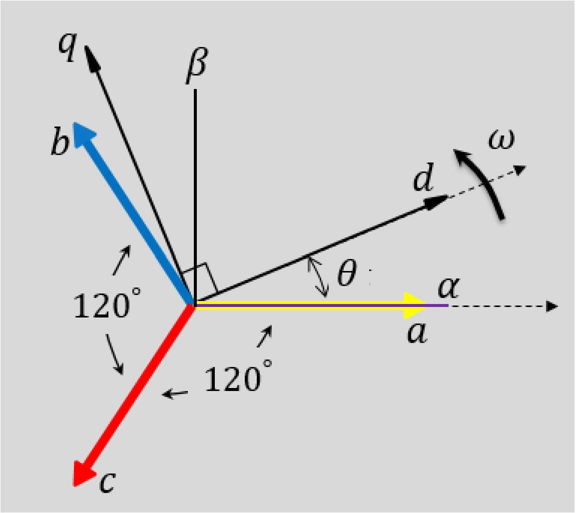
The Clarke to Park Angle Transform block converts the alpha, beta, and zero components in a stationary reference frame to direct, quadrature, and zero components in a rotating reference frame. For balanced three-phase systems, the zero components are equal to zero.

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 *αβ0* 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.



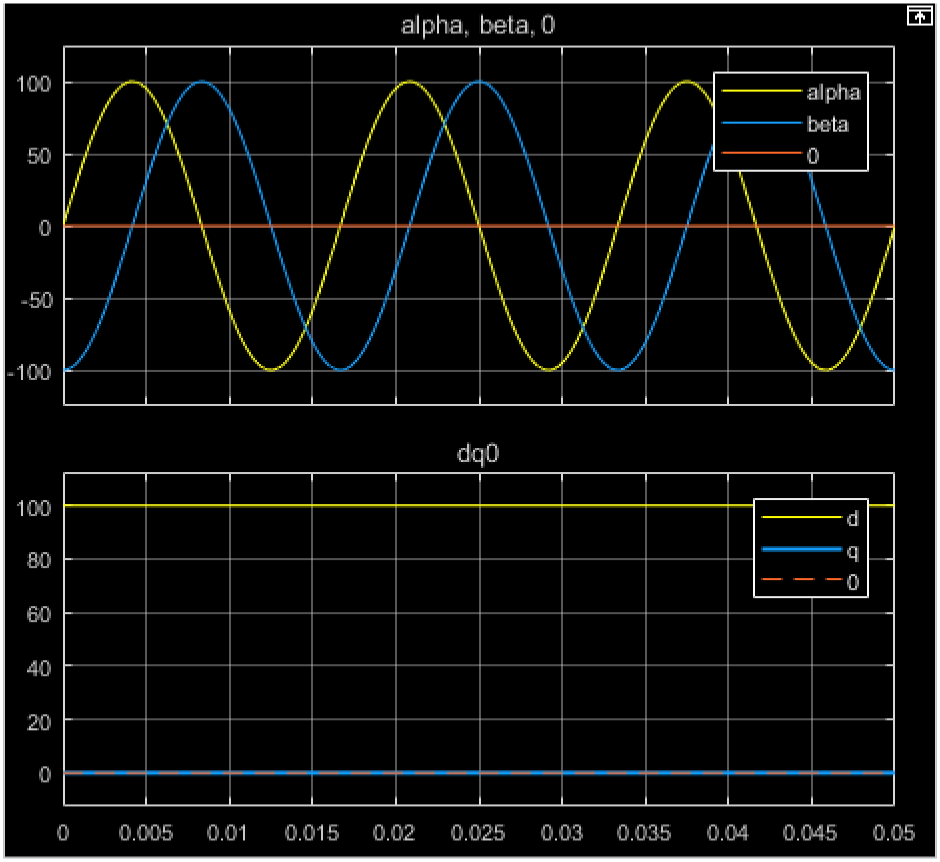
In both cases, the angle $\theta = \omega t$, where

- θ 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.

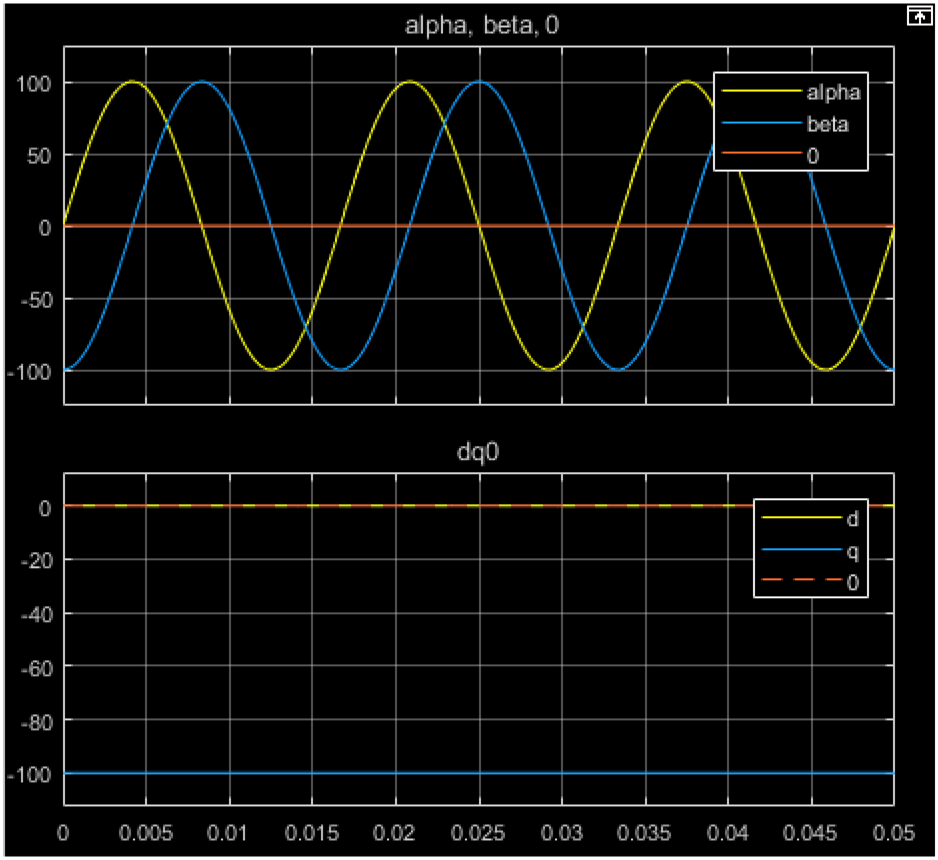
- ω is the rotational speed of the *d*-*q* reference frame.
- *t* is the time, in s, from the initial alignment.

The figures show the time-response of the individual components of equivalent balanced *αβ0* 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 Clarke to Park Angle Transform block implements the transform for an a-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:

- α and β 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 a-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

`αβ0` — *α-β* axis and zero components

vector

Alpha-axis, α, beta-axis, β, and zero components of the two-phase system in the stationary reference frame.

Data Types: `single` | `double`

θ_{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.

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.

See Also

Blocks

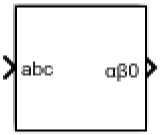
Clarke Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Clarke Transform

Implement *abc* to *αβ0* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

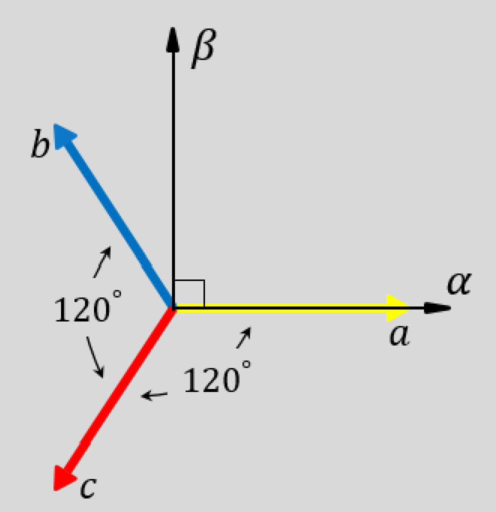


Description

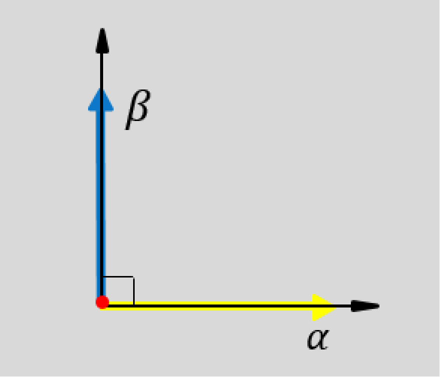
The Clarke Transform block converts the time-domain components of a three-phase system in an *abc* reference frame to components in a stationary *αβ0* reference frame. The block can preserve the active and reactive powers with the powers of the system in the *abc* reference frame by implementing a power invariant version of the Clarke transform. For a balanced system, the zero component is equal to zero.

The figures show:

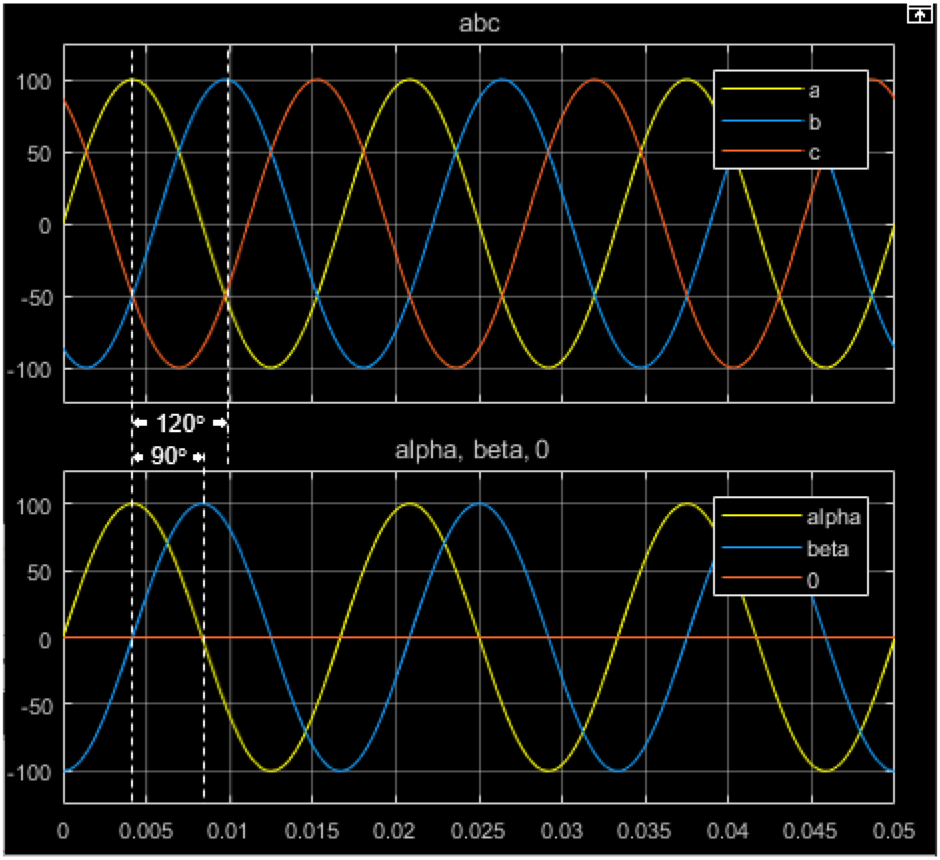
- The direction of the magnetic axes of the stator windings in the *abc* reference frame and the stationary *αβ0* reference frame



- Equivalent α, β, and zero components in the stationary reference frame



- The time-response of the individual components of equivalent balanced *abc* and $\alpha\beta0$ 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.
- α and β 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

$$\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

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`

Output

αβ0 — *α*-*β* axis and zero components

vector

Alpha-axis component,α, beta-axis component β, 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.

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.

See Also

Blocks

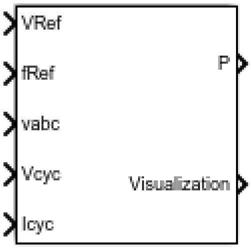
Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Three-Phase Bridge Cycloconverter Voltage Controller

PI-based RMS voltage control for three-phase bridge cycloconverters

Library: Simscape / Power Systems / Simscape Components / Control / Converter Control



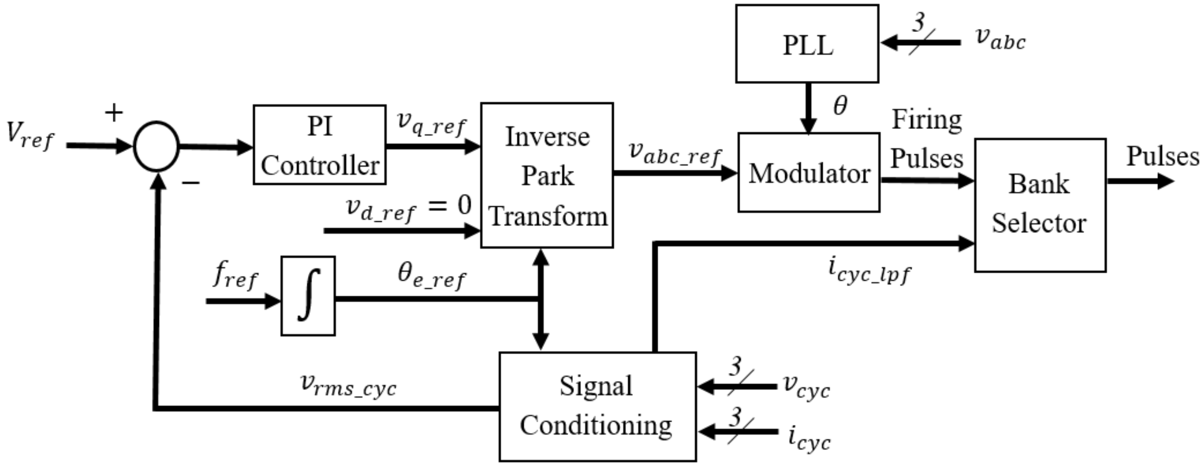
Description

The Three-Phase Bridge Cycloconverter Voltage Controller 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.

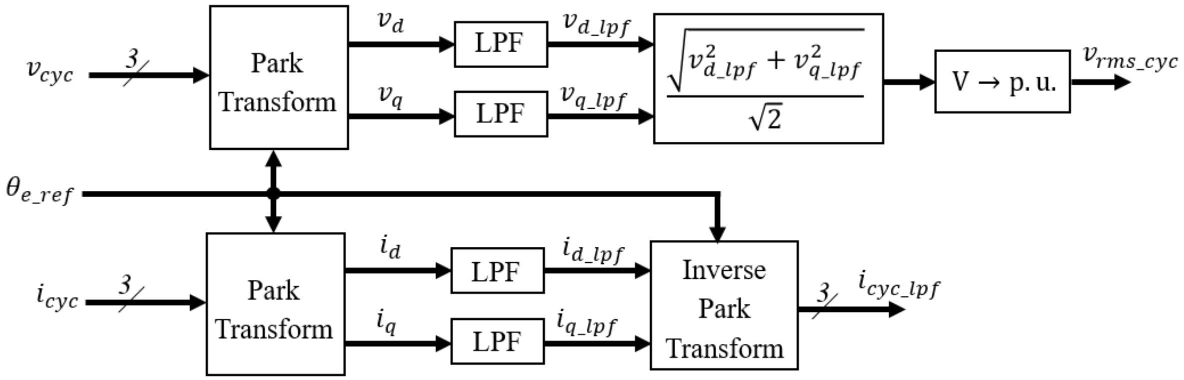


In the diagram:

- The controller integrates the desired output frequency f_{ref} to produce the reference electrical angle θ_{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 Three-Phase Sinusoidal Measurement (PLL) block estimates the phase angle θ 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 θ, 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 θ_{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 θ 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 1-158 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.

Model Examples

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.

See Also

Blocks

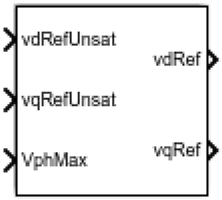
Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous) | Park Transform | Three-Phase Sinusoidal Measurement (PLL)

Introduced in R2017b

d-q Voltage Limiter

Limit voltage in the rotor direct-quadrature reference frame

Library: Simscape / Power Systems / Simscape Components / Control / Protection

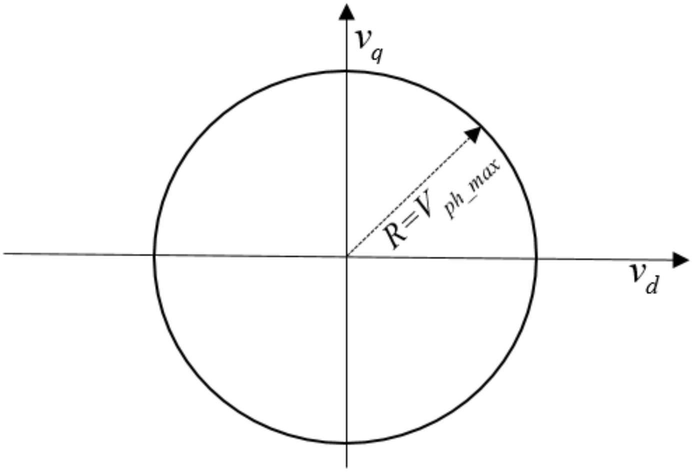


Description

The d-q Voltage Limiter block implements a voltage limiter in the rotor direct-quadrature (*d*-*q*) reference frame.

Electrical Defining Equations

The figure shows the circle that limits the *d*-*q* voltage vector.



That is,

$$\sqrt{v_d^2 + v_q^2} \le V_{ph\_max}$$

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

$$v_1^{sat} = \min\left(\max\left(v_1^{unsat}, -V_{ph\_max}\right), V_{ph\_max}\right)$$

and

$$v_2^{sat} = \min\left(\max\left(v_2^{unsat}, -V_{2\_\max}\right), V_{2\_\max}\right),$$

where:

- $$v_{2\_max} = \sqrt{\left(V_{ph\_max}\right)^2 - \left(v_1^{sat}\right)^2}$$
- 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

$$v_d^{sat} = \min\left(\max\left(v_d^{unsat}, -V_{d\_\max}\right), V_{d\_\max}\right)$$

and

$$v_q^{sat} = \min\left(\max\left(v_q^{unsat}, -V_{q\_\max}\right), V_{q\_\max}\right),$$

where:

- $$V_{d\_max} = \frac{V_{ph\_max}\left|v_d^{unsat}\right|}{\sqrt{\left(v_d^{unsat}\right)^2 + \left(v_q^{unsat}\right)^2}}$$
- $$V_{q\_max} = \frac{V_{ph\_max}\left|v_q^{unsat}\right|}{\sqrt{\left(v_d^{unsat}\right)^2 + \left(v_q^{unsat}\right)^2}}$$

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_d^{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?” (Simulink) and “Specify Sample Time” (Simulink).

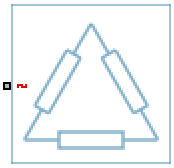
See Also

Simscape Blocks
Voltage Source

Introduced in R2017b

Delta-Connected Load

Three-phase load wired in delta configuration



Library

Passive Devices

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 or 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.

Parameters

- "Main Tab" on page 1-168
- "Parasitics Tab" on page 1-169
- "Initial Conditions Tab" on page 1-170

Main Tab

Parameterization

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

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel. The default is `R`, resistor.

Rated voltage

Voltage for which load powers are specified. This parameter is visible only when you specify values by rated power. The default value is `2.4e4 V`.

Real power

Total real power dissipated by three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes a resistor. The value must be greater than 0. The default value is `1000 W`.

Rated electrical frequency

Frequency for which reactive load powers are specified. This parameter is visible only when you specify values by rated power. The default value is `60 Hz`.

Inductive reactive power

Total inductive reactive power taken by the three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes an inductor. The value must be greater than 0. The default value is `100 V*A`.

Capacitive reactive power

Total capacitive reactive power taken by the three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes a capacitor. The value must be less than 0. The default value is `-100 V*A`.

Resistance

The resistance of each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes a resistor. The default value is `1 Ohm`.

Inductance

Inductance of each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes an inductor. The default value is `0.001 H`.

Capacitance

Capacitance in each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes a capacitor. The default value is `1e-6 F`.

Parasitics Tab

Parasitic series resistance

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load. The default value is `1e-6 Ohm`.

Parasitic parallel conductance

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load. The default value is `1e-6 1/Ohm`.

Initial Conditions Tab

Terminal voltage magnitude

Expected initial RMS line voltage at the load. This parameter is visible only when you specify values by rated power. The default value is `2.4e4 V`.

Terminal voltage angle

Expected initial phase of the voltage at the load. This parameter is visible only when you specify values by rated power. The default value is `0 deg`.

Frequency

Expected initial frequency at the load. This parameter is visible only when you specify values by rated power. The default value is `60 Hz`.

Initial inductor current [Ia Ib Ic]

Initial current in the a, b, and c phase inductors, respectively. This parameter is visible only when you specify component values directly and select a component structure that includes an inductor. The default value is `[0 0 0] A`.

Initial capacitor voltage [Va Vb Vc]

Initial voltage across the a, b, and c phase capacitors, respectively. This parameter is visible only when you specify component values directly and select a component structure that includes a capacitor. The default value is `[0 0 0] V`.

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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`Series LC`	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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]

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`Parallel RLC`	Resistance Inductance Capacitance	Parasitic series resistance	Initial inductor current [Ia Ib Ic] Initial capacitor voltage [Va Vb Vc]

Ports

The block has one expandable three-phase port, ~.

See Also

RLC | Wye-Connected Load

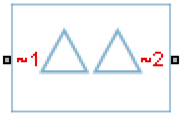
Topics

“Three-Phase Asynchronous Wind Turbine Generator”
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

Delta-Delta Transformer

Linear nonideal delta-delta transformer with three-limb core



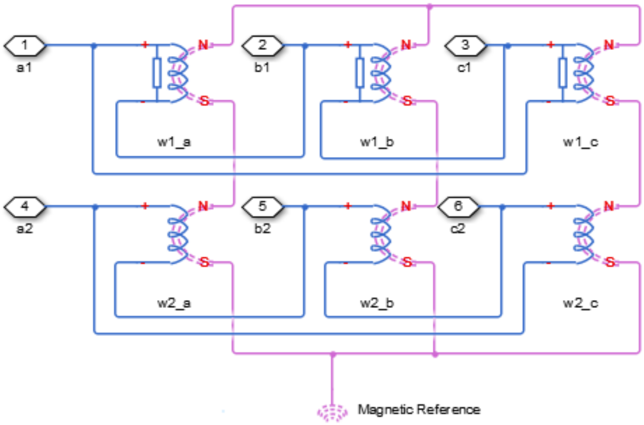
Library

Passive Devices / Transformers

Description

The Delta-Delta Transformer block models a linear, nonideal transformer with a three-limb core, in which both the primary and the secondary windings are configured in a delta connection. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the delta-delta transformer.



- $\texttt{w1}_a$ is the primary winding connected between the a-phase and the b-phase.
- $\texttt{w1}_b$ is the primary winding connected between the b-phase and the c-phase.
- $\texttt{w1}_c$ is the primary winding connected between the c-phase and the a-phase.
- $\texttt{w2}_a$ is the secondary winding connected between the a-phase and the b-phase.
- $\texttt{w2}_b$ is the secondary winding connected between the b-phase and the c-phase.
- $\texttt{w2}_c$ is the secondary winding connected between the c-phase and the a-phase.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-177
- "Impedances Tab" on page 1-178
- "Initial Conditions Tab" on page 1-178

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Secondary rated voltage

RMS line voltage applied to the secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Secondary leakage resistance (pu)

Power loss in the secondary winding. The default value is `0.01`.

Secondary leakage reactance (pu)

Magnetic flux loss in the secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial secondary currents

Current through the secondary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0]` `Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for secondary winding

See Also

Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

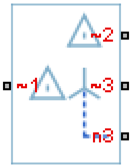
Topics

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

Introduced in R2013b

Delta1-Delta1-Wye Transformer

Linear nonideal delta1-delta1-wye transformer with three-limb core



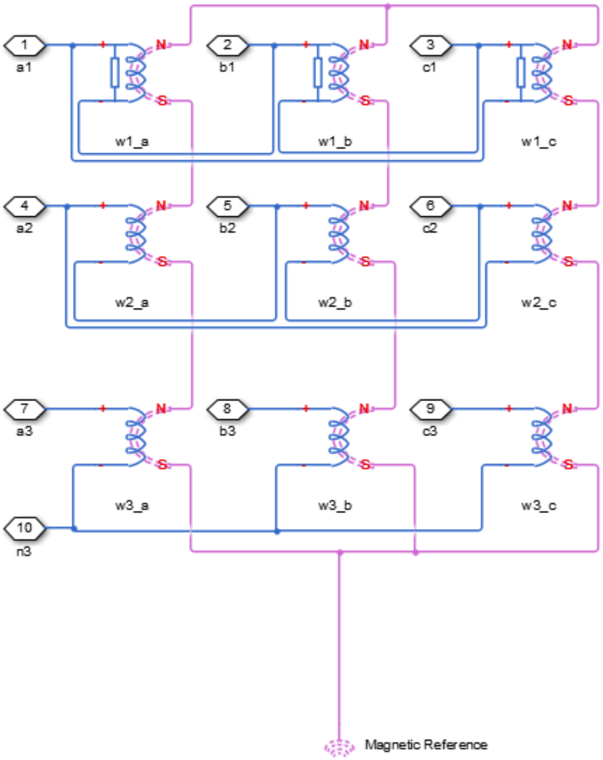
Library

Passive Devices / Transformers

Description

The Delta1-Delta1-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a delta connection and there are delta secondary windings and wye secondary windings. The delta voltages lag the wye voltages by 30 degrees, hence the name 1 o’clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the delta1-delta1-wye transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the b-phase.
- `w1`$_b$ is the primary winding connected between the b-phase and the c-phase.
- `w1`$_c$ is the primary winding connected between the c-phase and the a-phase.
- `w2`$_a$ is the delta secondary winding connected between the a-phase and the b-phase.
- `w2`$_b$ is the delta secondary winding connected between the b-phase and the c-phase.
- `w2`$_c$ is the delta secondary winding connected between the c-phase and the a-phase.
- `w3`$_a$ is the wye secondary winding connected between the a-phase and the secondary neutral point.
- `w3`$_b$ is the wye secondary winding connected between the b-phase and the secondary neutral point.
- `w3`$_c$ is the wye secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-182
- "Impedances Tab" on page 1-182
- "Initial Conditions Tab" on page 1-183

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `24e3 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see "Per-Unit System of Units".

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

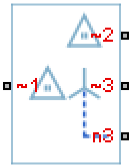
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Delta11-Delta11-Wye Transformer

Linear nonideal delta11-delta11-wye transformer with three-limb core



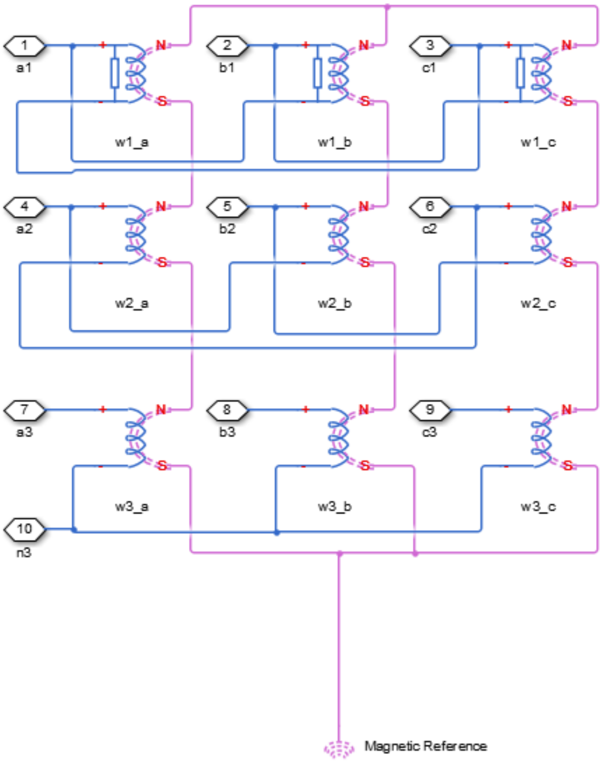
Library

Passive Devices / Transformers

Description

The Delta11-Delta11-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a delta connection and there are delta secondary windings and wye secondary windings. The delta voltages lead the wye voltages by 30 degrees, hence the name 11 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the delta11-delta11-wye transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the c-phase.
- `w1`$_b$ is the primary winding connected between the b-phase and the a-phase.
- `w1`$_c$ is the primary winding connected between the c-phase and the b-phase.
- `w2`$_a$ is the delta secondary winding connected between the a-phase and the c-phase.
- `w2`$_b$ is the delta secondary winding connected between the b-phase and the a-phase.
- `w2`$_c$ is the delta secondary winding connected between the c-phase and the b-phase.
- `w3`$_a$ is the wye secondary winding connected between the a-phase and the secondary neutral point.
- `w3`$_b$ is the wye secondary winding connected between the b-phase and the secondary neutral point.
- `w3`$_c$ is the wye secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-187
- "Impedances Tab" on page 1-187
- "Initial Conditions Tab" on page 1-188

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `24e3 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see "Per-Unit System of Units".

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

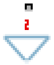
Topics

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

Introduced in R2013b

Delta Reference

Reference point for delta-connected network



Library

Connections

Description

In a Simscape Power Systems Simscape Components model, connect a Delta Reference 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 block to the part of the circuit between the two transformers.

Ports

The block has the following ports:

~

Expandable three-phase port

See Also

Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Diode

Piecewise linear diode

Library

Semiconductors / Fundamental Components

Description

The Diode block models a piecewise linear diode.

If the voltage across the diode exceeds the value specified in the block **Forward voltage** parameter, then the diode behaves like a linear resistor plus a series voltage source. The value of the **On resistance** parameter specifies the resistance of the linear resistor.

If the voltage across the diode is less than the forward voltage, the diode behaves like a linear resistor with low conductance specified by the value of the **Off conductance** parameter.

When forward biased, the series voltage source is given by

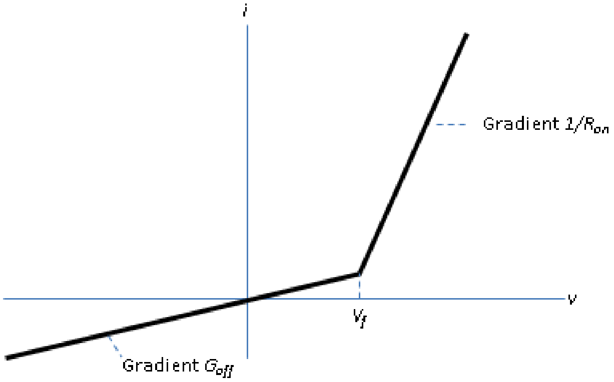
$V = V_f(1 - R_{on}G_{off}),$

where:

- V is the voltage supplied by the series voltage source.
- V_f is the forward voltage.
- R_{on} is the on resistance.
- G_{off} is the off conductance.

The $R_{on}*G_{off}$ term ensures that the diode current is exactly zero when the voltage across it is zero.

The figure shows a typical I-V characteristic for a diode device.



Modeling Variants

The block provides a thermal modeling variant. To select a variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

- **No thermal port** — This variant does not simulate heat generation in the device. This variant is the default.
- **Show thermal port** — This variant 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. The thermal port is hidden by default. When you select a thermal variant of the block, the thermal port appears.

Ports

\+

Electrical conserving port associated with the diode positive terminal

\-

Electrical conserving port associated with the diode negative terminal

H

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Number of series diodes

The number of diodes connected in series between the + and − block ports. Each diode has the **Forward voltage**, **On resistance**, and **Off conductance** that you specify. The default value is `1`.

Number of parallel diodes

The number of parallel diodes, or number of parallel paths formed by series-connected diodes, between the + and − block ports. Each diode has the **Forward voltage**, **On resistance**, and **Off conductance** that you specify. The default value is `1`.

See Also

Commutation Diode | GTO | IGBT | Ideal Semiconductor Switch | MOSFET | Thyristor

Topics

“Quantifying IGBT Thermal Losses”
“Simulate Thermal Losses in Semiconductors”

Introduced in R2013b

Discrete PI Controller

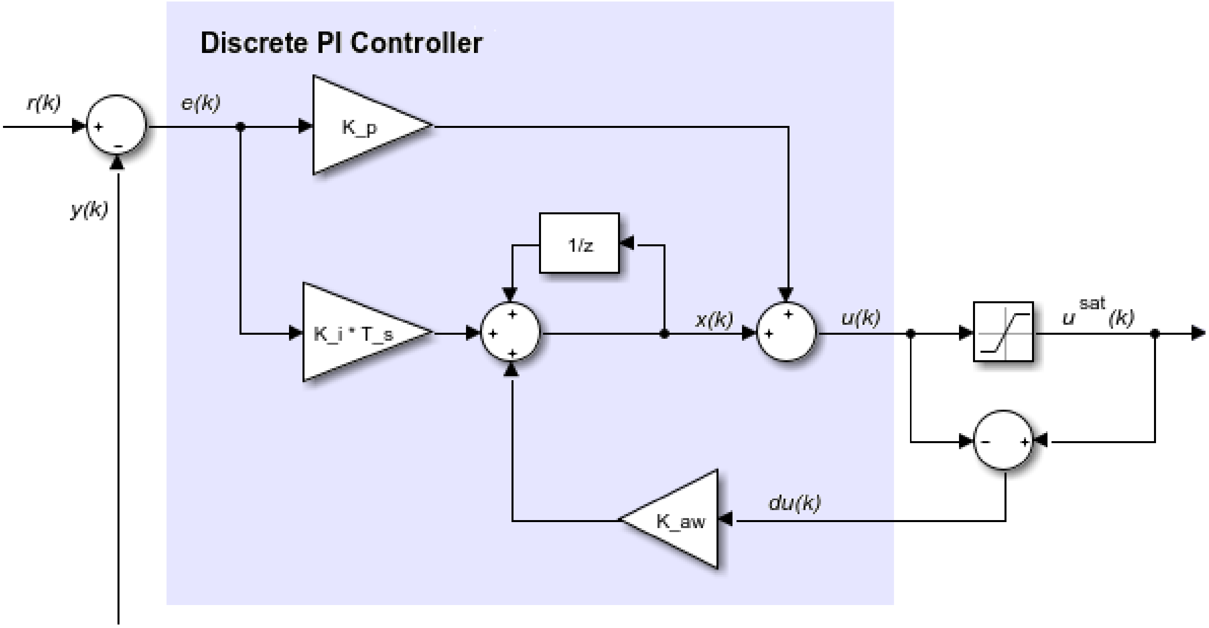
Discrete-time PI controller with external anti-windup input

Library: Simscape / Power Systems / Simscape Components / 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:

$$u(k) = \left[K_p + \left(K_i + du(k)K_{aw}\right)\frac{T_s z}{z-1}\right]\mathrm{e}(\mathrm{k}),$$

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?” (Simulink) and “Specify Sample Time” (Simulink).

Model Examples

DC Motor Control Energy Balance in a 48V Starter Generator IPMSG Voltage Stabilization Switched Reluctance Machine Speed Control IPMSM Torque Control in a Series-Parallel HEV Synchronous Reluctance Machine Torque Control

References

[1] Åström, K. and T. Hägglund. *Advanced PID Control.* Research Triangle Park, NC: ISA, 2005.

See Also

Blocks

Discrete PI Controller with Integral Anti-Windup

Introduced in R2017b

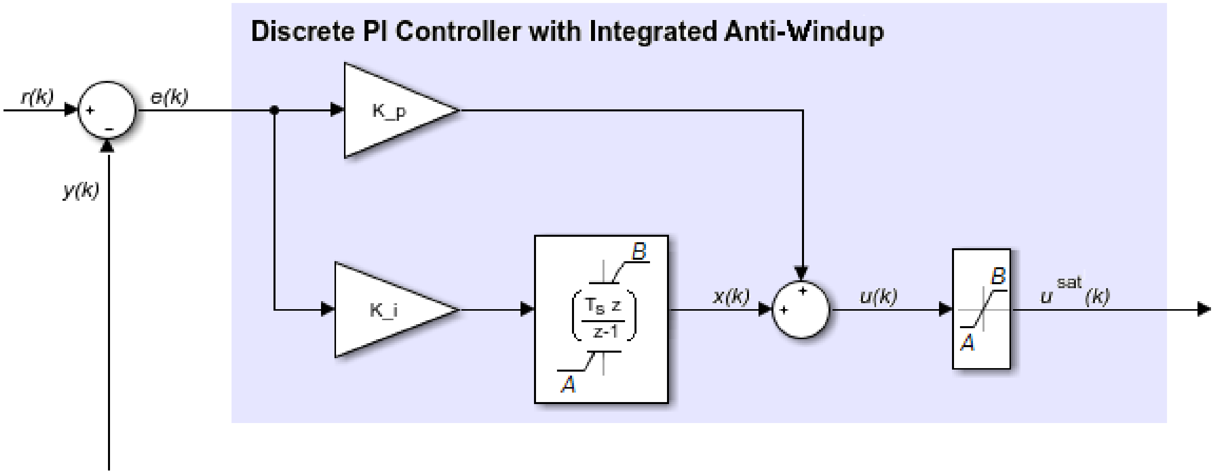
Discrete PI Controller with Integral Anti-Windup

Discrete-time PI control with integral anti-windup

Library: Simscape / Power Systems / Simscape Components / 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:

$$u(k) = \mathrm{sat}\left(K_p \mathrm{e}(\mathrm{k}) + \mathrm{sat}\left(K_i \frac{T_s z}{z-1} \mathrm{e}(\mathrm{k}), \mathrm{A}, \mathrm{B}\right), A, B\right),$$

$$sat(x,A,B)=\min\left(\max\left(x,\mathrm{A}\right),B\right),$$

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

Upper 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)` — 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?” (Simulink) and “Specify Sample Time” (Simulink).

Model Examples

Energy Balance in a 48V Starter Generator IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series-Parallel HEV Switched Reluctance Machine Speed Control Synchronous Reluctance Machine Torque Control

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5/D39. Piscataway, NJ: IEEE-SA, 2015.

See Also

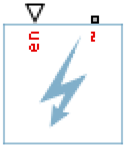
Blocks

Discrete PI Controller

Introduced in R2017b

Enabled Fault

Signal-enabled single-phase, two-phase, or three-phase grounded or ungrounded fault



Library

Passive Devices / Faults

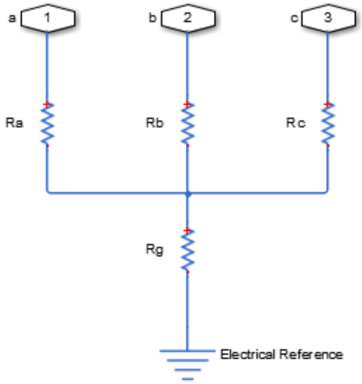
Description

The Enabled Fault models any permutation of a single-phase, two-phase, or three-phase grounded or ungrounded fault. You specify the fault activation threshold using the block **Threshold** parameter. An external control signal *en* enables the fault. The fault is active when *en* is greater than the threshold. The fault is inactive when *en* is less than or equal to the threshold.

You can set the Enabled Fault block to represent any of 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 Enabed Fault block.



You can determine the resistance in the equivalent circuit using the equations in the table.

Fault type	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
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}
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

Fault type	Value of R_a	Value of R_b	Value of R_c	Value of R_g
c-a	R_{pn}	$\frac{1}{G_{pn}}$	R_{pn}	Infinity / open circuit
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}
a-b-c	R_{pn}	R_{pn}	R_{pn}	Infinity / open circuit
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.

Parameters

- “Main Tab” on page 1-206
- “Parasitics Tab” on page 1-207

Main Tab

Fault type

Select one of the following:

- `None` — Specifies that the fault is not active. This is the default value.
- `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

Resistance between the phase connection and the neutral point when the fault is active. This parameter is visible if the **Fault type** parameter is set to anything other than `None`. The default value is `1e-3` `Ohm`.

Faulted neutral-ground resistance

Resistance between the neutral point and the electrical reference when fault is active. This parameter is visible if the **Fault type** parameter is set to any fault which includes a ground connection. The default value is `1e-3` `Ohm`.

Threshold

Threshold for activating the fault. If the input *en* is above the value for the **Threshold** parameter, then the fault is active. If the input *en* is equal to or less than the value for the **Threshold** parameter, then the fault is not active. This parameter is visible if the **Fault type** parameter is set to anything other than `None`. The default value is `0`.

Parasitics Tab

Unfaulted phase-neutral conductance

Conductance between the phase connections and the neutral point when a phase is not involved in the fault. The default value is `1e-6 1/Ohm`.

Ports

The block has the following ports:

`~`

Expandable three-phase port for connecting the fault to the system

`en`

Physical signal scalar control input port for enabling the fault

See Also

Time-Based Fault

Introduced in R2014a

Filtered Derivative (Discrete or Continuous)

Discrete-time or continuous-time filtered derivative

Library: Simscape / Power Systems / Simscape Components / Control / General Control

Description

The Filtered Derivative (Discrete or Continuous) block implements a filtered derivative in conformance with IEEE 421.5-2016[1].

You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

Equations

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 preceeding transfer function, the derivative defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}\left(Ku(t) - x(t)\right) \\ y(t) = \frac{1}{T}\left(Ku(t) - x(t)\right) \end{cases} \quad x(0) = u_0,\ y(0) = 0,$$

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.

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:

$$\begin{cases} x(n+1)=\left(1-\dfrac{T_s}{T}\right)x(n)+\left(\dfrac{T_s}{T}\right)u(n) \\ \quad y(n)=\dfrac{K}{T}\big(u(n)-x(n)\big) \end{cases} \quad x(0)=u_0,\ y(0)=0,$$

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

The block sets the state initial condition to the initial input, making the initial output zero.

Limiting the Integral

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 integrator 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.

`Sample time (-1 for inherited)` **— Sample time**

`-1` (default) | positive number

Filtered derivative sample time. Set this to `0` to implement a continuous derivative. Set this to `-1` or a positive number to implement a discrete derivative. For acceptable accuracy, set this value at least 10 times smaller than the **Time constant** parameter.

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.

See Also

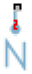
Blocks

Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous) | Washout (Discrete or Continuous)

Introduced in R2017b

Floating Neutral

Floating neutral point for phases of three-phase system



Library

Connections

Description

The Floating Neutral 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 block. If you want to create a neutral point that is connected to ground, use the Grounded Neutral block.

Ports

The block has the following ports:

~

  Expandable three-phase port

See Also

Grounded Neutral | Neutral Port

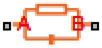
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Foster Thermal Model

Heat transfer through a semiconductor module

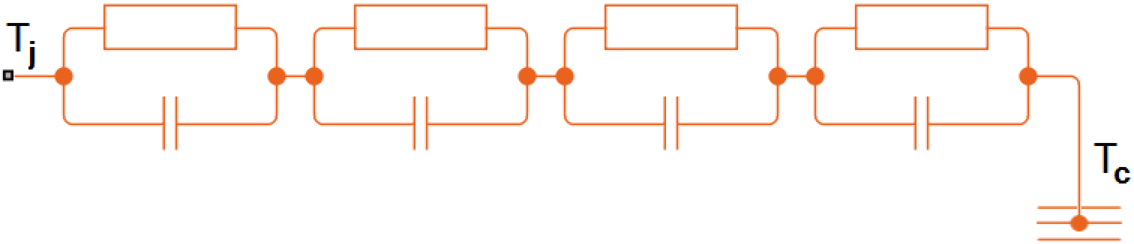


Library

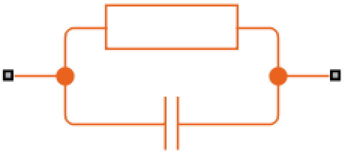
Semiconductors / Fundamental Components / Thermal

Description

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 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 nth 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.
- τ 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.

Parameters

Thermal resistance data

Thermal resistance values, $R_{thermal}$, of the semiconductor module, specified as a vector. The default value is `[ 0.0016 0.0043 0.0013 0.0014 ]` `K/W`.

Thermal time constant data

Thermal time constant values, τ, of the semiconductor module, specified as a vector. The default value is `[ 0.0068 0.064 0.32 2 ]` `s`.

Ports

The block has the following ports:

A

Thermal conserving port associated with the semiconductor junction.

B

Thermal conserving port associated with the base plate junction.

References

[1] Schütze, T. *AN2008-03: Thermal equivalent circuit models*. Application Note. V1.0. Germany: Infineon Technologies AG, 2008.

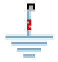
See Also

"Quantifying IGBT Thermal Losses" | Cauer Thermal Model Element | Thermal Resistor

Introduced in R2016a

Grounded Neutral

Ground connection for phases of three-phase system



Library

Connections

Description

The Grounded Neutral 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 block instead. If you want to create a floating neutral point, use the Floating Neutral block.

Ports

The block has the following ports:

~

Expandable three-phase port

See Also

Floating Neutral | Neutral Port

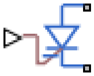
Topics

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

Introduced in R2013b

GTO

Gate Turn-Off Thyristor

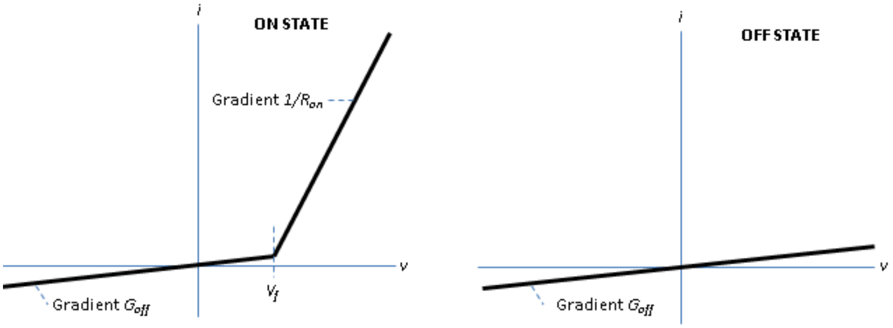


Library

Semiconductors / Fundamental Components

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 .



In the on state, the anode-cathode path behaves like a linear diode with forward-voltage drop, V_f, and on-resistance, R_{on}.

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.
- *Vf* is the forward voltage.
- *G* is the gate voltage.
- *Vgt* is the gate trigger voltage.
- *i* is the anode-cathode current.
- *Ih* is the holding current.
- *Vgt_off* is the gate turn-off voltage.
- *Ron* is the on-state resistance.
- *Goff* is the off-state conductance.

Using the Integral Diode tab of the block dialog box, 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. The block dialog box shows parameters relating to the Diode block.

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 Commutation Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Modeling Variants

The block provides four modeling variants. To select the desired variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

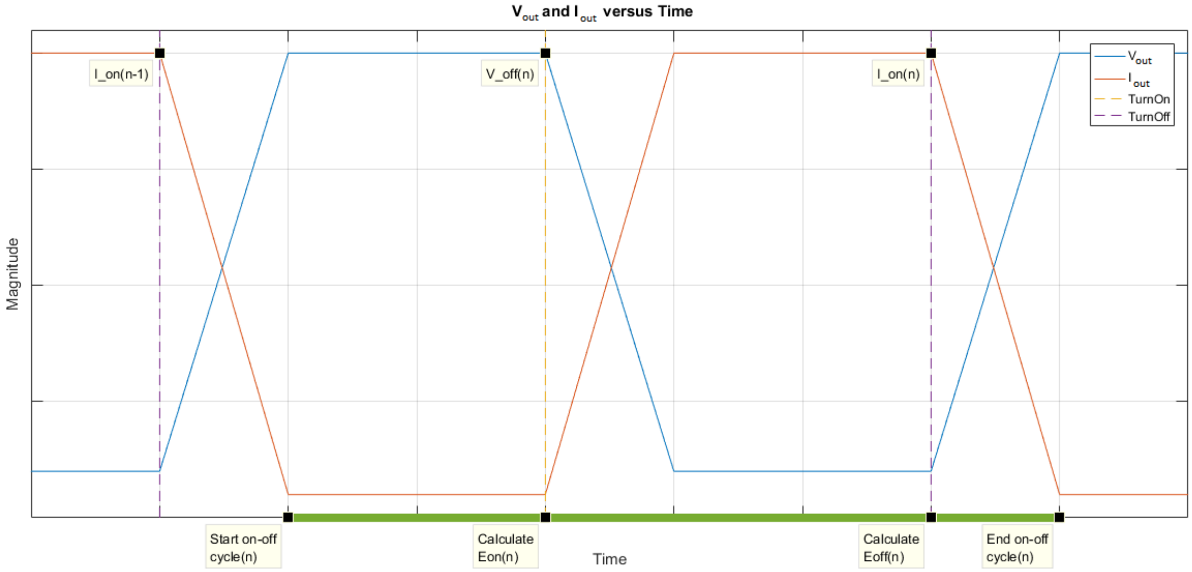
- **PS Control Port** — Contains a physical signal port that is associated with the gate terminal. This variant is the default.
- **Electrical Control Port** — Contains an electrical conserving port that is associated with the gate terminal.
- **PS Control Port | Thermal Port** — Contains a thermal port and a physical signal port that is associated with the gate terminal.
- **Electrical Control Port | Thermal Port** — Contains a thermal port and an electrical conserving port that is associated with the gate terminal.

The variants of this block without the thermal port do not simulate heat generation in the device.

The variants with the thermal port allow you to model the heat that switching events and conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block. The thermal port is hidden by default. To enable the thermal port, select a thermal block variant.

Thermal Loss Equations

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.



When the semiconductor turns on during the nth switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{on(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n-1)}),$$

where:

- $E_{on(n)}$ is the switch-on loss at the nth switch-on event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the nth switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n-1)}$ is the on-state output current, I_{out}, just before the device switches off during the cycle that precedes the nth switching cycle.

The function fcn is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{on\_data}, T, I_{on(n-1)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{on_data} is the **Switch-on loss, Eon=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns on during the nth switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{on(n)} = \left(\frac{V_{off(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{on\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{on_scalar} is the **Switch-on loss** parameter value.

When the semiconductor turns off during the nth switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{off(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n)}),$$

where:

- $E_{off(n)}$ is the switch-off loss at the nth switch-off event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the nth switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n)}$ is the on-state output current, I_{out}, just before the device switches off during the nth switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{off\_data}, T, I_{on(n)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{off_data} is the **Switch-off loss, Eoff=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns off during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{off\,(n)} = \left(\frac{V_{off\,(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{off\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{off_scalar} is the **Switch-off loss** parameter value.

If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int fcn(T, I_{out})\,dt,$$

where:

- $E_{conduction}$ is the heat loss due to electrical conduction.
- T is the device temperature.
- I_{out} is the device output current.

The function *fcn* is a 2-D lookup table:

$$Q_{conduction} = tablelookup(T_{j\_data}, I_{out\_data}, I_{out\_data\_repmat}\ .{*}\,V_{on\_data}, T, I_{out}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.

- I_{out_data} is the **Output current vector, Iout** parameter value.
- $I_{out\_data\_repmat}$ is a matrix that contains length, T_{j_data}, copies of I_{out_data}.
- V_{on_data} is the **On-state voltage, Von=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int \left(I_{out} * V_{on\_scalar} \right) dt,$$

where V_{on_scalar} is the **On-state voltage** parameter value.

The block uses the **Energy dissipation time constant** parameter to filter the amount of heat flow that the block outputs. The filtering allows the block to:

- Avoid discrete increments for the heat flow output
- Handle a variable switching frequency

The filtered heat flow is

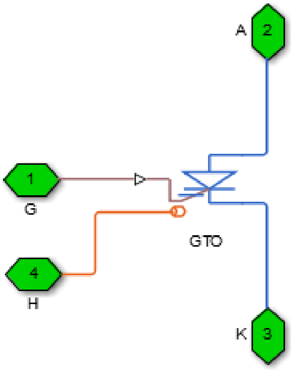
$$Q = \frac{1}{\tau} \left(\sum_{i=1}^{n} E_{on(i)} + \sum_{i=1}^{n} E_{off(i)} + E_{conduction} - \int Q\,dt \right),$$

where:

- Q is the heat flow from the component.
- τ is the **Energy dissipation time constant** parameter value.
- n is the number of switching cycles.
- $E_{on(i)}$ is the switch-on loss at the ith switch-on event.
- $E_{off(i)}$ is the switch-off loss at the ith switch-off event.
- $E_{conduction}$ is the heat loss due to electrical conduction.
- $\int Qdt$ is the total heat previously dissipated from the component.

Ports

The figure shows the block port names.



`G`

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

`A`

Electrical conserving port associated with the anode terminal.

`K`

Electrical conserving port associated with the cathode terminal.

`H`

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

- "Main Tab" on page 1-228
- "Integral Diode Tab" on page 1-228
- "Thermal Model Tab" on page 1-231

Main Tab

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Rate of change of voltage versus current above the forward voltage. The default value is `0.001 Ohm`.

Off-state conductance

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 1/Ohm`.

Gate trigger voltage, Vgt

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `1 V`.

Gate turn-off voltage, Vgt_off

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value. The default value is `-1 V`.

Holding current

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. The default value is `1 A`.

Integral Diode Tab

Integral protection diode

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`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 µAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Thermal Model Tab

The **Thermal Model** tab is enabled only when you select a block variant that includes a thermal port.

Thermal loss dependent on

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- `Voltage and current` — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data.
- `Voltage, current, and temperature` — Use vectors to specify the output current, switch-on loss, switch-off loss, on-state voltage, and temperature data. This is the default parameterization method.

Off-state voltage for losses data

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. The default value is `300 V`.

Energy dissipation time constant

Time constant used to average the switch-on losses, switch-off losses, and conduction losses. This value is equal to the period of the minimum switching frequency. The default value is `1e-4 s`.

Additional Parameters for Parameterizing by Voltage, Current, and Temperature

Temperature vector, Tj

Temperature values at which the switch-on loss, switch-off loss, and on-state voltage are specified. Specify this parameter using a vector quantity. The default value is `[ 298.15 398.15 ] K`.

Output current vector, Iout

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 vector quantity. The default value is `[ 0 10 50 100 200 400 600 ] A`.

Switch-on loss, Eon=fcn(Tj,Iout)

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. The default value is `[ 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`.

Switch-off loss, Eoff=fcn(Tj,Iout)

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. The default value is `[ 0 2.1e-4 0.00107 0.00214 0.00429 0.009859999999999999 0.01714; 0 4.3e-4 0.00197 0.00386 0.00771 0.01543 0.02271 ] J`.

On-state voltage, Von=fcn(Tj,Iout)

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. The default value is `[ 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`.

Additional Parameters for Parameterizing by Voltage and Current

Output current, Iout

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. The default value is `600 A`.

Switch-on loss

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. The default value is `0.02286 J`.

Switch-off loss

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. The default value is `0.01714 J`.

On-state voltage

Voltage drop across the block 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 scalar quantity. The default value is `2.7 V`.

See Also

Commutation Diode | Diode | IGBT | Ideal Semiconductor Switch | MOSFET | Thyristor

Topics

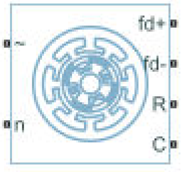
"Quantifying IGBT Thermal Losses"
"Simulate Thermal Losses in Semiconductors"
"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Hybrid Excitation Synchronous Machine

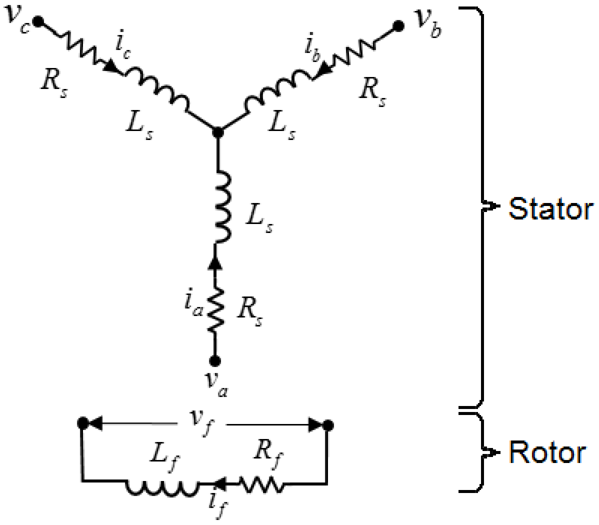
Hybrid excitation synchronous machine with three-phase wye-wound stator

Library: Simscape / Power Systems / Simscape Components /
Machines / Permanent Magnet Motor



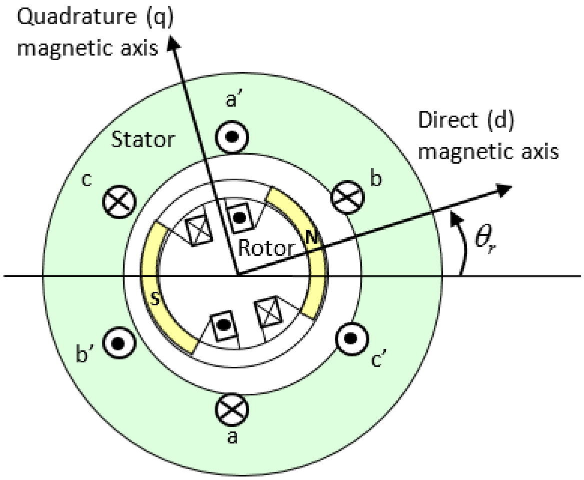
Description

The Hybrid Excitation Synchronous Machine 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 θ_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

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \dfrac{d\psi_a}{dt} \\ \dfrac{d\psi_b}{dt} \\ \dfrac{d\psi_c}{dt} \end{bmatrix},$$

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:

- ψ_a, ψ_b, and ψ_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.
- ψ_{am}, ψ_{bm}, and ψ_{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

$$\theta_e = N\theta_r,$$

$$L_{aa} = L_s + L_m \cos(2\theta_e),$$

$$L_{bb} = L_s + L_m \cos(2(\theta_e - 2\pi/3)),$$

$$L_{cc} = L_s + L_m \cos(2(\theta_e + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos\left(2(\theta_e + \pi/6)\right),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos\left(2(\theta_e + \pi/6 - 2\pi/3)\right),$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos\left(2(\theta_r + \pi/6 + 2\pi/3)\right),$$

where:

- N is the number of rotor pole pairs.
- θ_r is the rotor mechanical angle.
- θ_e is the rotor electrical angle.
- 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.

The magnetization flux linking winding, *a-a'* is a maximum when θ_r = 0° and zero when θ_r = 90°. Therefore:

$$\psi_m = \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \begin{bmatrix} \psi_m \cos\theta_r \\ \psi_m \cos(\theta_r - 2\pi/3) \\ \psi_m \cos(\theta_r + 2\pi/3) \end{bmatrix},$$

$$L_{mf} = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix} = \begin{bmatrix} L_{mf} \cos\theta_r \\ L_{mf} \cos(\theta_r - 2\pi/3) \\ L_{mf} \cos(\theta_r + 2\pi/3) \end{bmatrix},$$

and

$$\Psi_f = L_f i_f + L_{mf}^T \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix},$$

where:

- ψ_m is the linked motor flux.
- L_{mf} is the mutual field armature inductance.
- ψ_f is the flux linking the field winding.
- L_f is the field winding inductance.
- $\left[L_{mf}\right]^T$ is the transform of the L_{mf} vector, that is,

$$\left[L_{mf}\right]^T = \begin{bmatrix} L_{amf} \\ L_{bmf} \\ L_{cmf} \end{bmatrix}^T = \begin{bmatrix} L_{amf} & L_{bmf} & L_{cmf} \end{bmatrix}.$$

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

$$P = 2/3 \begin{bmatrix} \cos\theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin\theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

The inverse of the Park transformation is defined by

$$P^{-1} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e & 1 \\ \cos(\theta_e - 2\pi/3) & -\sin(\theta_e - 2\pi/3) & 1 \\ \cos(\theta_e + 2\pi/3) & -\sin(\theta_e + 2\pi/3) & 1 \end{bmatrix}.$$

Applying the Park transformation to the first two electrical defining equations produces equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} + L_{mf} \frac{di_f}{dt} - N\omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega(i_d L_d + \psi_m + i_f L_{mf}),$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt},$$

$$v_f = R_f i_f + L_f \frac{di_f}{dt} + \frac{3}{2} L_{mf} \frac{di_d}{dt},$$

$$T = \frac{3}{2} N \left(i_q \left(i_d L_d + \psi_m + i_f L_{mf} \right) - i_d i_q L_q \right),$$

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. $L_d = L_s + M_s + 3/2\ L_m$.
- ω 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 Synchronous Machine 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.

Assumptions

The block assumes that 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.

`n` — Neutral phase

electrical

Electrical conserving port associated with the neutral phase.

`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

Main

`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.

`Stator parameterization` — Parameterization method
`Specify Ld, Lq and L0` (default) | `Specify Ls, Lm, and Ms`

Method for parameterizing the stator.

Dependencies

Selecting `Specify Ld, Lq and L0` enables these parameters:

- **Stator d-axis inductance, Ld**
- **Stator q-axis inductance, Lq**
- **Stator zero-sequence inductance, L0**

Selecting `Specify Ls, Lm, and Ms` enables these parameters:

- **Stator self-inductance per phase, Ls**
- **Stator inductance fluctuation, Lm**
- **Stator mutual inductance, Ms**

`Stator d-axis inductance, Ld` — Inductance
`0.0031 H` (default)

Direct-axis inductance of the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator q-axis inductance, Lq` — Inductance
`0.0045 H` (default)

Quadrature-axis inductance of the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator zero-sequence inductance, L0` — Inductance
`0.0006 H` (default)

Zero-axis inductance for the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator self-inductance per phase, Ls` — Inductance
`0.0027 H` (default)

Average self-inductance of the three stator windings.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ls, Lm, and Ms`.

`Stator inductance fluctuation, Lm` — Inductance
`-0.0005 H` (default)

Amplitude of the fluctuation in self-inductance and mutual inductance with the rotor angle.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ls, Lm, and Ms`.

`Stator mutual inductance, Ms` — Inductance
`0.0011 H` (default)

Average mutual inductance between the stator windings.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ls, Lm, and Ms`.

`Field winding inductance, Lf` — Inductance
`0.06 H` (default)

Inductance of the field winding.

`Mutual field armature inductance, Lmf` — Inductance
`0.0067 H` (default)

Armature-field mutual inductance.

`Stator resistance per phase, Rs` — Resistance
`0.7 Ohm` (default)

Resistance of each of the stator windings.

`Field winding resistance, Rf` — Resistance
`2.85 Ohm` (default)

Resistance of the field winding.

Mechanical

`Rotor inertia` — Inertia
`0.01 kg*m^2` (default)

Inertia of the rotor.

`Rotor Damping` — Damping
`0 N*m/(rad/s)` (default)

Damping of the rotor.

Initial Conditions

`Initial currents, [i_d i_q i_0 i_f]` — Current
l`[0, 0, 0, 0] A` (default) | vector

Initial *d*-, *q*-, *0*-sequence and field winding currents.

`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.

`Initial rotor angle` — Angle
`0 deg` (default) | `0-360 deg`

Rotor angle at simulation start time.

`Initial rotor speed` — Angular Speed
`0 rpm` (default)

Rotor speed at simulation start time. If the rotor inertia, J, is zero, the initial speed of the rotor is zero rpm and the initial rotor speed is ignored.

Model Examples

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.

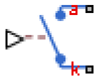
See Also

Brushless DC Motor | Permanent Magnet Synchronous Motor | Switched Reluctance Machine | Synchronous Machine Field Circuit (SI) | Synchronous Machine Field Circuit (pu) | Synchronous Machine Measurement | Synchronous Reluctance Machine

Introduced in R2017b

Ideal Semiconductor Switch

Ideal Semiconductor Switch



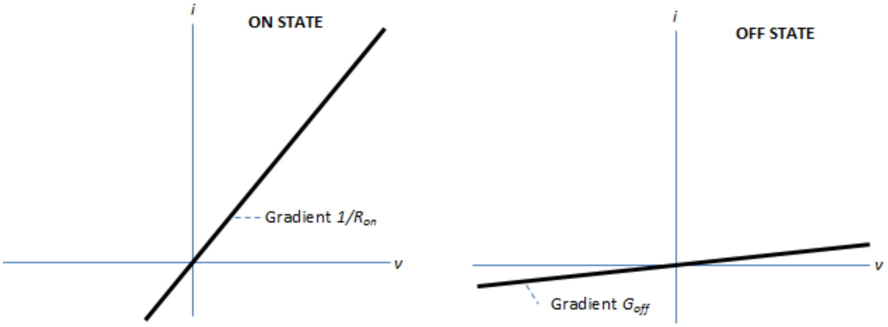
Library

Semiconductors / Fundamental Components

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 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. The block dialog box shows parameters relating to the Diode block.
Precisely specify reverse-mode charge dynamics.	`Protection diode with charge dynamics`	The block includes an integral copy of the Commutation Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Parameters

- "Main Tab" on page 1-247
- "Integral Diode Tab" on page 1-248

Main Tab

On-state resistance

Anode-cathode resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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-6 1/Ohm`.

Threshold voltage, Vth

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `0.5 V`.

Integral Diode Tab

Integral protection diode

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`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

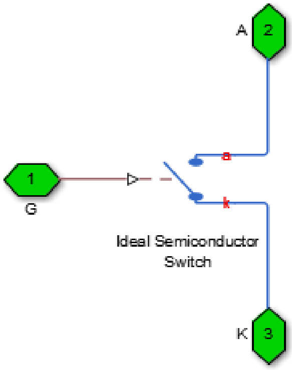
The default value is `1500 µAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Ports

This figure shows the block port names.



`G`

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

`A`

Electrical conserving port associated with the anode terminal.

`K`

Electrical conserving port associated with the cathode terminal.

See Also

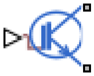
Topics

"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

IGBT

Insulated-Gate Bipolar Transistor

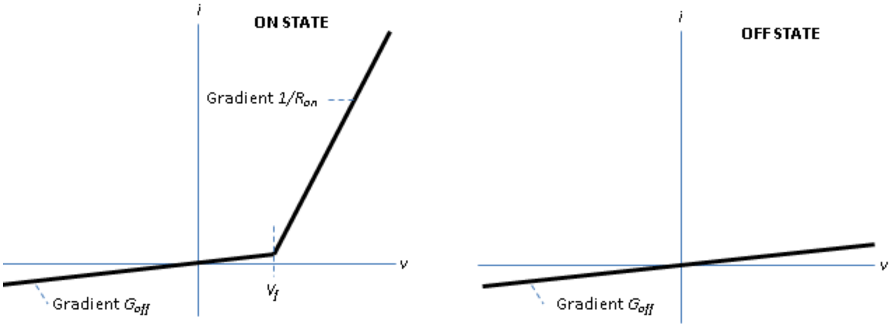


Library

Semiconductors / Fundamental Components

Description

The IGBT block models an insulated-gate bipolar transistor (IGBT). The I-V 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.



In the on state, the collector-emitter path behaves like a linear diode with forward-voltage drop, V_f, and on-resistance, R_{on}.

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.
- *Vf* is the forward voltage.
- *G* is the gate-emitter voltage.
- *Vth* is the threshold voltage.
- *i* is the collector-emitter current.
- *Ron* is the on-state resistance.
- *Goff* is the off-state conductance.

Integral Protection Diode Option

Using the Integral Diode tab of the block dialog box, 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. The block dialog box shows parameters relating to the Diode block.

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 Commutation Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Modeling Variants

The block provides four modeling variants. To select the desired variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

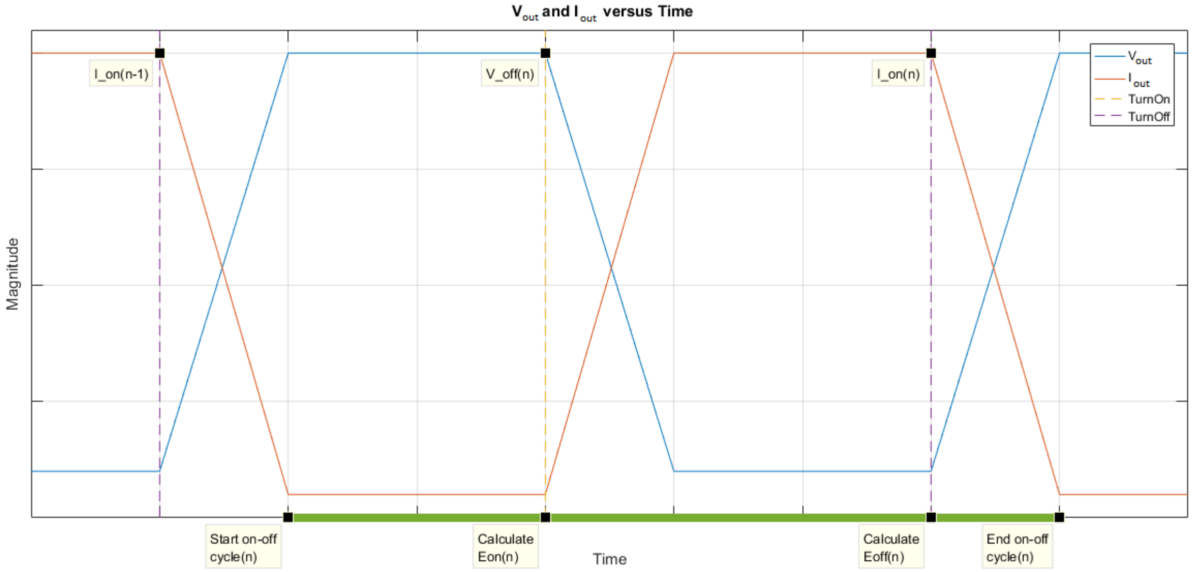
- **PS Control Port** — Contains a physical signal port that is associated with the gate terminal. This variant is the default.
- **Electrical Control Port** — Contains an electrical conserving port that is associated with the gate terminal.
- **PS Control Port | Thermal Port** — Contains a thermal port and a physical signal port that is associated with the gate terminal.
- **Electrical Control Port | Thermal Port** — Contains a thermal port and an electrical conserving port that is associated with the gate terminal.

The variants of this block without the thermal port do not simulate heat generation in the device.

The variants with the thermal port allow you to model the heat that switching events and conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block. The thermal port is hidden by default. To enable the thermal port, select a thermal block variant.

Thermal Loss Equations

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 nth switching cycle, during which the block turns off and then on.



When the semiconductor turns on during the *n*th switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{on(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n-1)}),$$

where:

- $E_{on(n)}$ is the switch-on loss at the *n*th switch-on event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the *n*th switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n\text{-}1)}$ is the on-state output current, I_{out}, just before the device switches off during the cycle that precedes the n*th* switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{on\_data}, T, I_{on(n-1)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{on_data} is the **Switch-on loss, Eon=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns on during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{on(n)} = \left(\frac{V_{off(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{on\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{on_scalar} is the **Switch-on loss** parameter value.

When the semiconductor turns off during the *n*th switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{off(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n)}),$$

where:

- $E_{off(n)}$ is the switch-off loss at the *n*th switch-off event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the *n*th switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n)}$ is the on-state output current, I_{out}, just before the device switches off during the *n*th switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{off\_data}, T, I_{on(n)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{off_data} is the **Switch-off loss, Eoff=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns off during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{off\,(n)} = \left(\frac{V_{off\,(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{off\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{off_scalar} is the **Switch-off loss** parameter value.

If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int fcn(T, I_{out})\,dt,$$

where:

- $E_{conduction}$ is the heat loss due to electrical conduction.
- T is the device temperature.
- I_{out} is the device output current.

The function *fcn* is a 2-D lookup table:

$$Q_{conduction} = tablelookup(T_{j\_data}, I_{out\_data}, I_{out\_data\_repmat} .{*} V_{on\_data}, T, I_{out}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.

- I_{out_data} is the **Output current vector, Iout** parameter value.
- $I_{out\_data\_repmat}$ is a matrix that contains length, T_{j_data}, copies of I_{out_data}.
- V_{on_data} is the **On-state voltage, Von=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int \left(I_{out} * V_{on\_scalar} \right) dt,$$

where V_{on_scalar} is the **On-state voltage** parameter value.

The block uses the **Energy dissipation time constant** parameter to filter the amount of heat flow that the block outputs. The filtering allows the block to:

- Avoid discrete increments for the heat flow output
- Handle a variable switching frequency

The filtered heat flow is

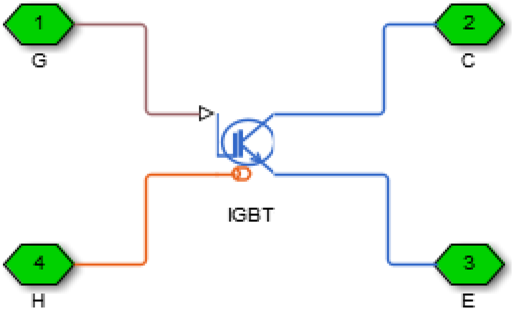
$$Q = \frac{1}{\tau} \left(\sum_{i=1}^{n} E_{on(i)} + \sum_{i=1}^{n} E_{off\,(i)} + E_{conduction} - \int Q\,dt \right),$$

where:

- Q is the heat flow from the component.
- τ is the **Energy dissipation time constant** parameter value.
- n is the number of switching cycles.
- $E_{on(i)}$ is the switch-on loss at the ith switch-on event.
- $E_{off(i)}$ is the switch-off loss at the ith switch-off event.
- $E_{conduction}$ is the heat loss due to electrical conduction.
- $\int Q dt$ is the total heat previously dissipated from the component.

Ports

The figure shows the block port names.



G

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port

C

Electrical conserving port associated with the collector terminal

E

Electrical conserving port associated with the emitter terminal

H

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

- “Main Tab” on page 1-260
- “Integral Diode Tab” on page 1-260
- “Thermal Model Tab” on page 1-263

Main Tab

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Collector-emitter resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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**. The default value is `1e-5 1/Ohm`.

Threshold voltage, Vth

Collector-emitter voltage at which the device turns on. The default value is `6 V`.

Integral Diode Tab

Integral protection diode

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`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 µAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Thermal Model Tab

The **Thermal Model** tab is enabled only when you select a block variant that includes a thermal port.

Thermal loss dependent on

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- `Voltage and current` — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data.
- `Voltage, current, and temperature` — Use vectors to specify the output current, switch-on loss, switch-off loss, on-state voltage, and temperature data. This is the default parameterization method.

Off-state voltage for losses data

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. The default value is `300 V`.

Energy dissipation time constant

Time constant used to average the switch-on losses, switch-off losses, and conduction losses. This value is equal to the period of the minimum switching frequency. The default value is `1e-4 s`.

Additional Parameters for Parameterizing by Voltage, Current, and Temperature

Temperature vector, Tj

Temperature values at which the switch-on loss, switch-off loss, and on-state voltage are specified. Specify this parameter using a vector quantity. The default value is `[ 298.15 398.15 ] K`.

Output current vector, Iout

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 vector quantity. The default value is `[ 0 10 50 100 200 400 600 ] A`.

Switch-on loss, Eon=fcn(Tj,Iout)

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. The default value is `[ 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`.

Switch-off loss, Eoff=fcn(Tj,Iout)

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. The default value is `[ 0 2.1e-4 0.00107 0.00214 0.00429 0.009859999999999999 0.01714; 0 4.3e-4 0.00197 0.00386 0.00771 0.01543 0.02271 ] J`.

On-state voltage, Von=fcn(Tj,Iout)

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. The default value is `[ 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`.

Additional Parameters for Parameterizing by Voltage and Current

Output current, Iout

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. The default value is `600 A`.

Switch-on loss

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. The default value is `0.02286 J`.

Switch-off loss

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. The default value is `0.01714 J`.

On-state voltage

Voltage drop across the block 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 scalar quantity. The default value is `2.7 V`.

See Also

Commutation Diode | Diode | GTO | Ideal Semiconductor Switch | MOSFET | Thyristor

Topics

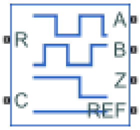
“Quantifying IGBT Thermal Losses”
“Simulate Thermal Losses in Semiconductors”
“Switch Between Physical Signal and Electrical Ports”

Introduced in R2013b

Incremental Shaft Encoder

Encoding

Library: Simscape / Power Systems / Simscape Components / Sensors



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.

See Also

Simscape Blocks

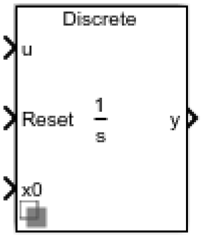
Ideal Rotational Motion Sensor

Introduced in R2017b

Integrator (Discrete or Continuous)

Discrete-time or continuous-time integrator

Library: Simscape / Power Systems / Simscape Components / Control / General Control



Description

The Integrator (Discrete or Continuous) block implements a simple integrator in conformance with IEEE 421.5-2016[1].

You can switch between continuous and discrete implementations of the integrator using the **Sample time** parameter.

Equations

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 preceeding 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.

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:

$A <= x <= B$.

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:
  $A <= y <= B$.

  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` — Sample time
`-1` (default) | positive number

Integrator sample time. Set this to `0` to implement a continuous integrator. Set this to `-1` or a positive number to implement a discrete integrator.

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

Blocks

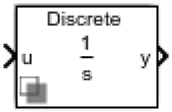
Filtered Derivative (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous) | Washout (Discrete or Continuous)

Introduced in R2017b

Integrator with Wrapped State (Discrete or Continuous)

Discrete-time or continuous-time integrator with wrapped state

Library: Simscape / Power Systems / Simscape Components / 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[1].

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

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 preceeding 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.

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:

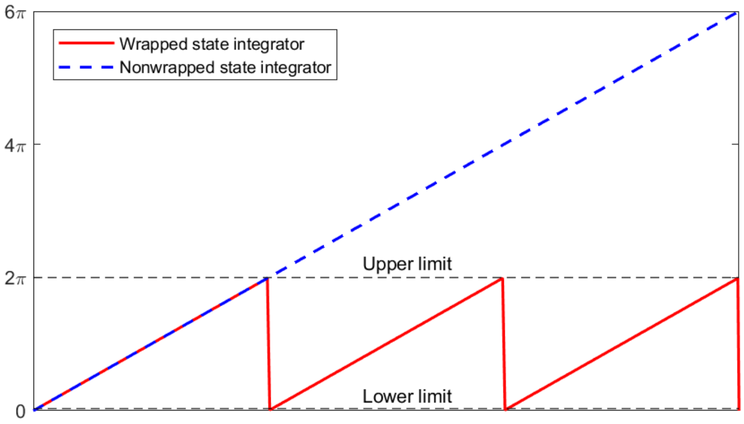
- u is the integrator input.
- x is the integrator state.
- y is the integrator output.
- n is the simulation time step.
- x_0 is the initial state of the integrator.

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 *2π*, 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` — Sample time

`-1` (default) | positive number

Integrator sample time. Set this to `0` to implement a continuous integrator. To implement a discrete integrator, set this to `-1` or a positive number.

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

Blocks

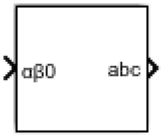
Filtered Derivative (Discrete or Continuous) | Integrator (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous) | Washout (Discrete or Continuous)

Introduced in R2017b

Inverse Clarke Transform

Implement *αβ0* to *abc* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

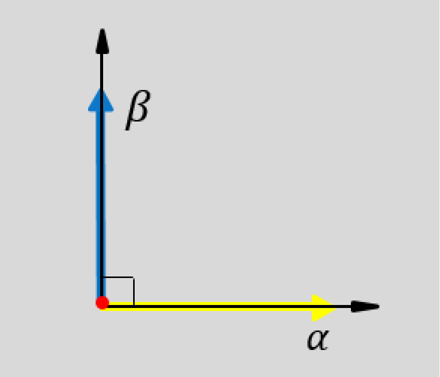


Description

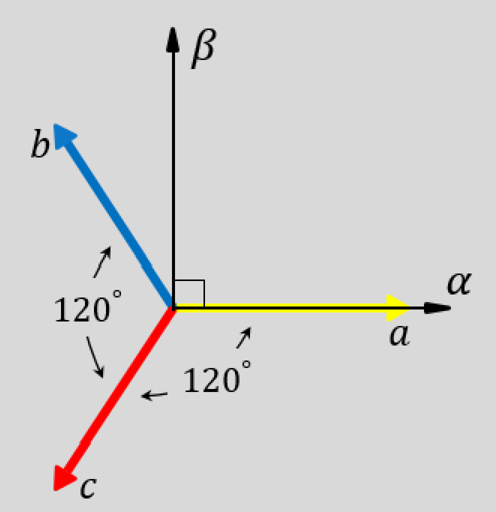
The Inverse Clarke Transform block converts the time-domain alpha, beta, and zero components in a stationary reference frame to three-phase components in an *abc* reference frame. The block can preserve the active and reactive powers with the powers of the system in the stationary reference frame by 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:

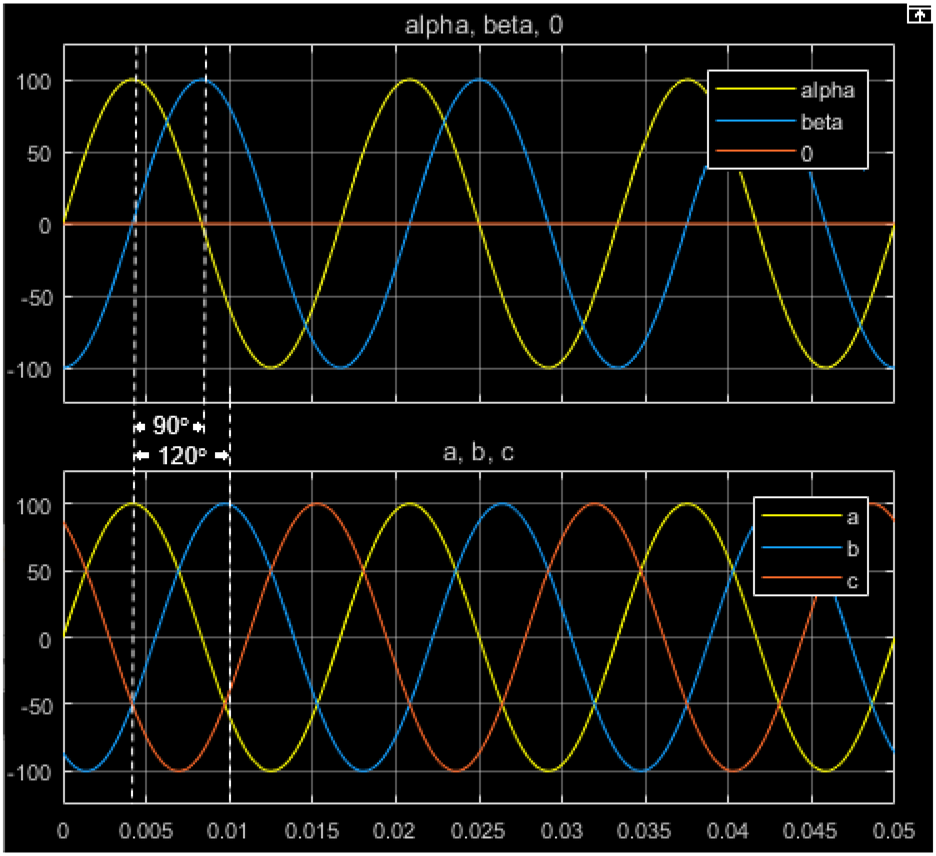
- Balanced α, β, and zero components in a stationary reference frame



- The direction of the magnetic axes of the stator windings in the stationary *αβ0* reference frame and the *abc* reference frame



- The time-response of the individual components of equivalent balanced *αβ0* 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:

- a and β 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

αβ0 — *α*-*β* axis and zero components
vector

Alpha-axis component,a, beta-axis component β, 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.

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.

See Also

Blocks

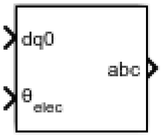
Clarke Transform | Clarke to Park Angle Transform | Inverse Park Transform | Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Inverse Park Transform

Implement *dq0* to *abc* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

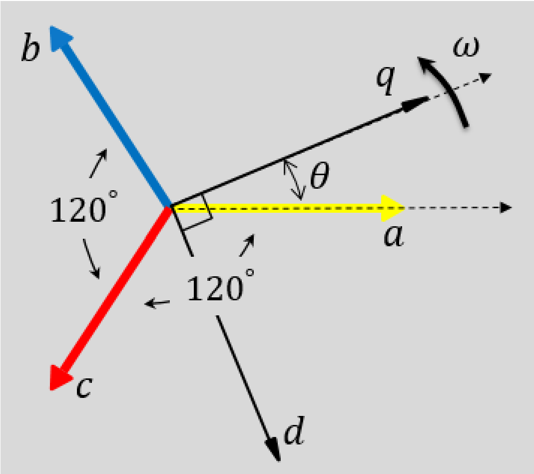


Description

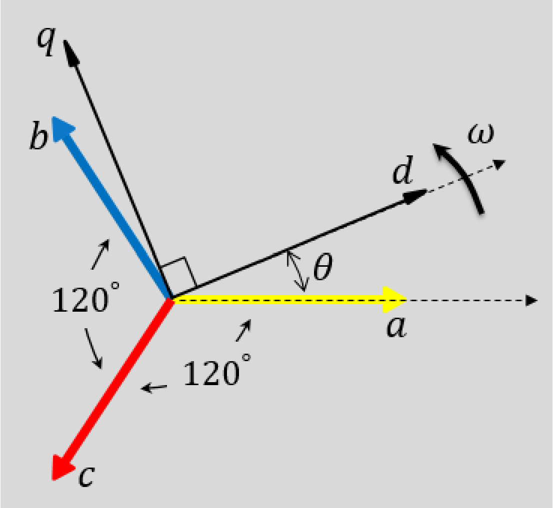
The Inverse Park Transform block converts the time-domain direct, quadrature, and zero components in a rotating reference frame to the components of a three-phase system in an *abc* reference frame. The block can preserve the active and reactive powers with the powers of the system in the rotating reference 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.

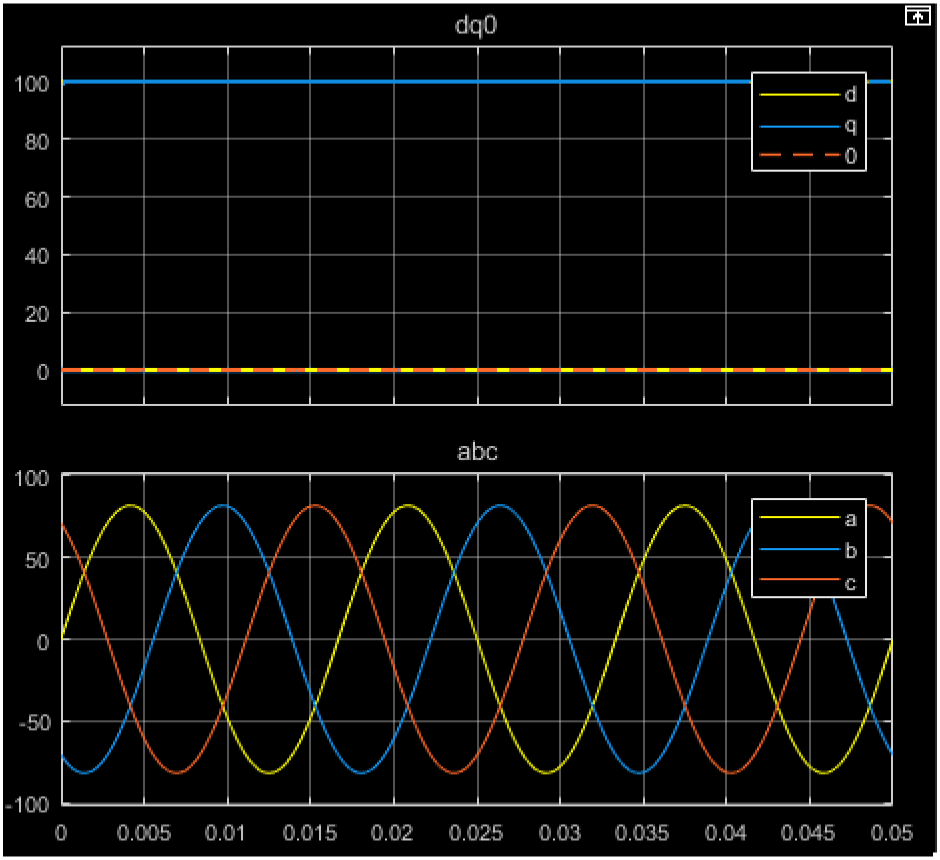


In both cases, the angle $\theta = \omega t$, where

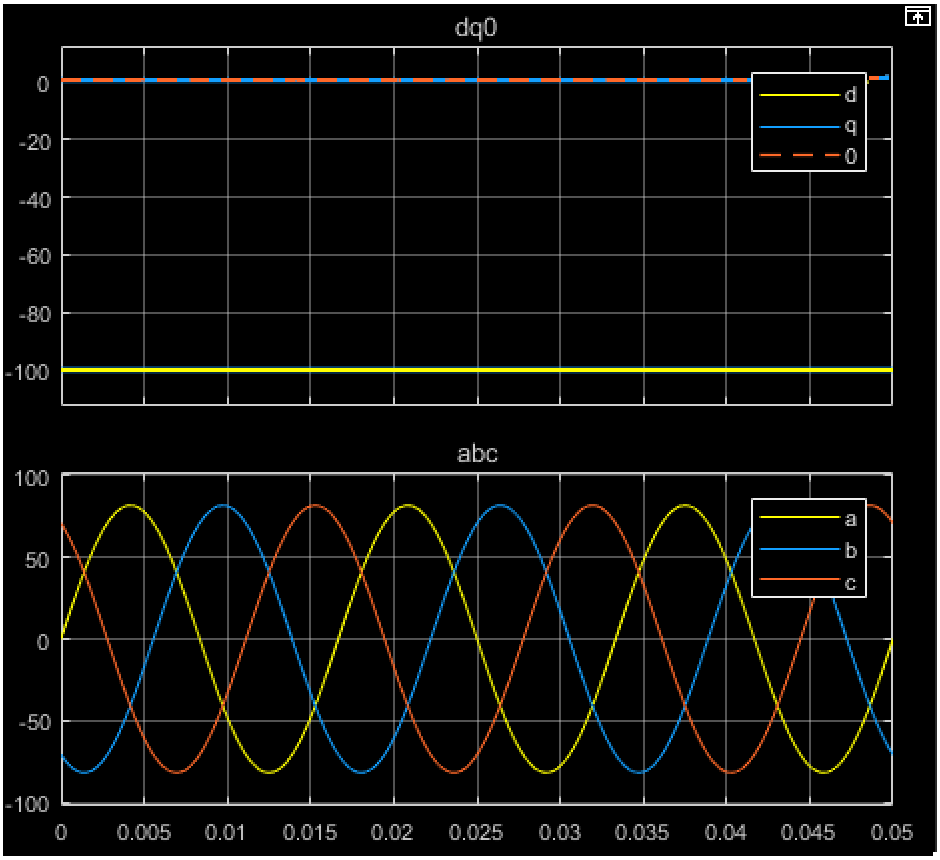
- θ is the angle between the a and q axes for the q-axis alignment or the angle between the a and d axes for the d-axis alignment.
- ω is the rotational speed of the d-q reference frame.
- t is the time, in s, from the initial alignment.

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`

θ_{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.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator HESM Torque Control HESM Velocity Control IPMSG Voltage Stabilization IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control SM Torque Control SM Velocity Control Switched Reluctance Machine Speed Control Synchronous Reluctance Machine Velocity Control Three-Phase Asynchronous Drive with Sensor Control Three-Phase Asynchronous Drive with Sensorless Control Three-Phase PMSM Drive

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.

See Also

Blocks

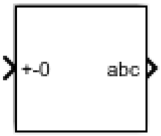
Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Inverse Symmetrical-Components Transform

Implement +-*0* to *abc* transform

Library: Simscape / Power Systems / Simscape Components / 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

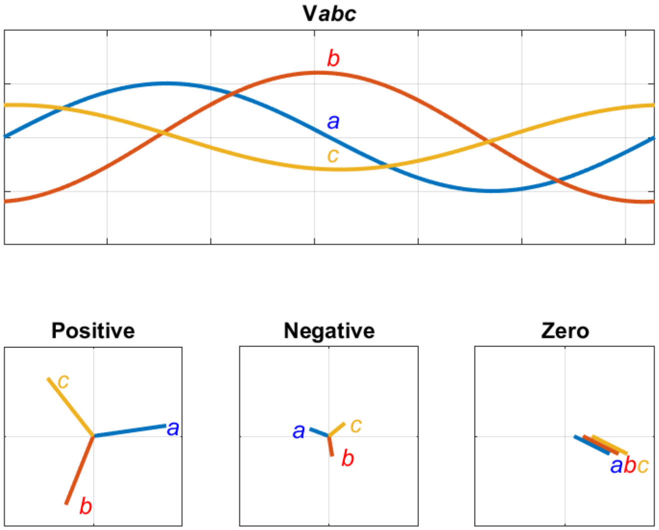
$$\begin{bmatrix} V_{c+} \\ V_{c-} \\ V_{c0} \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}.$$

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.

References

[1] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.

See Also

Blocks

Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform | Symmetrical-Components Transform

Introduced in R2017b

Lead-Lag (Discrete or Continuous)

Discrete-time or continuous-time lead-lag compensator

Library: Simscape / Power Systems / Simscape Components / Control / General Controls

Description

The Lead-Lag (Discrete or Continuous) block implements a lead-lag compensator in conformance with IEEE 421.5-2016[1].

You can switch between continuous and discrete implementations of the block using the **Sample time** parameter.

Equations

To configure the compensator for continuous time, set the **Sample time** property to `0`. This representation is equivalent to the continuous transfer function:

$$G(s) = \frac{T_1 s + 1}{T_2 s + 1},$$

where:

- T_1 is the lead time constant.
- T_2 is the lag time constant.

From the preceeding transfer function, the compensator defining equations are:

$$\begin{cases} \dot{x}(t) = \dfrac{1}{T_2}\left(u(t) - x(t)\right) \\ y(t) = \dfrac{T_1}{T_2} u(t) + \left(1 - \dfrac{T_1}{T_2}\right) x(t) \end{cases} \quad y(0) = x(0) = u_0,$$

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.

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

The block sets the state and output initial conditions to the initial input.

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.

`Sample time (-1 for inherited)` — Sample time
`-1` (default) | positive number

Compensator sample time. Set this to `0` to implement a continuous lead-lag compensator. Set this to `-1` or a positive number to implement a discrete lead-lag compensator.

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

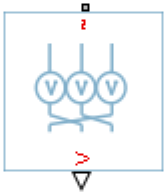
Blocks

Filtered Derivative (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous) | Washout (Discrete or Continuous)

Introduced in R2017b

Line Voltage Sensor

Measure line voltages in three-phase system



Library

Sensors

Description

The Line Voltage Sensor 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 phase voltages.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

V

Three-element physical signal vector output port associated with the voltages between the phases

See Also

Phase Voltage Sensor

Topics

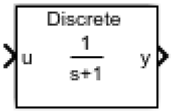
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

Low-Pass Filter (Discrete or Continuous)

Discrete-time or continuous-time low-pass filter

Library: Simscape / Power Systems / Simscape Components / Control / General Control



Description

The Low-Pass Filter (Discrete or Continuous) block implements a low-pass filter in conformance with IEEE 421.5-2016[1]. 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

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 preceeding transfer function, the filter defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}\left(Ku(t) - x(t)\right) \\ y(t) = x(t) \end{cases} \quad y(0) = x(0) = Ku_0,$$

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.

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

The block sets the state and output initial conditions proportionally to the initial input.

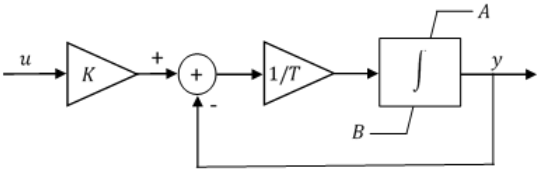
Limiting the Integral

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.

`Sample time` — Sample time

`-1` (default) | positive number

Low-pass filter sample time. Set this to `0` to implement a continuous low-pass filter. Set this to `-1` or a positive number to implement a discrete low-pass filter.

Model Examples

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

Blocks

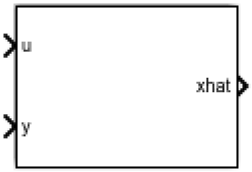
Filtered Derivative (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Washout (Discrete or Continuous)

Introduced in R2017b

Luenberger Observer

Discrete-time Luenberger observer

Library: Simscape / Power Systems / Simscape Components / 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:

$$\breve{x}(k+1)=A_d\breve{x}(k)+B_d u(k)+L_d(y(k)-\breve{y}(k)),$$

where:

- $\breve{x}(k)$ is the k^{th} estimated state vector.
- $\breve{y}(k)$ is the k^{th} estimated output vector.

- *u(k)* is the k^{th} input vector.
- *y(k)* is the k^{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:

$$e\left(k+1\right)=(A_d-L_dC_d)e(k),$$

where:

- *e(k)* is the k^{th} error vector.
- C_d is the output matrix.

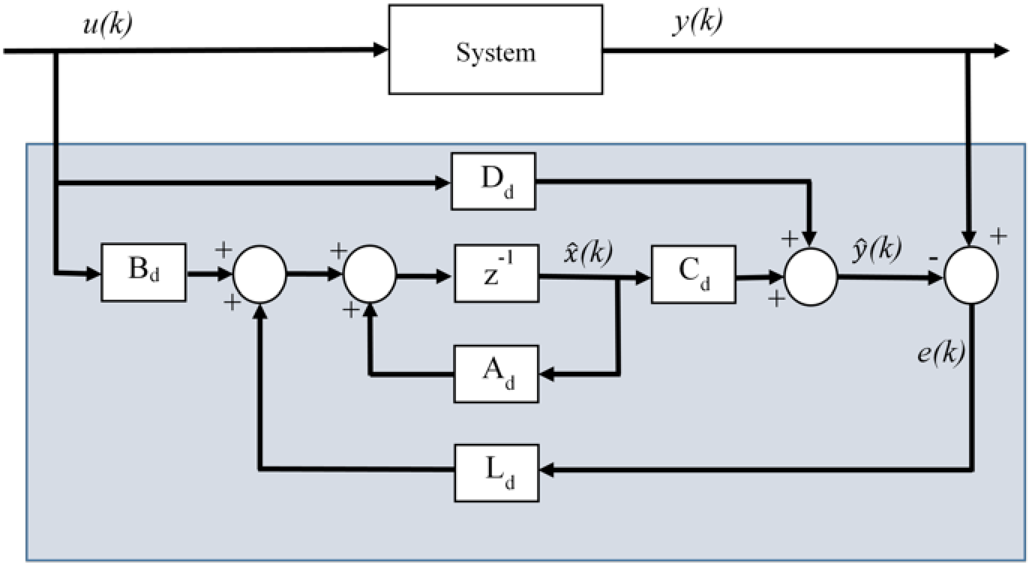
The estimation error converges to zero when A_d-L_dC_d has its eigenvalues inside the unit circle. Therefore, the value of L_d should be such that this goal is achieved. The block computes the observer gain by solving

$$L_d^T=GX^{-1},$$

where G is an arbitrary matrix and X is obtained by solving the Sylvester equation:

$$A_d^TX-X\Lambda=C_d^TG.$$

Here, Λ 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 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_dC_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.

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.

See Also

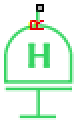
Blocks

ASM Flux Observer | State Feedback Controller

Introduced in R2017b

Machine Inertia

Machine inertia

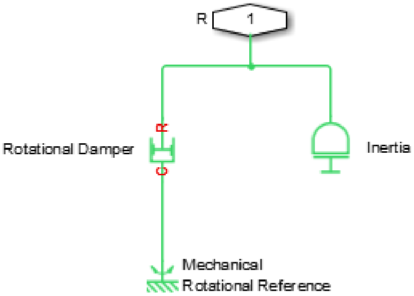


Library

Machines

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 kg·m^2.
- H is the inertia constant in sW/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:

- ω_{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).

Display Option

You can display machine parameters using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** menu, select **Display Parameters** to display the machine per-unit base values and inertia parameters in the MATLAB Command Window.

Parameters

- "Main Tab" on page 1-317
- "Inertia Tab" on page 1-317
- "Initial Conditions Tab" on page 1-318

Main Tab

Rated apparent power

Machine rated apparent power. The default value is `555e6 V*A`.

Rated electrical frequency

Nominal electrical frequency corresponding to the machine rated apparent power. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Inertia Tab

Specify inertia parameterization by

Inertia specification. The default value is `Inertia constant, H`.

Inertia constant, H

Inertia constant. This parameter is visible only if you set **Specify inertia parameterization by** to `Inertia constant, H`. The default value is `3.525 sW/VA`.

Actual inertia, J

Inertia. This parameter is visible only if you set **Specify inertia parameterization by** to `Actual inertia, J`. The default value is `27548` kg·m^2.

Specify damper parameterization by

Damping specification. The default value is `Per-unit damping coefficient, pu_D`.

Per-unit damping coefficient

Damping coefficient in per-unit. This parameter is visible only if you set **Specify damper parameterization by** to `Per-unit damping coefficient, pu_D`. The default value is `0.01`.

SI damping coefficient

Damping coefficient in SI units. This parameter is visible only if you set **Specify damper parameterization by** to `SI damping coefficient, D`. The default value is `39.0509` Nm/(rad/s).

Initial Conditions Tab

Specify initialization by

Frequency initialization. The default value is `Initial electrical frequency`.

Initial electrical frequency

Initial electrical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial electrical frequency`. The default value is `60 Hz`.

Initial mechanical frequency

Initial mechanical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial mechanical frequency`. The default value is `60 Hz`.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the machine rotor

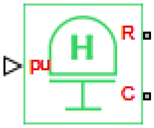
References

[1] Kundur, P. *Power System Stability and Control*. New York, NY: McGraw Hill, 1993.

Introduced in R2013b

Machine Mechanical Power (pu)

Machine mechanical power defined in the per-unit system



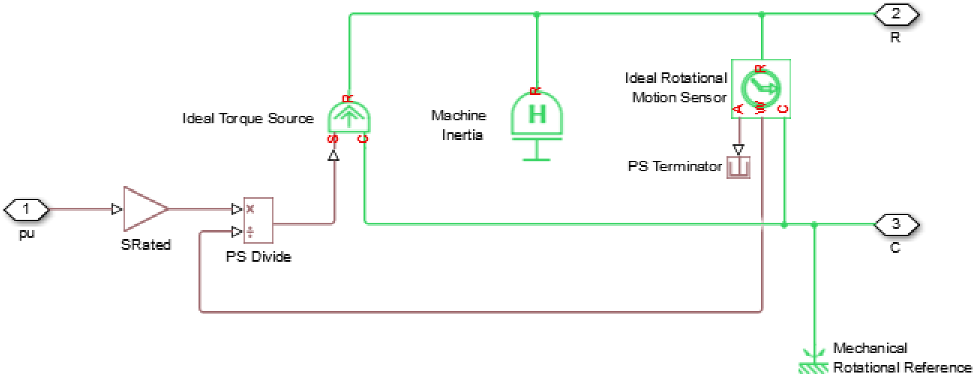
Library

Machines

Description

The Machine Mechanical Power (pu) 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 pu defines the per-unit mechanical power that is input to the machine. In motor mode, it defines the mechanical power output from the machine. The pu input must always be positive.

The figure shows an equivalent configuration to the Machine Mechanical Power (pu) block using Simscape mechanical rotational components.



Electrical Defining Equations

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^2.
- 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:

- ω_{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).

Parameters

- “Main Tab” on page 1-322
- “Inertia Tab” on page 1-323
- “Initial Conditions Tab” on page 1-324

Main Tab

Input power sign convention

Machine type specification. The choices are `Generator` and `Motor`. The default value is `Generator`.

Rated apparent power

Rated apparent power of the connected machine. The default value is `555e6 V*A`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power of the connected machine. The default value is `60 Hz`.

Number of pole pairs

Number of pole pairs of the connected machine. The default value is `1`.

Peak torque to rated torque ratio

Ratio that the block multiplies by the base torque to provide the upper limit for the torque that accelerates the inertia. The default value is `2`.

Inertia Tab

Specify inertia parameterization by

Inertia specification. The choices are between `Actual Inertia, J` and `Inertia constant, H`. The default value is `Inertia constant, H`.

Inertia constant, H

Inertia constant. This parameter is visible only if you set **Specify inertia parameterization by** to `Inertia constant, H`. The default value is `3.525 s*W/VA`.

Actual inertia, J

Inertia. This parameter is visible only if you set **Specify inertia parameterization by** to `Actual Inertia, J`. The default value is `27548` kg·m^2.

Specify damper parameterization by

Damping specification. The choices are `Per-unit damping coefficient, pu_D` and `SI damping coefficient, D`. The default value is `Per-unit damping coefficient, pu_D`.

Per-unit damping coefficient

Damping coefficient in per-unit. This parameter is visible only if you set **Specify damper parameterization by** to `Per-unit damping coefficient, pu_D`. The default value is `0.01`.

SI damping coefficient

Damping coefficient in SI units. This parameter is visible only if you set **Specify damper parameterization by** to `SI damping coefficient, D`. The default value is `39.0509` Nm/(rad/s).

Initial Conditions Tab

Specify initialization by

Frequency initialization. The choices are `Initial electrical frequency` and `Initial mechanical frequency`. The default value is `Initial electrical frequency`.

Initial electrical frequency

Initial electrical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial electrical frequency` . The default value is `60 Hz`.

Initial mechanical frequency

Initial mechanical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial mechanical frequency`. The default value is `60 Hz`.

Ports

The block has the following ports:

`pu`

Physical signal input port associated with mechanical power, per-unit

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

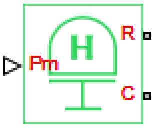
See Also

Machine Inertia | Machine Mechanical Power (SI)

Introduced in R2014b

Machine Mechanical Power (SI)

Machine mechanical power defined in SI units



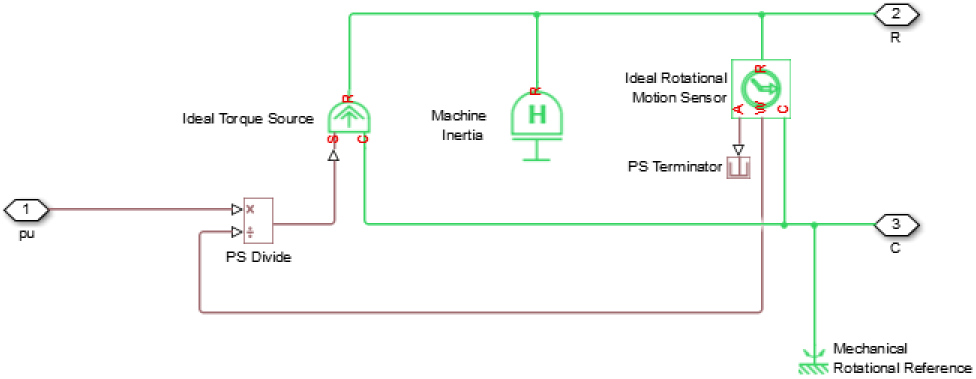
Library

Machines

Description

The Machine Mechanical Power (SI) 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 Pm defines the mechanical power in SI units that is input to the machine. In motor mode, it defines the mechanical power output from the machine. The Pm input must always be positive.

The figure shows an equivalent configuration to the Machine Mechanical Power (SI) block using Simscape mechanical rotational components.



Electrical Defining Equations

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^2.
- 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 pole pairs in the connected machine.

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:

- ω_{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).

Parameters

- "Main Tab" on page 1-327
- "Inertia Tab" on page 1-328
- "Initial Conditions Tab" on page 1-329

Main Tab

Input power sign convention

Machine type specification. The choices are `Generator` and `Motor`. The default value is `Generator`.

Rated apparent power

Rated apparent power of the connected machine. The default value is `555e6 V*A`.

Rated electrical frequency

Nominal electrical frequency corresponding to the rated apparent power of the connected machine. The default value is `60 Hz`.

Number of pole pairs

Number of pole pairs of the connected machine. The default value is `1`.

Peak torque to rated torque ratio

Ratio that the block multiplies by the base torque to provide the upper limit for the torque that accelerates the inertia. The default value is `2`.

Inertia Tab

Specify inertia parameterization by

Inertia specification. The choices are `Actual Inertia, J` and `Inertia constant, H`. The default value is `Inertia constant, H`.

Inertia constant, H

Inertia constant. This parameter is visible only if you set **Specify inertia parameterization by** to `Inertia constant, H`. The default value is `3.525 s*W/VA`.

Actual inertia, J

Inertia. This parameter is visible only if you set **Specify inertia parameterization by** to `Actual Inertia, J`. The default value is `27548` kg·m^2.

Specify damper parameterization by

Damping specification. The choices are `Per-unit damping coefficient, pu_D` and `SI damping coefficient, D`. The default value is `Per-unit damping coefficient, pu_D`.

Per-unit damping coefficient

Damping coefficient in per-unit. This parameter is visible only if you set **Specify damper parameterization by** to `Per-unit damping coefficient, pu_D`. The default value is `0.01`.

SI damping coefficient

Damping coefficient in SI units. This parameter is visible only if you set **Specify damper parameterization by** to `SI damping coefficient, D`. The default value is `39.0509` Nm/(rad/s).

Initial Conditions Tab

Specify initialization by

Frequency initialization. The choices are `Initial electrical frequency` and `Initial mechanical frequency`. The default value is `Initial electrical frequency`.

Initial electrical frequency

Initial electrical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial electrical frequency`. The default value is `60 Hz`.

Initial mechanical frequency

Initial mechanical frequency. This parameter is visible only if you set **Specify initialization by** to `Initial mechanical frequency`. The default value is `60 Hz`.

Ports

The block has the following ports:

`Pm`

Physical signal input port associated with mechanical power, W

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

See Also

Machine Inertia | Machine Mechanical Power (pu)

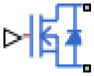
Topics

Three-Phase Asynchronous Machine Starting

Introduced in R2014b

MOSFET

N-channel MOSFET



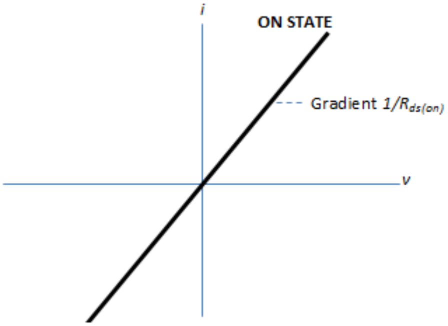
Library

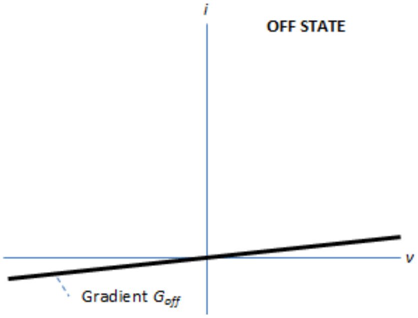
Semiconductors / Fundamental Components

Description

The MOSFET block models an n-channel metal-oxide-semiconductor field-effect transistor (MOSFET).

The I-V 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.





In the on state, the drain-source path behaves like a linear resistor with resistance, R_{ds_on}.

In the off state, the drain-source path behaves like a linear resistor with low off-state conductance, G_{off}.

The defining Simscape equations for the block are:

```
if G > Vth
  v == i*Rds_on;
else
  v == i/Goff;
end
```

where:

- *G* is the gate-source voltage.
- *Vth* is the threshold voltage.
- *v* is the drain-source voltage.
- *i* is the drain-source current.
- *Rds_on* is the on-state resistance.
- *Goff* is the off-state conductance.

Using the Integral Diode tab of the block dialog box, you can include an integral source-drain 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. The block dialog box shows parameters relating to the Diode block.

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 Commutation Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Modeling Variants

The block provides four modeling variants. To select the desired variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

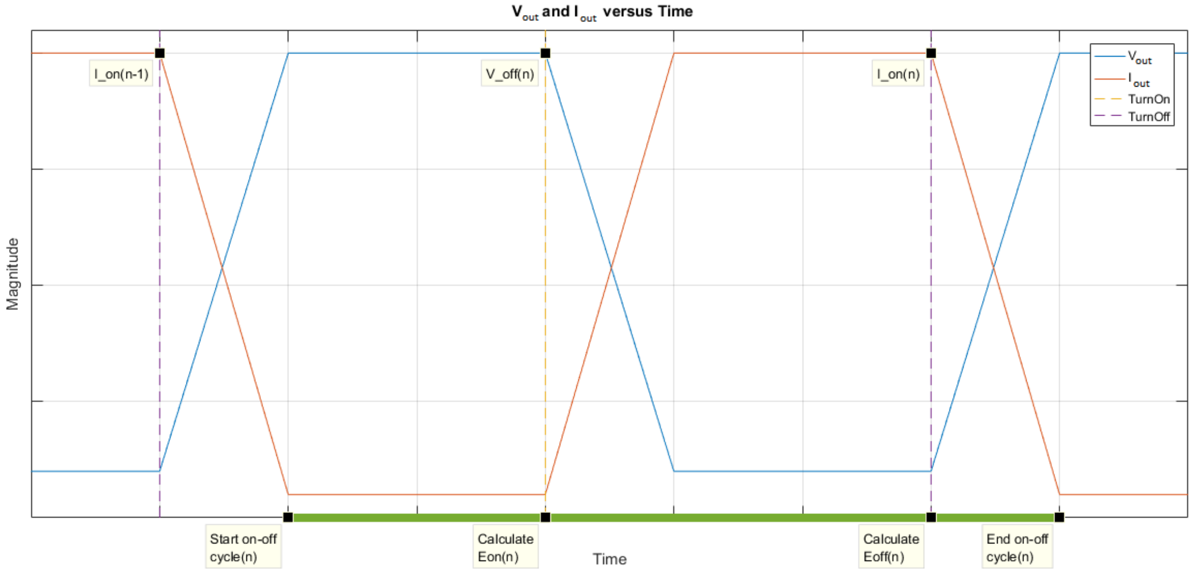
- **PS Control Port** — Contains a physical signal port that is associated with the gate terminal. This variant is the default.
- **Electrical Control Port** — Contains an electrical conserving port that is associated with the gate terminal.
- **PS Control Port | Thermal Port** — Contains a thermal port and a physical signal port that is associated with the gate terminal.
- **Electrical Control Port | Thermal Port** — Contains a thermal port and an electrical conserving port that is associated with the gate terminal.

The variants of this block without the thermal port do not simulate heat generation in the device.

The variants with the thermal port allow you to model the heat that switching events and conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block. The thermal port is hidden by default. To enable the thermal port, select a thermal block variant.

Thermal Loss Equations

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.



When the semiconductor turns on during the nth switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{on(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n-1)}),$$

where:

- $E_{on(n)}$ is the switch-on loss at the nth switch-on event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the nth switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n-1)}$ is the on-state output current, I_{out}, just before the device switches off during the cycle that precedes the nth switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{on\_data}, T, I_{on(n-1)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{on_data} is the **Switch-on loss, Eon=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns on during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{on(n)} = \left(\frac{V_{off(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{on\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{on_scalar} is the **Switch-on loss** parameter value.

When the semiconductor turns off during the *n*th switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{off(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n)}),$$

where:

- $E_{off(n)}$ is the switch-off loss at the *n*th switch-off event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the *n*th switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n)}$ is the on-state output current, I_{out}, just before the device switches off during the *n*th switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{off\_data}, T, I_{on(n)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{off_data} is the **Switch-off loss, Eoff=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns off during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{off(n)} = \left(\frac{V_{off(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{off\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{off_scalar} is the **Switch-off loss** parameter value.

If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int fcn(T, I_{out})\,dt,$$

where:

- $E_{conduction}$ is the heat loss due to electrical conduction.
- T is the device temperature.
- I_{out} is the device output current.

The function *fcn* is a 2-D lookup table:

$$Q_{conduction} = tablelookup(T_{j\_data}, I_{out\_data}, I_{out\_data\_repmat} .* V_{on\_data}, T, I_{out}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.

- I_{out_data} is the **Output current vector, Iout** parameter value.
- $I_{out\_data\_repmat}$ is a matrix that contains length, T_{j_data}, copies of I_{out_data}.
- V_{on_data} is the **On-state voltage, Von=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int \left(I_{out} * V_{on\_scalar} \right) dt,$$

where V_{on_scalar} is the **On-state voltage** parameter value.

The block uses the **Energy dissipation time constant** parameter to filter the amount of heat flow that the block outputs. The filtering allows the block to:

- Avoid discrete increments for the heat flow output
- Handle a variable switching frequency

The filtered heat flow is

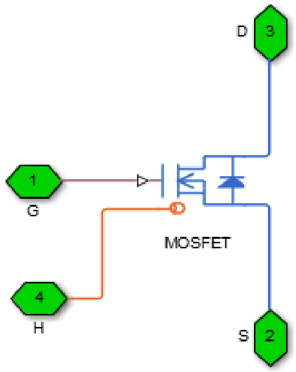
$$Q = \frac{1}{\tau} \left(\sum_{i=1}^{n} E_{on(i)} + \sum_{i=1}^{n} E_{off\,(i)} + E_{conduction} - \int Q\,dt \right),$$

where:

- Q is the heat flow from the component.
- τ is the **Energy dissipation time constant** parameter value.
- n is the number of switching cycles.
- $E_{on(i)}$ is the switch-on loss at the ith switch-on event.
- $E_{off(i)}$ is the switch-off loss at the ith switch-off event.
- $E_{conduction}$ is the heat loss due to electrical conduction.
- $\int Qdt$ is the total heat previously dissipated from the component.

Ports

The figure shows the block port names.



G

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

S

Electrical conserving port associated with the source terminal.

D

Electrical conserving port associated with the drain terminal.

H

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

- “Main Tab” on page 1-337
- “Integral Diode Tab” on page 1-338
- “Thermal Model Tab” on page 1-341

Main Tab

On-state resistance, R_DS(on)

Drain-source resistance when the device is on. The default value is `0.01 Ohm`.

Off-state conductance

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**. The default value is `1e-6 1/Ohm`.

Threshold voltage, Vth

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value. The default value is `2 V`.

Integral Diode Tab

Integral protection diode

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`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 μAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Thermal Model Tab

The **Thermal Model** tab is enabled only when you select a block variant that includes a thermal port.

Thermal loss dependent on

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- `Voltage and current` — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data.
- `Voltage, current, and temperature` — Use vectors to specify the output current, switch-on loss, switch-off loss, on-state voltage, and temperature data. This is the default parameterization method.

Off-state voltage for losses data

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. The default value is `300` `V`.

Energy dissipation time constant

Time constant used to average the switch-on losses, switch-off losses, and conduction losses. This value is equal to the period of the minimum switching frequency. The default value is `1e-4` `s`.

Additional Parameters for Parameterizing by Voltage, Current, and Temperature

Temperature vector, Tj

Temperature values at which the switch-on loss, switch-off loss, and on-state voltage are specified. Specify this parameter using a vector quantity. The default value is `[ 298.15 398.15 ]` `K`.

Output current vector, Iout

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 vector quantity. The default value is `[ 0 10 50 100 200 400 600 ]` `A`.

Switch-on loss, Eon=fcn(Tj,Iout)

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. The default value is `[ 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`.

Switch-off loss, Eoff=fcn(Tj,Iout)

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. The default value is `[ 0 2.1e-4 0.00107 0.00214 0.00429 0.009859999999999999 0.01714; 0 4.3e-4 0.00197 0.00386 0.00771 0.01543 0.02271 ] J`.

On-state voltage, Von=fcn(Tj,Iout)

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. The default value is `[ 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`.

Additional Parameters for Parameterizing by Voltage and Current

Output current, Iout

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. The default value is `600 A`.

Switch-on loss

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. The default value is `0.02286 J`.

Switch-off loss

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. The default value is `0.01714 J`.

On-state voltage

Voltage drop across the block 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 scalar quantity. The default value is `2.7 V`.

See Also

Commutation Diode | Diode | GTO | IGBT | Ideal Semiconductor Switch | Thyristor

Topics

“Quantifying IGBT Thermal Losses”
“Simulate Thermal Losses in Semiconductors”
“Switch Between Physical Signal and Electrical Ports”

Introduced in R2013b

Neutral Port

Connect phases of three-phase system to electrical conserving port

Library

Connections

Description

The Neutral Port block connects the phases of a three-phase system to an electrical conserving port. You can connect the electrical port to electrical components from the Simscape and Simscape Electronics™ libraries.

Note If you do not need to connect the neutral port to other blocks, use a Floating Neutral block instead. If you want to ground the neutral port, use a Grounded Neutral block.

Ports

The block has the following ports:

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral point

See Also

Floating Neutral | Neutral Port

Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Nonlinear Inductor

Model inductor with nonideal core

Library

Passive Devices / Fundamental Components

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 1-346
- “Single Saturation Point” on page 1-347
- “Magnetic Flux Versus Current Characteristic” on page 1-348
- “Magnetic Flux Density Versus Magnetic Field Strength Characteristic” on page 1-348
- “Magnetic Flux Density Versus Magnetic Field Strength Characteristic with Hysteresis” on page 1-349

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.
- Φ 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.
- Φ is the magnetic flux.
- Φ_{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:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = f\left(i_L\right)$$

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.
- Φ 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 is 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.
- Φ 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 is 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:

$$i = i_L + vG_p$$

$$v = N_w \frac{d\Phi}{dt}$$

$$\Phi = B \cdot A_e$$

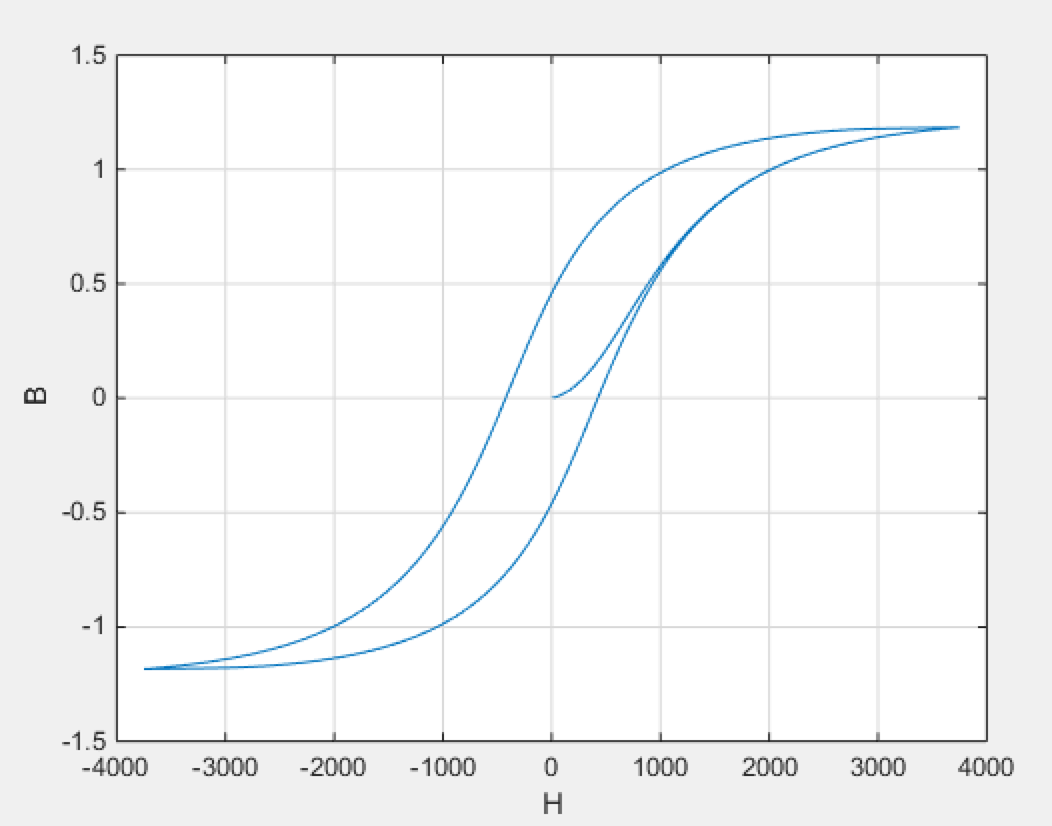
$$B = \mu_0 (H + M)$$

$$H = \frac{N_w}{l_e} i_L$$

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.
- Φ is the magnetic flux.
- B is the magnetic flux density.
- μ_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], [2] equations are used to determine M at any given time. The figure 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 one that is 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}:

$$M_{an} = M_s \left(\coth\left(\frac{H_{eff}}{\alpha} \right) - \frac{\alpha}{H_{eff}} \right)$$

This function defines a saturation curve with limiting values $\pm M_s$ and point of saturation determined by the value of α, the anhysteretic shape factor. It can be 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. These values are used to determine values for α 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 $1 / \left(K\delta - \alpha\left(M_{an} - M_{irr}\right)\right)$. The tracking error acts to create the hysteresis at the points where δ changes sign. The main parameter that shapes the irreversible characteristic is K, which is called the bulk coupling coefficient. The parameter α 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 α 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 α 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 $H_{\mathrm{eff}} = 0$) 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 a 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.

Sometimes iteration on these four steps is required to get a good match against a predefined B-H curve.

Parameters

- “Main Tab” on page 1-353
- “Initial Conditions Tab” on page 1-357

Main Tab

Parameterized by

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

The total number of turns of wire wound around the inductor core. The default value is `10`.

Unsaturated inductance

The value of inductance used when the inductor is operating in its linear region. This parameter is visible only when you select `Single inductance (linear)` or `Single saturation point` for the **Parameterized by** parameter. The default value is `2e-4 H`.

Saturated inductance

The value of inductance used when the inductor is operating beyond its saturation point. This parameter is visible only when you select `Single saturation point` for the **Parameterized by** parameter. The default value is `1e-4 H`.

Saturation magnetic flux

The value of magnetic flux at which the inductor saturates. This parameter is visible only when you select `Single saturation point` for the **Parameterized by** parameter. The default value is `1.3e-5` `Wb`.

Current, i

The current data used to populate the magnetic flux versus current lookup table. This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Parameterized by** parameter. The default value is `[ 0 0.64 1.28 1.92 2.56 3.20 ]` `A`.

Magnetic flux vector, phi

The magnetic flux data used to populate the magnetic flux versus current lookup table. This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Parameterized by** parameter. The default value is `[0 1.29 2.00 2.27 2.36 2.39 ].*1e-5` `Wb`.

Magnetic field strength vector, H

The magnetic field strength data used to populate the magnetic flux density versus magnetic field strength lookup table. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Parameterized by** parameter. The default value is `[ 0 200 400 600 800 1000 ]` `A/m`.

Magnetic flux density vector, B

The magnetic flux density data used to populate the magnetic flux density versus magnetic field strength lookup table. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Parameterized by** parameter. The default value is `[ 0 0.81 1.25 1.42 1.48 1.49 ]` `T`.

Effective length

The effective core length, that is, the average distance of the magnetic path. 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. The default value is `0.032` `m`.

Effective cross-sectional area

The effective core cross-sectional area, that is, the average area of the magnetic path. 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. The default value is `1.6e-5 m^2`.

Anhysteretic B-H gradient when H is zero

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. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `0.005 m*T/A`.

Flux density point on anhysteretic B-H curve

Specify a point on the anhysteretic 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. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `1.49 T`.

Corresponding field strength

The corresponding field strength for the point that you define by the **Flux density point on anhysteretic B-H curve** parameter. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `1000 A/m`.

Coefficient for reversible magnetization, c

The proportion of the magnetization that is reversible. The value should be greater than zero and less than one. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `0.1`.

Bulk coupling coefficient, K

The Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `200 A/m`.

Inter-domain coupling factor, alpha

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. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter. The default value is `1e-4`.

Parasitic parallel conductance

Use this parameter to represent small parasitic effects. A small parallel conductance may be required for the simulation of some circuit topologies. The default value is `1e-9 1/Ohm`.

Interpolation option

The lookup table interpolation option. 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. Select one of the following interpolation methods:

- `Linear` — Uses a linear interpolation function.
- `Cubic` — Uses the Piecewise Cubic Hermite Interpolation Polynomial (PCHIP).

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Initial Conditions Tab

Specify initial state by

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.

This parameter is not visible when you select `Magnetic flux density versus magnetic field strength characteristic with hysteresis` for the **Parameterized by** parameter on the **Main** tab.

Initial current

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. This parameter is visible only when you select `Current` for the **Specify initial state by** parameter. The default value is `0 A`.

Initial magnetic flux

The value of magnetic flux at time zero. This parameter is visible only when you select `Magnetic flux` for the **Specify initial state by** parameter. The default is `0 Wb`.

Initial magnetic flux density

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 **Parameterized by** parameter on the **Main** tab. The default is `0 T`.

Initial field strength

The value of magnetic field strength at time zero. 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. The default is `0 A/m`.

Ports

The block has the following ports:

`+`

Electrical conserving port associated with the positive terminal of the inductor winding

`-`

Electrical conserving port associated with the negative terminal of the inductor winding

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.

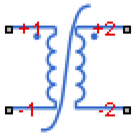
See Also

Nonlinear Transformer

Introduced in R2015b

Nonlinear Transformer

Model transformer with nonideal core



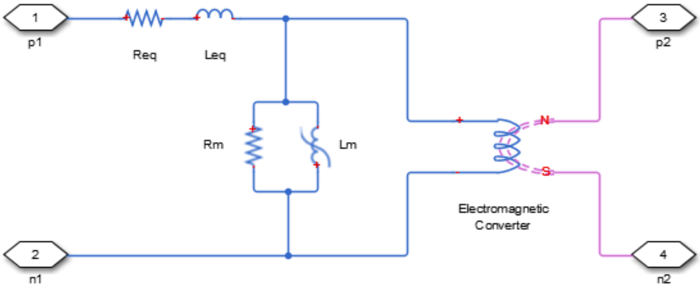
Library

Passive Devices / Fundamental Components

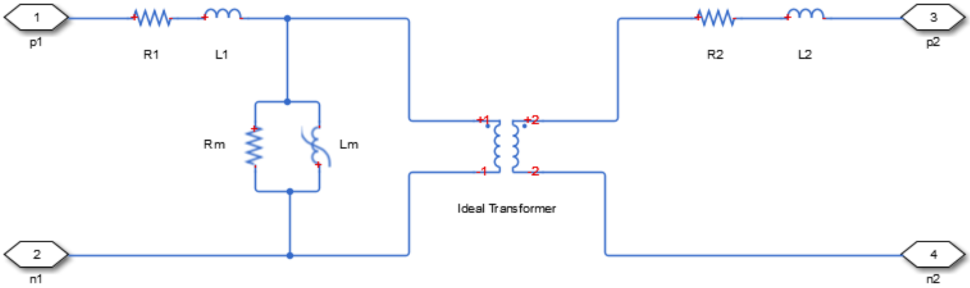
Description

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:

- *Req* is the combined leakage resistance.
- *Leq* is the combined leakage inductance.
- *R1* is the primary leakage resistance.
- *L1* is the primary leakage inductance.
- *R2* is the secondary leakage resistance.
- *L2* is the 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.

Parameters

- "Main Tab" on page 1-362
- "Magnetization Tab" on page 1-363
- "Initial Conditions Tab" on page 1-367
- "Parasitics Tab" on page 1-368

Main Tab

Primary number of turns

The number of turns of wire on the primary winding of the transformer. The default value is `100`.

Secondary number of turns

The number of turns of wire on the secondary winding of the transformer. The default value is `200`.

Winding parameterized by

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 leakage resistance

The lumped equivalent resistance *Req*, which represents the combined power loss of the primary and secondary windings. This parameter is visible only when you select `Combined primary and secondary values` for the **Winding parameterized by** parameter. The default value is `0.01 Ohm`.

Combined leakage inductance

The lumped equivalent inductance *Leq*, which represents the combined magnetic flux loss of the primary and secondary windings. This parameter is visible only when you select `Combined primary and secondary values` for the **Winding parameterized by** parameter. The default value is `1e-4 H`.

Primary leakage resistance

The resistance *R1*, which represents the power loss of the primary winding. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter. The default value is `0.01 Ohm`.

Primary leakage inductance

The inductance *L1*, which represents the magnetic flux loss of the primary winding. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter. The default value is `1e-4 H`.

Secondary leakage resistance

The resistance *R2*, which represents the power loss of the secondary winding. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter. The default value is `0.01 Ohm`.

Secondary leakage inductance

The inductance *L2*, which represents the magnetic flux loss of the secondary winding. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter. The default value is `1e-4 H`.

Magnetization Tab

Magnetization resistance

The resistance *Rm*, which represents the magnetic losses in the transformer core. The default value is `100 Ohm`.

Magnetization inductance parameterized by

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 magnetic 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 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.

Unsaturated inductance

The value of inductance used when the magnetization inductance *Lm* is operating in its linear region. This parameter is visible only when you select `Single inductance (linear)` or `Single saturation point` for the **Magnetization inductance parameterized by** parameter. The default value is `0.04 H`.

Saturated inductance

The value of inductance used when the magnetization inductance *Lm* is operating beyond its saturation point. This parameter is visible only when you select `Single saturation point` for the **Magnetization inductance parameterized by** parameter. The default value is `0.01 H`.

Saturation magnetic flux

The value of magnetic flux at which the magnetization inductance *Lm* saturates. This parameter is visible only when you select `Single saturation point` for the **Magnetization inductance parameterized by** parameter. The default value is `1.6e-4 Wb`.

Current, i

The current data used to populate the magnetic flux versus current lookup table. This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `[ 0 0.4 0.8 1.2 1.6 2.0 ] A`.

Magnetic flux vector, phi

The magnetic flux data used to populate the magnetic flux versus current lookup table. This parameter is visible only when you select `Magnetic flux versus current characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `[0 0 0.161 0.25 0.284 0.295 0.299 ].*1e-3` `Wb`.

Magnetic field strength vector, H

The magnetic field strength data used to populate the magnetic flux density versus magnetic field strength lookup table. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `[ 0 200 400 600 800 1000 ]` `A/m`.

Magnetic flux density vector, B

The magnetic flux density data used to populate the magnetic flux density versus magnetic field strength lookup table. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `[ 0 0.81 1.25 1.42 1.48 1.49 ]` `T`.

Effective length

The effective core length, that is, the average distance of the magnetic path around the transformer core. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `0.2` `m`.

Effective cross-sectional area

The effective core cross-sectional area, that is, the average area of the magnetic path around the transformer core. This parameter is visible only when you select `Magnetic flux density versus magnetic field strength characteristic` for the **Magnetization inductance parameterized by** parameter. The default value is `2e-4` `m^2`.

Anhysteretic B-H gradient when H is zero

The gradient of the anhysteretic (no hysteresis) B-H curve around zero field strength. Set it to the average gradient of the positive-going and negative-going hysteresis curves. 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. The default value is `0.005 m*T/A`.

Flux density point on anhysteretic B-H curve

Specify a point on the anhysteretic 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. 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. The default value is `1.49 T`.

Corresponding field strength

The corresponding field strength for the point that you define by the **Flux density point on anhysteretic B-H curve** parameter. 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. The default value is `1000 A/m`.

Coefficient for reversible magnetization, c

The proportion of the magnetization that is reversible. The value should be greater than zero and less than one. 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. The default value is `0.1`.

Bulk coupling coefficient, K

The Jiles-Atherton parameter that primarily controls the field strength magnitude at which the B-H curve crosses the zero flux density line. 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. The default value is `200 A/m`.

Inter-domain coupling factor, alpha

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. 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. The default value is `1e-4`.

Interpolation option

The lookup table interpolation option. This parameter is visible only when you select `Magnetic flux versus current characteristic` or `Magnetic flux density versus magnetic field strength characteristic` for the **Magnetization inductance parameterized by** parameter. Select one of the following interpolation methods:

- `Linear` — Uses a linear interpolation function.
- `Cubic` — Uses the Piecewise Cubic Hermite Interpolation Polynomial (PCHIP).

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Initial Conditions Tab

Combined leakage inductance initial current

The value of current through the combined leakage inductance *Leq* at time zero. This parameter is visible only when you select `Combined primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `0 A`.

Primary leakage inductance initial current

The value of current through the primary leakage inductance *L1* at time zero. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `0 A`.

Secondary leakage inductance initial current

The value of current through the secondary leakage inductance *L2* at time zero. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `0 A`.

Specify magnetization inductance initial state by

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.

This parameter is not visible 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

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*. This parameter is visible only when you select `Current` for the **Specify magnetization inductance initial state by** parameter. The default value is `0 A`.

Magnetization inductance initial magnetic flux

The value of the magnetic flux in the magnetization inductance *Lm* at time zero. This parameter is visible only when you select `Magnetic flux` for the **Specify magnetization inductance initial state by** parameter. The default is `0 Wb`.

Magnetization inductance initial magnetic flux density

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. The default is `0 T`.

Magnetization inductance initial field strength

The value of magnetic field strength 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. The default is `0 A/m`.

Parasitics Tab

Combined leakage inductance parasitic parallel conductance

Use this parameter to represent small parasitic effects in parallel to the combined leakage inductance *Leq*. A small parallel conductance may be required for the simulation of some circuit topologies. This parameter is visible only when you select `Combined primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `1e-9 1/Ohm`.

Primary leakage inductance parasitic parallel conductance

Use this parameter to represent 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. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `1e-9` `1/Ohm`.

Secondary leakage inductance parasitic parallel conductance

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. This parameter is visible only when you select `Separate primary and secondary values` for the **Winding parameterized by** parameter on the **Main** tab. The default value is `1e-9` `1/Ohm`.

Ports

The block has the following ports:

`+1`

Electrical conserving port associated with the positive terminal of the primary winding

`+2`

Electrical conserving port associated with the positive terminal of the secondary winding

`-1`

Electrical conserving port associated with the negative terminal of the primary winding

`-2`

Electrical conserving port associated with the negative terminal of the secondary winding

See Also

Nonlinear Inductor

Introduced in R2015b

Open Circuit

Three-phase connection that draws no current

Library

Connections

Description

The Open Circuit block models a three-phase connection that draws no current on any of the three phases. In Simscape, physical network block diagrams do not allow unconnected conserving ports. Therefore, use the Open Circuit block to terminate three-phase electrical ports on other blocks that you want to leave open-circuit.

Ports

The block has the following ports:

~

Expandable three-phase port

See Also

Topics

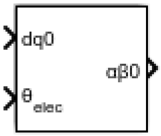
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

Park to Clarke Angle Transform

Implement *dq0* to *αβ0* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

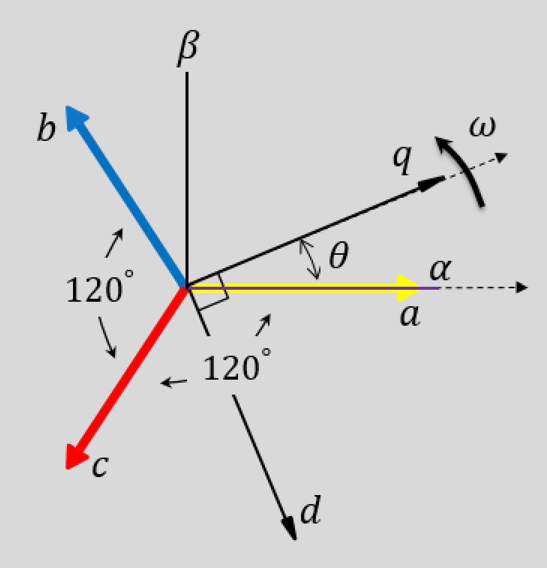


Description

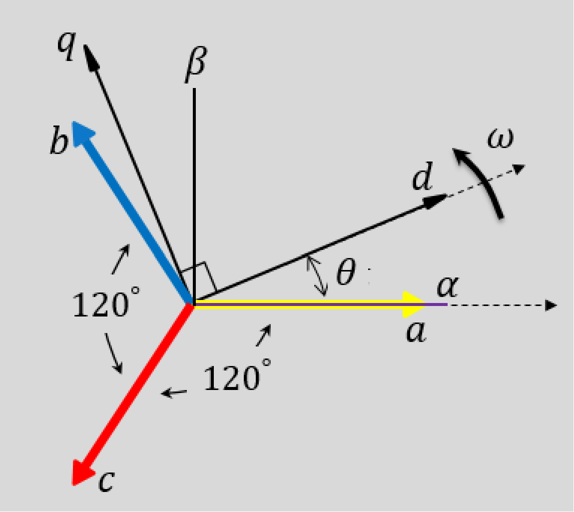
The Park to Clarke Angle Transform block converts the direct, quadrature, and zero components in a rotating reference frame to alpha, beta, and zero components in a stationary reference frame. For balanced systems, the zero components are equal to zero.

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 *αβ0* 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.



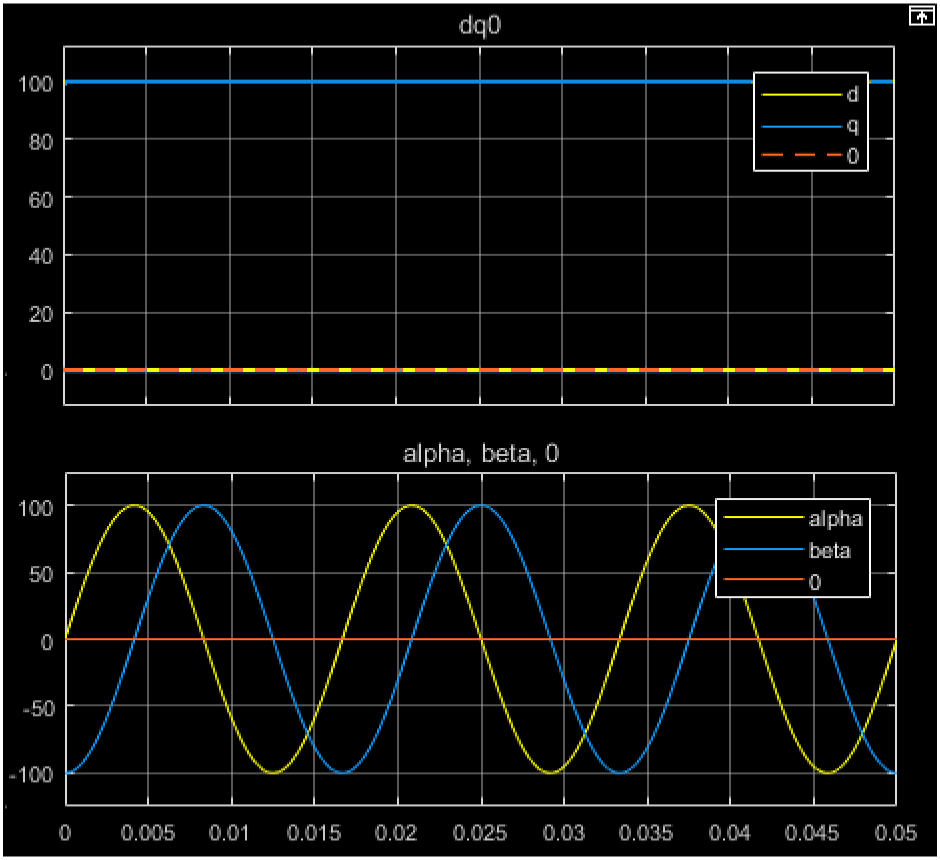
In both cases, the angle $\theta = \omega t$, where

- θ 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.

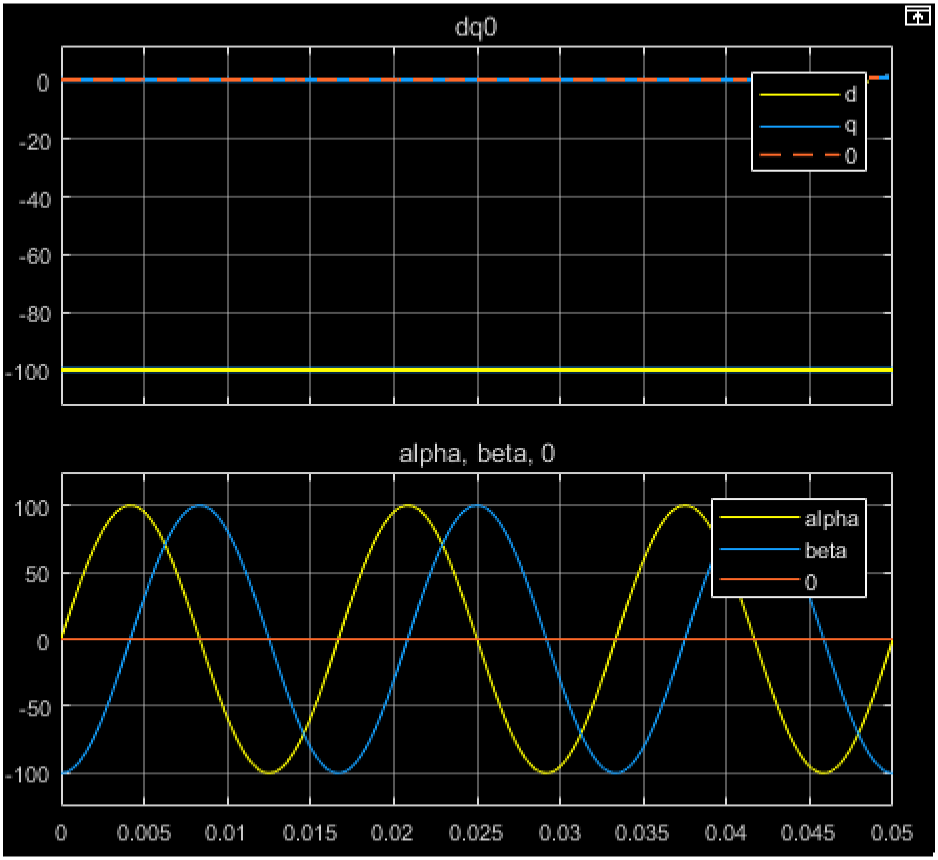
- ω is the rotational speed of the d-q reference frame.
- t is the time, in s, from the initial alignment.

The figures show the time-response of the individual components of equivalent balanced *dq0* and *αβ0* 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.
- α and β 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`

θ_{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

`αβ0` — α-β axis and zero components

vector

Alpha-axis component,α, beta-axis component, β, 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.

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.

See Also

Blocks

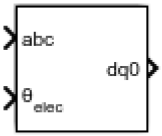
Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park Transform

Introduced in R2017b

Park Transform

Implement *abc* to *dq0* transform

Library: Simscape / Power Systems / Simscape Components / Control / Mathematical Transforms

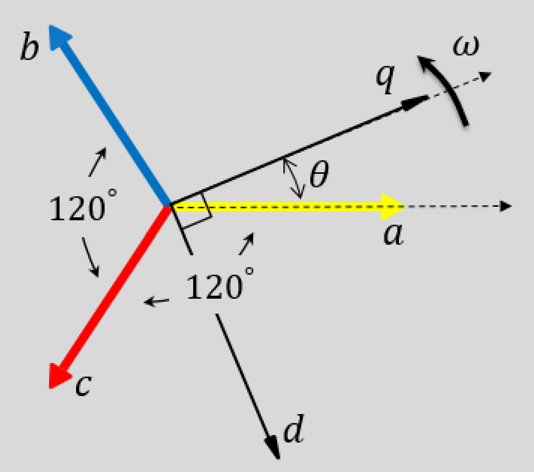


Description

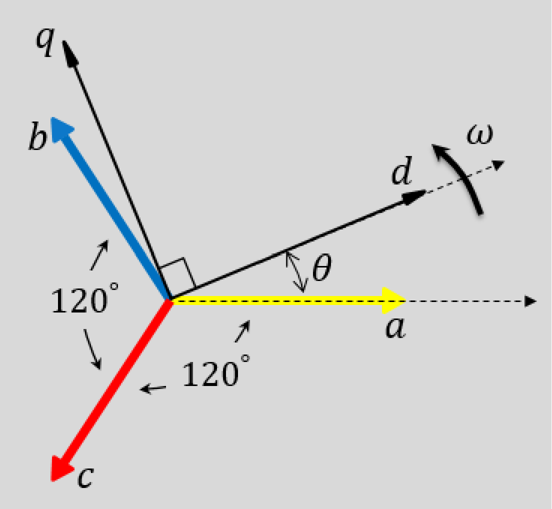
The Park Transform block converts the time-domain components of a three-phase system in an *abc* reference frame to direct, quadrature, and zero components in a rotating reference frame. The block can preserve the active and reactive powers with the powers of the system in the *abc* reference frame by 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.

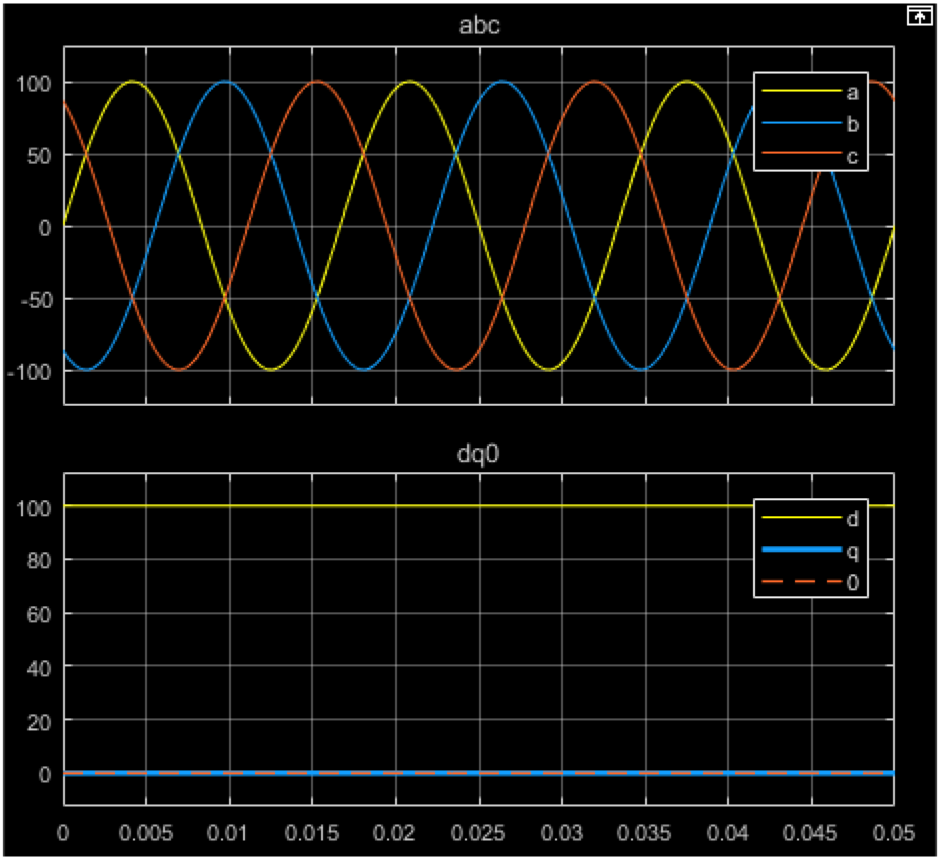


In both cases, the angle $\theta = \omega t$, where:

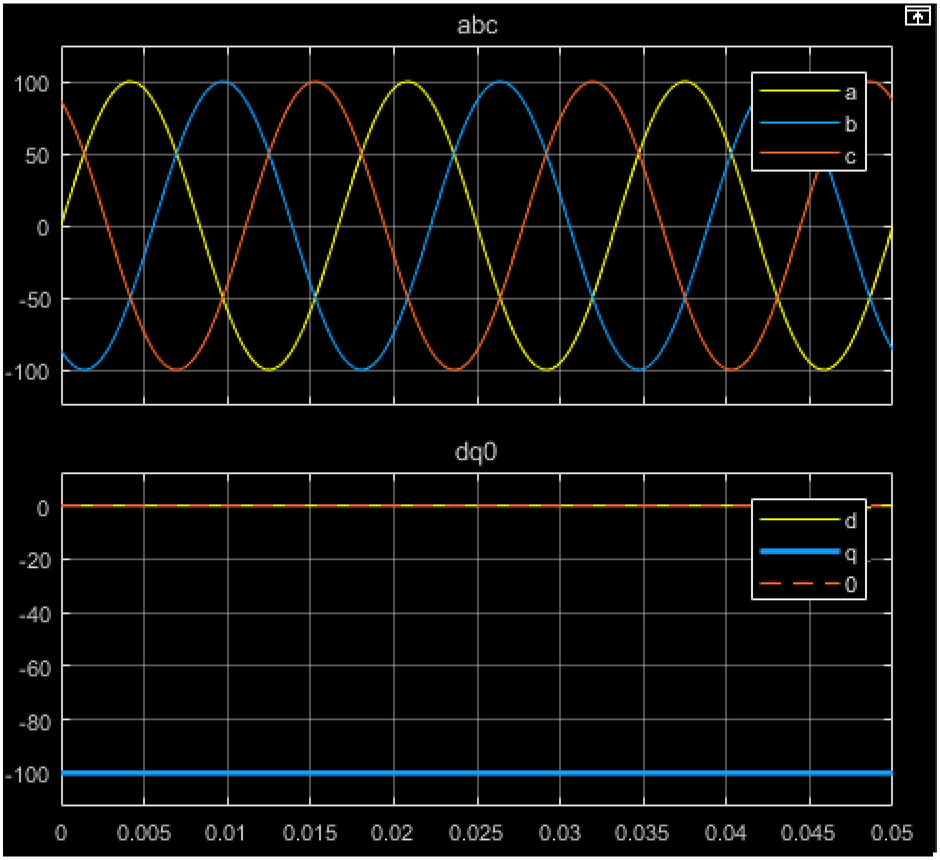
- θ is the angle between the a and q axes for the q-axis alignment or the angle between the a and d axes for the d-axis alignment.
- ω is the rotational speed of the d-q reference frame.
- t is the time, in s, from the initial alignment.

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:

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

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`

θ_{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.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator HESM Torque Control HESM Velocity Control IPMSG Voltage Stabilization IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control SM Torque Control SM Velocity Control Switched Reluctance Machine Speed Control Synchronous Reluctance Machine Velocity Control Three-Phase Asynchronous Drive with Sensor Control Three-Phase Asynchronous Drive with Sensorless Control Three-Phase PMSM Drive

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.

See Also

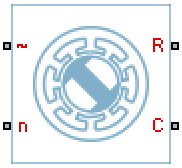
Blocks

Clarke Transform | Clarke to Park Angle Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Permanent Magnet Synchronous Motor

Permanent magnet synchronous motor with sinusoidal flux distribution

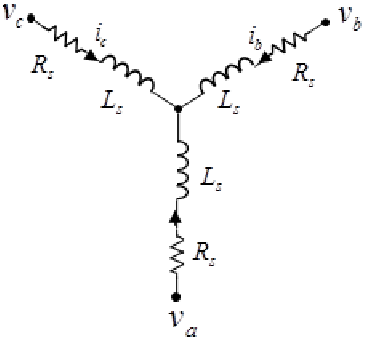


Library

Machines / Permanent Magnet Rotor

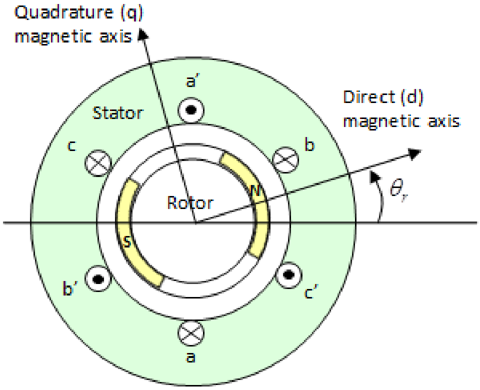
Description

The Permanent Magnet Synchronous Motor 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.



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 with rotor angle.

For the axes convention in the preceding figure, the a-phase and permanent magnet fluxes are aligned when rotor mechanical angle, θ_r, is zero. The block supports a second rotor axis definition in which rotor mechanical angle is defined as the angle between the a-phase magnetic axis and the rotor q-axis.

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 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:

- ψ_a, ψ_b, and ψ_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.
- ψ_{am}, ψ_{bm}, and ψ_{cm} are the permanent magnet fluxes linking the stator windings.

The inductances in the stator windings are functions of rotor electrical angle, defined by:

$$\theta_e = N\theta_r,$$

$$L_{aa} = L_s + L_m\cos(2\theta_e),$$

$$L_{bb} = L_s + L_m\cos(2(\theta_e - 2\pi/3)),$$

$$L_{cc} = L_s + L_m\cos(2(\theta_e + 2\pi/3)),$$

$$L_{ab} = L_{ba} = -M_s - L_m\cos(2(\theta_e + \pi/6)),$$

$$L_{bc} = L_{cb} = -M_s - L_m\cos(2(\theta_e + \pi/6 - 2\pi/3)),$$

and

$$L_{ca} = L_{ac} = -M_s - L_m\cos(2(\theta_e + \pi/6 + 2\pi/3)),$$

where:

- θ_r is the rotor mechanical angle.
- θ_e is the rotor electrical angle.
- 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.

The permanent magnet flux linking winding a is a maximum when $\theta_e = 0°$ and zero when $\theta_e = 90°$. 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 ψ_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:

$$P = 2/3 \begin{bmatrix} \cos\theta_e & \cos(\theta_e - 2\pi/3) & \cos(\theta_e + 2\pi/3) \\ -\sin\theta_e & -\sin(\theta_e - 2\pi/3) & -\sin(\theta_e + 2\pi/3) \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

where θ_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:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

and

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

The inverse of Park's transformation is defined by:

$$P^{-1} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e & 1 \\ \cos(\theta_e - 2\pi/3) & -\sin(\theta_e - 2\pi/3) & 1 \\ \cos(\theta_e + 2\pi/3) & -\sin(\theta_e + 2\pi/3) & 1 \end{bmatrix}.$$

Applying Park’s transformation to the first two electrical equations produces the following equations that define the block behavior:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N\omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega(i_d L_d + \psi_m),$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt},$$

and

$$T = \frac{3}{2} N\left(i_q\left(i_d L_d + \psi_m\right) - i_d i_q L_q\right),$$

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.
- ω 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** option.

The back EMF constant is defined as the peak voltage induced by the permanent magnet in each of the phases per unit rotational speed. It is related to peak permanent magnet flux linkage by:

$$k_e = N\psi_m.$$

From this definition, it follows that the back EMF e_{ph} for one phase is given by:

$$e_{ph} = k_e \omega.$$

The torque constant is defined as the peak torque induced by each of the phases per unit current. 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 when the currents in all three phases are balanced, it follows that the combined torque T is given by:

$$T = \frac{3}{2} k_t i_q = \frac{3}{2} k_t I_{pk},$$

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 could also be defined as:

$$k_t = \frac{2}{3}\left(\frac{T}{I_{pk}}\right),$$

where T is the measured total torque when testing with a balanced three-phase current with peak line voltage I_{pk}. Writing in terms of RMS line voltage:

$$k_t = \sqrt{\frac{2}{3}}\left(\frac{T}{i_{line,rms}}\right).$$

Parameters

Main

Number of pole pairs

Number of permanent magnet pole pairs on the rotor. The default value is `6`.

Permanent magnet flux linkage parameterization

Choose `Specify flux linkage`, the default value, `Specify torque constant`, or `Specify back EMF constant`.

Permanent magnet flux linkage

Peak permanent magnet flux linkage with any of the stator windings. This parameter is visible only if you set **Permanent magnet flux linkage** to `Specify flux linkage`.The default value is `0.03` `Wb`.

Torque constant

Torque constant with any of the stator windings. This parameter is visible only if you set **Permanent magnet flux linkage** to `Specify torque constant`. The default value is `0.18` `N*m/A`.

Back EMF constant

Back EMF constant with any of the stator windings. This parameter is visible only if you set **Permanent magnet flux linkage** to `Specify back EMF constant`. The default value is `0.18` `V*s/rad`.

Stator parameterization

Choose `Specify Ld, Lq, and L0`, the default value, or `Specify Ls, Lm, and Ms`.

Stator d-axis inductance, Ld

Direct-axis inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00019` `H`.

Stator q-axis inductance, Lq

Quadrature-axis inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00025` `H`.

Stator zero-sequence inductance, L0

Zero-sequence inductance. This parameter is visible only if you set **Stator parameterization** to `Specify Ld, Lq, and L0`. The default value is `0.00016` `H`.

Stator self-inductance per phase, Ls

Average self-inductance of each of the three stator windings. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `0.0002` `H`.

Stator inductance fluctuation, Lm

Amplitude of the fluctuation in self-inductance and mutual inductance of the stator windings with rotor angle. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `-0.00002` `H`.

Stator mutual inductance, Ms

Average mutual inductance between the stator windings. This parameter is visible only if you set **Stator parameterization** to `Specify Ls, Lm, and Ms`. The default value is `0.00002 H`.

Stator resistance per phase, Rs

Resistance of each of the stator windings. The default value is `0.013 Ohm`.

Initial Conditions

Initial currents, [i_d i_q i_0]

Initial *d*-axis, *q*-axis, and zero-sequence currents. The default value is `[0, 0, 0] A`.

Rotor angle definition

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 1-386 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.

Initial rotor angle

Initial angle of the rotor. The default value is `0 deg`.

Ports

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral phase

`R`

Mechanical rotational conserving port associated with the motor rotor

`C`

Mechanical rotational conserving port associated with the motor case

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.

See Also

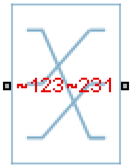
Topics

“Expand and Collapse Three-Phase Ports on a Block”
“Electric Power Assisted Steering”
“Three-Phase PMSM Drive”
“Parameterize a Permanent Magnet Synchronous Motor”

Introduced in R2013b

Phase Permute

Permute phases of three-phase system



Library

Connections

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
a1	c2
b1	a2
c1	b2

Ports

The block has the following ports:

`~123`

Expandable three-phase port

`~231`

Expandable three-phase port

See Also

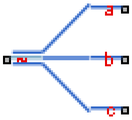
Topics

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

Introduced in R2013b

Phase Splitter

Expand composite three-phase port



Library

Connections

Description

The Phase Splitter block expands a composite three-phase port into its constituent phases.

The expanded output ports are electrical conserving ports. Therefore, you can connect the output ports to electrical components from the Simscape and Simscape Electronics libraries.

Ports

The block has the following ports:

`~`

Composite three-phase port

`a,b,c`

Constituent phases of the expanded three-phase port

See Also

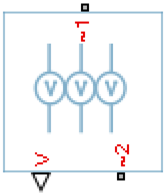
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Phase Voltage Sensor

Measure phase voltages in three-phase system



Library

Sensors

Description

The Phase Voltage Sensor 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.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

`~2`

Expandable three-phase port

`V`

Three-element physical signal vector output port associated with the phase voltages

See Also

Line Voltage Sensor

Topics

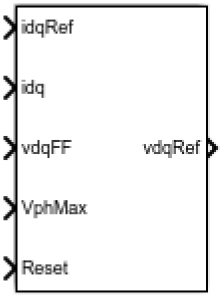
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

PMSM Current Controller

Permanent magnet synchronous machine current controller

Library: Simscape / Power Systems / Simscape Components / Control / PMSM Control



Description

The PMSM Current Controller block implements a discrete-time PI-based 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:

$$v_d^{ref} = \left(K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) \left(i_d^{ref} - i_d \right) + v_{d\_FF},$$

and

$$v_q^{ref} = \left(K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) \left(i_q^{ref} - i_q \right) + v_{q\_FF},$$

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-cancelation block in the feedforward path. The zero cancellation transfer functions in discrete time are:

$$G_{ZC\_id}(z) = \frac{\dfrac{T_s K_{i\_id}}{K_{p\_id}}}{z + \left(\dfrac{T_s - \dfrac{K_{p\_id}}{K_{i\_id}}}{\dfrac{K_{p\_id}}{K_{i\_id}}} \right)},$$

and

$$G_{ZC\_iq}(z) = \frac{\dfrac{T_s K_{i\_iq}}{K_{p\_iq}}}{z + \left(\dfrac{T_s - \dfrac{K_{p\_iq}}{K_{i\_iq}}}{\dfrac{K_{p\_iq}}{K_{i\_iq}}} \right)}.$$

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:

$$v_1^{sat} = \min\left(\max\left(v_1^{unsat}, -V_{ph\_max}\right), V_{ph\_max}\right)$$

and

$$v_2^{sat} = \min\left(\max\left(v_2^{unsat}, -V_{2\_max}\right), V_{2\_max}\right),$$

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 $v_{2\_max} = \sqrt{\left(V_{ph\_max}\right)^2 - \left(v_1^{sat}\right)^2}$.

In the case that the direct and quadrature axes have the same priority (d-q equivalence) the constrained voltages are obtained as follows:

$$v_d^{sat} = \min\left(\max\left(v_d^{unsat}, -V_{d\_max}\right), V_{d\_max}\right)$$

and

$$v_q^{sat} = \min\left(\max\left(v_q^{unsat}, -V_{q\_max}\right), V_{q\_max}\right),$$

where

$$V_{d\_max} = \frac{V_{ph\_max}\left|v_d^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}\left|v_q^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

Integral Anti-Windup

An anti-windup mechanism is employed to avoid saturation of integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}\left(v_d^{sat} - v_d^{unsat}\right)$$

and

$$K_{i\_iq} + K_{aw\_iq}\left(v_q^{sat} - v_q^{unsat}\right),$$

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.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator IPMSG Voltage Stabilization IPMSM Torque Control IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control Switched Reluctance Machine Speed Control Synchronous Reluctance Machine Velocity Control

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.

See Also

Blocks

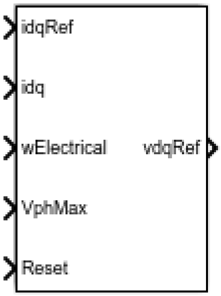
PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

Introduced in R2017b

PMSM Current Controller with Pre-Control

Permanent magnet synchronous machine current controller with pre-control

Library: Simscape / Power Systems / Simscape Components / Control / PMSM Control



Description

The PMSM Current Controller with Pre-Control block implements a discrete-time PI-based 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:

$$v_d^{ref} = \left(K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1} \right) \left(i_d^{ref} - i_d \right) + v_{d\_FF} ,$$

and

$$v_q^{ref} = \left(K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1} \right) \left(i_q^{ref} - i_q \right) + v_{q\_FF} ,$$

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:

$$v_{d\_FF} = \omega_e L_q i_q ,$$

and

$$v_{q\_FF} = -\omega_e \left(L_d i_d + \psi_m \right),$$

where:

- ω_e is the rotor electrical velocity.
- L_d and L_q are the *d*-axis and *q*-axis inductances, respectively.
- ψ_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-cancelation 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}:

$$\sqrt{v_d^2 + v_q^2} \le V_{ph\_max},$$

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:

$$v_1^{sat} = \min\left(\max\left(v_1^{unsat}, -V_{ph\_max}\right), V_{ph\_max}\right)$$

and

$$v_2^{sat} = \min\left(\max\left(v_2^{unsat}, -V_{2\_max}\right), V_{2\_max}\right),$$

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 $v_{2\_max} = \sqrt{\left(V_{ph\_max}\right)^2 - \left(v_1^{sat}\right)^2}$.

In the case that the direct and quadrature axes have the same priority (d-q equivalence), the constrained voltages are obtained as follows:

$$v_d^{sat} = \min\left(\max\left(v_d^{unsat}, -V_{d\_max}\right), V_{d\_max}\right)$$

and

$$v_q^{sat} = \min\left(\max\left(v_q^{unsat}, -V_{q\_max}\right), V_{q\_max}\right),$$

where:

$$V_{d\_max} = \frac{V_{ph\_max}\left|v_d^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}$$

and

$$V_{q\_max} = \frac{V_{ph\_max}\left|v_q^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

Integral Anti-Windup

An anti-windup mechanism is employed to avoid saturation of integrator output. In such a situation, the integrator gains become:

$$K_{i\_id} + K_{aw\_id}\left(v_d^{sat} - v_d^{unsat}\right)$$

and

$$K_{i\_iq} + K_{aw\_iq}\left(v_q^{sat} - v_q^{unsat}\right),$$

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, id (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, iq (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, Ld(id,iq) (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 * `ones(3, 3)`.

`Lq matrix, Lq(id,iq) (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, e.g., L_q * `ones(3, 3)`.

`Permanent magnet flux linkage matrix, PM(id,iq) (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* * `ones(3, 3)`.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator IPMSG Voltage Stabilization IPMSM Velocity Control IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control

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.

See Also

Blocks

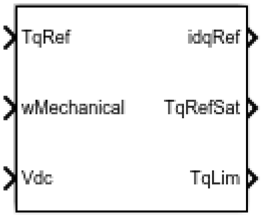
PMSM Current Controller | PMSM Current Reference Generator

Introduced in R2017b

PMSM Current Reference Generator

Permanent magnet synchronous machine current reference generator

Library: Simscape / Power Systems / Simscape Components / Control / PMSM Control



Description

The PMSM Current Reference Generator block implements a current reference generator for 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.
- ψ_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:

$$i_d^{ref} = \mathrm{f}\left(n_m, T_{ref}, v_{dc}\right),$$

and

$$i_q^{ref} = \mathrm{g}\left(n_m, T_{ref}, v_{dc}\right),$$

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 factor, which can be computed as follows:

$$M_f = \frac{V_s}{V_{ph\_max}},$$

where V_s is the stator voltage amplitude and V_{ph_max} is the maximum allowable phase voltage. In the case that the modulation factor 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:

- β 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:

$$T_e = \frac{3p}{2}\psi_m I_s \cos\beta + \frac{3p}{4}\left(L_q - L_d\right) I_s{}^2 \sin 2\beta,$$

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

$$-\frac{3p}{2}\psi_m I_s \sin\beta + \frac{3p}{2}\left(L_q - L_d\right) I_s^2 \left(\cos^2\beta - \sin^2\beta\right) = 0.$$

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:

$$i_{d\_mtpa} = \frac{\psi_m}{2\left(L_q - L_d\right)} - \sqrt{\frac{\psi_m^2}{4\left(L_q - L_d\right)^2} + i_{q\_mtpa}^2}.$$

Finally, by plugging the previous equation into the d-q variant of the PMSM torque equation, the following polynomial is obtained:

$$9p^2\left(L_q - L_d\right)^2 i_{q\_mtpa}^4 + 6T_{ref}\, p\psi_m i_{q\_mtpa} - 4T_{ref}^2 = 0.$$

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:

$$V_s = \sqrt{v_d^2 + v_q^2} \le V_{ph\_max},$$

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

$$v_q = R_s i_q + \omega_e\left(L_d i_d + \psi_m\right).$$

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:

$$i_{d\_fw} = -\frac{\psi_m}{L_d} + \frac{1}{L_d}\sqrt{\frac{V_{ph\_max}^2}{\omega_e^{\,2}} - \left(L_q i_{q\_fw}\right)^2},$$

Finally, by plugging the i_{d_fw} equation into the PMSM torque equation, the following polynomial is obtained:

$$9p^2\left(L_d - L_q\right)^2 L_q^2 \omega_e^{\,2} i_{q\_fw}^4 + \left(9p^2\psi_m^2 L_q^2 \omega_e^{\,2} - 9p^2\left(L_d - L_q\right)^2 V_{ph\_max}^2\right) i_{q\_fw}^2 - 12T_{ref}\, p\psi_m L_d L_q \omega_e^{\,2} i_{q_{fw}} + 4T_{ref}^2 L_d^2$$

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(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.

`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.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator IPMSG Voltage Stabilization IPMSM Velocity Control IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control

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] 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.

[3] 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.

See Also

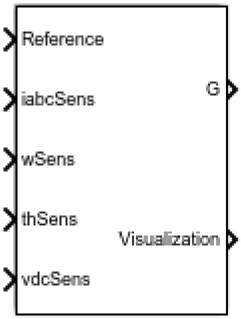
Blocks

PMSM Current Controller | PMSM Current Controller with Pre-Control

Introduced in R2017b

PMSM Field-Oriented Control

Permanent magnet synchronous machine field-oriented control
Library: Simscape / Power Systems / Simscape Components / Control / PMSM Control



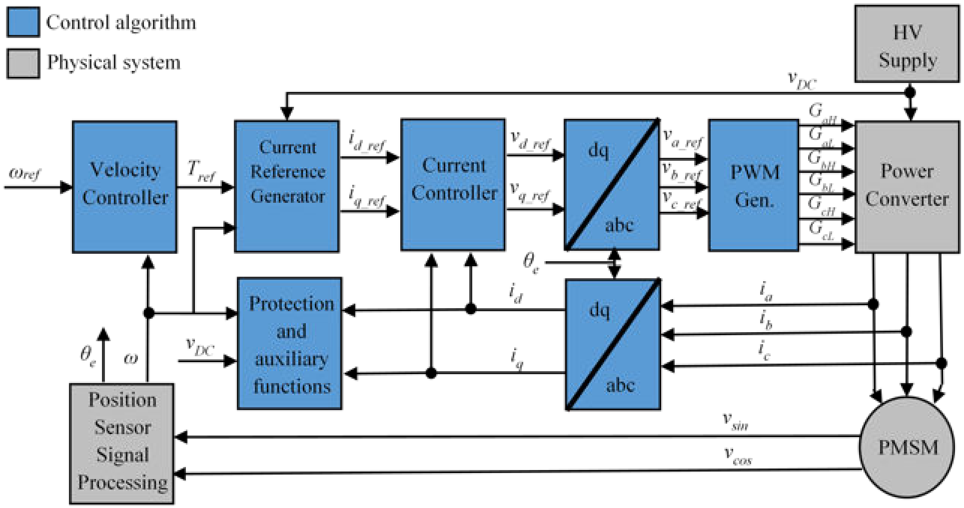
Description

The PMSM Field-Oriented Control block implements a PMSM 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 measurable 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:

- ω and ω_{ref} are the measured and reference angular velocities, respectively.
- T_{ref} is the reference rotor 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.
- θ_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 of 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

`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
`300V` (default) | positive number

Nominal DC-link voltage of the electrical source.

`Maximum power (W)` — Maximum power
`35000W` (default) | positive number

Maximum machine power.

`Maximum torque (N*m)` — Maximum torque
`250N*m` (default) | positive number

Maximum machine torque.

`Inverter dc-link voltage threshold (V)` — DC-link voltage threshold
`100V` (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-PI control`

Specify the type of the control strategy.

`PI controller proportional gain` — Proportional gain of PI controller
`1` (default) | positive number

Proportional gain of the PI controller.

`PI 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.

`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, 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.

`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.

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
`off` (default) | `on`

Enable or disable precontrol voltage.

`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, id (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, iq (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.

`Ld matrix, Ld(id,iq) (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 * `ones(3, 3)`.

`Lq matrix, Lq(id,iq) (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 * `ones(3, 3)`.

`Permanent magnet flux linkage matrix, PM(id,iq) (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* * `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.

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.

See Also

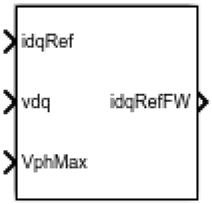
Blocks

PMSM Current Controller | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

Introduced in R2017b

PMSM Field-Weakening Controller

Permanent magnet synchronous machine field-weakening controller
Library: Simscape / Power Systems / Simscape Components / Control / PMSM Control



Description

The PMSM Field-Weakening Controller block implements a field-weakening controller for PMSM machines.

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:

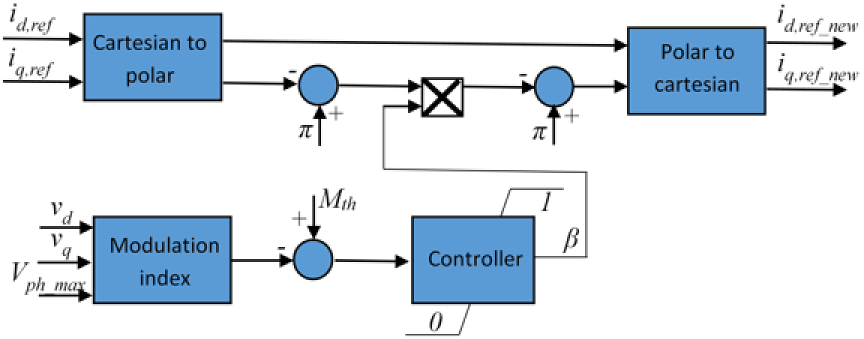
- Use the zero d-axis control technique to generate an unconstrained current reference vector to drive the PMSM. You can implement this strategy with the PMSM Current Reference Generator block.
- Use this block to adjust the angle of the current reference vector in order to satisfy voltage phase constraints.
- 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, β tends to 0, decreasing the q-axis current.
- When the required stator voltage is within its limit, β 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.

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.

See Also

Blocks

PMSM Current Controller | PMSM Current Reference Generator

Introduced in R2017b

PMSM Torque Estimator

Permanent magnet synchronous machine torque estimator
Library: Simscape / Power Systems / Simscape Components / 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:

$$T_e = \frac{3p}{2}\left(\psi_m i_q + \left(L_d - L_q\right) i_d i_q\right),$$

where

- p is the number of pole pairs of the PMSM.
- ψ_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, id (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, iq (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, Ld(id,iq) (H)` — D-axis inductance lookup data
`0.0002 * ones(3, 3)H` (default) | positive matrix

L_d matrix used as lookup table data.

`Lq matrix, Lq(id,iq) (H)` — Q-axis inductance lookup data
`0.0002 * ones(3, 3)H` (default) | positive matrix

L_q matrix used as lookup table data.

`Permanent magnet flux linkage matrix, PM(id,iq) (Wb)` — Flux linkage lookup data
`0.04 * 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.

Model Examples

Electric Engine Dyno Energy Balance in a 48V Starter Generator IPMSG Voltage Stabilization IPMSM Velocity Control IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV

See Also

Blocks

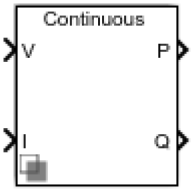
PMSM Current Controller | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

Introduced in R2017b

Power Measurement

Measure single-phase real and reactive power

Library: Simscape / Power Systems / Simscape Components / 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:

$$P_k + jQ_k = G(V_k e^{j\theta_{V_k}})(\overline{I_k e^{j\theta_{I_k}}}),$$

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:

$$V_k e^{j\theta_{V_k}} = \frac{2}{T}\int_{t-T}^{t} V(t)\sin(2\pi kFt)dt + j\frac{2}{T}\int_{t-T}^{t} V(t)\cos(2\pi kFt)dt$$

$$I_k e^{j\theta_{I_k}} = \frac{2}{T}\int_{t-T}^{t} I(t)\sin(2\pi kFt)dt + j\frac{2}{T}\int_{t-T}^{t} I(t)\cos(2\pi kFt)dt.$$

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
vector

Real power for selected frequency components, in W.

Data Types: `single` | `double`

`P` — Reactive power
vector

Reactive power for selected frequency components, in var.

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

Sample time for the block. For continuous operation, set this property to `0`. For discrete operation, specify the sample time explicitly. This block does not support inherited sample time.

See Also

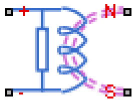
Blocks

RMS Measurement | Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL) | Three-Phase Power Measurement

Introduced in R2017b

Primary Winding

Linear nonideal transformer winding

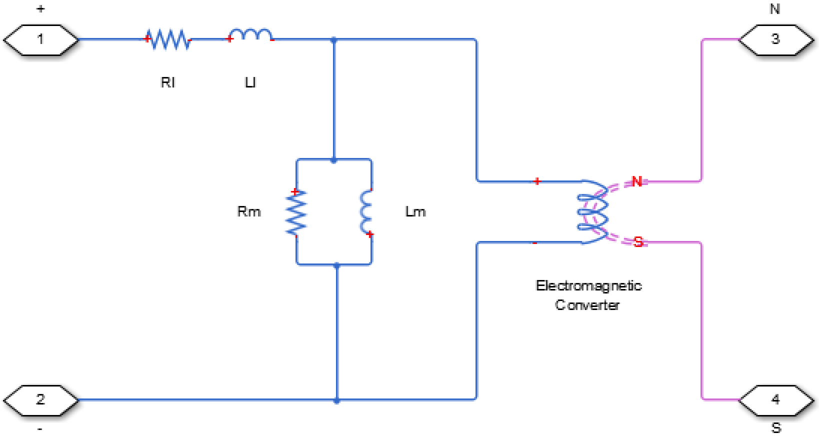


Library

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.

Parameters

- “Main Tab” on page 1-451
- “Variables Tab” on page 1-451

Main Tab

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`.

Variables Tab

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

+

    Positive electrical conserving port

-

    Negative electrical conserving port

N

    North magnetic conserving port

S

    South magnetic conserving port

See Also

Electromagnetic Converter | Secondary Winding

Introduced in R2013b

PWM Generator

Generate pulse width modulated signal

Library: Simscape / Power Systems / Simscape Components / Control / Pulse Width Modulation

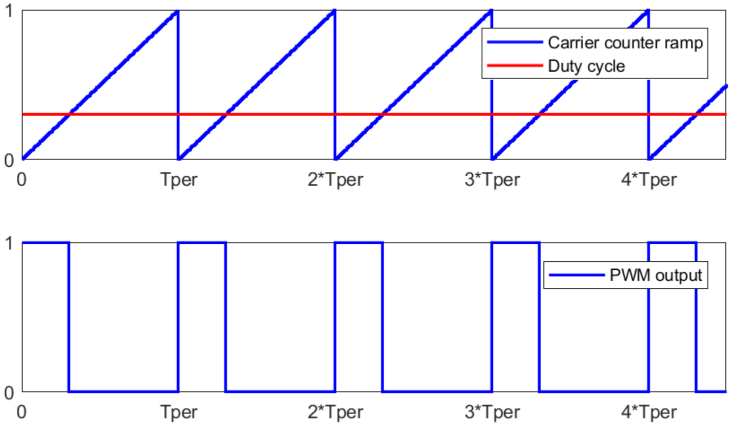
Description

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.

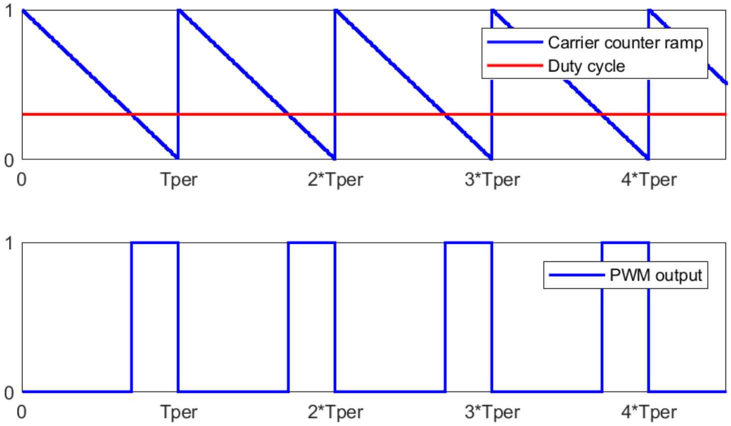
Working Principle

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 *Tper*. You can change the initial output, or phase, of the PWM output by specifying one of three types of carrier counters:

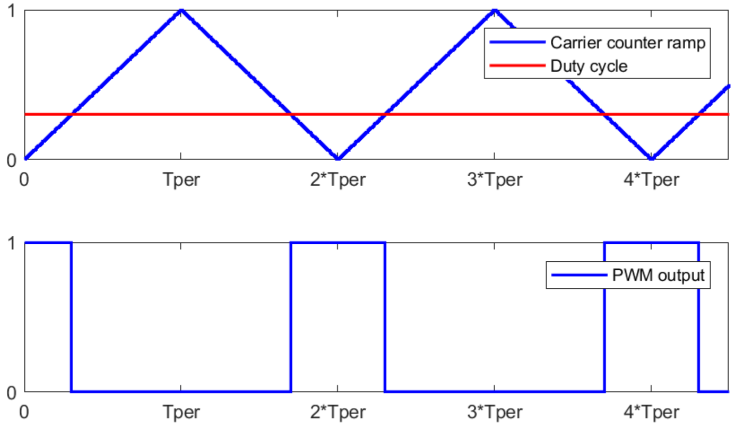
- Up counter — The PWM output signal initializes at the start of the `on` cycle. This graphic shows the carrier counter signal and the corresponding PWM output.



- 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 behaviour 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 scalar

PWM timer period.

`Sample time (s)` **— Block sample time**

`5e-5` (default) | positive number

Sample time for the block. If this block is used in a continuous variable-step model, then the sample time can be explicitly specified.

Model Examples

DC Motor Control HESM Torque Control HESM Velocity Control SM Torque Control SM Velocity Control

See Also

Blocks

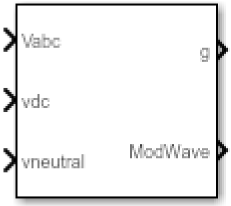
PWM Generator | PWM Generator (Three-phase, Three-level) | PWM Generator (Three-phase, Two-level) | Thyristor 6-Pulse Generator

Introduced in R2017b

PWM Generator (Three-phase, Three-level)

Generate three-phase, three-level pulse width modulated waveform

Library: Simscape / Power Systems / Simscape Components / 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.

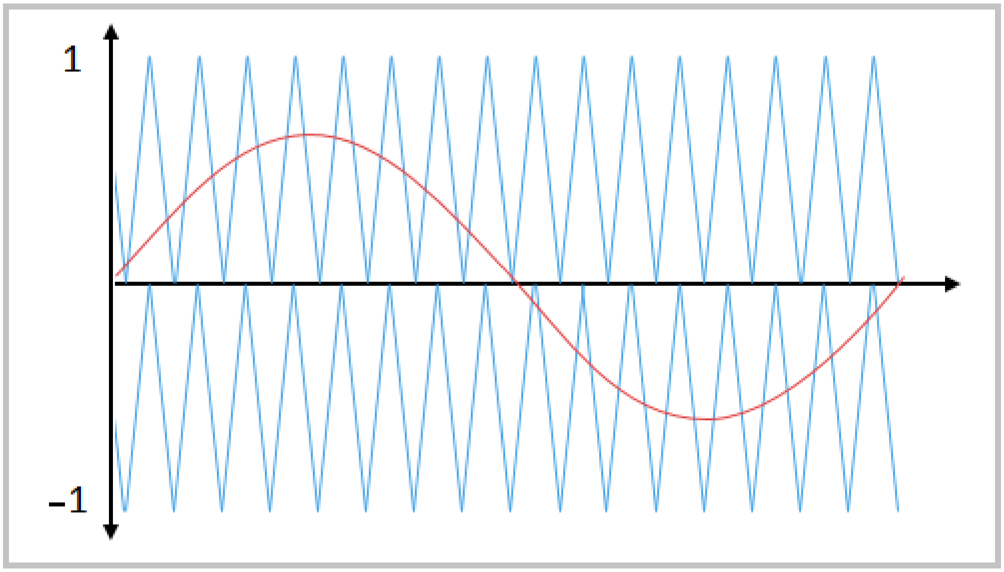
Sampling Mode

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:

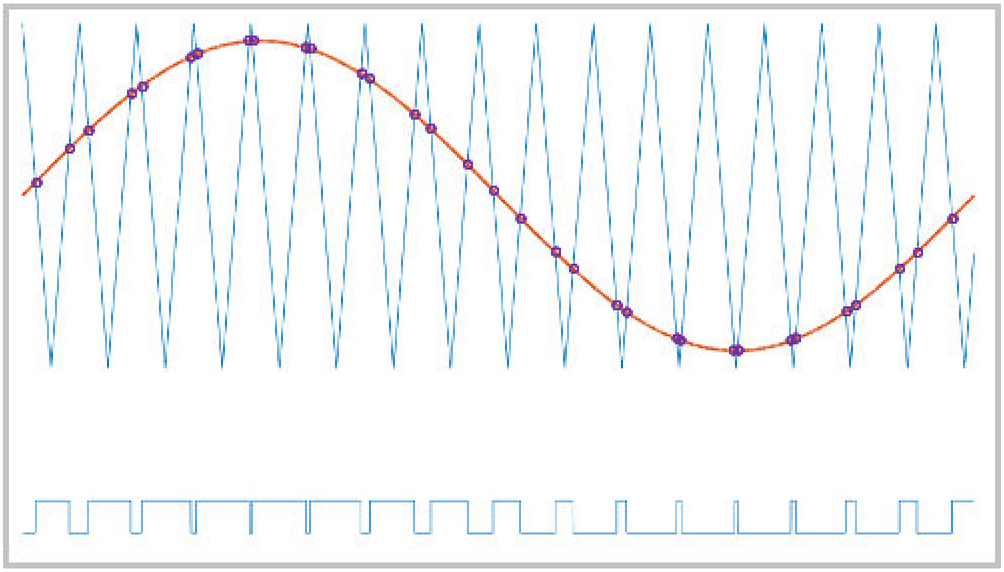
1. Samples a reference wave.
2. Compares the sample to two parallel triangle carrier waves, separated by one level.



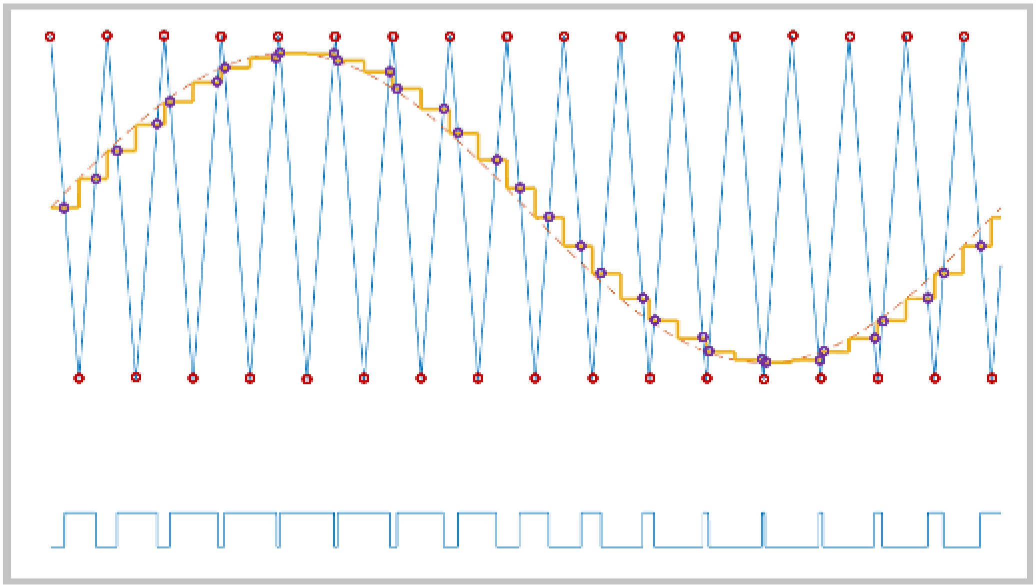
3. Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave.

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:

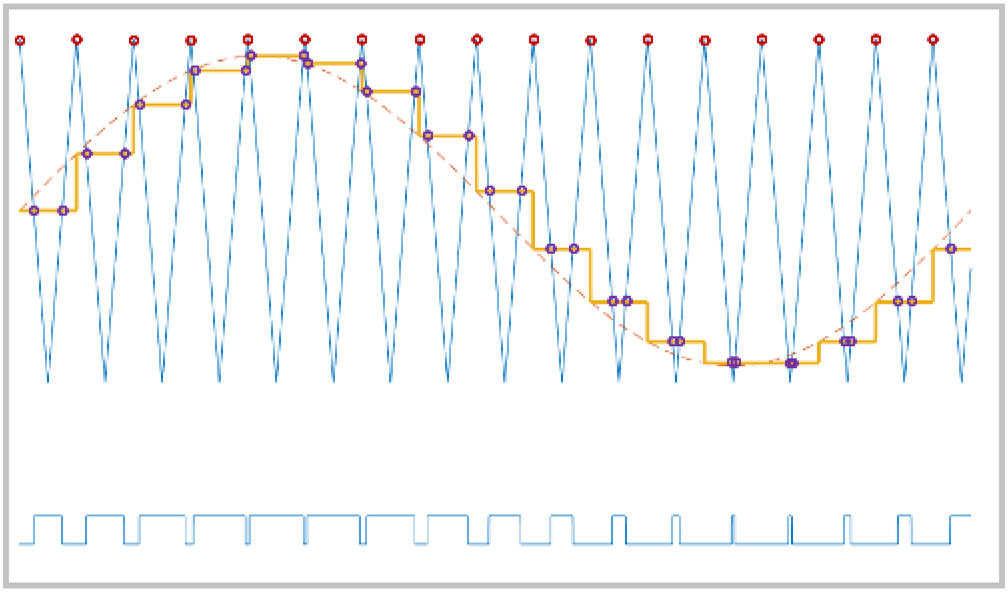
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- 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 or the lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling. The figure show upper boundary sampling.



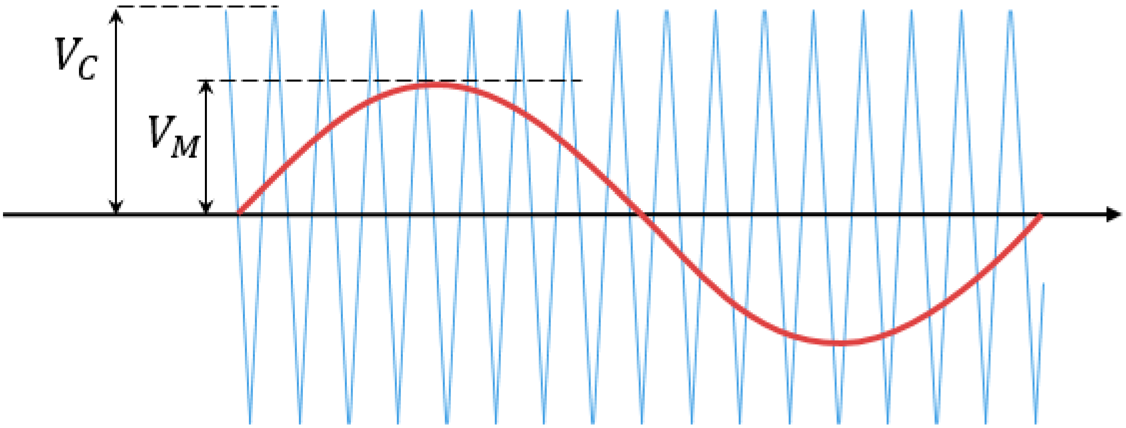
Overmodulation

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,

$$V_{peak} = m\frac{v_{dc}}{2},$$

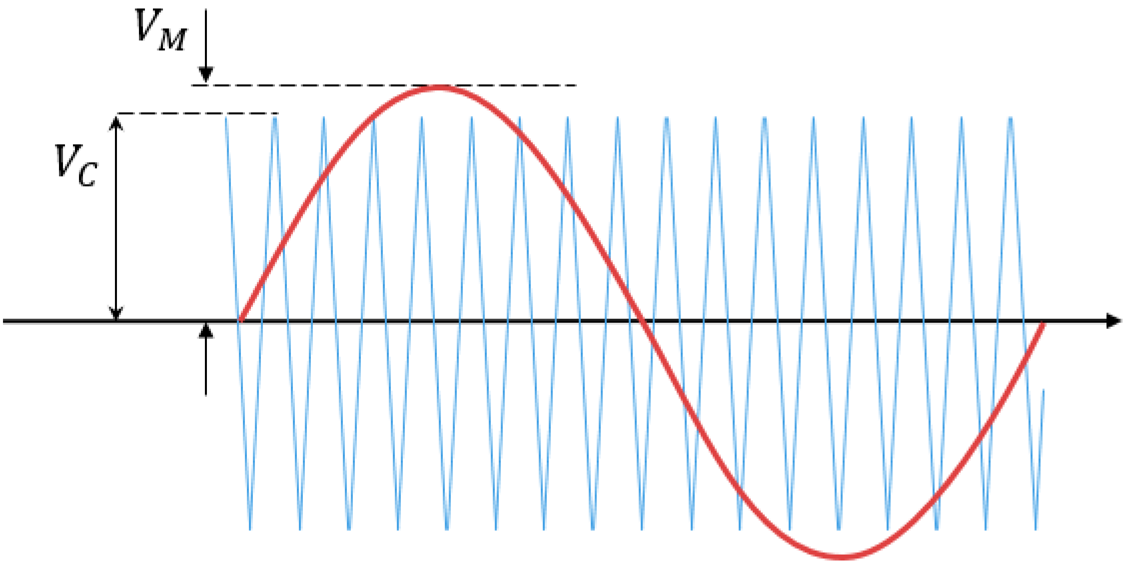
where

- 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),

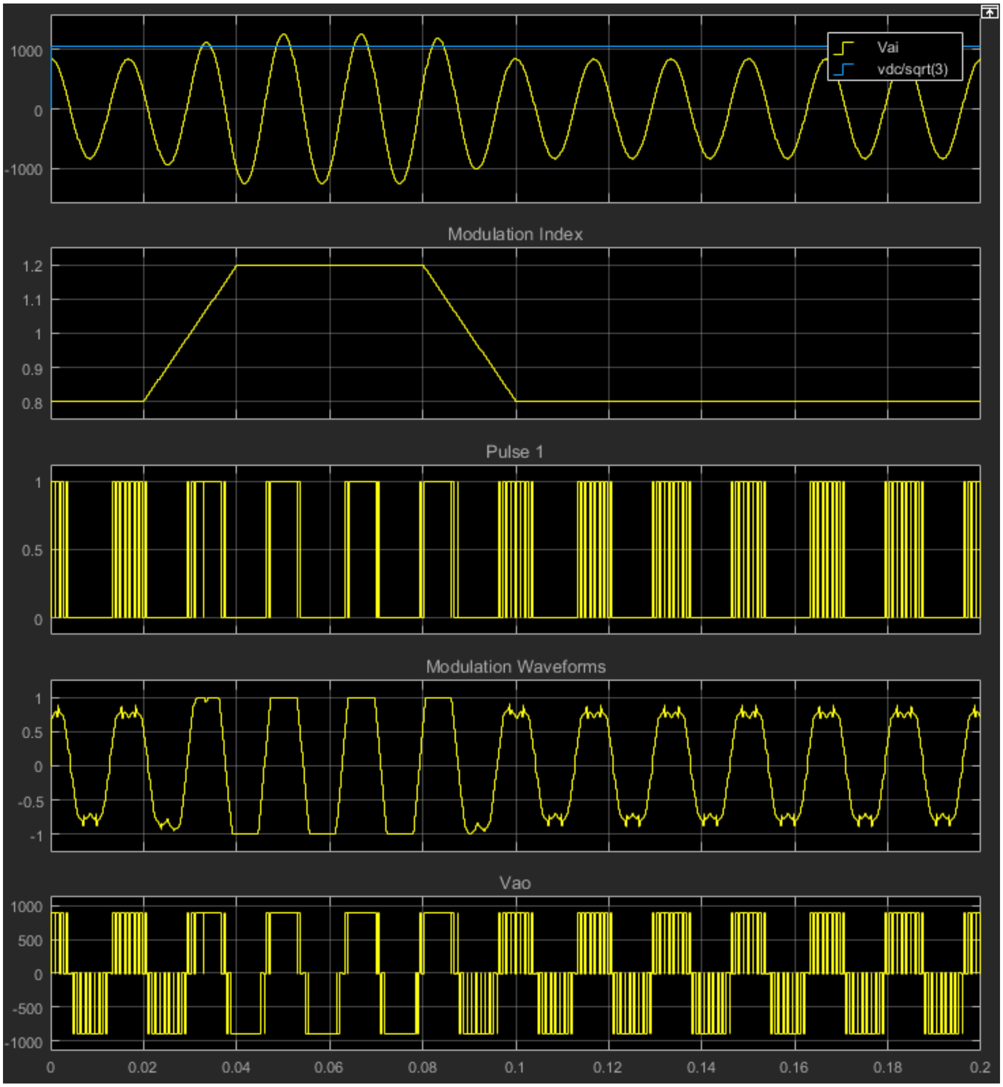
$$V_{peak} = m\frac{v_{dc}}{\sqrt{3}}.$$

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)

Specify the rate at which you want the switches in the power converter to switch.

`Sample time (s)` — Block sample time

`5e-5` (default)

Specify the time interval between successive block executions (output calculations).

Model Examples

Three-Phase Three-Level PWM Generator

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.

See Also

Simscape Blocks

Converter

Blocks

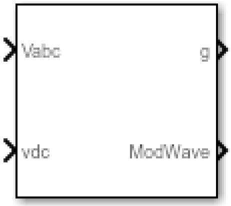
PWM Generator | PWM Generator (Three-phase, Two-level) | Thyristor 6-Pulse Generator

Introduced in R2016b

PWM Generator (Three-phase, Two-level)

Generate three-phase, two-level pulse width modulated waveform

Library: Simscape / Power Systems / Simscape Components / Control / Pulse Width Modulation



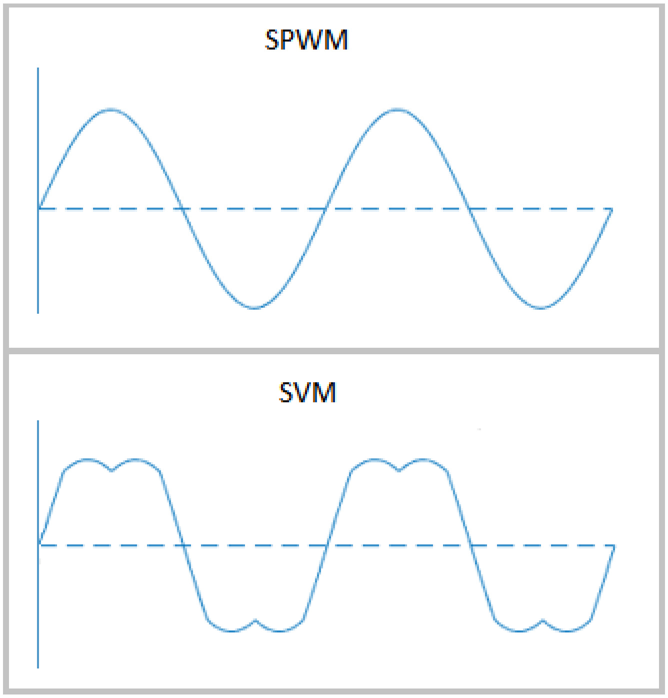
Description

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.

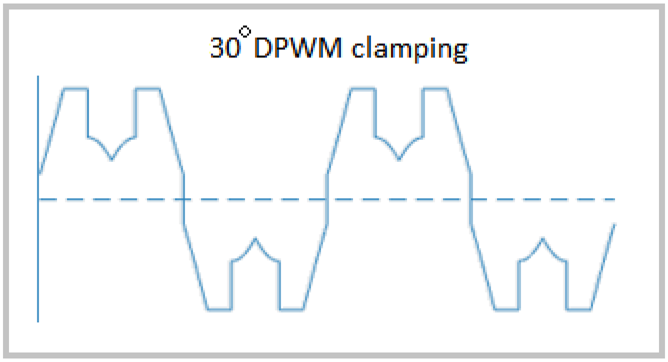
Continuous and Discontinuous PWM

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.

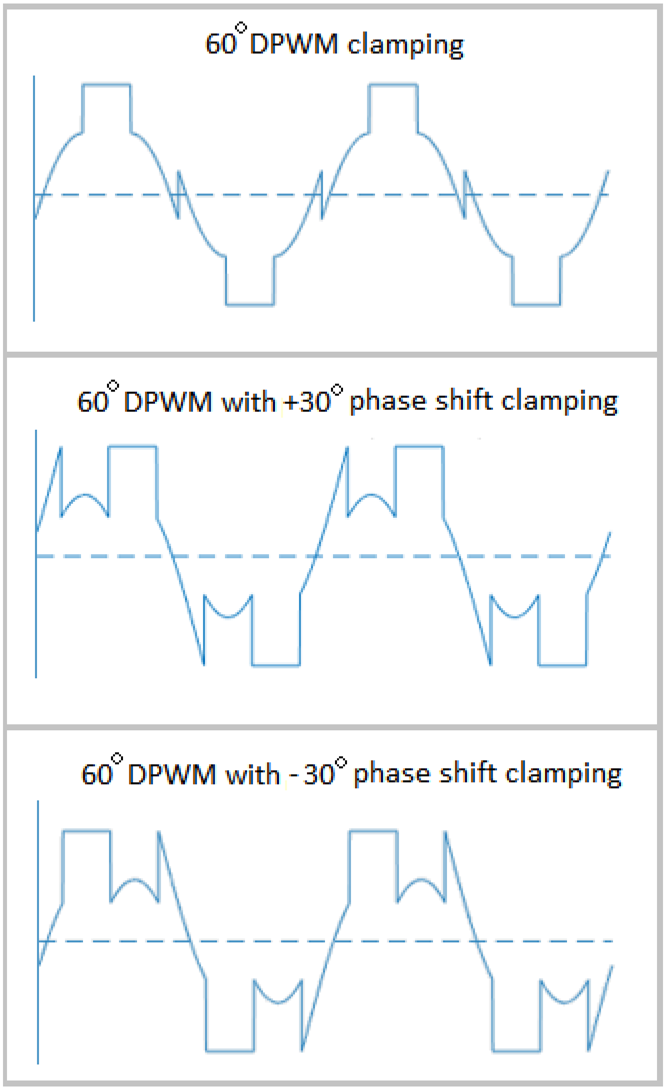


For discontinuous PWM (DPWM), the block clamps the modulation wave to the positive or negative DC rail for a total of 120 degrees during each fundamental period. During the clamping intervals, modulation discontinues.

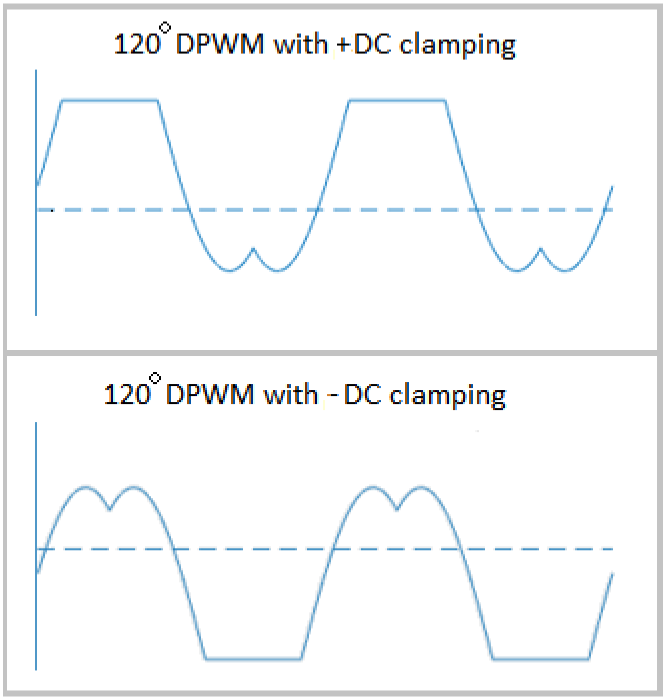
A waveform with 30-degree DPWM has four 30-degree intervals per fundamental period.



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.

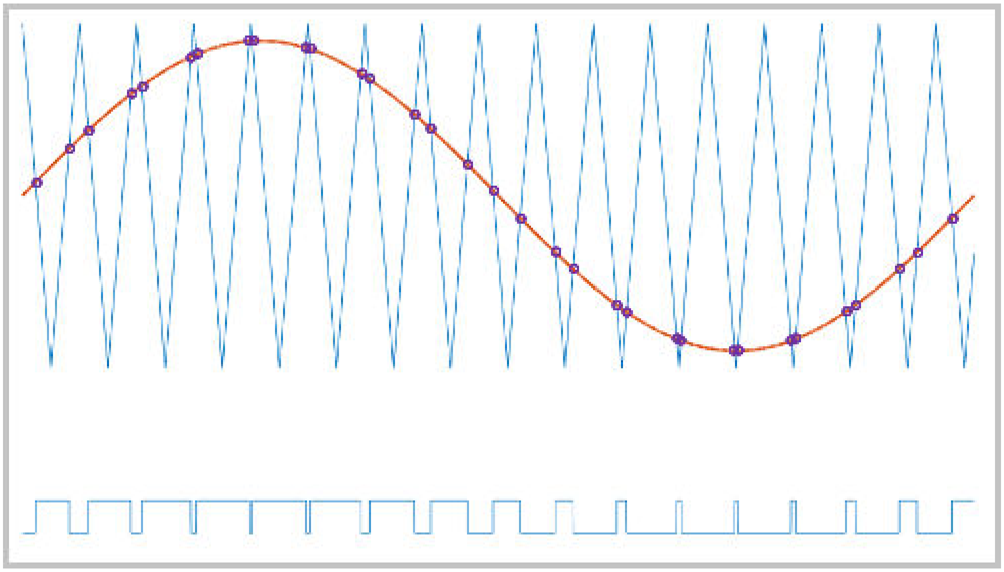
Carrier-based PWM is, however, useful for showing how the sampling mode that you select relates to the switch-on and switch-off behavior of the pulses that the block generates. A generator that uses a two-level, carrier-based PWM method:

1. Samples a reference wave.
2. Compares the sample to a triangle carrier wave.

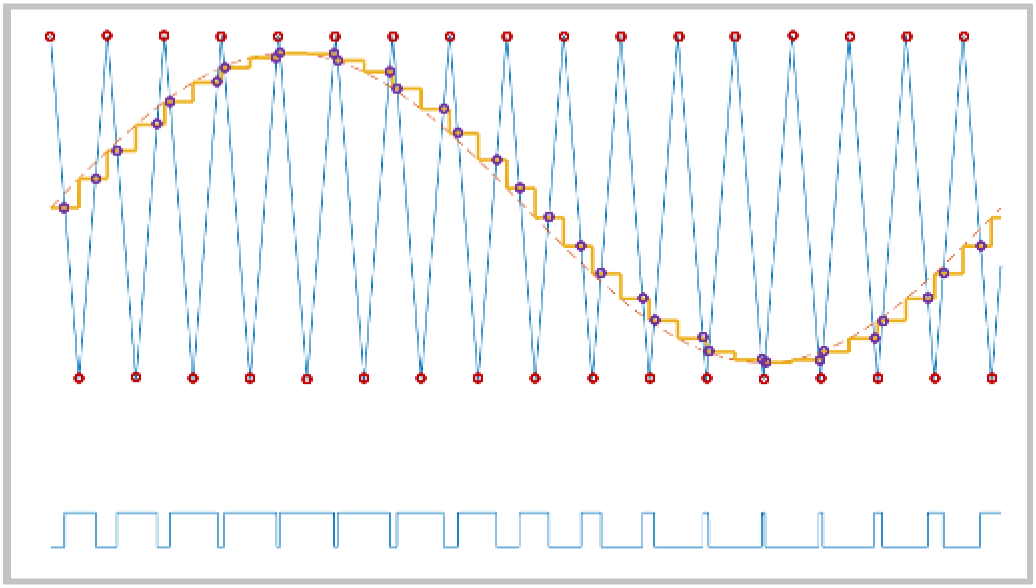
3 Generates a switch-on pulse if a sample is higher than the carrier signal or a switch-off pulse if a sample is lower than the carrier wave.

To determine switch-on and switch-off pulse behavior, a two-level carrier-based PWM generator uses these methods to sample the triangle wave:

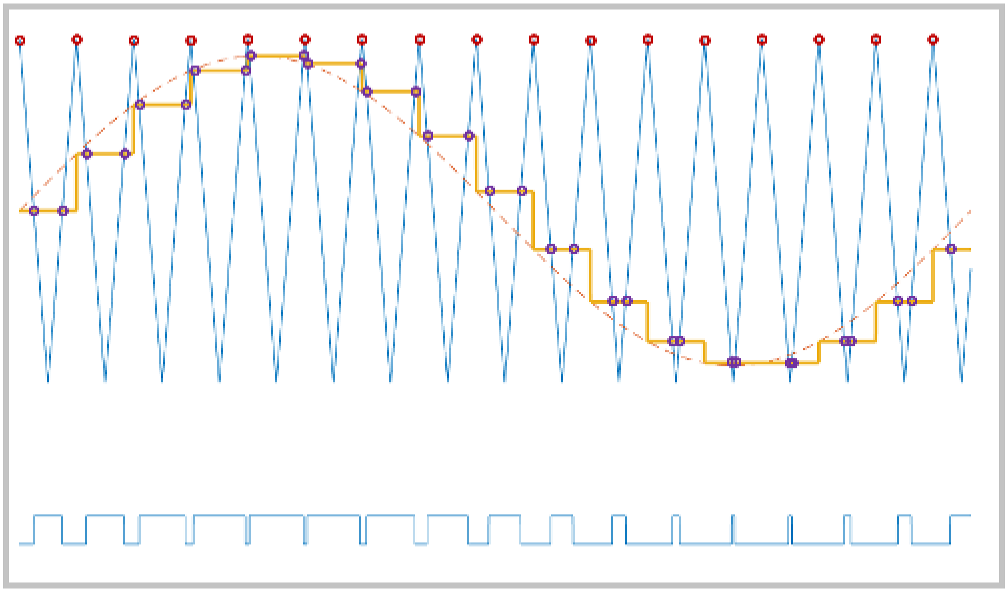
- Natural — The sampling and comparison occur at the intersection points of the modulation wave and the carrier wave.



- 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 or the lower boundaries of the carrier wave. The comparison occurs at the intersection that follows the sampling. The figure show upper boundary sampling.



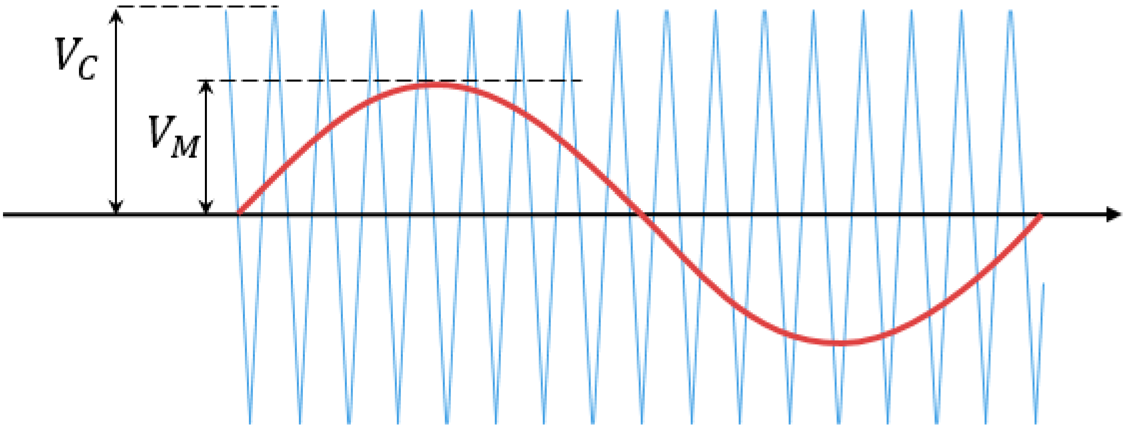
Overmodulation

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,

$$V_{peak} = m\frac{v_{dc}}{2},$$

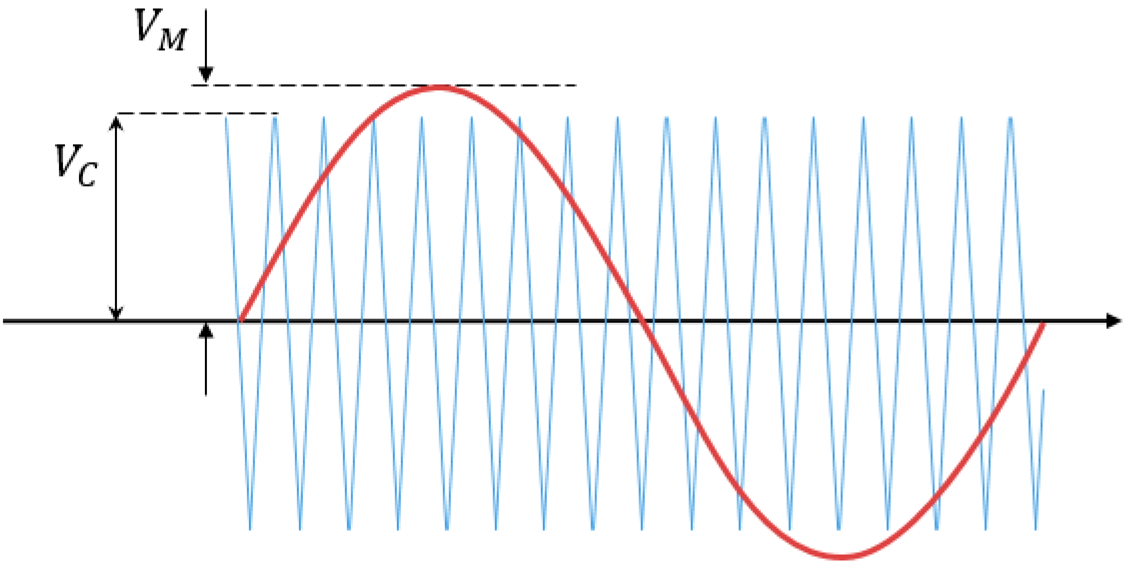
where

- 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,

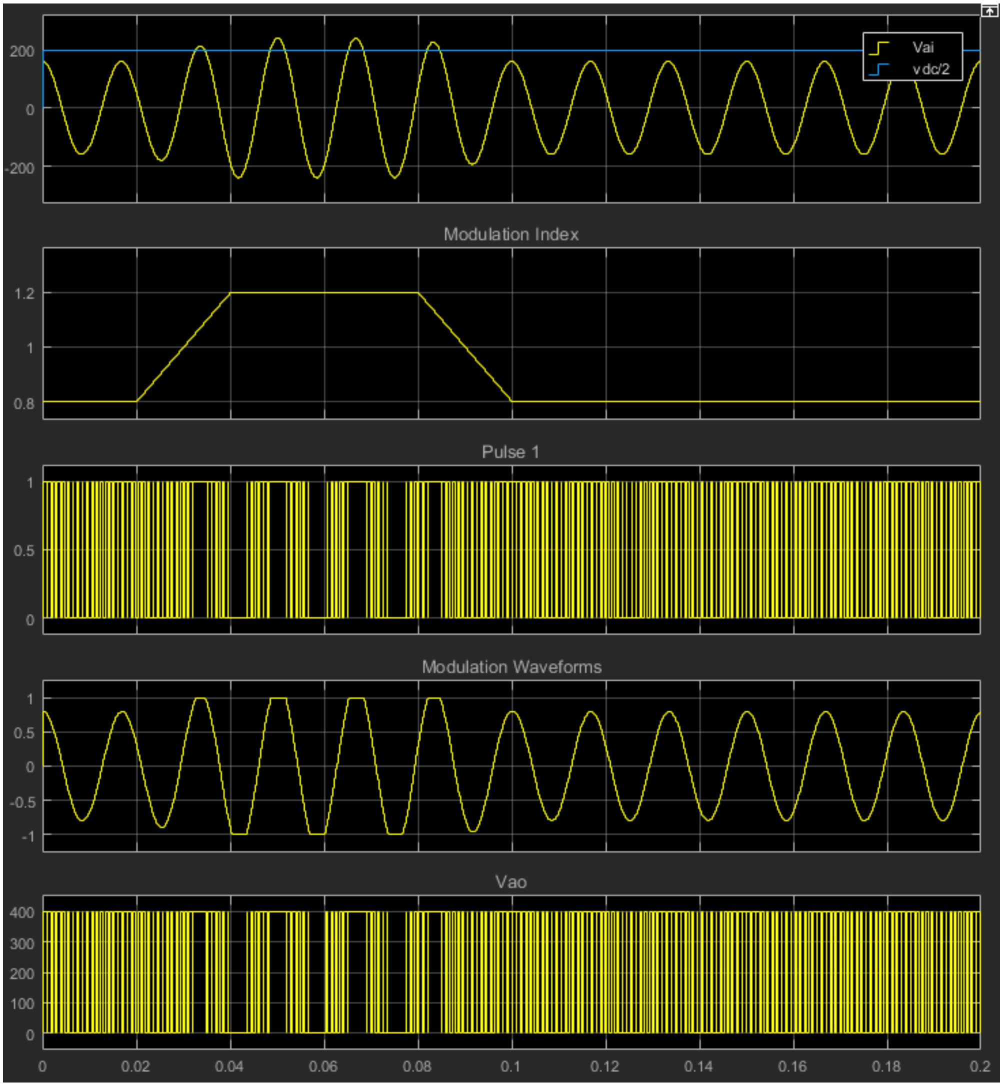
$$V_{peak} = m\frac{v_{dc}}{\sqrt{3}}.$$

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 V/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)

Specify the rate at which you want the switches in the power converter to switch.

`Sample time (s)` — Block sample time

`5e-5` (default)

Specify the time interval between successive block executions (output calculations).

`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)`.

Model Examples

Asynchronous Machine Scalar Control Electric Engine Dyno Energy Balance in a 48V Starter Generator HESM Torque Control HESM Velocity Control IPMSG Voltage Stabilization IPMSM Torque Control in a Parallel HEV IPMSM Torque Control in a Series HEV IPMSM Torque Control in a Series-Parallel HEV IPMSM Torque Control in an Axle-Drive HEV IPMSM Velocity Control SM Torque Control SM Velocity Control Switched Reluctance Machine Speed Control Synchronous Reluctance Machine Velocity Control Three-Phase Asynchronous Drive with Sensor Control Three-Phase Asynchronous Drive with Sensorless Control Three-Phase PMSM Drive Three-Phase Two-Level PWM Generator

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.

See Also

Simscape Blocks

Converter

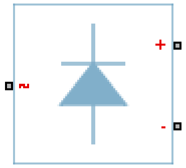
Blocks

PWM Generator | PWM Generator (Three-phase, Three-level) | Thyristor 6-Pulse Generator

Introduced in R2016b

Rectifier

Convert three-phase AC voltage to DC voltage

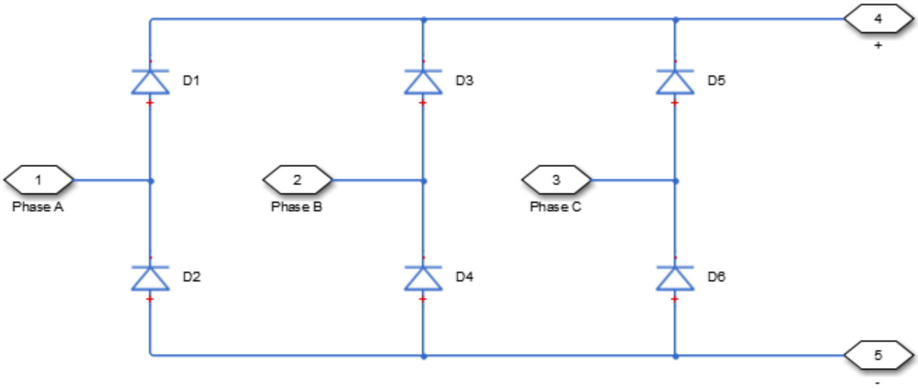


Library

Semiconductors

Description

The Rectifier 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 Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Parameters

- "Main Tab" on page 1-481
- "Charge Dynamics Tab" on page 1-482

Main Tab

Forward voltage

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**. The default forward voltage value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the forward voltage for each diode. The default value is `0.001 Ohm`.

Off conductance

Conductance of each reverse-biased diode. The default value is `1e-5 1/Ohm`.

Charge Dynamics Tab

Model dynamics

Diode charge dynamics. The default value is `No dynamics`.

The charge dynamics options you can select are:

- `No dynamics`
- `Model charge dynamics`

When you select `Model charge dynamics`, additional parameters appear.

Additional Parameters for Model charge dynamics

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 μAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Ports

The block has the following ports:

~

Expandable three-phase port

+

Electrical conserving port associated with the positive terminal

–

Electrical conserving port associated with the negative terminal

See Also

Average-Value Inverter | Average-Value Rectifier | Converter

Topics

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

Introduced in R2013b

Reluctance with Hysteresis

Nonlinear reluctance with magnetic hysteresis

Library: Simscape / Power Systems / Simscape Components / Passive Devices / Fundamental Components

Description

The Reluctance with Hysteresis block models a 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** section 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, respectively:

$$B = \Phi / A_e$$

$$MMF = l_e \cdot H$$

where:

- MMF is magnetomotive force (mmf) across the component.
- Φ is flux through the component.
- B is flux density.
- H is field strength.
- A_e is the effective cross-sectional area of the section being modeled.
- l_e is the effective length of the section being modeled.

The block then implements the relationship between B and H according to the Jiles-Atherton [1 on page 1-490, 2 on page 1-490] equations. The equation that relates B and H to the magnetization of the core is:

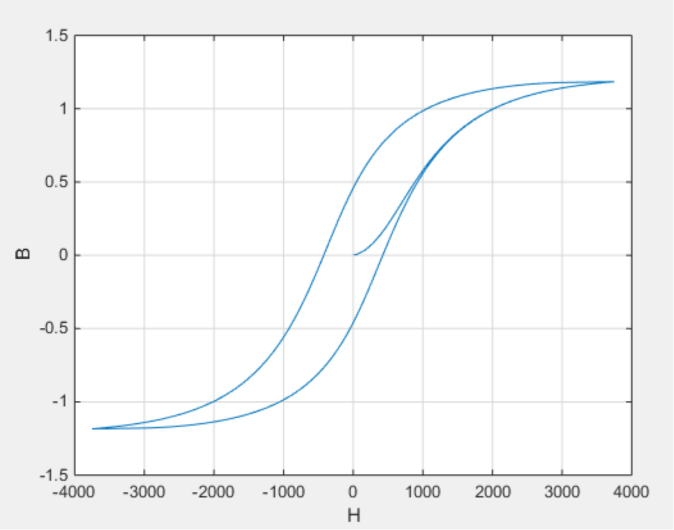
$$B = \mu_0 \left(H + M \right)$$

where:

- μ_0 is the 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 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}:

$$M_{an} = M_s \left(\coth\left(\frac{H_{eff}}{\alpha} \right) - \frac{\alpha}{H_{eff}} \right)$$

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 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 $1 / \left(K\delta - \alpha(M_{an} - M_{irr}) \right)$. The tracking error acts to create the hysteresis at the points where δ changes sign. The main parameter that shapes the irreversible characteristic is K, which is called the bulk coupling coefficient. The parameter α 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 α 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 α 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 $H_{\text{eff}} = 0$) 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 a 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.

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” (Simscape).

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

Geometry

`Effective length` — Effective length of the section being modeled
0.032 m (default)

Effective length of the section being modeled, that is, the average distance of the magnetic path.

`Effective cross-sectional area` — Effective cross-sectional area of the section being modeled
1.6e-5 m^2 (default)

Effective cross-sectional area of the section being modeled, that is, the average area of the magnetic path.

B-H Curve

`Anhysteretic B-H gradient when H is zero` — Gradient of the anhysteretic B-H curve around zero field strength
0.005 m*T/A (default)

The gradient of the anhysteretic (no hysteresis) B-H curve around zero field strength. Set it to the average gradient of the positive-going and negative-going hysteresis curves.

`Flux density point on anhysteretic B-H curve` — Flux density of the point for field strength measurement
1.49 T (default)

Specify a point on the anhysteretic 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.

`Corresponding field strength` — Field strength at measurement point
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.

`Coefficient for reversible magnetization, c` — Proportion of magnetization that is reversible
0.1 (default)

The proportion of the magnetization that is reversible. The value must be greater than zero and less than one.

`Bulk coupling coefficient, K` — Jiles-Atherton equations parameter
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.

`Inter-domain coupling factor, alpha` — Jiles-Atherton equations parameter
0.0001 (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.

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.

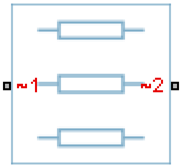
See Also

Nonlinear Inductor | Nonlinear Transformer

Introduced in R2017b

RLC

Three-phase line impedance



Library

Passive Devices

Description

The RLC 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.

Parameters

- "Main Tab" on page 1-492
- "Parasitics Tab" on page 1-492

- “Initial Conditions Tab” on page 1-492

Main Tab

Component structure

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel. The default is `R`, resistor.

Resistance

Resistance of each of the line impedances. This parameter is visible only when you select a component structure that includes a resistor. The default value is `1 Ohm`.

Inductance

Inductance of each of the line impedances. This parameter is visible only when you select a component structure that includes an inductor. The default value is `0.001 H`.

Capacitance

Capacitance in each of the line impedances. This parameter is visible only when you select a component structure that includes a capacitor. The default value is `1e-6 F`.

Parasitics Tab

Parasitic series resistance

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load. The default value is `1e-6 Ohm`.

Parasitic parallel conductance

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load. The default value is `1e-6 1/Ohm`.

Initial Conditions Tab

Initial inductor current [Ia Ib Ic]

Initial current in the a, b, and c phase inductors, respectively. This parameter is visible only when you select a component structure that includes an inductor. The default value is `[0 0 0] A`.

Initial capacitor voltage [Va Vb Vc]

Initial voltage across the a, b, and c phase capacitors, respectively. This parameter is visible only when you select a component structure that includes a capacitor. The default value is `[0 0 0]` `V`.

Block Parameterization

The following table lists the block parameters for each of the configurations, based on the selected **Component structure** option.

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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	Parxasitic 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]

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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]

Ports

The block has two expandable three-phase ports, ~1 and ~2, representing the two terminals of the three-phase line.

See Also

Delta-Connected Load | Wye-Connected Load

Topics

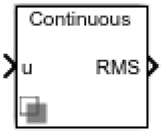
“Three-Phase Asynchronous Machine Starting”
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

RMS Measurement

Measure root-mean-square (RMS) properties of a signal

Library: Simscape / Power Systems / Simscape Components / 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:

$$x_{RMS}(t) = \sqrt{\frac{1}{T}\int_{t-T}^{t} x(t)^2\, dt},$$

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:

$$x_{k,RMS}(t) = G\left(\frac{2}{T}\right)\sqrt{\left(\int_{t-T}^{t} x(t)\sin\left(\frac{2\pi kt}{T}\right)dt\right)^2 + \left(\int_{t-T}^{t} x(t)\cos\left(\frac{2\pi kt}{T}\right)dt\right)^2},$$

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

Sample time for the block. For continuous operation, set this property to `0`. For discrete operation, specify the sample time explicitly. This block does not support inherited sample time.

See Also

Blocks

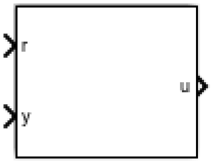
Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL)

Introduced in R2017b

RST Controller

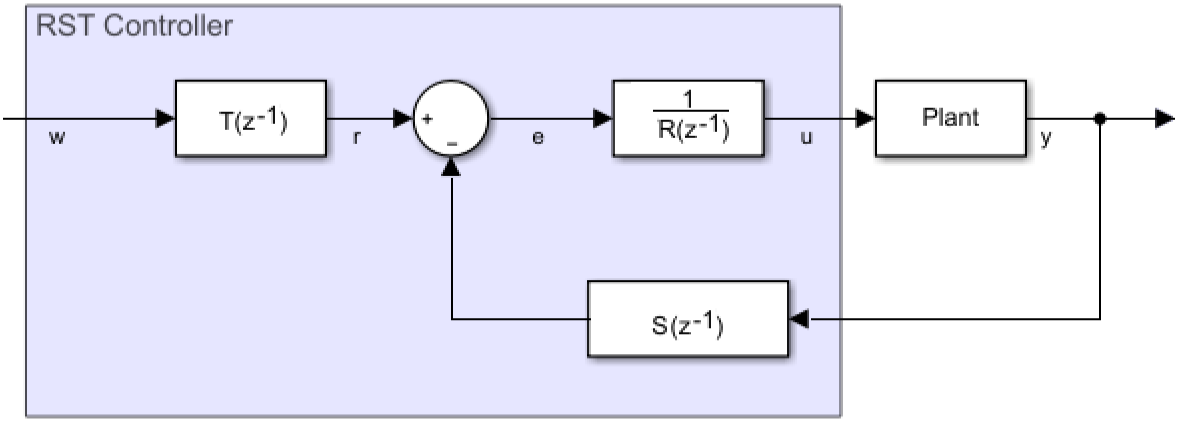
Predictive control using a polynomial representation

Library: Simscape / Power Systems / Simscape Components / 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:

$$A\left(z^{-1}\right)y(k) = z^{-d}B\left(z^{-1}\right)u(k-1) + \frac{e(k)C\left(z^{-1}\right)}{D\left(z^{-1}\right)}$$

$$A\left(z^{-1}\right) = 1 + a_1 z^{-1} + \cdots + a_{n_A} z^{-n_A}$$

$$B\left(z^{-1}\right) = b_0 + b_1 z^{-1} + \cdots + b_{n_B} z^{-n_B}$$

$$C\left(z^{-1}\right) = 1$$

$$D\left(z^{-1}\right) = 1 - z^{-1},$$

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

$$\hat{y}(k+j\,|\,k) = G_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)z^{-d-1}u(k+j) + \frac{H_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)}{C\left(z^{-1}\right)}u(k-1) + \frac{F_{j-d}\left(z^{-1}\right)}{C\left(z^{-1}\right)}y(k)$$

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

$\hat{y}(k + j \mid k)$.

To determine the system polynomials, $F_{j-d}\left(z^{-1}\right)$, $G_{j-d}\left(z^{-1}\right)$, and $H_{j-d}\left(z^{-1}\right)$, the block uses two Diophantine equations. The first Diophantine equation is

$$\frac{C\left(z^{-1}\right)}{A\left(z^{-1}\right)D\left(z^{-1}\right)} = E_{j-d}\left(z^{-1}\right) + z^{-j+d}\frac{F_{j-d}\left(z^{-1}\right)}{A\left(z^{-1}\right)D\left(z^{-1}\right)},$$

where:

$$E_{j-d}\left(z^{-1}\right) = 1 + 1 + e_1 z^{-1} + \cdots + e_{n_E} z^{-n_E}$$

$$F_{j-d}\left(z^{-1}\right) = f_0 + f_1 z^{-1} + \cdots + f_{n_F} z^{-n_F}$$

$$n_E = j - d - 1$$

$$n_F = \max(n_A + n_D - 1, n_C - j + d)$$

The second Diophantine equation is

$$E_{j-d}\left(z^{-1}\right)B\left(z^{-1}\right) = C\left(z^{-1}\right)G_{j-d}\left(z^{-1}\right) + z^{-j+d}H_{j-d}\left(z^{-1}\right),$$

where:

$$G_{j-d}\left(z^{-1}\right) = g_0 + g_1 z^{-1} + \cdots + g_{n_G} z^{-n_G}$$

$$H_{j-d}\left(z^{-1}\right) = h_0 + h_1 z^{-1} + \cdots + h_{n_H} z^{-n_H}$$

$$n_G = j - d - 1$$

$$n_H = \max(n_C, n_B + d) - 1$$

The resulting prediction model is

$$\hat{y}(k + j \mid k) = G_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)z^{-d-1}u(k + j) + \hat{y}_0(k + j \mid k),$$

where

$$\hat{y}_0(k+j\mid k)=\frac{H_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)}{C\left(z^{-1}\right)}u(k-1)+\frac{F_{j-d}\left(z^{-1}\right)}{C\left(z^{-1}\right)}y(k)$$

represents the free response of the system.

Using the matrix notation, the prediction model can be written as

$$\hat{\mathrm{y}}=\mathrm{Gu}_d+\hat{\mathrm{y}}_0,$$

where:

$$\hat{\mathrm{y}}=\left[\hat{y}(k+hi\mid k),\hat{y}(k+hi+1\mid k),\cdots,\hat{y}(k+hp\mid k)\right]^T$$

$$\mathrm{G}=\begin{bmatrix} g_{hi-d-1} & \cdots & g_0 & 0 & \cdots & 0 \\ g_{hi-d} & \cdots & g_1 & g_0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{hc-1} & \cdots & \cdots & \cdots & \cdots & g_0 \\ g_{hp-d-1} & \cdots & \cdots & \cdots & \cdots & g_{hp-hc-1} \end{bmatrix}$$

$$\mathrm{u}_d=\left[D\left(z^{-1}\right)u(k),\cdots,D\left(z^{-1}\right)u(k+hc-1)\right]^T$$

$$\hat{\mathrm{y}}_0=\left[\hat{y}_0(k+hi\mid k),\hat{y}_0(k+hi+1\mid k),\cdots,\hat{y}_0(k+hp\mid k)\right]^T$$

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, λ, such that

$$J=(\mathrm{Gu}_d+\hat{\mathrm{y}}_0-\mathrm{w})^T(\mathrm{Gu}_d+\hat{\mathrm{y}}_0-\mathrm{w})+\lambda\mathrm{u}_d^T\mathrm{u}_d$$

for

$$D\left(z^{-1}\right)u(k+i)=0$$

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:

$$\mathrm{u}_d^* = \left(\mathrm{G}^T\mathrm{G} + \lambda\mathrm{I}_{hc}\right)\mathrm{G}^T\left[\mathrm{w} - \mathrm{y}_0\right].$$

As γ_j and $j = \overline{hi,hp}$ are elements in the first row of the matrix $\left(\mathrm{G}^T\mathrm{G} + \lambda\mathrm{I}_{hc}\right)^{-1}\mathrm{G}^T$, applying the receding horizon principle yields the control algorithm equation as

$$D\left(z^{-1}\right)u(k) = \sum_{j=hi}^{hp}\gamma_j\left[w(k+j|k) - \hat{y}_0(k+j\,|\,k)\right].$$

Substitution using

$$\hat{y}_0(k+j\,|\,k) = \frac{H_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)}{C\left(z^{-1}\right)}u(k-1) + \frac{F_{j-d}\left(z^{-1}\right)}{C\left(z^{-1}\right)}y(k)$$

yields this form of the control algorithm equation:

$$C\left(z^{-1}\right)D\left(z^{-1}\right)u(k) = -\sum_{j=hi}^{hp}\gamma_j H_{j-d}\left(z^{-1}\right)D\left(z^{-1}\right)u(k-1) - \sum_{j=hi}^{hp}\gamma_j F_{j-d}\left(z^{-1}\right)y(k) + \sum_{j=hi}^{hp}\gamma_j C\left(z^{-1}\right)w(k+j).$$

The polynomial form of the control algorithm follows as

$$R\left(z^{-1}\right)u(k) + S\left(z^{-1}\right)y(k) = T\left(z^{-1}\right)w(k+hp),$$

where:

$$R\left(z^{-1}\right) = \left(C\left(z^{-1}\right) + \sum_{j=hi}^{hp}\gamma_j z^{-1}H_{j-d}\left(z^{-1}\right)\right)D\left(z^{-1}\right),$$

$$S\left(z^{-1}\right) = \sum_{j=hi}^{hp}\gamma_j F_{j-d}\left(z^{-1}\right),$$

and

$$T\left(z^{-1}\right) = C\left(z^{-1}\right)\sum_{j=hi}^{hp}\gamma_j z^{-hp+j}.$$

Limitations

To obtain the *R*, *R*, 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?” (Simulink) and “Specify Sample Time” (Simulink).

References

[1] Camacho, E. F. and C. Bordons. *Model Predictive Control.* Second Edition, London: Springer, 2007.

See Also

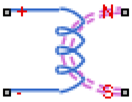
Blocks

Smith Predictor Controller | State-Feedback Controller

Introduced in R2017b

Secondary Winding

Linear nonideal transformer winding

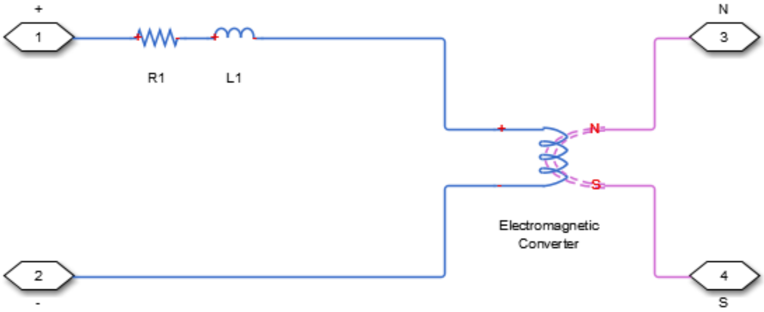


Library

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.

Parameters

- “Main Tab” on page 1-508
- “Variables Tab” on page 1-508

Main Tab

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`.

Variables Tab

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

+

Positive electrical conserving port

-

Negative electrical conserving port

N

North magnetic conserving port

S

South magnetic conserving port

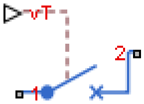
See Also

Electromagnetic Converter | Primary Winding

Introduced in R2013b

Single-Phase Circuit Breaker

Single-pole single-throw circuit breaker



Library

Switches & Breakers / Fundamental Components

Description

The Single-Phase 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` controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
`vT` < **Threshold**	The breaker is closed. Port `1` connects to port `2`.	**Closed Resistance**
`vT` ≥ **Threshold**	When the current in port `1` goes through zero, the phase disconnects from port `2`. The breaker is open.	**Open Conductance**

Parameters

Closed resistance

Resistance between ports `1` and `2` when the breaker is closed. The default value is `0.001 Ohm`.

Open conductance

Conductance between ports `1` and `2` when the breaker is open. The default value is `1e-6 1/Ohm`.

Threshold

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. The default value is `0 V`.

Ports

The block has the following ports:

`1`

Electrical conserving port

`2`

Electrical conserving port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Circuit Breaker | Single-Phase Circuit Breaker (with arc)

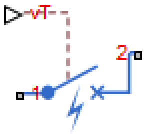
Topics

"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Single-Phase Circuit Breaker (with arc)

Single-pole single-throw circuit breaker with Mayr arc representation



Library

Switches & Breakers / Fundamental Components

Description

The Single-Phase 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` < **Threshold**	The circuit breaker is closed. Port 1 is connected to port 2.
`vT` ≥ **Threshold**	The circuit breaker is either opening or open. Port 1 is connected to port 2 via a nonlinear conductance.

The Single-Phase Circuit Breaker (with arc) block has a higher computational overhead than the Single-Phase Circuit Breaker block. If the fidelity of the representation of arc current or voltage is your overriding requirement, use the Single-Phase Circuit Breaker (with arc) block and use a global Simulink® variable-step solver. Otherwise, use the Single-Phase 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 open,

$$\frac{dx}{dt} = 0.$$

When the breaker is closed,

$$\frac{dx}{dt} = \frac{1}{\tau}\left(\frac{gv^2}{P} - 1\right),$$

where:

- τ is the arc time constant.
- P is the cooling power.

Parameters

Arc time constant, tau

The default value is `0.3e-6 s`.

Cooling power, P

The default value is `30900 W`.

Initial arc conductance, g0

Conductance between ports `1` and `2` when the breaker is closed. The default value is `1e4 S`.

Threshold

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. The default value is `0 V`.

Ports

The block has the following ports:

`1`

Electrical conserving port

`2`

Electrical conserving port

`vT`

Scalar control port, which is either a physical signal or an electrical port

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.

See Also

Circuit Breaker | Single-Phase Circuit Breaker

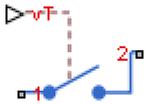
Topics

“Switch Between Physical Signal and Electrical Ports”

Introduced in R2015b

Single-Phase Switch

Single-pole single-throw switch



Library

Switches & Breakers / Fundamental Components

Description

The Single-Phase Switch block models a single-pole single-throw switch that uses an external signal to connect port `1` to port `2` via internal resistance.

The table shows how the external signal `vT` controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
`vT` ≤ **Threshold**	The switch is open. Port `1` is connected to port `2` via large internal resistance.	**Open conductance**
`vT` > **Threshold**	The switch is closed. Port `1` is connected to port `2` via small internal resistance.	**Closed resistance**

Parameters

Closed resistance

Resistance between ports `1` and `2` when the switch is closed. The default value is `0.001 Ohm`.

Open conductance

Conductance between ports `1` and `2` when the switch is open. The default value is `1e-6` `1/Ohm`.

r **Threshold**

Threshold voltage for the control port `vT`. When the voltage is above the threshold, the switch is closed. The default value is `0` `V`.

Ports

The block has the following ports:

`1`

Electrical conserving port

`2`

Electrical conserving port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Single-Phase Two-Way Switch | Switch | Two-Way Switch

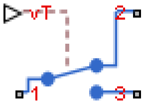
Topics

"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Single-Phase Two-Way Switch

Single-pole double-throw switch



Library

Switches & Breakers / Fundamental Components

Description

The Single-Phase Two-Way Switch block models a single-pole double-throw switch that uses an external signal to connect the port `1` to either of two ports `2` or `3` via internal resistance.

The table shows how the external signal `vT` controls the block behavior.

Condition	Block Behavior	Resistance Parameter Used
`vT` ≤ **Threshold**	Port `1` is connected to port `2` via internal resistance. Port `3` is unconnected.	**Open conductance** (port `1` to port `3`).**Closed resistance** (port `1` to port `2`)
`vT` > **Threshold**	Port `1` is connected to port `3` via internal resistance. Port `2` is unconnected.	**Open conductance** (port `1` to port `2`).**Closed resistance** (port `1` to port `3`)

Parameters

Closed resistance

Resistance between ports `1` and `3` when the switch is closed. The default value is `0.001` `Ohm`.

Open conductance

Conductance between ports `1` and `2` when the switch is open. The default value is `1e-6` `1/Ohm`.

Threshold

Threshold voltage for the control port `vT`. When the voltage is above the threshold, the switch is closed. The default value is `0` `V`.

Ports

The block has the following ports:

`1`

Electrical conserving port

`2`

Electrical conserving port

`3`

Electrical conserving port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Single-Phase Switch | Switch | Two-Way Switch

Topics

"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Sinusoidal Measurement (PLL)

Estimate sinusoidal characteristics using a phase-locked loop

Library: Simscape / Power Systems / Simscape Components / 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:

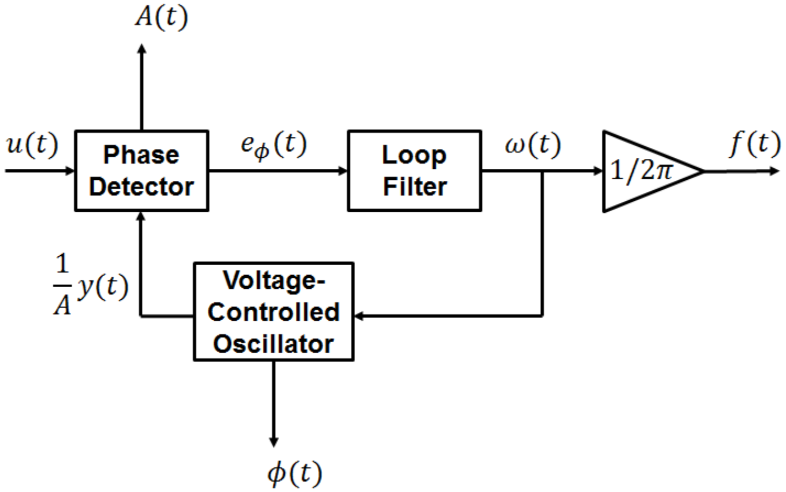
- y is the estimate of the input signal.
- A is the estimate of the amplitude of the input signal.
- ϕ_0 is the initial phase angle of the input signal.

The estimated phase angle ϕ is the angle of this generated sinusoid:

$$\phi(t) = \phi_0 + \int 2\pi f(t)dt,$$

where f if the frequency of the sinusoid, and ϕ_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_ϕ 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 ω 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 ϕ. 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) | `-1` or positive number

Sample time for the block (-1 for inherited). Set this to `0` for continuous operation, or explicitly for discrete operation. 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.

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..

See Also

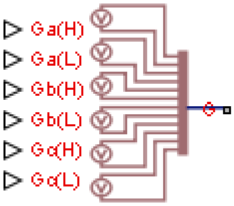
Blocks

RMS Measurement | Three-Phase Sinusoidal Measurement (PLL)

Introduced in R2017b

Six-Pulse Gate Multiplexer

Multiplex gate input signals to Converter block



Library

Semiconductors

Description

The Six-Pulse Gate Multiplexer block routes gate voltage signals to the six switching devices in a Converter block. The block multiplexes the six separate gate signals into a single vector.

If you want to use Simscape Electronics to model the electronics that drive the Converter block, you can switch the input ports of the Six-Pulse Gate Multiplexer block from physical signal ports to electrical ports.

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 block and the negative DC voltage common to each low-side switching device in the Converter block.

Ports

The block has the following ports:

`Ga(H),Gb(H),Gc(H)`

Ports associated with the gate terminals of the Converter block high-side switching devices. You can set the ports to either physical signal or electrical ports.

`Ga(L),Gb(L),Gc(L)`

Ports associated with the gate terminals of the Converter block low-side switching devices. You can set the ports to either physical signal or electrical ports.

`G`

Vector output port associated with the multiplexed gate signals. Connect this port to the `G` port of the Converter block.

`a,b,c`

Electrical conserving ports associated with the individual phase voltages that connect to the high-side switching devices of the Converter block. These ports are visible only if you set the input ports of the Six-Pulse Gate Multiplexer block to electrical ports.

`L`

Electrical conserving port associated with the negative DC voltage common to each low-side switching device in the Converter block. These ports are visible only if you set the input ports of the Six-Pulse Gate Multiplexer block to electrical ports.

See Also

Converter

Topics

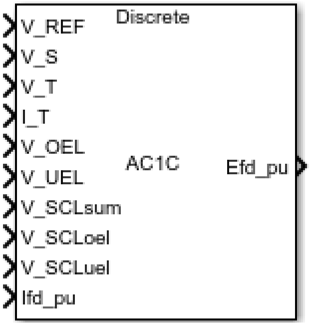
“Switch Between Physical Signal and Electrical Ports”

Introduced in R2013b

SM AC1C

Synchronous machine AC1C excitation system including an Automatic Voltage Regulator (AVR) and an exciter

Library: Simscape / Power Systems / Simscape Components / Control / SM Control



Description

The SM AC1C block implements a synchronous machine type AC1C excitation system model in conformance with IEEE 421.5-2016[1].

Use this block to model the control and regulation of the field voltage of a synchronous machine operating as a generator using an AC rotating exciter.

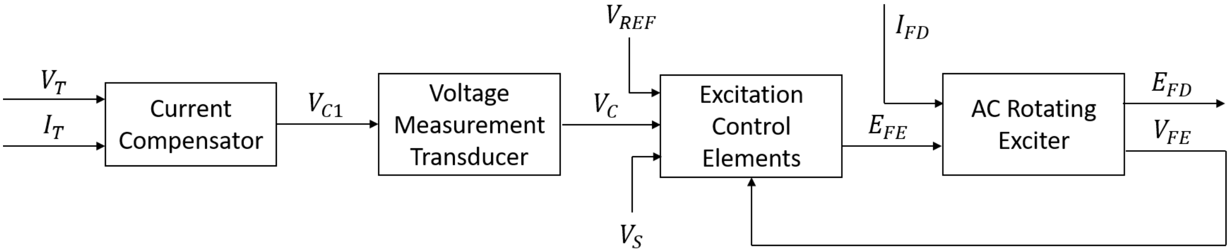
You can switch between continuous and discrete implementations of the block by using the Sample time parameter. To configure the integrator for continuous time, set the **Sample time** property to `0`. To configure the integrator for discrete time, set the **Sample time** property to a positive, nonzero value, or to `-1` to inherit the sample time from an upstream block.

The 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, producing a field voltage to be applied to the controlled synchronous machine. The block also feeds the exciter field current (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:

$$V_{C1} = V_T + I_T \sqrt{R_C^2 + X_C^2},$$

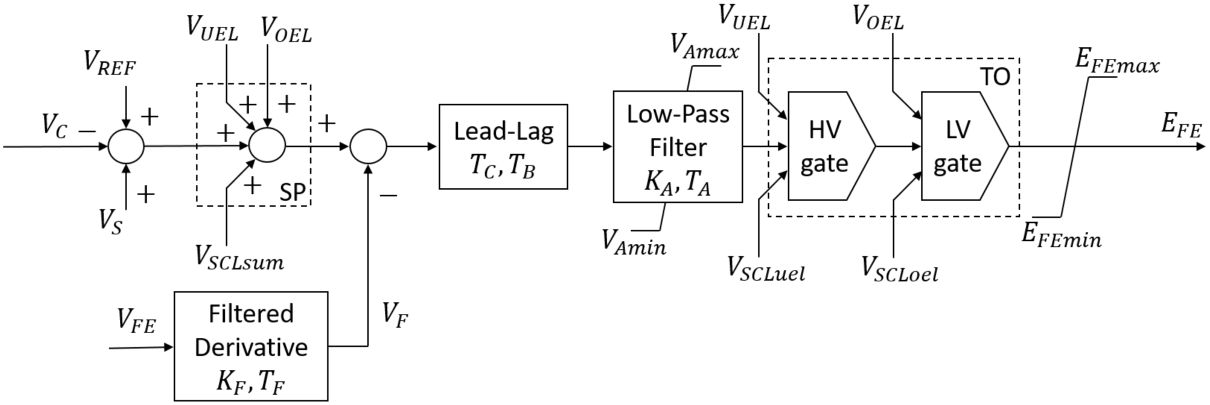
where:

- 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 exact 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 1-530.
- 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_{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 1-530.
- 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.

- 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, risking desynchronization.
- Use a stator current limiter to prevent overheating of the stator windings due to excessive current.

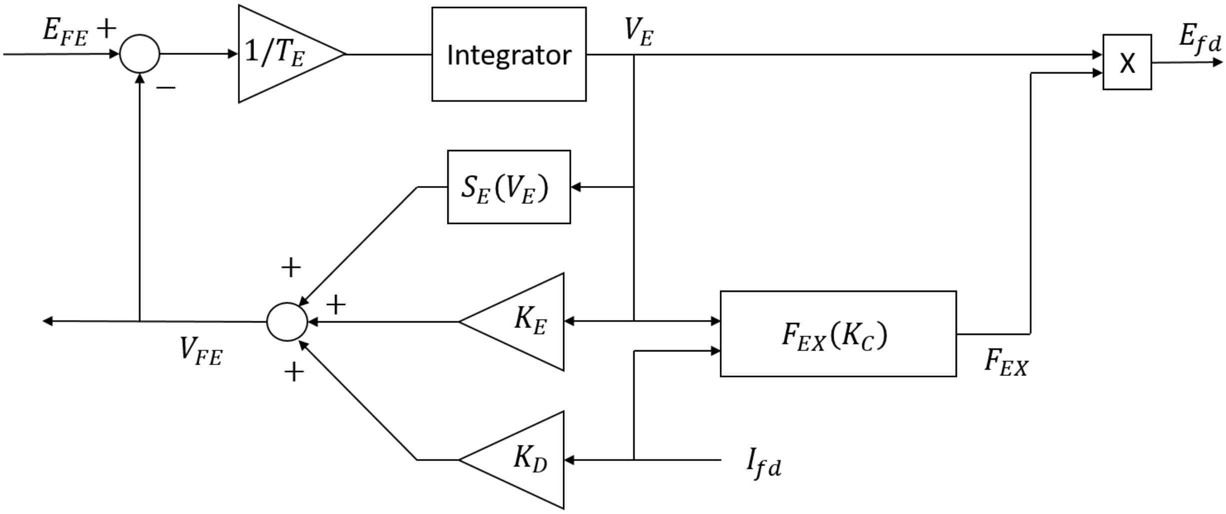
Attach the output of any of these limiters at one of these points:

- The summation point as part of the AVR feedback loop
- The take-over point to override the usual behaviour 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_{SCLoel} and an 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 $S_E(V_E)$ 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 modelled using the demagnetization constant K_D in the feedback loop.
- The Integrator block 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 rectifier regulation. This function depends on the constant K_C which itself is a function of commutating reactance.

Ports

Input

`V_REF` — Voltage reference

scalar

Voltage regulator reference set point, in per-unit representation.

Data Types: `single` | `double`

`V_S` — Input from stabilizer
scalar

Input from the power system stabilizer, in per-unit representation.

Data Types: `single` | `double`

`V_T` — Terminal voltage
scalar

Terminal voltage magnitude in per-unit representation.

Data Types: `single` | `double`

`I_T` — Terminal current
scalar

Terminal current magnitude in per-unit representation.

Data Types: `single` | `double`

`V_OEL` — Overexcitation limit signal
scalar

Input from the overexcitation limiter, in per-unit representation.

Dependencies

- To ignore the input from the overexcitation limiter, set **Alternate OEL input locations** to `Unused`.
- To use the input from the overexcitation limiter at the summation point, set **Alternate OEL input locations** to `Summation point`.
- To use the input from the overexcitation limiter at the take-over point, set **Alternate OEL input locations** to `Take-over`.

Data Types: `single` | `double`

`V_UEL` — Underexcitation limit signal
scalar

Input from the underexcitation limiter, in per-unit representation.

Dependencies

- To ignore the input from the underexcitation limiter, set **Alternate UEL input locations** to `Unused`.
- To use the input from the underexcitation limiter at the summation point, set **Alternate UEL input locations** to `Summation point`.
- To use the input from the underexcitation limiter at the take-over point, set **Alternate UEL input locations** 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.

Dependencies

- To ignore the input from the stator current limiter, set **Alternate SCL input locations** 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 to prevent field overexcitation when using the take-over point, in per-unit representation.

Dependencies

- To ignore the input from the stator current limiter, set **Alternate SCL input locations** to `Unused`.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations** to `Take-over`.

Data Types: `single` | `double`

`V_SCLuel` — 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.

Dependencies

- To ignore the input from the stator current limiter, set **Alternate SCL input locations** to `Unused`.
- To use the input from the stator current limiter at the take-over point, set **Alternate SCL input locations** 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 be applied to the field circuit of the synchronous machine.

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)` — Sample time
`-1` (default) | `-1` or positive number

Block sample time. Set this to `0` to implement a continuous AC1C system. Set this to `-1` or a positive number to implement a discrete system.

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 X_C 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 R_C 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 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 excitation system stabilizer gain, K_F (pu)` — Rate feedback gain

`0.03` (default) | positive number

Rate feedback block gain for stabilization of excitation system.

`Rate feedback time constant, T_F (s)` — Rate feedback time constant

`1` (default) | positive number

Rate feedback block time constant for stabilization of excitation system.

`Maximum regulator output, V_Amax (pu)` — Regulator output upper limit

`14.5` (default) | real number

Maximum per-unit output voltage of the regulator.

`Minimum regulator output, V_Amin (pu)` — Regulator output lower limit

`-14.5` (default) | real number

Minimum per-unit output voltage of the regulator.

`Maximum exciter field voltage, E_FEmax (pu)` — Exciter voltage upper limit

`6.03` (default) | real number

Maximum per unit field voltage to be applied to exciter.

`Minimum exciter field voltage, E_FEmin (pu)` — Exciter voltage lower limit

`-5.43` (default) | real number

Minimum per unit field voltage to be applied to exciter.

`Alternate OEL input locations (V_OEL)` — OEL input location
`Unused` (default) | `Summation point` | `Take-over`

Select overexcitation limiter input location.

`Alternate UEL input locations (V_UEL)` — UEL input location
`Unused` (default) | `Summation point` | `Take-over`

Select underexcitation limiter input location.

`Alternate SCL input locations (V_SCL)` — SCL 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** inport port.
- If you select `Take-over`, use the **V_SCLoel** and **V_SCLuel** inport 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
`0.8` (default) | positive number

Time constant for 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 commutating reactance.

`Demagnetizing factor, function of exciter alternator reactances, K_D (pu)` — Demagnetization factor
`0.38` (default) | positive number

Demagnetization factor related to 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 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

First exciter saturation factor.

`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 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

Second exciter saturation factor.

Model Examples

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

Blocks

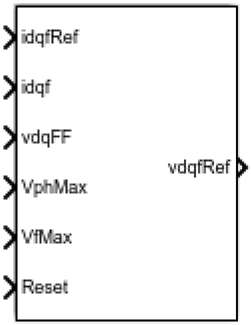
Lead-Lag (Discrete or Continuous) | PMSM Current Controller with Pre-Control | PMSM Current Reference Generator

Introduced in R2017b

SM Current Controller

Synchronous machine current controller

Library: Simscape / Power Systems / Simscape Components / Control / SM Control



Description

The SM Current Controller block implements a discrete time PI-based 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:

$$v_d^{ref} = \left(K_{p\_id} + K_{i\_id} \frac{T_s z}{z-1}\right)\left(i_d^{ref} - i_d\right) + v_{d\_FF},$$

$$v_q^{ref} = \left(K_{p\_iq} + K_{i\_iq} \frac{T_s z}{z-1}\right)\left(i_q^{ref} - i_q\right) + v_{q\_FF},$$

and

$$v_f^{ref} = \left(K_{p\_if} + K_{i\_if} \frac{T_s z}{z-1}\right)\left(i_f^{ref} - i_f\right),$$

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-cancelation block in the feedforward path. The zero cancellation transfer functions in discrete time are:

$$G_{ZC\_id}(z) = \frac{\dfrac{T_s K_{i\_id}}{K_{p\_id}}}{z + \left(\dfrac{T_s - \dfrac{K_{p\_id}}{K_{i\_id}}}{\dfrac{K_{p\_id}}{K_{i\_id}}}\right)},$$

$$G_{ZC\_iq}(z) = \frac{\dfrac{T_s K_{i\_iq}}{K_{p\_iq}}}{z + \left(\dfrac{T_s - \dfrac{K_{p\_iq}}{K_{i\_iq}}}{\dfrac{K_{p\_iq}}{K_{i\_iq}}}\right)},$$

and

$$G_{ZC\_if}(z) = \frac{\dfrac{T_s K_{i\_if}}{K_{p\_if}}}{z + \left(\dfrac{T_s - \dfrac{K_{p\_if}}{K_{i\_if}}}{\dfrac{K_{p\_if}}{K_{i\_if}}} \right)}.$$

Saturation must be imposed when the stator voltage vector exceeds the voltage phase limit V_{ph_max}:

$$\sqrt{v_d^2 + v_q^2} \le V_{ph\_max},$$

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:

$$v_1^{sat} = \min\left(\max\left(v_1^{unsat}, -V_{ph\_max}\right), V_{ph\_max}\right),$$

and

$$v_2^{sat} = \min\left(\max\left(v_2^{unsat}, -V_{2\_max}\right), V_{2\_max}\right),$$

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 $v_{2\_max} = \sqrt{\left(V_{ph\_max}\right)^2 - \left(v_1^{sat}\right)^2}$.

In the case that the direct and quadrature axes have the same priority (d-q equivalence) the constrained voltages are obtained as follows:

$$v_d^{sat} = \min\left(\max\left(v_d^{unsat}, -V_{d\_max}\right), V_{d\_max}\right),$$

and

$$v_q^{sat} = \min\left(\max\left(v_q^{unsat}, -V_{q\_max}\right), V_{q\_max}\right),$$

where

$$V_{d\_max} = \frac{V_{ph\_max}\left|v_d^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}},$$

and

$$V_{q\_max} = \frac{V_{ph\_max}\left|v_q^{unsat}\right|}{\sqrt{(v_d^{unsat})^2 + (v_q^{unsat})^2}}.$$

The constrained (saturated) field voltage v_f^{sat} is limited according to the maximum admissible value:

$$v_f^{sat} = \min\left(\max\left(v_f^{unsat}, -V_{f\_max}\right), V_{f\_max}\right),$$

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:

$$K_{i\_id} + K_{aw\_id}\left(v_d^{sat} - v_d^{unsat}\right),$$

$$K_{i\_iq} + K_{aw\_iq}\left(v_q^{sat} - v_q^{unsat}\right),$$

and

$$K_{i\_if} + K_{aw\_if}\left(v_f^{sat} - v_f^{unsat}\right),$$

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) | `-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.

`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.

Model Examples

HESM Torque Control HESM Velocity Control SM Torque Control SM Velocity Control

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.*

See Also

Blocks

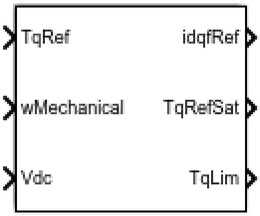
SM AC1C | SM Current Reference Generator

Introduced in R2017b

SM Current Reference Generator

Synchronous machine current reference generator

Library: Simscape / Power Systems / Simscape Components / Control / SM Control



Description

The SM Current Reference Generator block implements a current reference generator for 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:

  $$i_f^{ref} = \frac{\left|T_{ref}\, i_{f,max}\right|}{T_{max}},$$

  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:

$$i_d^{ref} = \mathrm{f}\left(n_m, T_{ref}, v_{dc}\right),$$

$$i_q^{ref} = \mathrm{g}\left(n_m, T_{ref}, v_{dc}\right),$$

and

$$i_f^{ref} = \mathrm{h}\left(n_m, T_{ref}, v_{dc}\right).$$

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.

Model Examples

HESM Torque Control HESM Velocity Control SM Torque Control SM Velocity Control

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.*

See Also

Blocks

SM AC1C | SM Current Controller

Introduced in R2017b

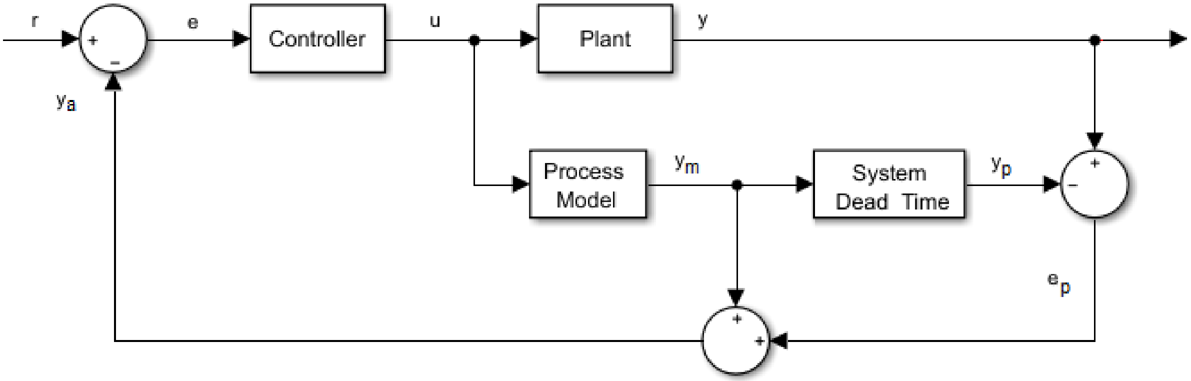
Smith Predictor Controller

Discrete-time Smith dead-time compensator

Library: Simscape / Power Systems / Simscape Components / 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:

- τ 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?” (Simulink) and “Specify Sample Time” (Simulink).

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.

See Also

Blocks

RST Controller | State-Feedback Controller

Introduced in R2017b

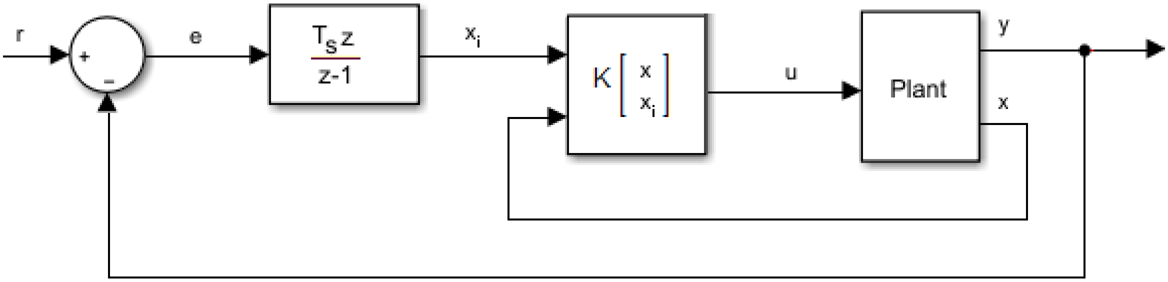
State-Feedback Controller

Discrete-time state-feedback controller with integral action

Library: Simscape / Power Systems / Simscape Components / Control / General Control

Description

The State-Feedback Controller block implements a discrete-time state-feedback controller with integral action.



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

`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.

`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?” (Simulink) and “Specify Sample Time” (Simulink).

See Also

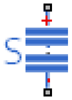
Blocks

RST Controller | Smith Predictor Controller

Introduced in R2017b

Supercapacitor

Represent an electrochemical double-layer capacitor
Library: Sources



Description

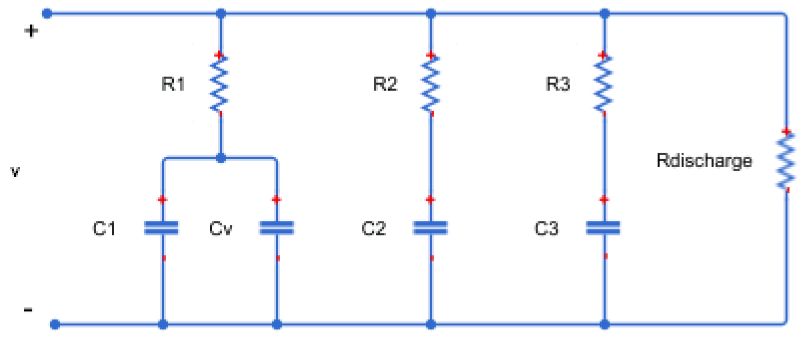
The Supercapacitor block represents an electrochemical double-layer capacitor (ELDC), which is commonly referred to as a supercapacitor or an ultracapacitor. The capacitance values for supercapacitors are orders of magnitude larger than the values for regular capacitors. Supercapacitors can provide bursts 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 nth branch.
- R_n is the resistance in the nth branch.
- V_{cn} is voltage across the capacitor in the nth 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 nth 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 1-566.

`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" (Simscape).

Model Examples

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.

See Also

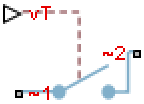
Simscape Blocks

Battery | Current Source | Voltage Source

Introduced in R2016b

Switch

Three-phase single-throw switch



Library

Switches & Breakers

Description

The Switch 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` ≤ **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` > **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**

Parameters

Closed resistance

Resistance between ports `~1` and `~2` when the switch is closed. The default value is `0.001 Ohm`.

Open conductance

Conductance between ports `~1` and `~2` when the switch is open. The default value is `1e-6 1/Ohm`.

Threshold

Threshold voltage for the control port `vT`. When the voltage is above the threshold, the switch is closed. The default value is `0 V`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

`~2`

Expandable three-phase port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Single-Phase Switch | Single-Phase Two-Way Switch | Two-Way Switch

Topics

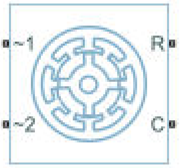
"Expand and Collapse Three-Phase Ports on a Block"
"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Switched Reluctance Machine

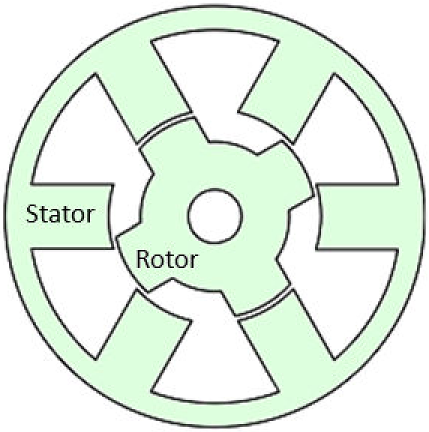
Switched reluctance machine (SRM)

Library: Simscape / Power Systems / Simscape Components / Machines / Reluctance Machines



Description

The Switched Reluctance Machine block represents a three-phase switched reluctance machine (SRM). The diagram shows the motor construction.



Equations

The rotor stroke angle for a three-phase machine is

$$\theta_{st} = \frac{2\pi}{3N_r},$$

where:

- θ_{st} is the stoke angle.
- N_r is the number of rotor poles.

The torque production capability, β, 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

$$v_{ph} = R_s i_{ph} + \frac{d\lambda_{ph}\left(i_{ph},\theta_{ph}\right)}{dt}$$

where:

- v_{ph} is the voltage per phase.
- R_s is the stator resistance per phase.
- i_{ph} is the current per phase.
- λ_{ph} is the flux linkage per phase.
- θ_{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

$$L_t\left(i_{ph},\theta_{ph}\right) = \frac{\partial\lambda_{ph}\left(i_{ph},\theta_{ph}\right)}{\partial i_{ph}},$$

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:

$$v_{ph} = R_s i_{ph} + L_t\left(i_{ph}, \theta_{ph}\right)\frac{di_{ph}}{dt} + E_{ph}.$$

Applying the co-energy formula to equations for torque,

$$T_{ph} = \frac{\partial W\left(\theta_{ph}\right)}{\partial \theta_r},$$

and energy,

$$W\left(i_{ph}, \theta_{ph}\right) = \int_0^{i_{ph}} \lambda_{ph}\left(i_{ph}, \theta_{ph}\right) di_{ph}.$$

yields an integral equation that defines the instantaneous torque per phase, that is,

$$T_{ph}\left(i_{ph}, \theta_{ph}\right) = \int_0^{i_{ph}} \frac{\partial \lambda_{ph}\left(i_{ph}, \theta_{ph}\right)}{\partial \theta_{ph}} di_{ph}.$$

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.
- ω 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,

$$\lambda_{ph}\left(i_{ph},\theta_{ph}\right).$$

For low-fidelity modeling, you can also approximate the curve using analytical techniques. One such technique [2] uses this exponential function:

$$\lambda_{ph}\left(i_{ph},\theta_{ph}\right)=\lambda_{sat}\left(1-e^{-i_{ph}f\left(\theta_{ph}\right)}\right),$$

where:

- λ_{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:

$$f\left(\theta_{ph}\right)=a+b\cos\left(N_r\theta_{ph}\right)$$

where $a > b$,

$$a=\frac{L_{\min}+L_{\max}}{2\lambda_{sat}},$$

and

$$b=\frac{L_{max}-L_{min}}{2\lambda_{sat}}.$$

Assumptions

The block assumes that a zero rotor angle corresponds to a rotor pole that is aligned perfectly with the a-phase.

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`

`~1` — Three-phase composite
electrical

Expandable three-phase port.

Data Types: `double`

`~2` — Three-phase composite
electrical

Expandable three-phase port.

Data Types: `double`

Parameters

Main

`Number of rotor poles` — Rotor pole
`4` (default) | integer

Number of pole pairs on the rotor.

`Stator resistance per phase` — Resistance

`3 Ohm` (default)

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)

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)

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)

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)

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, 45, 90] deg` (default)

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)

Flux linkage matrix that defines the flux linkage curve family.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify flux characteristic`.

Mechanical

`Rotor inertia` — *J*
`0.1 kg*m^2` (default)

Moment of inertia of the rotor.

`Rotor Damping` — Define B_m
`0 N*m/(rad/s)` (default)

Damping of the rotor.

Initial Conditions

`Initial currents, [i_a i_b i_c]` — Currents at simulation start time
`l[0, 0, 0] A` (default) | vector

Initial a-, b-, and c-phase currents.

`Initial rotor angle` — Rotor angle at simulation start time
`0 deg` (default) | `0-360 deg`

The initial angle of the rotor.

`Initial rotor speed` — Rotor speed at simulation start time
`0 rpm` (default)

Initial speed of the rotor. If the rotor inertia, J, is zero, the initial speed of the rotor is zero rpm and the initial rotor speed is ignored.

References

[1] Boldea, I. and S. A. Nasar. *Electric Drives, Second Edition*. CRC Press, New York, 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.

See Also

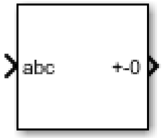
Brushless DC Motor | Permanent Magnet Synchronous Motor | Synchronous Machine Field Circuit (SI) | Synchronous Machine Field Circuit (pu) | Synchronous Machine Measurement | Synchronous Reluctance Machine

Introduced in R2017b

Symmetrical-Components Transform

Implement *abc* to *+-0* transform

Library: Simscape / Power Systems / Simscape Components / 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

$$\begin{bmatrix} V_{c+} \\ V_{c-} \\ V_{c0} \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_{a+} \\ V_{a-} \\ V_{a0} \end{bmatrix}.$$

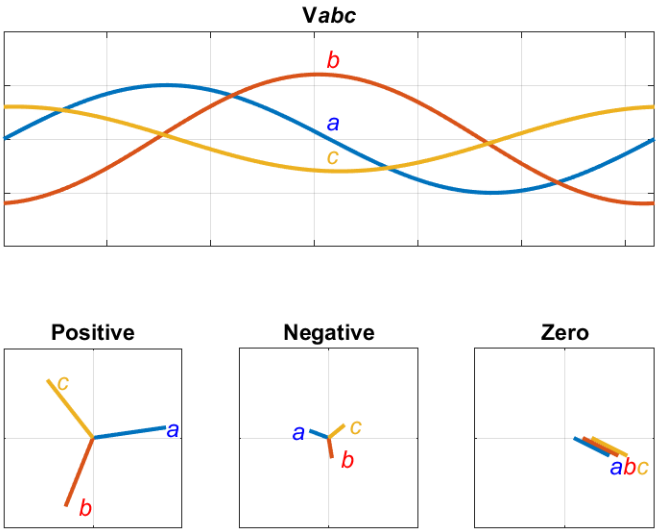
Operating Principle

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.

References

[1] Anderson, P. M. *Analysis of Faulted Power Systems*. Hoboken, NJ: Wiley-IEEE Press, 1995.

See Also

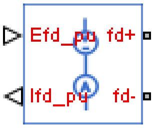
Blocks

Clarke Transform | Clarke to Park Angle Transform | Inverse Symmetrical-Components Transform | Inverse Clarke Transform | Inverse Park Transform | Park to Clarke Angle Transform

Introduced in R2017b

Synchronous Machine Field Circuit (pu)

Synchronous machine field circuit per-unit voltage supply and current measurement



Library

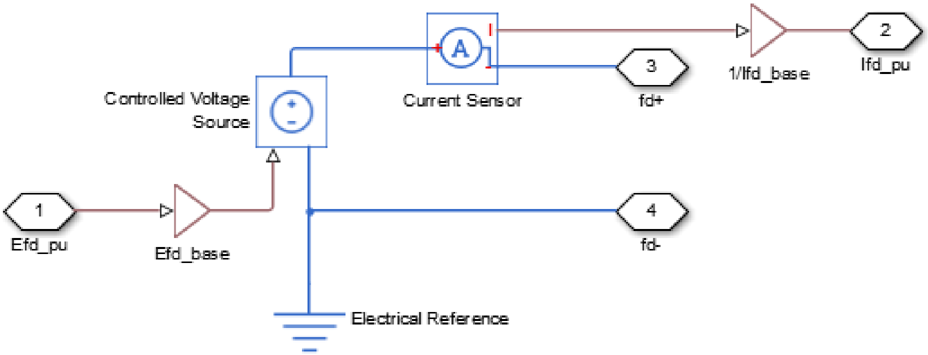
Machines

Description

The Synchronous Machine Field Circuit (pu) block applies specified voltage to, and measures current through, the field circuit of the synchronous machine that it is connected to. It 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 per-unit bases are the nonreciprocal per-unit system, Efd and Ifd, rather than the reciprocal per-unit system, efd and ifd.

The figure shows the schematic for the Synchronous Machine Field Circuit (pu) block.



Parameters

- "Main Tab" on page 1-584
- "Machine Parameters Tab" on page 1-584

Main Tab

Rated apparent power

Rated apparent power of the connected machine. The default value is `555e6 V*A`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power of the connected machine is quoted. The default value is `60 Hz`.

Specify field circuit input required to produce rated terminal voltage at no load by

Choose between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This value is used to calculate the per-unit bases for the field circuit (nonreciprocal per-unit system). This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1300 A`.

Field circuit voltage

This value is used to calculate the per-unit bases for the field circuit (nonreciprocal per-unit system). This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `92.95 V`.

Machine Parameters Tab

Specify parameterization by

Choose between `Fundamental Parameters` and `Standard Parameters`. The default value is `Fundamental Parameters`.

Stator d-axis mutual inductance (unsaturated), Ladu

Unsaturated stator d-axis mutual inductance. This parameter is visible only when **Specify parameterization by** is set to `Fundamental Parameters`. The default value is `1.66` pu.

Rotor field circuit resistance, Rfd

Rotor field-circuit resistance. This parameter is visible only when **Specify parameterization by** is set to `Fundamental Parameters`. The default value is `0.0006` pu.

Stator leakage reactance, Xl

Stator leakage reactance. This parameter is visible only when **Specify parameterization by** is set to `Standard Parameters`. The default value is `0.15` pu.

d-axis synchronous reactance, Xd

The d-axis synchronous reactance. This parameter is visible only when **Specify parameterization by** is set to `Standard Parameters`. The default value is `1.81` pu.

d-axis transient reactance, Xd'

The d-axis transient reactance. This parameter is visible only when **Specify parameterization by** is set to `Standard Parameters`. The default value is `0.3` pu.

Specify d-axis transient time constant by

This parameter is visible only when **Specify parameterization by** is set to `Standard Parameters`. Choose between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis transient open-circuit, Td0'

The d-axis transient open-circuit time constant. This parameter is visible only when **Specify d-axis transient time constant by** is set to `Open-circuit value`. The default value is `8 s`.

d-axis transient short-circuit, Td'

The d-axis transient short-circuit time constant. This parameter is visible only when **Specify d-axis transient time constant by** is set to `Short-circuit value`. The default value is `1.326 s`.

Ports

The block has the following ports:

`Efd_pu`

Field voltage input, per-unit

`Ifd_pu`

Field current output, per-unit

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

See Also

Synchronous Machine Salient Pole (standard) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Field Circuit (SI) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Round Rotor (fundamental)

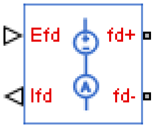
Topics

Three-Phase Synchronous Machine Control

Introduced in R2014b

Synchronous Machine Field Circuit (SI)

Synchronous machine field circuit voltage supply and current measurement in SI units



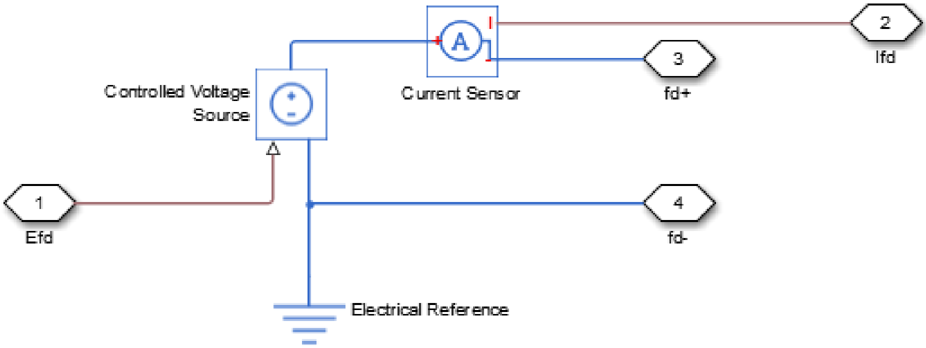
Library

Machines

Description

The Synchronous Machine Field Circuit (SI) block applies specified voltage to, and measures current through, the field circuit of the synchronous machine that it is connected to. It includes an electrical reference. The physical signal input Efd defines the voltage, in Volts, and the physical signal output Ifd provides the current, in Amperes.

The figure shows the schematic for the Synchronous Machine Field Circuit (SI) block.



Ports

The block has the following ports:

`Efd`

Field voltage input, V

`Ifd`

Field current output, A

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

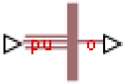
See Also

Synchronous Machine Salient Pole (standard) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Field Circuit (pu) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Round Rotor (fundamental)

Introduced in R2014b

Synchronous Machine Measurement

Per-unit measurement from synchronous machine



Library

Machines

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)`
Terminal voltage	$\sqrt{(pu\_ed^2 + pu\_eq^2)}$
Terminal current	$\sqrt{(pu\_id^2 + pu\_iq^2)}$
Power factor angle (rad)	`power_factor_angle = atan2(pu_Qt, pu_Pt)`
Power factor	`cos(power_factor_angle)`
Load angle (rad)	`load_angle(rad) = atan2(pu_ed, pu_eq)`

Parameters

Output

Per-unit measurement from synchronous machine. The default value is `Field voltage (field circuit base, Efd)`.

Ports

The block has the following ports:

`pu`

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`

`o`

Per-unit measurement output port.

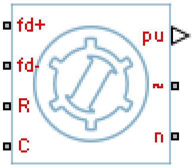
See Also

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

Introduced in R2013b

Synchronous Machine Model 2.1 (fundamental)

Synchronous machine with simplified transformation, simplified representation, and fundamental parameterization



Library

Machines / Synchronous Machine (Simplified)

Description

The Synchronous Machine Model 2.1 (fundamental) 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 parameters to define the characteristics of the machine. This block contains a dq Park transformation, so use it only for balanced operation.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.
- θ_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 ~ 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 ~ 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

$$\frac{de''_d}{dt} = \frac{\left(x_q - x''_q\right)i_q - e''_d}{T''_{q0}},$$

$$\frac{de'_q}{dt} = \frac{E_{fd} - \left(x_d - x'_d\right)i_d - e'_q}{T'_{d0}},$$

and

$$\frac{de''_q}{dt} = \frac{e'_q - \left(x'_d - x''_d\right)i_d - e''_q}{T''_{d0}},$$

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

$$T_e = e''_d i_d + e''_q i_q - \left(x''_d - x''_q\right)i_d i_q.$$

These defining equations do not describe the parameters you can set in the dialog box. To see their relationship with the equation coefficients, see [1].

Display Options

You can perform display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

- “Main Tab” on page 1-596
- “Impedances Tab” on page 1-597
- “Initial Conditions Tab” on page 1-598

Main Tab

Rated apparent power

Rated apparent power. The default value is `555e6 V*A`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Specify field circuit input required to produce rated terminal voltage at no load by

Select between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1300 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `92.95` V.

Impedances Tab

Stator d-axis mutual inductance (unsaturated), Ladu

Unsaturated stator *d*-axis mutual inductance. The default value is `1.66` pu.

Stator q-axis mutual inductance (unsaturated), Laqu

Unsaturated stator *q*-axis mutual inductance. The default value is `1.61` pu.

Stator leakage inductance, Ll

Stator leakage inductance. The default value is `0.15` pu.

Stator resistance, Ra

Stator resistance. The default value is `0.003` pu.

Rotor field circuit inductance, Lfd

Rotor field circuit inductance. The default value is `0.165` pu.

Rotor field circuit resistance, Rfd

Rotor field circuit resistance. The default value is `0.0006` pu.

Rotor d-axis damper winding 1 inductance, L1d

Rotor *d*-axis damper winding 1 inductance. The default value is `0.1713` pu.

Rotor d-axis damper winding 1 resistance, R1d

Rotor *d*-axis damper winding 1 resistance. The default value is `0.0284` pu.

Rotor q-axis damper winding 1 inductance, L1q

Rotor *q*-axis damper winding 1 inductance. The default value is `0.1066` pu.

Rotor q-axis damper winding 1 resistance, R1q

Rotor *q*-axis damper winding 1 resistance. The default value is `0.0650` pu.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and voltage states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3 V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 deg`.

Terminal active power

Initial active power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `500e6 V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0 deg`.

Initial voltage behind d-axis subtransient reactance

Initial voltage behind d-axis subtransient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Initial voltage behind q-axis transient reactance

Initial voltage behind q-axis transient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Initial voltage behind q-axis subtransient reactance

Initial voltage behind *q*-axis subtransient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal.

`fd-`

Electrical conserving port corresponding to the field winding negative terminal.

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`pu`

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- `pu_fd_Efd`
- `pu_fd_Ifd`
- `pu_torque`
- `pu_velocity`
- `pu_ed`
- `pu_eq`
- `pu_e0` — This port is provided to maintain a compatible interface for existing machine models. Its value is always zero.
- `pu_id`
- `pu_iq`
- `pu_i0` — This port is provided to maintain a compatible interface for existing machine models. Its value is always zero.

`~`

Expandable three-phase port associated with the stator windings.

`n`

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.

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] Pal, M. K. *Lecture Notes on Power System Stability*. http://www.mkpalconsulting.com/files/stabilitybook.pdf, 2007.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (standard) | Synchronous Machine Round Rotor (fundamental) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Salient Pole (standard)

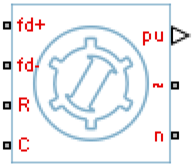
Topics

"Expand and Collapse Three-Phase Ports on a Block"
Three-Phase Synchronous Machine Control

Introduced in R2015a

Synchronous Machine Model 2.1 (standard)

Synchronous machine with simplified transformation, simplified representation, and standard parameterization



Library

Machines / Synchronous Machine (Simplified)

Description

The Synchronous Machine Model 2.1 (standard) 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 standard parameters to define the characteristics of the machine. The block converts the standard parameters to fundamental parameters. This block contains a dq Park transformation, so use it only for balanced operation.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.

- θ_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 ~ 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 ~ 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

$$\frac{de_d''}{dt} = \frac{\left(x_q - x_q''\right)i_q - e_d''}{T_{q0}''},$$

$$\frac{de_q'}{dt} = \frac{E_{fd} - \left(x_d - x_d'\right)i_d - e_q'}{T_{d0}'},$$

and

$$\frac{de_q''}{dt} = \frac{e_q' - \left(x_d' - x_d''\right)i_d - e_q''}{T_{d0}''},$$

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

$$T_e = e_d'' i_d + e_q'' i_q - \left(x_d'' - x_q''\right) i_d i_q.$$

These defining equations do not describe the short-circuit time constants you can set in the dialog box. To see their relationship with the equation coefficients, see [1].

Display Options

You can perform display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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.

Parameters

- “Main Tab” on page 1-604
- “Impedances Tab” on page 1-605
- “Time Constants Tab” on page 1-606
- “Initial Conditions Tab” on page 1-607

Main Tab

Rated apparent power

Rated apparent power. The default value is `555e6 VA`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Specify field circuit input required to produce rated terminal voltage at no load by

Select between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1300 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `92.95` V.

Impedances Tab

Stator resistance, Ra

Stator resistance. The default value is `0.003` pu.

Stator leakage reactance, Xl

Stator leakage reactance. The default value is `0.15` pu.

d-axis synchronous reactance, Xd

The *d*-axis synchronous reactance. The default value is `1.81` pu.

q-axis synchronous reactance, Xq

The *q*-axis synchronous reactance. The default value is `1.76` pu.

d-axis transient reactance, Xd'

The *d*-axis transient reactance. The default value is `0.3` pu.

d-axis subtransient reactance, Xd''

The *d*-axis subtransient reactance. The default value is `0.23` pu.

q-axis subtransient reactance, Xq''

The *q*-axis subtransient reactance. The default value is `0.25` pu.

Time Constants Tab

Specify d-axis transient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis transient open-circuit, Td0'

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`. The default value is `8 s`.

d-axis transient short-circuit, Td'

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`. The default value is `1.326 s`.

Specify d-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis subtransient open-circuit, Td0''

The *d*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.03 s`.

d-axis subtransient short-circuit, Td''

The *d*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.023 s`.

Specify q-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

q-axis subtransient open-circuit, Tq0''

The *q*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.07 s`.

q-axis subtransient short-circuit, Tq"

The *q*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.0269 s`.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and voltage states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3 V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 deg`.

Terminal active power

Initial active power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `500e6 V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0 deg`.

Initial voltage behind d-axis subtransient reactance

Initial voltage behind *d*-axis subtransient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Initial voltage behind q-axis transient reactance

Initial voltage behind *q*-axis transient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Initial voltage behind q-axis subtransient reactance

Initial voltage behind *q*-axis subtransient reactance. This parameter is visible only when you set **Specify initialization by** to `Mechanical and voltage states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal.

`fd-`

Electrical conserving port corresponding to the field winding negative terminal.

`R`

Mechanical rotational conserving port associated with the machine rotor.

`C`

Mechanical rotational conserving port associated with the machine case.

`pu`

Physical signal vector port associated with the machine per-unit measurements. The vector elements are:

- `pu_fd_Efd`
- `pu_fd_Ifd`
- `pu_torque`
- `pu_velocity`

- `pu_ed`
- `pu_eq`
- `pu_e0` — This port is provided to maintain a compatible interface for existing machine models. Its value is always zero.
- `pu_id`
- `pu_iq`
- `pu_i0` — This port is provided to maintain a compatible interface for existing machine models. Its value is always zero.

`~`

Expandable three-phase port associated with the stator windings.

`n`

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.

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] Pal, M. K. *Lecture Notes on Power System Stability*. http://www.mkpalconsulting.com/files/stabilitybook.pdf, 2007.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (fundamental) | Synchronous Machine Round Rotor (fundamental) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Salient Pole (standard)

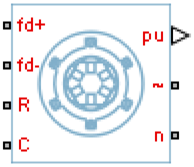
Topics

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

Introduced in R2015a

Synchronous Machine Round Rotor (fundamental)

Round-rotor synchronous machine with fundamental parameterization



Library

Machines / Synchronous Machine (Round Rotor)

Description

The Synchronous Machine Round Rotor (fundamental) block models a round-rotor synchronous machine using fundamental parameters.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.
- θ_r is the rotor angle.

Park's transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. Park's transformation is defined by

$$P_s = \frac{2}{3}\begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_d}{\mathrm{d}t} - \Psi_q\omega_r - R_a i_d,$$

$$e_q = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_q}{\mathrm{d}t} + \Psi_d\omega_r - R_a i_q,$$

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 ~ to neutral port n.
- ω_{base} is the per-unit base electrical speed.
- ψ_d, ψ_q, and ψ_0 are the d-axis, q-axis, and zero-sequence stator flux linkages.
- ω_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 ~ to port 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.
- ψ_{fd}, ψ_{1d}, ψ_{1q}, and ψ_{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),

$K_s = f\left(\psi_{at}\right)$ (If saturation is enabled),

$$L_{ad} = K_s * L_{adu},$$

and

$$L_{aq} = K_s * L_{aqu},$$

where:

- ψ_{ad} is the d-axis air-gap or mutual flux linkage.
- ψ_{aq} is the q-axis air-gap or mutual flux linkage.
- ψ_{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

$$\Psi_0 = -L_0 i_0,$$

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}, L_{11d}, L_{11q}, and L_{22q} are the self-inductances of the rotor field circuit, d-axis damper winding 1, q-axis damper winding 1 and q-axis damper winding 2. L_{f1d} is the rotor field circuit and d-axis damper winding 1 mutual inductance. They are defined by the following equations.

  $$L_{ffd} = L_{ad} + L_{fd}$$

  $$L_{f1d} = L_{ffd} - L_{fd}$$

  $$L_{11d} = L_{f1d} + L_{1d}$$

  $$L_{11q} = L_{aq} + L_{1q}$$

  $$L_{22q} = L_{aq} + L_{2q}$$

These 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 **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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), Ladu** 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, ψ_{at}, characteristic.
- **Plot Saturation Factor (pu)** plots saturation factor,K_s, versus magnetic flux linkage, ψ_{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), Ladu**
    - **Per-unit field current saturation data, ifd**
    - **Per-unit air-gap voltage saturation data, Vag**

Parameters

- “Main Tab” on page 1-618
- “Impedances Tab” on page 1-619

- “Saturation Tab” on page 1-620
- “Initial Conditions Tab” on page 1-620

Main Tab

Rated apparent power

Rated apparent power. The default value is `555e6 V*A`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Specify field circuit input required to produce rated terminal voltage at no load by

Select between

- `Field circuit voltage`
- `Field circuit current`

The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1300 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `92.95` V.

Impedances Tab

Stator d-axis mutual inductance (unsaturated), Ladu

Unsaturated stator *d*-axis mutual inductance. If **Magnetic saturation representation** is set to `None`, this is equivalent to the stator *d*-axis mutual inductance. The default value is `1.66` pu.

Stator q-axis mutual inductance (unsaturated), Laqu

Unsaturated stator *q*-axis mutual inductance. If **Magnetic saturation representation** is set to `None`, this is equivalent to the stator *q*-axis mutual inductance. The default value is `1.61` pu.

Stator zero-sequence inductance, L0

Stator zero-sequence inductance. The default value is `0.15` pu.

Stator leakage inductance, Ll

Stator leakage inductance. The default value is `0.15` pu.

Stator resistance, Ra

Stator resistance. The default value is `0.003` pu.

Rotor field circuit inductance, Lfd

Rotor field circuit inductance. The default value is `0.165` pu.

Rotor field circuit resistance, Rfd

Rotor field circuit resistance. The default value is `0.0006` pu.

Rotor d-axis damper winding 1 inductance, L1d

Rotor *d*-axis damper winding 1 inductance. The default value is `0.1713` pu.

Rotor d-axis damper winding 1 resistance, R1d

Rotor *d*-axis damper winding 1 resistance. The default value is `0.0284` pu.

Rotor q-axis damper winding 1 inductance, L1q

Rotor *q*-axis damper winding 1 inductance. The default value is `0.7252` pu.

Rotor q-axis damper winding 1 resistance, R1q

Rotor *q*-axis damper winding 1 resistance. The default value is `0.00619` pu.

Rotor q-axis damper winding 2 inductance, L2q

Rotor *q*-axis damper winding 2 inductance. The default value is `0.125` pu.

Rotor q-axis damper winding 2 resistance, R2q

Rotor q-axis damper winding 2 resistance. The default value is `0.02368` pu.

Saturation Tab

Magnetic saturation representation

Block magnetic saturation representation. Options are:

- `None`
- `Per-unit open-circuit lookup table (Vag versus ifd)`

The default value is `None`.

Per-unit field current saturation data, ifd

The field current, i_{fd}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.48, 0.76, 1.38, 1.79]` pu.

Per-unit air-gap voltage saturation data, Vag

The air-gap voltage, V_{ag}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.80, 1.08, 1.31, 1.40]` pu.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and magnetic states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3 V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0` `deg`.

Terminal active power

Initial active power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `500e6` `V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0` `V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` `deg`.

Initial stator d-axis magnetic flux linkage

Stator *d*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Stator *q*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Zero-sequence initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial field circuit magnetic flux linkage

Field circuit initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial d-axis damper winding 1 magnetic flux linkage

The *d*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 1 magnetic flux linkage

The *q*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 2 magnetic flux linkage

The *q*-axis damper winding 2 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

`pu`

Physical signal vector port associated with the machine per-unit measurements. 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`

`~`

Expandable three-phase port associated with the stator windings

`n`

Electrical conserving port associated with the neutral point of the wye winding configuration

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.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (fundamental) | Synchronous Machine Model 2.1 (standard) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Salient Pole (standard)

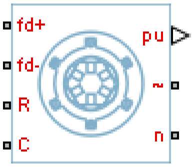
Topics

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

Introduced in R2013b

Synchronous Machine Round Rotor (standard)

Round-rotor synchronous machine with standard parameterization



Library

Machines / Synchronous Machine (Round Rotor)

Description

The Synchronous Machine Round Rotor (standard) block models a round-rotor synchronous machine using standard parameters.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.
- θ_r is the rotor angle.

Park's transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. Park's transformation is defined by

$$P_s = \frac{2}{3}\begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_d}{\mathrm{d}t} - \Psi_q\omega_r - R_a i_d,$$

$$e_q = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_q}{\mathrm{d}t} + \Psi_d\omega_r - R_a i_q,$$

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

  $$\begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = P_s \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}.$$

  where v_a, v_b, and v_c are the stator voltages measured from port ~ to neutral port n.
- ω_{base} is the per-unit base electrical speed.
- ψ_d, ψ_q, and ψ_0 are the d-axis, q-axis, and zero-sequence stator flux linkages.
- ω_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

  $$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = P_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}.$$

  where i_a, i_b, and i_c are the stator currents flowing from port ~ to port 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.
- ψ_{fd}, ψ_{1d}, ψ_{1q}, and ψ_{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:

- ψ_{ad} is the d-axis air-gap or mutual flux linkage.
- ψ_{aq} is the q-axis air-gap or mutual flux linkage.
- ψ_{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

$$\Psi_0 = -L_0 i_0,$$

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}, L_{11d}, L_{11q}, and L_{22q} are the self-inductances of the rotor field circuit, d-axis damper winding 1, q-axis damper winding 1, and q-axis damper winding 2. L_{f1d} is the rotor field circuit and d-axis damper winding 1 mutual inductance. They are defined by the following equations.

  $$L_{ffd} = L_{ad} + L_{fd}$$

  $$L_{f1d} = L_{ffd} - L_{fd}$$

  $$L_{11d} = L_{f1d} + L_{1d}$$

  $$L_{11q} = L_{aq} + L_{1q}$$

  $$L_{22q} = L_{aq} + L_{2q}$$

These 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

$$T_e = \Psi_d i_q - \Psi_q i_d \,.$$

These defining equations do not describe the time constants you can set in the dialog box. To see their relationship with the equation coefficients, see [1].

Plotting and Display Options

You can perform plotting and display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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), Ladu** 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, ψ_{at}, characteristic.
- **Plot Saturation Factor (pu)** plots saturation factor,K_s, versus magnetic flux linkage, ψ_{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), Ladu**
  - **Per-unit field current saturation data, ifd**
  - **Per-unit air-gap voltage saturation data, Vag**

Parameters

Main Tab

Rated apparent power

Rated apparent power. The default value is `555e6 V*A`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `1`.

Specify field circuit input required to produce rated terminal voltage at no load by

Choose between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1300 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `92.95` V.

Impedances Tab

Stator resistance, Ra

Stator resistance. The default value is `0.003` pu.

Stator leakage reactance, Xl

Stator leakage reactance. The default value is `0.15` pu.

d-axis synchronous reactance, Xd

The *d*-axis synchronous reactance. The default value is `1.81` pu.

q-axis synchronous reactance, Xq

The *q*-axis synchronous reactance. The default value is `1.76` pu.

zero-sequence reactance, X0

The zero-sequence reactance. The default value is `0.15` pu.

d-axis transient reactance, Xd'

The *d*-axis transient reactance. The default value is `0.3` pu.

q-axis transient reactance, Xq'

The *q*-axis transient reactance. The default value is `0` pu.

d-axis subtransient reactance, Xd''

The *d*-axis subtransient reactance. The default value is `0.23` pu.

q-axis subtransient reactance, Xq''

The *q*-axis subtransient reactance. The default value is `0.25` pu.

Time Constants Tab

Specify d-axis transient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis transient open-circuit, Td0'

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`. The default value is `8 s`.

d-axis transient short-circuit, Td'

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`. The default value is `1.326 s`.

Specify d-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis subtransient open-circuit, Td0''

The *d*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.03 s`.

d-axis subtransient short-circuit, Td''

The *d*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.023 s`.

Specify q-axis transient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

q-axis transient open-circuit, Tq0'

The *q*-axis transient open-circuit time constant. This parameter is visible only when **Specify q-axis transient time constant** is set to `Open-circuit value`. The default value is `1 s`.

q-axis transient short-circuit, Tq'

The *q*-axis transient short-circuit time constant. This parameter is visible only when **Specify q-axis transient time constant** is set to `Short-circuit value`. The default value is `0.3693 s`.

Specify q-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

q-axis subtransient open-circuit, Tq0''

The *q*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.07 s`.

q-axis subtransient short-circuit, Tq"

The *q*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.0269 s`.

Saturation Tab

Magnetic saturation representation

Block magnetic saturation representation. Options are:

- `None`
- `Per-unit open-circuit lookup table (Vag versus ifd)`

The default value is `None`.

Per-unit field current saturation data, ifd

The field current, i_{fd}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.48, 0.76, 1.38, 1.79]` pu.

Per-unit air-gap voltage saturation data, Vag

The air-gap voltage, V_{ag}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.80, 1.08, 1.31, 1.40]` pu.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and magnetic states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3 V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 deg`.

Terminal active power

Initial active power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `500e6 V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Stator d-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Stator q-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Zero-sequence initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial field circuit magnetic flux linkage

Field circuit initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial d-axis damper winding 1 magnetic flux linkage

The *d*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 1 magnetic flux linkage

The *q*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 2 magnetic flux linkage

The *q*-axis damper winding 2 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

`pu`

Physical signal vector port associated with the machine per-unit measurements. 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`

`~`

Expandable three-phase port associated with the stator windings

`n`

Electrical conserving port associated with the neutral point of the wye winding configuration

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.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (fundamental) | Synchronous Machine Model 2.1 (standard) | Synchronous Machine Round Rotor (fundamental) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Salient Pole (standard)

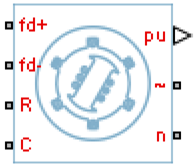
Topics

"Expand and Collapse Three-Phase Ports on a Block"
Three-Phase Synchronous Machine Control

Introduced in R2013b

Synchronous Machine Salient Pole (fundamental)

Salient-pole synchronous machine with fundamental parameterization



Library

Machines / Synchronous Machine (Salient Pole)

Description

The Synchronous Machine Salient Pole (fundamental) block models a salient-pole synchronous machine using fundamental parameters.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.
- θ_r is the rotor angle.

Park's transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. Park's transformation is defined by

$$P_s = \frac{2}{3}\begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_d}{\mathrm{d}t} - \Psi_q\omega_r - R_a i_d,$$

$$e_q = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_q}{\mathrm{d}t} + \Psi_d\omega_r - R_a i_q,$$

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 ~ to neutral port n.
- ω_{base} is the per-unit base electrical speed.
- ψ_d, ψ_q, and ψ_0 are the d-axis, q-axis, and zero-sequence stator flux linkages.
- ω_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 ~ to port 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.
- ψ_{fd}, ψ_{1d}, and ψ_{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

$$\psi_{ad} = \psi_d + L_l i_d,$$

$$\psi_{aq} = \psi_q + L_l i_q,$$

$$\psi_{at} = \sqrt{\psi_{ad}^2 + \psi_{aq}^2},$$

$K_s = 1$ (If saturation is disabled),

$K_s = f(\psi_{at})$ (If saturation is enabled),

and

$$L_{ad} = K_s * L_{adu},$$

where:

- ψ_{ad} is the d-axis air-gap or mutual flux linkage.
- ψ_{aq} is the q-axis air-gap or mutual flux linkage.
- ψ_{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

$$\Psi_0 = -L_0 i_0,$$

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}, L_{11d}, and L_{11q} are the self-inductances of the rotor field circuit, d-axis damper winding 1, and q-axis damper winding 1. L_{f1d} is the rotor field circuit and d-axis damper winding 1 mutual inductance. They are defined by the following equations.

  $$L_{ffd} = L_{ad} + L_{fd}$$

  $$L_{f1d} = L_{ffd} - L_{fd}$$

  $$L_{11d} = L_{f1d} + L_{1d}$$

  $$L_{11q} = L_{aq} + L_{1q}$$

These 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 **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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), Ladu** 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, ψ_{at}, characteristic.
- **Plot Saturation Factor (pu)** plots saturation factor,K_s, versus magnetic flux linkage, ψ_{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), Ladu**
    - **Per-unit field current saturation data, ifd**
    - **Per-unit air-gap voltage saturation data, Vag**

Parameters

- “Main Tab” on page 1-645
- “Impedances Tab” on page 1-646
- “Saturation Tab” on page 1-647
- “Initial Conditions Tab” on page 1-648

Main Tab

Rated apparent power

Rated apparent power. The default value is `300e6 V*A`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `10`.

Specify field circuit input required to produce rated terminal voltage at no load by

Choose between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1000 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `216.54` V.

Impedances Tab

Stator d-axis mutual inductance (unsaturated), Ladu

Unsaturated stator d-axis mutual inductance, L_{adu}. If **Magnetic saturation representation** is set to `None`, this is equivalent to the stator d-axis mutual inductance, L_{ad}. The default value is `0.9` pu.

Stator q-axis mutual inductance, Laq

Stator q-axis mutual inductance, Laq. The default value is `0.55` pu.

Stator zero-sequence inductance, L0

Stator zero-sequence inductance, L0. The default value is `0.15` pu.

Stator leakage inductance, Ll

Stator leakage inductance. The default value is `0.15` pu.

Stator resistance, Ra

Stator resistance. The default value is `0.011` pu.

Rotor field circuit inductance, Lfd

Rotor field circuit inductance. The default value is `0.2571` pu.

Rotor field circuit resistance, Rfd

Rotor field circuit resistance. The default value is `0.0006` pu.

Rotor d-axis damper winding 1 inductance, L1d

Rotor *d*-axis damper winding 1 inductance. The default value is `0.2` pu.

Rotor d-axis damper winding 1 resistance, R1d

Rotor *d*-axis damper winding 1 resistance. The default value is `0.0354` pu.

Rotor q-axis damper winding 1 inductance, L1q

Rotor *q*-axis damper winding 1 inductance. The default value is `0.2567` pu.

Rotor q-axis damper winding 1 resistance, R1q

Rotor *q*-axis damper winding 1 resistance. The default value is `0.0428` pu.

Saturation Tab

Magnetic saturation representation

Block magnetic saturation representation. Options are:

- `None`
- `Per-unit open-circuit lookup table (Vag versus ifd)`

The default value is `None`.

Per-unit field current saturation data, ifd

The field current, i_{fd}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.48, 0.76, 1.38, 1.79]` pu.

Per-unit air-gap voltage saturation data, Vag

The air-gap voltage, V_{ag}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00 0.43 0.59 0.71 0.76]` pu.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and magnetic states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3 V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 deg`.

Terminal active power

Initial active power. This parameter is visible only when **Specify initialization by** is set to `Electrical power and voltage output`. The default value is `270e6 V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0 V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0 deg`.

Initial stator d-axis magnetic flux linkage

Stator *d*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Stator *q*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Zero-sequence initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial field circuit magnetic flux linkage

Field circuit initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial d-axis damper winding 1 magnetic flux linkage

The d-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 1 magnetic flux linkage

The q-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 2 magnetic flux linkage

The q-axis damper winding 2 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

`pu`

Physical signal vector port associated with the machine per-unit measurements. 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`

`~`

Expandable three-phase port associated with the stator windings

`n`

Electrical conserving port associated with the neutral point of the wye winding configuration

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.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (fundamental) | Synchronous Machine Model 2.1 (standard) | Synchronous Machine Round Rotor (fundamental) | Synchronous Machine Round Rotor (standard) | Synchronous Machine Salient Pole (standard)

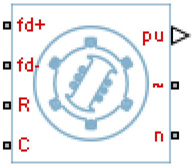
Topics

"Expand and Collapse Three-Phase Ports on a Block"
Three-Phase Synchronous Machine Control

Introduced in R2013b

Synchronous Machine Salient Pole (standard)

Salient-pole synchronous machine with standard parameterization



Library

Machines / Synchronous Machine (Salient Pole)

Description

The Synchronous Machine Salient Pole (standard) block models a salient-pole synchronous machine n using standard parameters.

Electrical Defining Equations

The synchronous machine equations are expressed with respect to a rotating reference frame defined by the equation

$$\theta_e(t) = N\theta_r(t),$$

where:

- θ_e is the electrical angle.
- N is the number of pole pairs.
- θ_r is the rotor angle.

Park's transformation maps the synchronous machine equations to the rotating reference frame with respect to the electrical angle. Park's transformation is defined by

$$P_s = \frac{2}{3}\begin{bmatrix} \cos\theta_e & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin\theta_e & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Park's transformation is used to define the per-unit synchronous machine equations. The stator voltage equations are defined by

$$e_d = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_d}{\mathrm{d}t} - \Psi_q\omega_r - R_a i_d\,,$$

$$e_q = \frac{1}{\omega_{base}}\frac{\mathrm{d}\psi_q}{\mathrm{d}t} + \Psi_d\omega_r - R_a i_q\,,$$

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 ~ to neutral port n.
- ω_{base} is the per-unit base electrical speed.
- ψ_d, ψ_q, and ψ_0 are the d-axis, q-axis, and zero-sequence stator flux linkages.
- ω_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 ~ to port 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.
- ψ_{fd}, ψ_{1d}, and ψ_{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

$$\psi_{ad} = \psi_d + L_l i_d,$$

$$\psi_{aq} = \psi_q + L_l i_q,$$

$$\psi_{at} = \sqrt{\psi_{ad}^2 + \psi_{aq}^2},$$

$K_s = 1$ (If saturation is disabled),

$K_s = f\left(\psi_{at}\right)$ (If saturation is enabled),

and

$$L_{ad} = K_s * L_{adu},$$

where:

- ψ_{ad} is the d-axis air-gap or mutual flux linkage.
- ψ_{aq} is the q-axis air-gap or mutual flux linkage.
- ψ_{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

$$\Psi_0 = -L_0 i_0,$$

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}, L_{11d}, and L_{11q} are the self-inductances of the rotor field circuit, d-axis damper winding 1, and q-axis damper winding 1. L_{f1d} is the rotor field circuit and d-axis damper winding 1 mutual inductance. They are defined by the following equations.

  $$L_{ffd} = L_{ad} + L_{fd}$$

  $$L_{f1d} = L_{ffd} - L_{fd}$$

  $$L_{11d} = L_{f1d} + L_{1d}$$

  $$L_{11q} = L_{aq} + L_{1q}$$

These 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

$$T_e = \Psi_d i_q - \Psi_q i_d .$$

These defining equations do not describe the time constants you can set in the dialog box. To see their relationship with the equation coefficients, see [1].

Plotting and Display Options

You can perform plotting and display actions using the **Power Systems** menu on the block context menu.

Right-click the block and, from the **Power Systems** 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), Ladu** 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, ψ_{at}, characteristic.
- **Plot Saturation Factor (pu)** plots saturation factor,K_s, versus magnetic flux linkage, ψ_{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), Ladu**
  - **Per-unit field current saturation data, ifd**
  - **Per-unit air-gap voltage saturation data, Vag**

Parameters

- “Main Tab” on page 1-658
- “Impedances Tab” on page 1-659
- “Time Constants Tab” on page 1-660
- “Saturation Tab” on page 1-661
- “Initial Conditions Tab” on page 1-662

Main Tab

Rated apparent power

Rated apparent power. The default value is `300e6 V*A`.

Rated voltage

RMS rated line-line voltage. The default value is `24e3 V`.

Rated electrical frequency

Nominal electrical frequency at which rated apparent power is quoted. The default value is `60 Hz`.

Number of pole pairs

Number of machine pole pairs. The default value is `10`.

Specify field circuit input required to produce rated terminal voltage at no load by

Choose between `Field circuit voltage` and `Field circuit current`. The default value is `Field circuit current`.

Field circuit current

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit current`. The default value is `1000 A`.

Field circuit voltage

This parameter is visible only when **Specify field circuit input required to produce rated terminal voltage at no load by** is set to `Field circuit voltage`. The default value is `216.54 V`.

Impedances Tab

Stator resistance, Ra

Stator resistance. The default value is `0.011` pu.

Stator leakage reactance, Xl

Stator leakage reactance. The default value is `0.15` pu.

d-axis synchronous reactance, Xd

The *d*-axis synchronous reactance. The default value is `1.05` pu.

q-axis synchronous reactance, Xq

The *q*-axis synchronous reactance. The default value is `0.7` pu.

zero-sequence reactance, X0

The zero-sequence reactance. The default value is `0.15` pu.

d-axis transient reactance, Xd'

The *d*-axis transient reactance. The default value is `0.35` pu.

d-axis subtransient reactance, Xd''

The *d*-axis subtransient reactance. The default value is `0.25` pu.

q-axis subtransient reactance, Xq''

The *q*-axis subtransient reactance. The default value is `0.325` pu.

Time Constants Tab

Specify d-axis transient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis transient open-circuit, Td0'

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`. The default value is `5.25 s`.

d-axis transient short-circuit, Td'

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`. The default value is `1.75 s`.

Specify d-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

d-axis subtransient open-circuit, Td0''

The *d*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.03 s`.

d-axis subtransient short-circuit, Td''

The *d*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify d-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.0214 s`.

Specify q-axis subtransient time constant

Select between `Open-circuit value` and `Short-circuit value`. The default value is `Open-circuit value`.

q-axis subtransient open-circuit, Tq0''

The *q*-axis subtransient open-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Open-circuit value`. The default value is `0.05 s`.

q-axis subtransient short-circuit, Tq''

The *q*-axis subtransient short-circuit time constant. This parameter is visible only when **Specify q-axis subtransient time constant** is set to `Short-circuit value`. The default value is `0.0232 s`.

Saturation Tab

Magnetic saturation representation

Block magnetic saturation representation. Options are:

- `None`
- `Per-unit open-circuit lookup table (Vag versus ifd)`

The default value is `None`.

Per-unit field current saturation data, ifd

The field current, i_{fd}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00, 0.48, 0.76, 1.38, 1.79]` pu.

Per-unit air-gap voltage saturation data, Vag

The air-gap voltage, V_{ag}, data populates the air-gap voltage, V_{ag}, versus field current, i_{fd}, lookup table. This parameter is only visible when you set **Magnetic saturation representation** to `Per-unit open-circuit lookup table (Vag versus ifd)`. This parameter must contain a vector with at least five elements. The default value is `[0.00 0.43 0.59 0.71 0.76]` pu.

Initial Conditions Tab

Specify initialization by

Select between `Electrical power and voltage output` and `Mechanical and magnetic states`. The default value is `Electrical power and voltage output`.

Terminal voltage magnitude

Initial RMS line-line voltage. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `24e3` `V`.

Terminal voltage angle

Initial voltage angle. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0` `deg`.

Terminal active power

Initial active power. This parameter is visible only when **Specify initialization by** is set to `Electrical power and voltage output`. The default value is `270e6` `V*A`.

Terminal reactive power

Initial reactive power. This parameter is visible only when you set **Specify initialization by** to `Electrical power and voltage output`. The default value is `0` `V*A`.

Initial rotor angle

Initial rotor angle. During steady-state operation, set this parameter to the sum of the load angle and required terminal voltage offset. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` `deg`.

Initial stator d-axis magnetic flux linkage

Stator *d*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator q-axis magnetic flux linkage

Stator *q*-axis initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial stator zero-sequence magnetic flux linkage

Zero-sequence initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial field circuit magnetic flux linkage

Field circuit initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial d-axis damper winding 1 magnetic flux linkage

The *d*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 1 magnetic flux linkage

The *q*-axis damper winding 1 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Initial q-axis damper winding 2 magnetic flux linkage

The *q*-axis damper winding 2 initial flux linkage. This parameter is visible only when you set **Specify initialization by** to `Mechanical and magnetic states`. The default value is `0` pu.

Ports

The block has the following ports:

`fd+`

Electrical conserving port corresponding to the field winding positive terminal

`fd-`

Electrical conserving port corresponding to the field winding negative terminal

`R`

Mechanical rotational conserving port associated with the machine rotor

`C`

Mechanical rotational conserving port associated with the machine case

`pu`

Physical signal vector port associated with the machine per-unit measurements. 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`

`~`

Expandable three-phase port associated with the stator windings

`n`

Electrical conserving port associated with the neutral point of the wye winding configuration

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.

See Also

Synchronous Machine Measurement | Synchronous Machine Model 2.1 (fundamental) | Synchronous Machine Model 2.1 (standard) | Synchronous Machine Round Rotor (fundamental) | Synchronous Machine Salient Pole (fundamental) | Synchronous Machine Salient Pole (fundamental)

Topics

"Expand and Collapse Three-Phase Ports on a Block"
Three-Phase Synchronous Machine Control

Introduced in R2013b

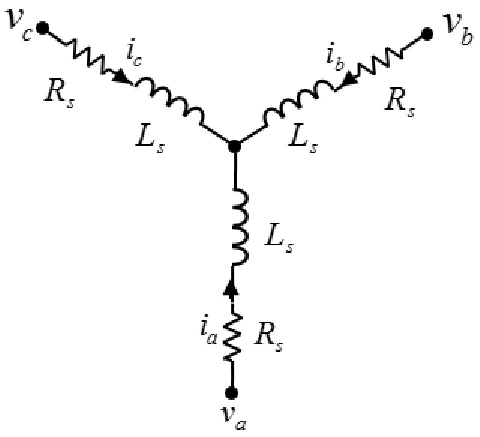
Synchronous Reluctance Machine

Synchronous reluctance machine with sinusoidal flux distribution

Library: Simscape / Power Systems / Simscape Components / Machines / Reluctance Machines

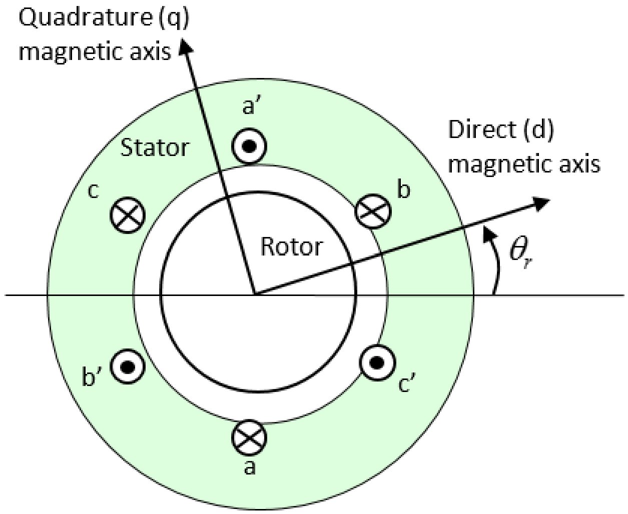
Description

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 θ_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

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \dfrac{d\psi_a}{dt} \\ \dfrac{d\psi_b}{dt} \\ \dfrac{d\psi_c}{dt} \end{bmatrix},$$

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.
- ψ_a, ψ_b, and ψ_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.

The inductances in the stator windings are functions of rotor electrical angle and are defined as

$$L_{aa} = L_s + L_m \cos(2\theta_r),$$

$$L_{bb} = L_s + L_m \cos\left(2\left(\theta_r - \frac{2\pi}{3}\right)\right),$$

$$L_{cc} = L_s + L_m \cos\left(2\left(\theta_r + \frac{2\pi}{3}\right)\right),$$

$$L_{ab} = L_{ba} = -M_s - L_m \cos\left(\theta_r + \frac{\pi}{6}\right),$$

$$L_{bc} = L_{cb} = -M_s - L_m \cos\left(\theta_r + \frac{\pi}{6} - \frac{2\pi}{3}\right),$$

$$L_{ca} = L_{ac} = -M_s - L_m \cos\left(\theta_r + \frac{\pi}{6} + \frac{2\pi}{3}\right),$$

where:

- 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.
- θ_r is the rotor mechanical 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, K, is defined as

$$K = \frac{2}{3}\begin{bmatrix} \cos\theta_e & \cos\left(\theta_e - \frac{2\pi}{3}\right) & \cos\left(\theta_e + \frac{2\pi}{3}\right) \\ -\sin\theta_e & -\sin\left(\theta_e - \frac{2\pi}{3}\right) & -\sin\left(\theta_e + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

where θ_e is the electrical angle. The electrical angle depends on the rotor mechanical angle and the number of pole pairs such that

$$\theta_e = N\theta_r,$$

where:

- N is the number of pole pairs.
- θ_r is the rotor mechanical angle.

The inverse of the Park transformation is defined by

$$K^{-1} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e & 1 \\ \cos\left(\theta_e - \frac{2\pi}{3}\right) & -\sin\left(\theta_e - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_e + \frac{2\pi}{3}\right) & -\sin\left(\theta_e + \frac{2\pi}{3}\right) & 1 \end{bmatrix}.$$

Applying the Park transformation to the first two electrical defining equations produces equations that define the behavior of the block:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - N\omega i_q L_q,$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + N\omega i_d L_d,$$

$$v_0 = R_s i_0 + L_0 \frac{di_0}{dt},$$

$$T = \frac{3}{2} N\left(i_q i_d L_d - i_d i_q L_q\right)$$

$$J \frac{d\omega}{dt} = T - T_L - B_m \omega,$$

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.
- ω is the rotor mechanical rotational speed.
- J is the rotor inertia.

Assumptions

The block assumes that 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.

`n` — Neutral phase
electrical

Electrical conserving port associated with the neutral phase.

Parameters

Main

`Number of pole pairs` — Rotor pole pairs
`6` (default) | integer

Number of permanent magnet pole pairs on the rotor.

`Stator parameterization` — Parameterization method
`Specify Ld, Lq and L0` (default) | `Specify Ls, Lm, and Ms`

Method for parameterizing the stator.

Dependencies

Selecting `Specify Ld, Lq and L0` enables these parameters:

- **Stator d-axis inductance, Ld**
- **Stator q-axis inductance, Lq**
- **Stator zero-sequence inductance, L0**

Selecting `Specify Ls, Lm, and Ms` enables these parameters:

- **Stator self-inductance per phase, Ls**
- **Stator inductance fluctuation, Lm**
- **Stator mutual inductance, Ms**

`Stator d-axis inductance, Ld` — Inductance
`0.0031 H` (default)

Direct-axis inductance of the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator q-axis inductance, Lq` — Inductance
`0.004 H` (default)

Quadrature-axis inductance of the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator zero-sequence inductance, L0` — Inductance

`0.0005 H` (default)

Zero-axis inductance for the machine stator.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ld, Lq and L0`.

`Stator self-inductance per phase, Ls` — Inductance

`0.0025 H` (default)

Average self-inductance of the three stator windings.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ls, Lm, and Ms`.

`Stator inductance fluctuation, Lm` — Inductance

`-0.0003 H` (default)

Amplitude of the fluctuation in self-inductance and mutual inductance with the rotor angle.

Dependencies

To enable this parameter, set **Stator parameterization** to `Specify Ls, Lm, and Ms`.

`Stator mutual inductance, Ms` — Inductance

`0.0010 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` — Resistance

`0.7 Ohm` (default)

Resistance of each of the stator windings.

`Field winding resistance, Rf` — Resistance
`2.85 Ohm` (default)

Resistance of the field winding.

Mechanical

`Rotor inertia` — Inertia
`0.01 kg*m^2` (default)

Inertia of the rotor.

`Rotor Damping` — Damping
`0 N*m/(rad/s)` (default)

Damping of the rotor.

Initial Conditions

`Initial currents, [i_a i_b i_c]` — Current
`l[0, 0, 0] A` (default) | vector

The *a*-, *b*-, and *c*-phase currents at simulation start time.

`Initial rotor angle` — Angle
`0 deg` (default) | `0-360 deg`

Rotor angle at simulation start time.

`Initial rotor speed` — Speed
`0 rpm` (default)

Initial speed of the rotor. If the rotor inertia, *J*, is zero, the initial speed of the rotor is zero rpm and the initial rotor speed is ignored.

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, Neuville sur Oise, France, 2012.

See Also

Simscape Blocks

Brushless DC Motor | Hybrid Excitation Synchronous Machine | Permanent Magnet Synchronous Motor | Switched Reluctance Machine | Synchronous Machine Field Circuit (pu) | Synchronous Machine Measurement

Introduced in R2017b

Thermal Resistor

Heat transfer by conduction through a layer of material

Library

Semiconductors / Fundamental Components / 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.

Parameters

- “Parameters Tab” on page 1-677
- “Variables Tab” on page 1-677

Parameters Tab

Thermal resistance

The default value for the thermal resistance, $R_{thermal}$, is `1e-3` `K/W`.

Variables Tab

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

A

Thermal conserving port associated with surface A of the material that the heat flows through.

B

Thermal conserving port associated with surface B of the material that the heat flows through.

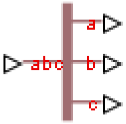
See Also

Cauer Thermal Model Element | Foster Thermal Model

Introduced in R2016a

Three Element Demux

Convert three-element physical signal vector into scalar physical signals



Library

Sensors

Description

The Three Element Demux block splits a three-element physical signal vector into three scalar physical signals.

Ports

The block has the following ports:

`abc`

Three-element physical signal input port.

`a`

Scalar physical signal output port.

`b`

Scalar physical signal output port.

`c`

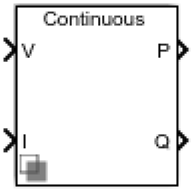
Scalar physical signal output port.

Introduced in R2013b

Three-Phase Power Measurement

Measure three-phase real and reactive power

Library: Simscape / Power Systems / Simscape Components / Control / Measurements



Description

The Three-Phase Power Measurement 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:

$$P_k + jQ_k = \frac{3}{2}(V_k e^{j\theta_{V_k}})(\overline{I_k e^{j\theta_{I_k}}}),$$

where:

- $V_k e^{j\theta_{V_k}}$ is the phasor representing the k-component voltage of the selected sequence.
- $\overline{I_k e^{j\theta_{I_k}}}$ 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`:

  $$V_k e^{j\theta_{V_k}} = V_{k+} e^{j\theta_{V_{k+}}}, I_k e^{j\theta_{I_k}} = I_{k+} e^{j\theta_{I_{k+}}}$$

- `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:

$$\begin{bmatrix} V_{ka} e^{j\theta_{V_{ka}}} \\ V_{kb} e^{j\theta_{V_{kb}}} \\ V_{kc} e^{j\theta_{V_{kc}}} \end{bmatrix} = \frac{2}{T}\int_{t-T}^{t} V(t)\sin(2\pi kFt)dt + j\frac{2}{T}\int_{t-T}^{t} V(t)\cos(2\pi kFt)dt,$$

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

vector

Real power for selected frequency components, in W.

Data Types: `single` | `double`

`P` **— Reactive power**

vector

Reactive power for selected frequency components, in var.

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) | 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

Sample time for the block. For continuous operation, set this property to `0`. For discrete operation, specify the sample time explicitly. This block does not support inherited sample time.

See Also

Blocks

Power Measurement | RMS Measurement | Sinusoidal Measurement (PLL) | Three-Phase Sinusoidal Measurement (PLL)

Introduced in R2017b

Three-Phase Sinusoidal Measurement (PLL)

Estimate three-phase sinusoidal characteristics using a phase-locked loop

Library: Simscape / Power Systems / Simscape Components / Control / Measurements

Description

The Three-Phase Sinusoidal Measurement (PLL) 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:

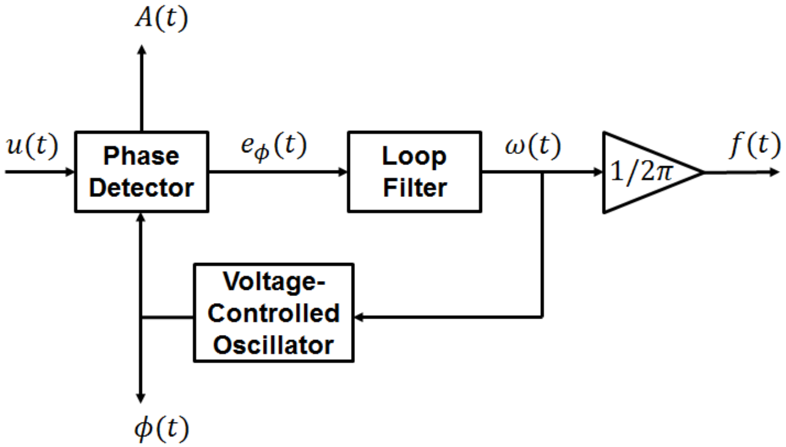
- y is the estimate of the input signal.
- A is the amplitude of the input signal.
- ϕ_0 is the initial phase angle of the input signal.

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 ϕ is the angle of this generated sinusoid:

$$\phi(t) = \phi_0 + \int 2\pi f(t)dt,$$

where *f* if the frequency of the sinusoid, and ϕ_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_ϕ 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 ω 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 ϕ 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) | `-1` or positive number

Sample time for the block (-1 for inherited). Set this to `0` for continuous operation, or explicitly for discrete operation. 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.

Model Examples

See Also

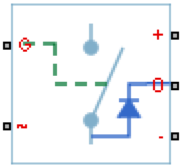
Blocks

RMS Measurement | Sinusoidal Measurement (PLL)

Introduced in R2017b

Three-Level Converter

Connect three-phase AC network to three-level DC network



Library

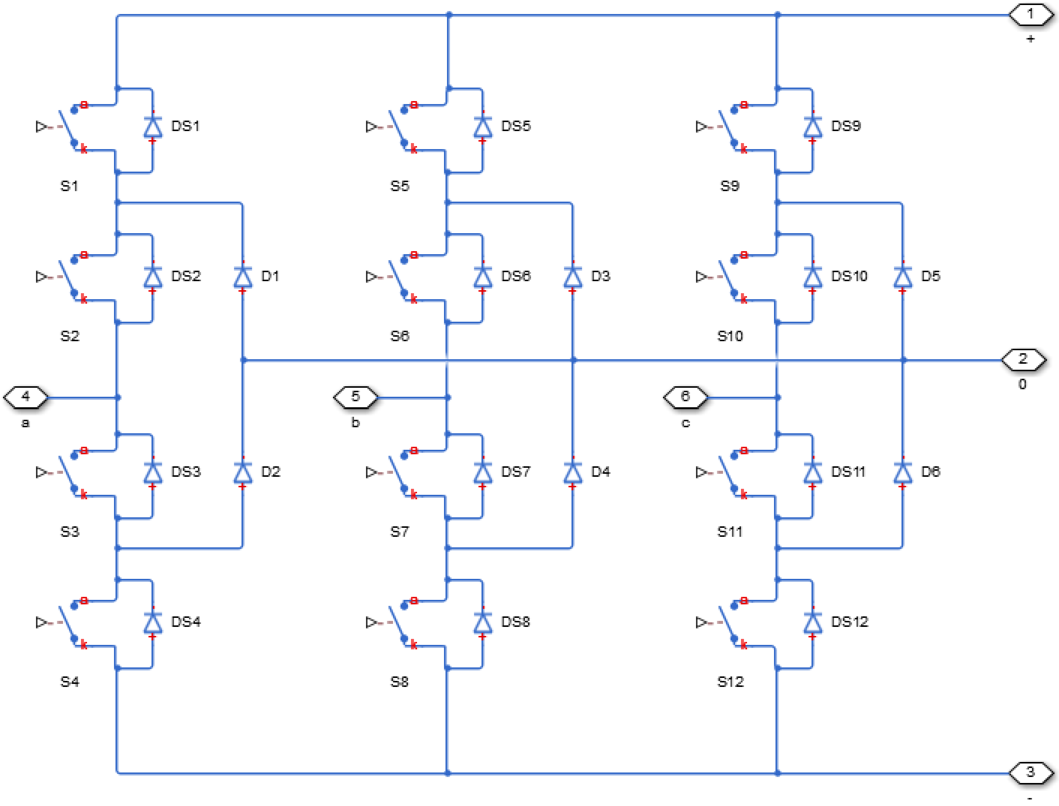
Semiconductors

Description

The Three-Level Converter block models a three-arm converter circuit that connects a three-phase AC network to a three-level DC network.

Each component in the three-arm circuit is the same switching device, which you specify using an option in the Converter block dialog box. The switching devices are 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 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 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.

Goal	Value to Select	Block Behavior
Prioritize simulation speed.	`Protection diode with no dynamics`	The block includes an integral copy of the Diode block. The block dialog box shows parameters relating to the Diode block.
Precisely specify reverse-mode charge dynamics.	`Protection diode with charge dynamics`	The block includes an integral copy of the Commutation Diode block. The block dialog box shows parameters relating to the Commutation 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.

Parameters

- "Switching Devices Tab" on page 1-692
- "Diodes Tab" on page 1-695
- "Snubbers Tab" on page 1-698

Switching Devices Tab

Switching device

Converter switching device. The default value is `Ideal Semiconductor Switch`.

The switching devices you can select are:

- `GTO`

- `Ideal Semiconductor Switch`
- `IGBT`
- `MOSFET`

When you select `GTO`, parameters for the GTO block appear.

Additional GTO Parameters

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Rate of change of voltage versus current above the forward voltage. The default value is `0.001 Ohm`.

Off-state conductance

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-6 1/Ohm`.

Gate trigger voltage, Vgt

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `1 V`.

Gate turn-off voltage, Vgt_off

Gate-cathode voltage threshold. The device turns off when the gate-cathode voltage is below this value. The default value is `-1 V`.

Holding current

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. The default value is `1 A`.

For more information, see GTO.

When you select `Ideal Semiconductor Switch`, parameters for the Ideal Semiconductor Switch block appear.

Additional Ideal Semiconductor Switch Parameters

On-state resistance

Anode-cathode resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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-6 1/Ohm`.

Threshold voltage, Vth

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `6 V`.

For more information, see Ideal Semiconductor Switch.

When you select `IGBT`, parameters for the IGBT block appear.

Additional IGBT Parameters

Forward voltage, Vf

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**. The default value is `0.8 V`.

On-state resistance

Collector-emitter resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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**. The default value is `1e-6 1/Ohm`.

Threshold voltage, Vth

Collector-emitter voltage at which the device turns on. The default value is `6 V`.

For more information, see IGBT.

When you select `MOSFET`, parameters for the MOSFET block appear.

Additional MOSFET Parameters

On-state resistance, R_DS(on)

Drain-source resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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**. The default value is `1e-6 1/Ohm`.

Threshold voltage, Vth

Gate-source voltage threshold. The device turns on when the gate-source voltage is above this value. The default value is `6` V.

For more information, see MOSFET.

Diodes Tab

Integral protection diode

Integral protection diode for each switching device. Choose between `Diode with no dynamics` and `Diode with charge dynamics`. The default value is `Diode with no dynamics`.

When you select `Diode with no dynamics`, additional parameters appear.

Additional Parameters for Diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

The default value is `1500 μAs`.

The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Snubbers Tab

Snubber

Snubber for each switching device. The default value is `None`.

Snubber resistance

This parameter is visible only if you set **Snubber** to `RC snubber`. The default value is `0.1 Ohm`.

Snubber capacitance

This parameter is visible only if you set **Snubber** to `RC snubber`. The default value is `1e-7 F`.

Ports

The block has the following ports:

`G`

Vector input port associated with the gate terminals of the switching devices. Connect this port to a Twelve-Pulse Gate Multiplexer block.

`~`

Expandable three-phase port.

`+`

Electrical conserving port associated with the DC positive terminal.

`0`

Electrical conserving port associated with the DC neutral terminal.

`-`

Electrical conserving port associated with the DC negative terminal.

See Also

Average-Value Inverter | Average-Value Rectifier | Converter | Rectifier | Twelve-Pulse Gate Multiplexer

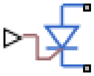
Topics

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

Introduced in R2014b

Thyristor

Thyristor

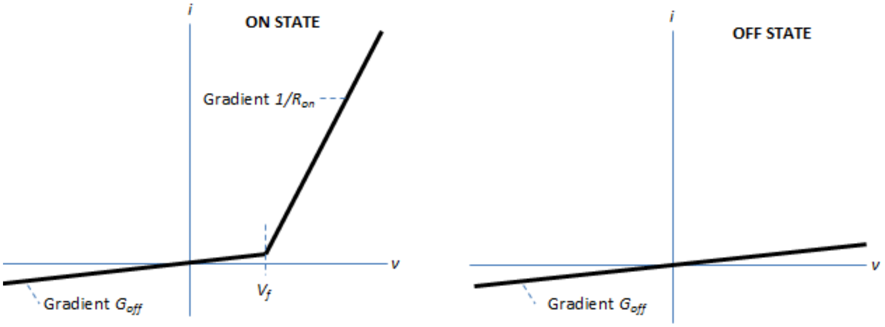


Library

Semiconductors / Fundamental Components

Description

The Thyristor 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.



In the on state, the anode-cathode path behaves like a linear diode with forward-voltage drop, V_f, and 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}.

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.
- *Vf* is the forward voltage.
- *G* is the gate voltage.
- *Vgt* is the gate trigger voltage.
- *i* is the anode-cathode current.
- *Ih* is the holding current.
- *Ron* is the on-state resistance.
- *Goff* 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. The block dialog box shows parameters relating to the Diode block.

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 Commutation Diode block. The block dialog box shows parameters relating to the Commutation Diode block.

Modeling Variants

The block provides four modeling variants. To select the desired variant, right-click the block in your model. From the context menu, select **Simscape** > **Block choices**, and then one of these variants:

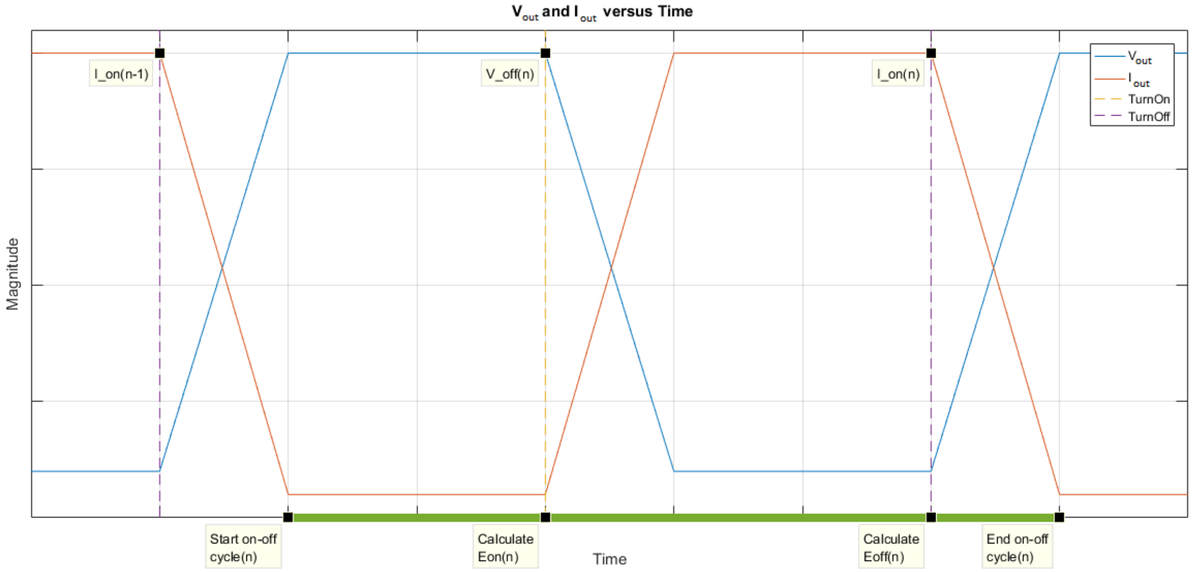
- **PS Control Port** — Contains a physical signal port that is associated with the gate terminal. This variant is the default.
- **Electrical Control Port** — Contains an electrical conserving port that is associated with the gate terminal.
- **PS Control Port | Thermal Port** — Contains a thermal port and a physical signal port that is associated with the gate terminal.
- **Electrical Control Port | Thermal Port** — Contains a thermal port and an electrical conserving port that is associated with the gate terminal.

The variants of this block without the thermal port do not simulate heat generation in the device.

The variants with the thermal port allow you to model the heat that switching events and conduction losses generate. For numerical efficiency, the thermal state does not affect the electrical behavior of the block. The thermal port is hidden by default. To enable the thermal port, select a thermal block variant.

Thermal Loss Equations

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 nth switching cycle, during which the block turns off and then on.



When the semiconductor turns on during the *n*th switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{on(n)} = \frac{V_{off\,(n)}}{V_{off\,\_\,data}}\,fcn(T, I_{on(n-1)}),$$

where:

- $E_{on(n)}$ is the switch-on loss at the *n*th switch-on event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the *n*th switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n-1)}$ is the on-state output current, I_{out}, just before the device switches off during the cycle that precedes the n*th* switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{on\_data}, T, I_{on(n-1)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{on_data} is the **Switch-on loss, Eon=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns on during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{on(n)} = \left(\frac{V_{off(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{on\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{on_scalar} is the **Switch-on loss** parameter value.

When the semiconductor turns off during the *n*th switching cycle, the amount of thermal energy that the device dissipates increments by a discrete amount. If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, the equation for the incremental change is

$$E_{off(n)} = \frac{V_{off(n)}}{V_{off\_data}} fcn(T, I_{on(n)}),$$

where:

- $E_{off(n)}$ is the switch-off loss at the *n*th switch-off event.
- $V_{off(n)}$ is the off-state output voltage, V_{out}, just before the device switches on during the *n*th switching cycle.
- V_{off_data} is the **Off-state voltage for losses data** parameter value.
- T is the device temperature.
- $I_{on(n)}$ is the on-state output current, I_{out}, just before the device switches off during the *n*th switching cycle.

The function *fcn* is a 2-D lookup table with linear interpolation and linear extrapolation:

$$E = tablelookup(T_{j\_data}, I_{out\_data}, E_{off\_data}, T, I_{on(n)}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.
- I_{out_data} is the **Output current vector, Iout** parameter value.
- E_{off_data} is the **Switch-off loss, Eoff=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, when the semiconductor turns off during the *n*th switching cycle, the equation that the block uses to calculate the incremental change in the discrete amount of thermal energy that the device dissipates is

$$E_{off\,(n)} = \left(\frac{V_{off\,(n)}}{V_{off\_data}}\right)\left(\frac{I_{on(n-1)}}{I_{out\_scalar}}\right)(E_{off\_scalar})$$

where:

- I_{out_scalar} is the **Output current, Iout** parameter value.
- E_{off_scalar} is the **Switch-off loss** parameter value.

If you select `Voltage, current, and temperature` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int fcn(T, I_{out})\, dt,$$

where:

- $E_{conduction}$ is the heat loss due to electrical conduction.
- T is the device temperature.
- I_{out} is the device output current.

The function *fcn* is a 2-D lookup table:

$$Q_{conduction} = tablelookup(T_{j\_data}, I_{out\_data}, I_{out\_data\_repmat} .{*} V_{on\_data}, T, I_{out}),$$

where:

- T_{j_data} is the **Temperature vector, Tj** parameter value.

- I_{out_data} is the **Output current vector, Iout** parameter value.
- $I_{out\_data\_repmat}$ is a matrix that contains length, T_{j_data}, copies of I_{out_data}.
- V_{on_data} is the **On-state voltage, Von=fcn(Tj,Iout)** parameter value.

If you select `Voltage and current` for the **Thermal loss dependent on** parameter, then, for both the on state and the off state, the heat loss due to electrical conduction is

$$E_{conduction} = \int \left(I_{out} * V_{on\_scalar} \right) dt,$$

where V_{on_scalar} is the **On-state voltage** parameter value.

The block uses the **Energy dissipation time constant** parameter to filter the amount of heat flow that the block outputs. The filtering allows the block to:

- Avoid discrete increments for the heat flow output
- Handle a variable switching frequency

The filtered heat flow is

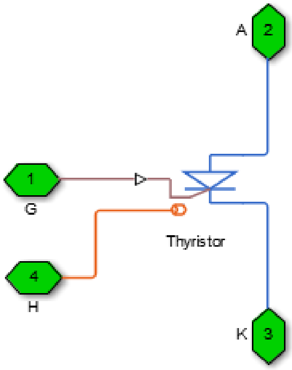
$$Q = \frac{1}{\tau} \left(\sum_{i=1}^{n} E_{on(i)} + \sum_{i=1}^{n} E_{off\,(i)} + E_{conduction} - \int Q\,dt \right),$$

where:

- Q is the heat flow from the component.
- τ is the **Energy dissipation time constant** parameter value.
- n is the number of switching cycles.
- $E_{on(i)}$ is the switch-on loss at the ith switch-on event.
- $E_{off(i)}$ is the switch-off loss at the ith switch-off event.
- $E_{conduction}$ is the heat loss due to electrical conduction.
- $\int Qdt$ is the total heat previously dissipated from the component.

Ports

The figure shows the block port names.



`G`

Port associated with the gate terminal. You can set the port to either a physical signal or electrical port.

`A`

Electrical conserving port associated with the anode terminal.

`K`

Electrical conserving port associated with the cathode terminal.

`H`

Thermal conserving port. The thermal port is optional and is hidden by default. To enable this port, select a variant that includes a thermal port.

Parameters

- “Main Tab” on page 1-707
- “Integral Diode Tab” on page 1-708
- “Thermal Model Tab” on page 1-711

Main Tab

Forward voltage, Vf

Forward voltage at which the device turns on. The default value is `0.8` `V`.

On-state resistance

Anode-cathode resistance when the device is on. The default value is `0.001 Ohm`.

Off-state conductance

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 1/Ohm`.

Gate trigger voltage, Vgt

Gate-cathode voltage threshold. The device turns on when the gate-cathode voltage is above this value. The default value is `6 V`.

Holding current

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. The default value is `1 A`.

Integral Diode Tab

Integral protection diode

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`

When you select `Protection diode with no dynamics`, additional parameters appear.

Additional Parameters for Protection diode with no dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

For more information on these parameters, see Diode.

When you select `Protection diode with charge dynamics`, additional parameters appear.

Additional Parameters for Protection diode with charge dynamics

Forward voltage

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**. The default value is `0.8 V`.

On resistance

Rate of change of voltage versus current above the **Forward voltage**. The default value is `0.001 Ohm`.

Off conductance

Conductance of the reverse-biased diode. The default value is `1e-5 1/Ohm`.

Junction capacitance

Diode junction capacitance. The default value is `50 nF`.

Peak reverse current, iRM

Peak reverse current measured by an external test circuit. This value must be less than zero. The default value is `-235 A`.

Initial forward current when measuring iRM

Initial forward current when measuring peak reverse current. This value must be greater than zero. The default value is `300 A`.

Rate of change of current when measuring iRM

Rate of change of current when measuring peak reverse current. This value must be less than zero. The default value is `-50 A/μs`.

Reverse recovery time parameterization

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 “Alternatives to Specifying trr Directly” on page 1-120.

Reverse recovery time, trr

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 default value is `15 μs`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery time directly`.

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.

Reverse recovery time stretch factor

Value that the block uses to calculate **Reverse recovery time, trr**. This value must be greater than `1`. The default value is `3`.

This parameter is visible only if you set **Reverse recovery time parameterization** to `Specify stretch factor`.

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.

Reverse recovery charge, Qrr

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 $-\frac{i^2_{RM}}{2a}$,

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**.

  The default value is `1500` `μAs`.

  The parameter is visible only if you set **Reverse recovery time parameterization** to `Specify reverse recovery charge`.

For more information on these parameters, see Commutation Diode.

Thermal Model Tab

The **Thermal Model** tab is enabled only when you select a block variant that includes a thermal port.

Thermal loss dependent on

Select a parameterization method. The option that you select determines which other parameters are enabled. Options are:

- `Voltage and current` — Use scalar values to specify the output current, switch-on loss, switch-off loss, and on-state voltage data.
- `Voltage, current, and temperature` — Use vectors to specify the output current, switch-on loss, switch-off loss, on-state voltage, and temperature data. This is the default parameterization method.

Off-state voltage for losses data

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. The default value is `300` `V`.

Energy dissipation time constant

Time constant used to average the switch-on losses, switch-off losses, and conduction losses. This value is equal to the period of the minimum switching frequency. The default value is `1e-4` `s`.

Additional Parameters for Parameterizing by Voltage, Current, and Temperature

Temperature vector, Tj

Temperature values at which the switch-on loss, switch-off loss, and on-state voltage are specified. Specify this parameter using a vector quantity. The default value is `[ 298.15 398.15 ]` `K`.

Output current vector, Iout

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 vector quantity. The default value is `[ 0 10 50 100 200 400 600 ] A`.

Switch-on loss, Eon=fcn(Tj,Iout)

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. The default value is `[ 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`.

Switch-off loss, Eoff=fcn(Tj,Iout)

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. The default value is `[ 0 2.1e-4 0.00107 0.00214 0.00429 0.009859999999999999 0.01714; 0 4.3e-4 0.00197 0.00386 0.00771 0.01543 0.02271 ] J`.

On-state voltage, Von=fcn(Tj,Iout)

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. The default value is `[ 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`.

Additional Parameters for Parameterizing by Voltage and Current

Output current, Iout

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. The default value is `600 A`.

Switch-on loss

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. The default value is `0.02286 J`.

Switch-off loss

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. The default value is `0.01714 J`.

On-state voltage

Voltage drop across the block 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 scalar quantity. The default value is `2.7 V`.

See Also

Commutation Diode | Diode | GTO | IGBT | Ideal Semiconductor Switch | MOSFET

Topics

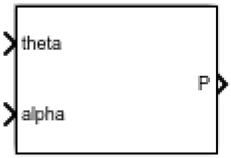
"Quantifying IGBT Thermal Losses"
"Simulate Thermal Losses in Semiconductors"
"Switch Between Physical Signal and Electrical Ports"

Introduced in R2013b

Thyristor 6-Pulse Generator

Generate thyristor 6-pulse waveform in single-pulsing mode

Library: Simscape / Power Systems / Simscape Components / Control / Pulse Width Modulation



Description

The Thyristor 6-Pulse Generator block implements a thyristor 6-pulse waveform generator in single-pulsing mode.

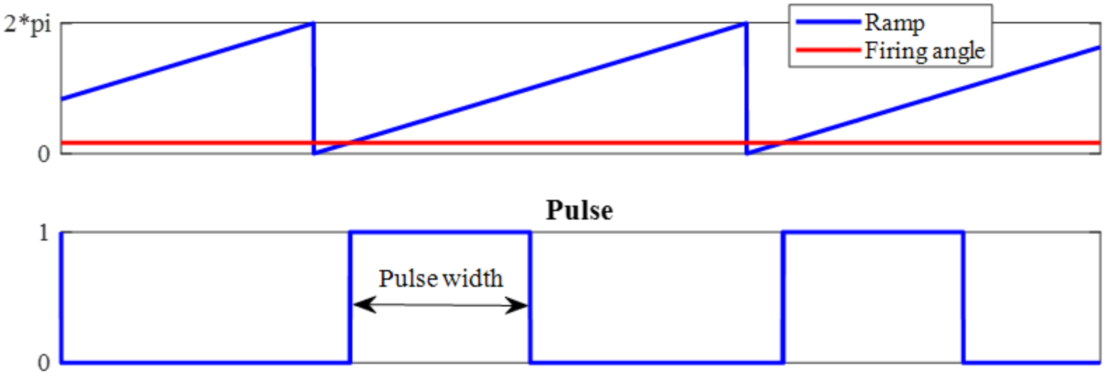
You can use this block to perform phase-controlled AC-to-DC conversion by:

- 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

Working Principles

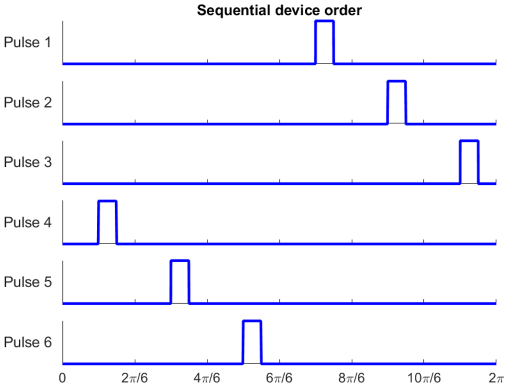
Using the synchronization angle, the block internally generates six ramps, one for each of the pulse elements in its output vector.

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.

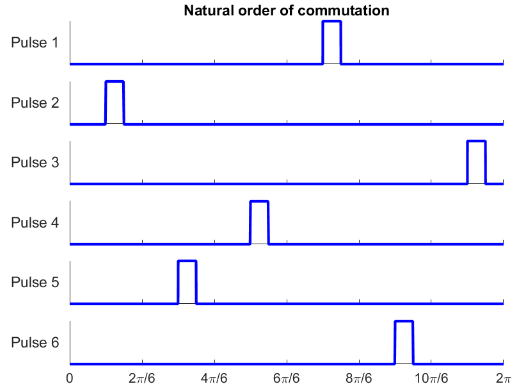


Set the pulse ordering strategy to modify the distinct phase-shift of each ramp, and as a result, the order of generated pulses:

- Set the `Pulse ordering` property to `Sequential device order` to generate pulses in sequential order. Use this strategy to generate pulses for the Converter 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

scalar

Synchronization angle in the range `[0, 2*pi]`, in radians.

Data Types: `single` | `double`

`alpha` — Firing angle

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 number

Specify the width of each pulse in the range `[0, pi]`.

`Sample time (-1 for inherited)` — Block sample time
`1e-5` (default) | `-1` or positive number

Specify sample time for the block (`-1` for inherited). If this block is used inside a triggered subsystem, set the sample time should to `-1`. If this block is used in a continuous variable-step model, specify the sample time explicitly.

Model Examples

References

[1] Pelly, B. R. *Thyristor Phase-Controlled Converters and Cycloconverters: Operation, Control, and Performance*. Hoboken, NJ: John Wiley & Sons, 1971.

See Also

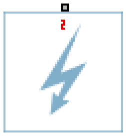
Blocks

Converter | Six-Pulse Gate Multiplexer | Three-Phase Sinusoidal Measurement (PLL)

Introduced in R2017b

Time-Based Fault

Time-based single-phase, two-phase, or three-phase grounded or ungrounded fault



Library

Passive Devices / Faults

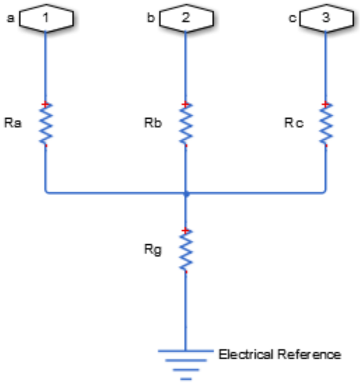
Description

The Time-Based Fault block models any permutation of a single-phase, two-phase, or three-phase grounded or ungrounded fault. You specify the fault activation time using the block **Fault start time** parameter. The fault becomes inactive when the fault duration that you specify elapses.

You can set the Time-Based Fault block to represent any of 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 Time-Based Fault block.



You can determine the resistance in the equivalent circuit using the equations in the table.

Fault type	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
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}
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

Fault type	Value of R_a	Value of R_b	Value of R_c	Value of R_g
c-a	R_{pn}	$\frac{1}{G_{pn}}$	R_{pn}	Infinity / open circuit
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}
a-b-c	R_{pn}	R_{pn}	R_{pn}	Infinity / open circuit
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.

Parameters

- “Main Tab” on page 1-722
- “Parasitics Tab” on page 1-723

Main Tab

Fault type

Select one of the following:

- `None` — Specifies that the fault is not active. This is the default value.
- `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

Resistance between the phase connection and the neutral point when the fault is active. This parameter is visible if the **Fault type** parameter is set to anything other than `None`. The default value is `1e-3 Ohm`.

Faulted neutral-ground resistance

Resistance between the neutral point and the electrical reference when fault is active. This parameter is visible if the **Fault type** parameter is set to any fault which includes a ground connection. The default value is `1e-3 Ohm`.

Fault start time

Simulation time when the fault becomes active. This parameter is visible if the **Fault type** parameter is set to anything other than `None`. The default value is `1 s`.

Fault duration

Period of time that the fault is active. This parameter is visible if the **Fault type** parameter is set to anything other than `None`. The default value is `0.1 s`.

Parasitics Tab

Unfaulted phase-neutral conductance

Conductance between the phase connections and the neutral point when a phase is not involved in the fault. The default value is `1e-6 1/Ohm`.

Ports

The block has one expandable three-phase port for connecting the fault to the system.

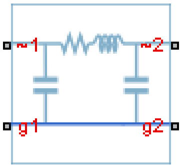
See Also

Enabled Fault

Introduced in R2014a

Transmission Line

Three-phase transmission line using lumped-parameter pi-line model



Library

Passive Devices

Description

The Transmission Line 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.
- 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.
- *I1* is the three-phase current flowing into the `~1` port.
- *I2* is the three-phase current flowing into the `~2` port.
- *V1* is the three-phase voltage at the `~1` port.
- *V2* 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:

$$R = \frac{2R_1 + R_0}{3}$$

$$R_m = \frac{R_0 - R_1}{3}$$

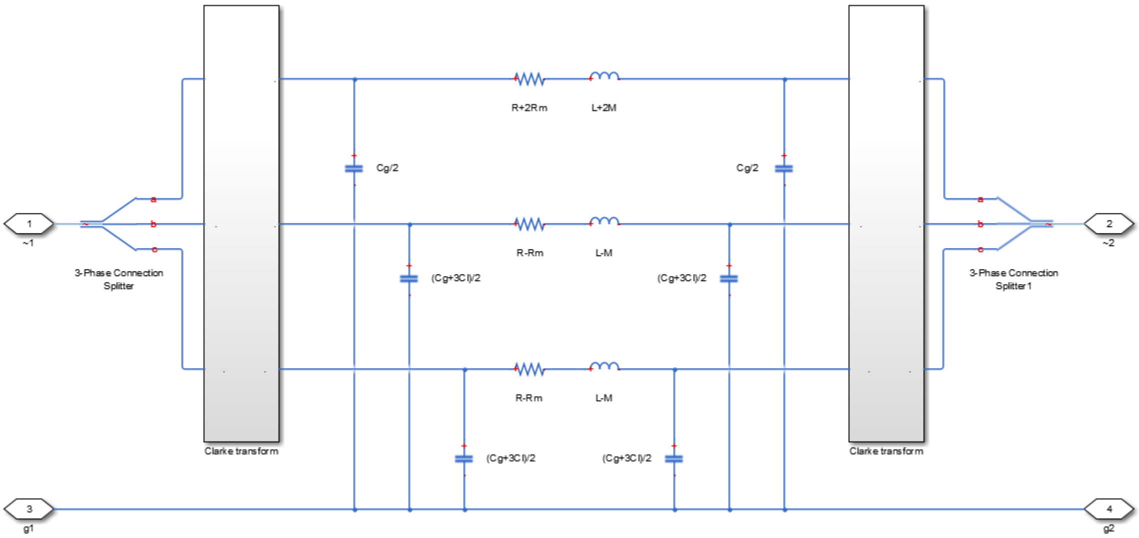
$$L = \frac{2L_1 + L_0}{3}$$

$$M = \frac{L_0 - L_1}{3}$$

$$C_l = \frac{C_1 - C_0}{3}$$

$$C_g = C_0$$

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.

Parameters

- "Main tab" on page 1-727
- "Parasitics tab" on page 1-728

Main tab

Line length

Length of the transmission line. The default value is `1 km`.

Resistance

Resistance of the transmission line per phase per-unit length. The default value is `0.02 Ohm/km`.

Inductance

Self-inductance of the transmission line per phase per-unit length. The default value is `0.5 mH/km`.

Mutual inductance

Line-line mutual inductance per-unit length. Set this to `0` to remove mutual inductance. The default value is `0.1 mH/km`.

Line-line capacitance

Line-line capacitance per-unit length. The default value is `0.3 µF/km`.

Line-ground capacitance

Line-ground capacitance per-unit length. The default value is `0 µF/km` (no line-ground capacitance).

Mutual Resistance

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 in the pi-line model. The default value is `1`.

Parasitics tab

Parasitic series resistance

Resistance value, divided by the number of segments, that is added in series with every capacitor in the model. The default value is `1e-6 Ohm`.

Parasitic parallel conductance

Conductance value, divided by the number of segments, that is added in parallel with every series resistor and inductor in the model. The default value is `1e-6 1/Ohm`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

`~2`

Expandable three-phase port

`g1`

Electrical conserving port corresponding to ground connection at `~1` end of the transmission line

`g2`

Electrical conserving port corresponding to ground connection at `~2` end of the transmission line

See Also

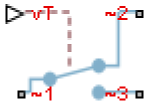
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Two-Way Switch

Three-phase double-throw switch



Library

Switches & Breakers

Description

The Two-Way Switch block models a three-phase 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` ≤ **Threshold**	Each phase of port `~1` is connected to the corresponding phase of port `~2` via internal resistance. Port `~3` is unconnected.	**Open conductance** (port `~1` to port `~3`). **Closed resistance** (port `~1` to port `~2`)
`vT` > **Threshold**	Each phase of port `~1` is connected to the corresponding phase of port `~3` via internal resistance. Port `~2` is unconnected.	**Open conductance** (port `~1` to port `~2`). **Closed resistance** (port `~1` to port `~3`)

Parameters

Closed resistance

Resistance between ports `~1` and `~3` when the switch is closed. The default value is `0.001` `Ohm`.

Open conductance

Conductance between ports `~1` and `~2` when the switch is open. The default value is `1e-6` `1/Ohm`.

Threshold

Threshold voltage for the control port `vT`. When the voltage is above the threshold, the switch is closed. The default value is `0` `V`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port

`~2`

Expandable three-phase port

`~3`

Expandable three-phase port

`vT`

Scalar control port, which is either a physical signal or an electrical port.

See Also

Single-Phase Switch | Single-Phase Two-Way Switch | Switch

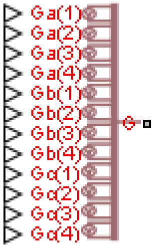
Topics

“Expand and Collapse Three-Phase Ports on a Block”
“Switch Between Physical Signal and Electrical Ports”

Introduced in R2013b

Twelve-Pulse Gate Multiplexer

Multiplex gate input signals to Three-Level Converter block



Library

Semiconductors

Description

The Twelve-Pulse Gate Multiplexer block routes gate voltage signals to the 12 switching devices in a Three-Level Converter block. The block multiplexes the 12 gate signals into a single vector. Gate signals are ordered as A-phase, Behaves, and then Chafes, with four gate signals per phase.

If you want to use Simscape Electronics to model the electronics that drive the Three-Level Converter block, you can switch the input ports of the Twelve-Pulse Gate Multiplexer block from physical signal ports to electrical ports.

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

The block has the following ports:

`Ga(1),Ga(2),Ga(3),Ga(4)`

Ports associated with the gate terminals of the Three-Level Converter A-phase switching devices. You can set the ports to either physical signal or electrical ports.

`Gb(1),Gb(2),Gb(3),Gb(4)`

Ports associated with the gate terminals of the Three-Level Converter Behaves switching devices. You can set the ports to either physical signal or electrical ports.

`Gc(1),Gc(2),Gc(3),Gc(4)`

Ports associated with the gate terminals of the Three-Level Converter Chafes switching devices. You can set the ports to either physical signal or electrical ports.

`G`

Vector output port associated with the multiplexed gate signals. Connect this port to the G port of the Three-Level Converter block.

`Ka(1),Ka(2),Ka(3),Ka(4)`

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter block A-phase switching devices. These ports are visible only if you set the input ports of the Twelve-Pulse Gate Multiplexer block to electrical ports.

`Kb(1),Kb(2),Kb(3),Kb(4)`

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter block Behaves switching devices. These ports are visible only if you set the input ports of the Twelve-Pulse Gate Multiplexer block to electrical ports.

`Kc(1),Kc(2),Kc(3),Kc(4)`

Electrical conserving ports associated with the individual cathode terminals corresponding to the Three-Level Converter block Chafes switching devices. These ports are visible only if you set the input ports of the Twelve-Pulse Gate Multiplexer block to electrical ports.

See Also

Six-Pulse Gate Multiplexer | Three-Level Converter

Topics

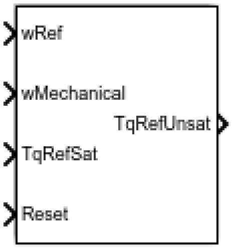
“Switch Between Physical Signal and Electrical Ports”

Introduced in R2014b

Velocity Controller

Discrete-time velocity controller

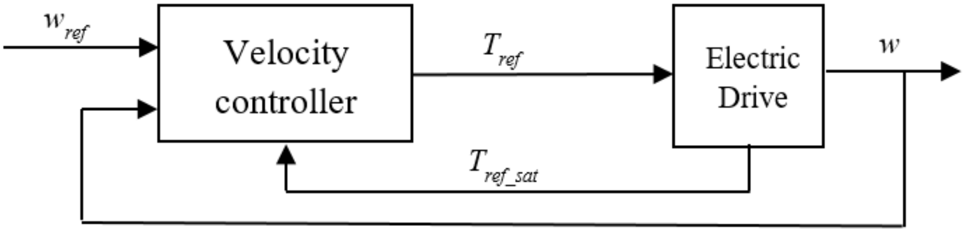
Library: Simscape / Power Systems / Simscape Components / Control / General Machine Control



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} to be given to 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:

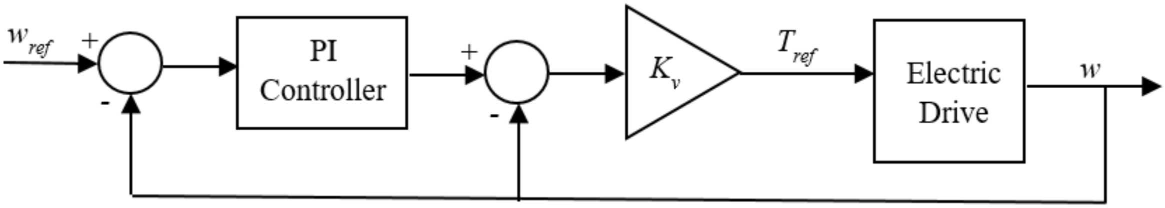
- Proportional-integral (PI) control, with proportional and integral gains K_{p_w} and K_{i_w}:

  $$T_{ref} = \left(K_{p\_w} + K_{i\_w}\frac{T_s z}{z-1}\right)\left(w_{ref} - w\right)$$

- Proportional (P) control, with proportional gain K_{p_w}:

  $$T_{ref} = K_{p\_w}\left(w_{ref} - w\right)$$

- 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.

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 strategy**
`PI control` (default) | `P control` | `P-PI control`

Type of controller.

`Controller proportional gain` **— Proportional gain**
`1` (default) | positive scalar

Proportional gain of PI or P controller.

`Controller integral gain` **— PI integral gain**
`1` (default) | positive scalar

Integral gain of PI controller.

Dependencies

This parameter is enabled 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 of P controller.

Dependencies

This parameter is enabled when the **Control type** is set to `P-PI control`.

`Anti-windup gain` — PI anti-windup gain

`1` (default) | positive scalar

Anti-windup gain of PI controller.

Dependencies

This parameter is enabled when the **Control type** is set to `PI control` or `P-PI control`.

`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?” (Simulink) and “Specify Sample Time” (Simulink).

`Discretization sample time` — Discrete sample time

`0.001` (default) | positive scalar

Specify the discretization sample time when the zero-cancellation is active and simulation sample time is set to `-1`.

Dependencies

This parameter is enabled when:

- **Control type** is set to `PI control` or `P-PI control`.

- **Sample time** is set to `-1`.

`Enable zero cancellation` — Feedforward zero-cancellation
`off` (default) | `on`

Enable or disable zero-cancellation on the feedforward path.

Dependencies

This parameter is enabled when the **Control type** is set to `PI control` or `P-PI control`.

Model Examples

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.

See Also

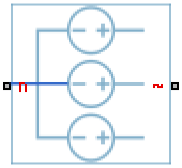
Blocks

SM Current Controller | SM Current Reference Generator

Introduced in R2017b

Voltage Source

Three-phase voltage source



Library

Simscape / Power Systems / Simscape Components / Sources

Description

The Voltage Source 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 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, $V(+) – V\text{n}$.

Equations

The output voltage for the Voltage Source block is defined by these equations:

$$V_0 = \frac{\sqrt{2}}{\sqrt{3}} v_{line\_rms}$$

$$v_a = V_0 \sin(2\pi ft + \varphi)$$

$$v_b = V_0 \sin(2\pi ft + \varphi - 120^\circ)$$

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

where:

- V_0 is the peak phase voltage.
- $v_{\mathrm{line\_rms}}$ is the root-mean square (RMS) phase-to-phase voltage.
- v_a, v_b, v_c are the respective phase voltages.
- f is frequency.
- φ 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:

$$V_0 = \frac{\sqrt{2}}{\sqrt{3}} v_{line\_rms} H_{ratios}$$

$$v_a = V_0 \sin((2\pi ft + \varphi) H'_{orders})$$

$$v_b = V_0 \sin((2\pi ft + \varphi - \theta) H'_{orders})$$

$$v_c = V_0 \sin((2\pi ft + \varphi + \theta) H'_{orders}),$$

where:

- V_0 is a row-vector containing the peak voltage of the fundamental and harmonic sinusoids.
- $v_{\mathrm{line\_rms}}$ 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.
- φ is a column-vector of harmonic phase shifts. The first element is the fundamental phase shift.
- θ is a column-vector of harmonic phase offsets. The first element is `120°`.
- 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.
- ϕ 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.

Parameters

Main

Voltage (phase-to-phase RMS)

RMS phase-to-phase, or line, voltage. The default value is `sqrt(3)*100/sqrt(2)`, or 122.4745, `V`.

Phase shift

Phase shift in angular units. The default value is `0 deg`.

Frequency

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. The default value is `60 Hz`.

Source Impedance

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

Selecting `X/R Ratio` for the **Source Impedance** parameter enables this parameter. The default value is `1e6 V*A`.

Source X/R ratio

Complex impedance, that is, the reactance-to-resistance ratio. Selecting `X/R Ratio` for the **Source Impedance** parameter enables this parameter. The default value is `15`.

Source Resistance

Selecting `Series R` or `Series RL` for the **Source Impedance** parameter enables this parameter. The default value is `0.01 Ohm`.

Source Inductance

Selecting `Series L` or `Series RL` for the **Source Impedance** parameter enables this parameter. The default value is `3.97e-4 H`.

Harmonics

Source Representation

Choose between `None` and `Generate harmonics`. The default value is `None`.

Harmonic orders

A row-vector of additional integer harmonic orders at which harmonics are to be generated. This parameter is only visible when you set the **Source representation** parameter to `Generate harmonics`. The default value is `[5, 7, 11, 13]`.

Harmonic magnitude to peak magnitude ratios

A row-vector of ratios of harmonic magnitudes relative to the fundamental magnitude. This parameter is only visible when you set the **Source representation** parameter to `Generate harmonics`. The default value is `[0.1, 0.1, 0.1, 0.1]`.

Parasitics

Source impedance parasitic parallel conductance

Selecting `X/R Ratio`, `Series L`, or `Series RL` for the **Source Impedance** parameter enables this parameter. The default value is `0 1/Ohm`.

Variables

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” (Simscape).

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the center of the wye

See Also

Current Source

Introduced in R2013b

Washout (Discrete or Continuous)

Discrete-time or continuous-time washout or high-pass filter

Library: Simscape / Power Systems / Simscape Components / Control / General Control

Description

The Washout (Discrete or Continuous) block implements a washout filter in conformance with IEEE 421.5-2016[1]. 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

To configure the washout 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 preceeding transfer function, the washout defining equations are:

$$\begin{cases} \dot{x}(t) = \frac{1}{T}\left(-x(t)+u(t)\right) \\ y(t) = -x(t)+u(t) \end{cases} \quad x(0) = u_0,\ y(0) = 0,$$

where:

- u is the washout input.
- x is the washout state.

- y is the washout output.
- t is the simulation time.
- u_0 is the initial input to the washout block.

To configure the washout block 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:

$$\begin{cases} x(n+1)=\left(1-\dfrac{T_s}{T}\right)x(n)+\left(\dfrac{T_s}{T}\right)u(n) & x(0)=u_0,\ y(0)=0, \\ y(n)=u(n)-x(n) & \end{cases}$$

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 washout block.

Initial Conditions

The block sets the state initial condition to the initial input, making the initial output zero.

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:

$$T\le T_s \rightarrow y=u.$$

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.

`Sample time (-1 for inherited)` — Sample time
`-1` (default) | positive number

Washout filter sample time. Set this to `0` to implement a continuous washout block. Set this to `-1` or a positive number to implement a discrete washout block.

References

[1] *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. IEEE Std 421.5-2016. Piscataway, NJ: IEEE-SA, 2016.

See Also

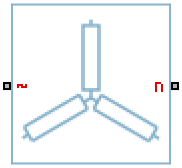
Blocks

Filtered Derivative (Discrete or Continuous) | Integrator (Discrete or Continuous) | Integrator with Wrapped State (Discrete or Continuous) | Lead-Lag (Discrete or Continuous) | Low-Pass Filter (Discrete or Continuous)

Introduced in R2017b

Wye-Connected Load

Three-phase load wired in wye configuration



Library

Passive Devices

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 or 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.

Parameters

- "Main Tab" on page 1-752
- "Parasitics Tab" on page 1-753
- "Initial Conditions Tab" on page 1-754

Main Tab

Parameterization

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

Select the desired combination of a resistor (R), capacitor (C), and inductor (L), connected in series or in parallel. The default is `R`, resistor.

Rated voltage

Voltage for which load powers are specified. This parameter is visible only when you specify values by rated power. The default value is `2.4e4 V`.

Real power

Total real power dissipated by three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes a resistor. The value must be greater than 0. The default value is `1000 W`.

Rated electrical frequency

Frequency for which reactive load powers are specified. This parameter is visible only when you specify values by rated power. The default value is `60 Hz`.

Inductive reactive power

Total inductive reactive power taken by the three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes an inductor. The value must be greater than 0. The default value is `100 V*A`.

Capacitive reactive power

Total capacitive reactive power taken by the three-phase load when supplied at the rated voltage. This parameter is visible only when you specify values by rated power and select a component structure that includes a capacitor. The value must be less than 0. The default value is `-100 V*A`.

Resistance

Resistance of each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes a resistor. The default value is `1 Ohm`.

Inductance

Inductance of each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes an inductor. The default value is `0.001 H`.

Capacitance

Capacitance in each of the load limbs. This parameter is visible only when you specify component values directly and select a component structure that includes a capacitor. The default value is `1e-6 F`.

Parasitics Tab

Parasitic series resistance

Represents small parasitic effects. The parameter value corresponds to the series resistance value added to all instances of capacitors in the load. The default value is `1e-6 Ohm`.

Parasitic parallel conductance

Represents small parasitic effects. The parameter value corresponds to the parallel conductance value added across all instances of inductors in the load. The default value is `1e-6 1/Ohm`.

Initial Conditions Tab

Terminal voltage magnitude

Expected initial RMS line voltage at the load. This parameter is visible only when you specify values by rated power. The default value is `2.4e4 V`.

Terminal voltage angle

Expected initial phase of the voltage at the load. This parameter is visible only when you specify values by rated power. The default value is `0 deg`.

Frequency

Expected initial frequency at the load. This parameter is visible only when you specify values by rated power. The default value is `60 Hz`.

Initial inductor current [Ia Ib Ic]

Initial current in the a, b, and c phase inductors, respectively. This parameter is visible only when you specify component values directly and select a component structure that includes an inductor. The default value is `[0 0 0] A`.

Initial capacitor voltage [Va Vb Vc]

Initial voltage across the a, b, and c phase capacitors, respectively. This parameter is visible only when you specify component values directly and select a component structure that includes a capacitor. The default value is `[0 0 0] V`.

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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`Series LC`	Rated voltage Rated electrical frequency Inductive reactive power Capacitive reactive power	Parasitic parallel conductance	Terminal voltage magnitude Terminal voltage angle Frequency
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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
`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

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`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]

Component Structure	Main Tab	Parasitics Tab	Initial Conditions Tab
`Parallel RLC`	Resistance Inductance Capacitance	Parasitic series resistance	Initial inductor current [Ia Ib Ic] Initial capacitor voltage [Va Vb Vc]

Ports

The block has the following ports:

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral phase

See Also

Delta-Connected Load | RLC

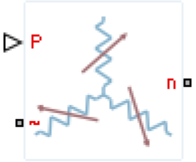
Topics

“Three-Phase Asynchronous Wind Turbine Generator”
“Expand and Collapse Three-Phase Ports on a Block”

Introduced in R2013b

Wye-Connected Variable Load

Three-phase variable load wired in wye configuration



Library

Passive Devices / Time Varying

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.

Electrical Defining 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.

Parameters

Main Tab

Rated voltage

RMS, rated line-line voltage for the resistance equation. The default value is `24e3 V`.

Minimum real power

Minimum real power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0. The default value is `1e3 W`.

Ports

The block has the following ports:

`P`

Physical signal input port for real power

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral phase

See Also

Wye-Connected Load | Wye-Connected Variable Load (lagging)

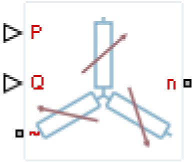
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2014b

Wye-Connected Variable Load (lagging)

Three-phase variable, lagging load wired in wye configuration



Library

Passive Devices / Time Varying

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

$$L = \frac{QV_{Rated}^2}{2\pi F_{Rated}\left(P^2 + Q^2\right)},$$

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, φ, to the steady-state current:

$$L(i) = \frac{\phi(i)}{i}.$$

Therefore the current-voltage relationship for the inductor is:

$$v = \frac{dL}{dt}i + L\frac{di}{dt}.$$

Parameters

- “Main Tab” on page 1-763
- “Parasitics” on page 1-764
- “Variables Tab” on page 1-764

Main Tab

Rated voltage

RMS, rated line-line voltage for the resistance equation. The default value is `24e3 V`.

Rated electrical frequency

Nominal AC electrical frequency for the inductance equation. The default value is `60 Hz`.

Minimum real power

Minimum real power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0. The default value is `1e3 W`.

Minimum reactive power

Minimum reactive power that the three-phase load dissipates when supplied at the rated voltage. The value must be greater than 0. The default value is `1e3 V*A`.

Parasitics

Parasitic parallel conductance

Conductance that the block adds, in parallel, to the series RL. The default value is `1e-6 1/Ohm`.

Variables Tab

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” (Simscape) .

Unlike block parameters, variables do not have conditional visibility. The **Variables** tab lists all the existing block variables. If a variable is not used in the set of equations corresponding to the selected block configuration, the values specified for this variable are ignored.

Ports

The block has the following ports:

`P`

Physical signal input port for real power

`Q`

Physical signal input port for reactive power

`~`

Expandable three-phase port

`n`

Electrical conserving port associated with the neutral phase.

See Also

Wye-Connected Load | Wye-Connected Variable Load

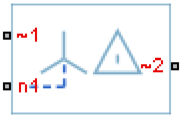
Topics

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

Introduced in R2014b

Wye-Delta1 Transformer

Linear nonideal wye-delta1 transformer with three-limb core



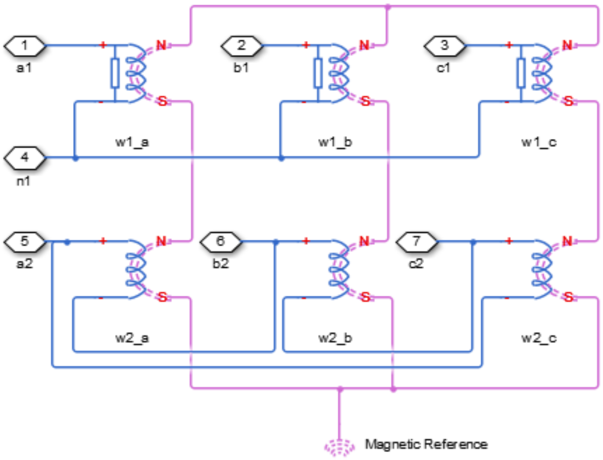
Library

Passive Devices / Transformers

Description

The Wye-Delta1 Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a wye connection and the secondary windings are configured in a delta connection. The delta voltages lag the wye voltages by 30 degrees, hence the name 1 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the wye-delta1 transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the primary neutral point.
- `w1`$_b$ is the primary winding connected between the b-phase and the primary neutral point.
- `w1`$_c$ is the primary winding connected between the c-phase and the primary neutral point.
- `w2`$_a$ is the secondary winding connected between the a-phase and the b-phase.
- `w2`$_b$ is the secondary winding connected between the b-phase and the c-phase.
- `w2`$_c$ is the secondary winding connected between the c-phase and the a-phase.

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 **Power Systems** > **Display Base Values**.

Parameters

- “Main Tab” on page 1-768

- “Impedances Tab” on page 1-768
- “Initial Conditions Tab” on page 1-769

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Secondary rated voltage

RMS line voltage applied to the secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Secondary leakage resistance (pu)

Power loss in the secondary winding. The default value is `0.01`.

Secondary leakage reactance (pu)

Magnetic flux loss in the secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial secondary currents

Current through the secondary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0]` `Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for secondary winding

`n1`

Electrical conserving port associated with the primary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

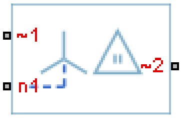
Topics

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

Introduced in R2013b

Wye-Delta11 Transformer

Linear nonideal wye-delta11 transformer with three-limb core



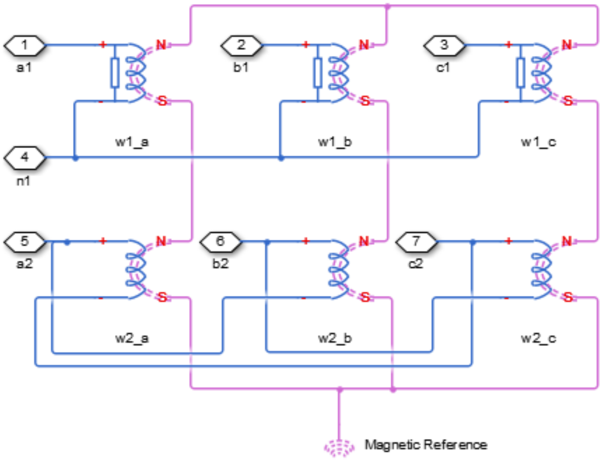
Library

Passive Devices / Transformers

Description

The Wye-Delta11 Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a wye connection and the secondary windings are configured in a delta connection. The delta voltages lead the wye voltages by 30 degrees, hence the name 11 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the wye-delta11 transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the primary neutral point.
- `w1`$_b$ is the primary winding connected between the b-phase and the primary neutral point.
- `w1`$_c$ is the primary winding connected between the c-phase and the primary neutral point.
- `w2`$_a$ is the secondary winding connected between the a-phase and the c-phase.
- `w2`$_b$ is the secondary winding connected between the b-phase and the a-phase.
- `w2`$_c$ is the secondary winding connected between the c-phase and the b-phase.

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 **Power Systems** > **Display Base Values**.

Parameters

- “Main Tab” on page 1-773
- “Impedances Tab” on page 1-773

- “Initial Conditions Tab” on page 1-774

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Secondary rated voltage

RMS line voltage applied to the secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Secondary leakage resistance (pu)

Power loss in the secondary winding. The default value is `0.01`.

Secondary leakage reactance (pu)

Magnetic flux loss in the secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial secondary currents

Current through the secondary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0]` `Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for secondary winding

`n1`

Electrical conserving port associated with the primary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

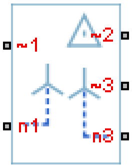
Topics

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

Introduced in R2013b

Wye-Delta1-Wye Transformer

Linear nonideal wye-delta1-wye transformer with three-limb core



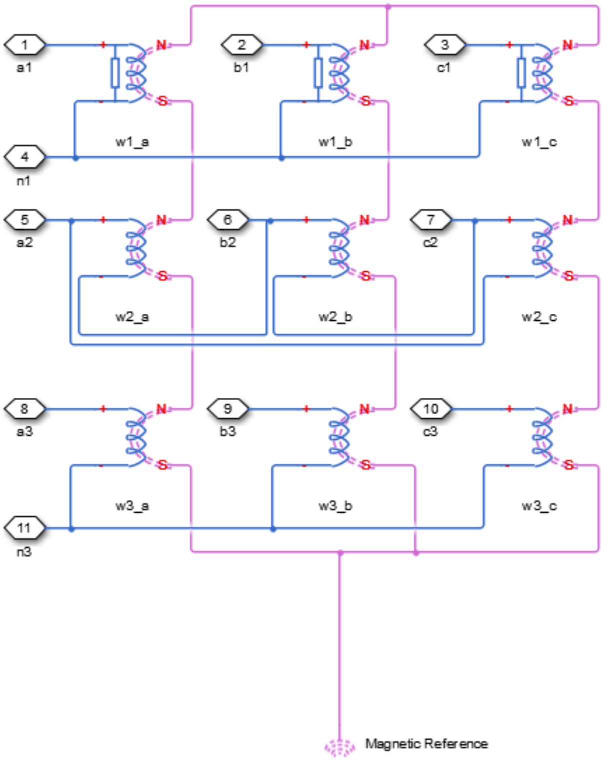
Library

Passive Devices / Transformers

Description

The Wye-Delta1-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a wye connection and there are delta secondary windings and wye secondary windings. The delta voltages lag the wye voltages by 30 degrees, hence the name 1 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the wye-delta1-wye transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the primary neutral point.
- `w1`$_b$ is the primary winding connected between the b-phase and the primary neutral point.
- `w1`$_c$ is the primary winding connected between the c-phase and the primary neutral point.
- `w2`$_a$ is the secondary winding connected between the a-phase and the b-phase.
- `w2`$_b$ is the secondary winding connected between the b-phase and the c-phase.
- `w2`$_c$ is the secondary winding connected between the c-phase and the a-phase.
- `w3`$_a$ is the secondary winding connected between the a-phase and the secondary neutral point.

- `w3`$_b$ is the secondary winding connected between the b-phase and the secondary neutral point.
- `w3`$_c$ is the secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-778
- "Impedances Tab" on page 1-779
- "Initial Conditions Tab" on page 1-779

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `24e3 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0]` `A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0]` `Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n1

Electrical conserving port associated with the primary winding neutral point

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

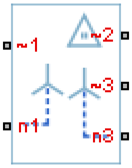
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Wye-Delta11-Wye Transformer

Linear nonideal wye-delta11-wye transformer with three-limb core



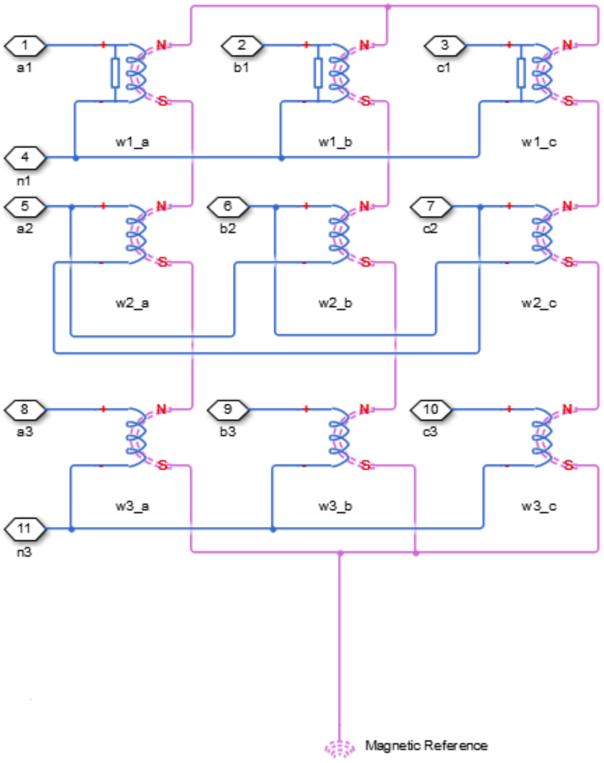
Library

Passive Devices / Transformers

Description

The Wye-Delta11-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a wye connection and there are delta secondary windings and wye secondary windings. The delta voltages lead the wye voltages by 30 degrees, hence the name 11 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the wye-delta11-wye transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the primary neutral point.
- `w1`$_b$ is the primary winding connected between the b-phase and the primary neutral point.
- `w1`$_c$ is the primary winding connected between the c-phase and the primary neutral point.
- `w2`$_a$ is the delta secondary winding connected between the a-phase and the c-phase.
- `w2`$_b$ is the delta secondary winding connected between the b-phase and the a-phase.
- `w2`$_c$ is the delta secondary winding connected between the c-phase and the b-phase.
- `w3`$_a$ is the wye secondary winding connected between the a-phase and the secondary neutral point.

- `w3`$_b$ is the wye secondary winding connected between the b-phase and the secondary neutral point.
- `w3`$_c$ is the wye secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-783
- "Impedances Tab" on page 1-784
- "Initial Conditions Tab" on page 1-784

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `24e3 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n1

Electrical conserving port associated with the primary winding neutral point

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

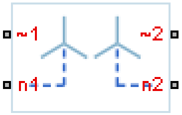
Topics

"Expand and Collapse Three-Phase Ports on a Block"

Introduced in R2013b

Wye-Wye Transformer

Linear nonideal wye-wye transformer with three-limb core



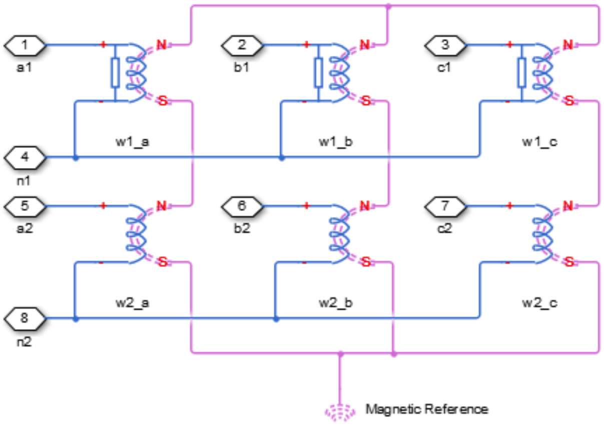
Library

Passive Devices / Transformers

Description

The Wye-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which both the primary and the secondary windings are configured in a wye connection. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the wye-wye transformer.



- `w1`$_a$ is the primary winding connected between the a-phase and the primary neutral point.
- `w1`$_b$ is the primary winding connected between the b-phase and the primary neutral point.
- `w1`$_c$ is the primary winding connected between the c-phase and the primary neutral point.
- `w2`$_a$ is the secondary winding connected between the a-phase and the secondary neutral point.
- `w2`$_b$ is the secondary winding connected between the b-phase and the secondary neutral point.
- `w2`$_c$ is the secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- "Main Tab" on page 1-787
- "Impedances Tab" on page 1-788
- "Initial Conditions Tab" on page 1-788

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `4160 V`.

Secondary rated voltage

RMS line voltage applied to the secondary winding under normal operating conditions. The default value is `24e3 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see "Per-Unit System of Units"

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Secondary leakage resistance (pu)

Power loss in the secondary winding. The default value is `0.01`.

Secondary leakage reactance (pu)

Magnetic flux loss in the secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial secondary currents

Current through the secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for secondary winding

`n1`

Electrical conserving port associated with the primary winding neutral point

`n2`

Electrical conserving port associated with the secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Zigzag-Delta1-Wye Transformer | Zigzag-Delta11-Wye Transformer

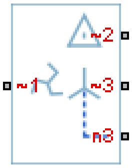
Topics

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

Introduced in R2013b

Zigzag-Delta1-Wye Transformer

Linear nonideal zigzag-delta1-wye transformer with three-limb core



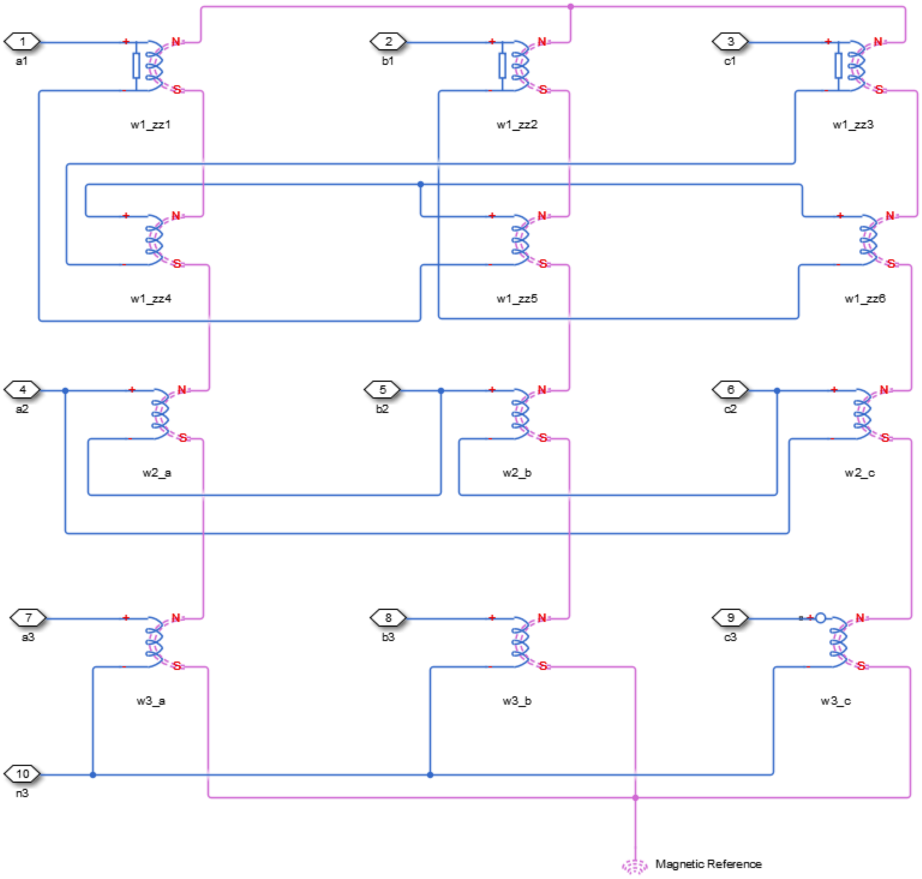
Library

Passive Devices / Transformers

Description

The Zigzag-Delta1-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a zigzag connection and there are delta secondary windings and wye secondary windings. You can specify the phase offset between the zigzag and wye windings via a block parameter. The delta voltages lag the wye voltages by 30 degrees, hence the name one o’clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the zigzag-delta1-wye transformer.



- `w1`$_{zz1}$ is the primary winding located on the first limb of the core, connected between the a-phase and the negative terminal of winding `w1`$_{zz5}$.
- `w1`$_{zz2}$ is the primary winding located on the second limb of the core, connected between the b-phase and the negative terminal of winding `w1`$_{zz6}$.
- `w1`$_{zz3}$ is the primary winding located on the third limb of the core, connected between the c-phase and the negative terminal of winding `w1`$_{zz4}$.
- `w1`$_{zz4}$ is the primary winding located on the first limb of the core, connected between the zigzag neutral point and the negative terminal of winding `w1`$_{zz3}$.

- $w1_{zz5}$ is the primary winding located on the second limb of the core, connected between the zigzag neutral point and the negative terminal of winding $w1_{zz1}$.
- $w1_{zz6}$ is the primary winding located on the third limb of the core, connected between the zigzag neutral point and the negative terminal of winding $w1_{zz2}$.
- $w2_a$ is the secondary winding connected between the a-phase and the b-phase.
- $w2_b$ is the secondary winding connected between the b-phase and the c-phase.
- $w2_c$ is the secondary winding connected between the c-phase and the a-phase.
- $w3_a$ is the secondary winding connected between the a-phase and the secondary neutral point.
- $w3_b$ is the secondary winding connected between the b-phase and the secondary neutral point.
- $w3_c$ is the secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- “Main Tab” on page 1-792
- “Impedances Tab” on page 1-793
- “Initial Conditions Tab” on page 1-794

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6` `V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `24e3` `V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `4160 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `4160 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Wye secondary phase shift

The phase offset between the zigzag and wye secondary windings. The default value is `-7.5 deg`.

Impedances Tab

Parameters in this tab are expressed in per unit (pu). For more information, see “Per-Unit System of Units”.

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta11-Wye Transformer

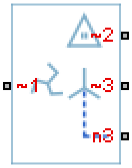
Topics

"Expand and Collapse Three-Phase Ports on a Block"
Custom Zigzag Transformer

Introduced in R2015a

Zigzag-Delta11-Wye Transformer

Linear nonideal zigzag-delta11-wye transformer with three-limb core



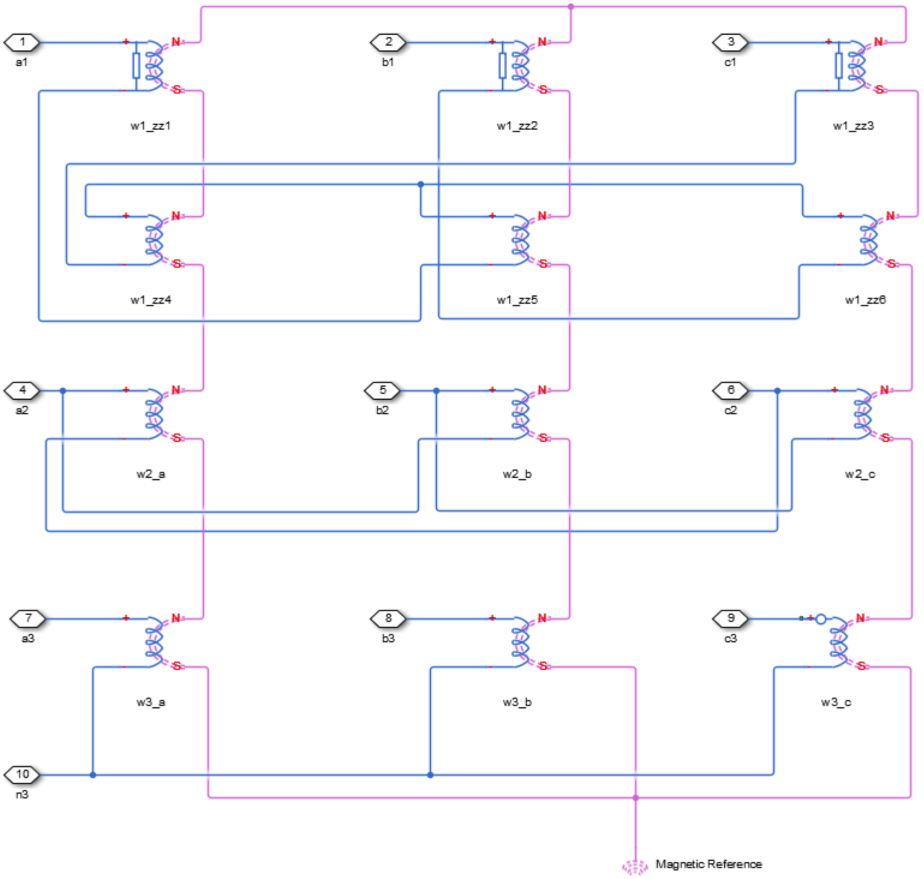
Library

Passive Devices / Transformers

Description

The Zigzag-Delta11-Wye Transformer block models a linear, nonideal transformer with a three-limb core, in which the primary windings are configured in a zigzag connection and there are delta secondary windings and wye secondary windings. You can specify the phase offset between the zigzag and wye windings via a block parameter. The delta voltages lead the wye voltages by 30 degrees, hence the name 11 o'clock delta. The block includes linear winding leakage and linear core magnetization effects.

The figure shows the equivalent circuit diagram for the zigzag-delta11-wye transformer.



- `w1`$_{zz1}$ is the primary winding located on the first limb of the core, connected between the a-phase and the negative terminal of winding `w1`$_{zz5}$.
- `w1`$_{zz2}$ is the primary winding located on the second limb of the core, connected between the b-phase and the negative terminal of winding `w1`$_{zz6}$.
- `w1`$_{zz3}$ is the primary winding located on the third limb of the core, connected between the c-phase and the negative terminal of winding `w1`$_{zz4}$.
- `w1`$_{zz4}$ is the primary winding located on the first limb of the core, connected between the zigzag neutral point and the negative terminal of winding `w1`$_{zz3}$.

- `w1`$_{zz5}$ is the primary winding located on the second limb of the core, connected between the zigzag neutral point and the negative terminal of winding `w1`$_{zz1}$.
- `w1`$_{zz6}$ is the primary winding located on the third limb of the core, connected between the zigzag neutral point and the negative terminal of winding `w1`$_{zz2}$.
- `w2`$_{a}$ is the secondary winding connected between the a-phase and the c-phase.
- `w2`$_{b}$ is the secondary winding connected between the b-phase and the a-phase.
- `w2`$_{c}$ is the secondary winding connected between the c-phase and the b-phase.
- `w3`$_{a}$ is the secondary winding connected between the a-phase and the secondary neutral point.
- `w3`$_{b}$ is the secondary winding connected between the b-phase and the secondary neutral point.
- `w3`$_{c}$ is the secondary winding connected between the c-phase and the secondary neutral point.

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 **Power Systems** > **Display Base Values**.

Parameters

- “Main Tab” on page 1-798
- “Impedances Tab” on page 1-799
- “Initial Conditions Tab” on page 1-800

Main Tab

Rated apparent power

Apparent power flowing through the transformer when operating at rated capacity. The default value is `100e6 V*A`.

Primary rated voltage

RMS line voltage applied to the primary winding under normal operating conditions. The default value is `24e3 V`.

Delta secondary rated voltage

RMS line voltage applied to the delta secondary winding under normal operating conditions. The default value is `4160 V`.

Wye secondary rated voltage

RMS line voltage applied to the wye secondary winding under normal operating conditions. The default value is `4160 V`.

Rated electrical frequency

Rated or nominal frequency of the AC network to which the transformer is connected. The default value is `60 Hz`.

Wye secondary phase shift

The phase offset between the zigzag and wye secondary windings. The default value is `-7.5 deg`.

Impedances Tab

Parameters in this tab are expressed in per-unit (pu). For more information, see "Per-Unit System of Units".

Primary leakage resistance (pu)

Power loss in the primary winding. The default value is `0.01`.

Primary leakage reactance (pu)

Magnetic flux loss in the primary winding. The default value is `0.001`.

Delta secondary leakage resistance (pu)

Power loss in the delta secondary winding. The default value is `0.01`.

Delta secondary leakage reactance (pu)

Magnetic flux loss in the delta secondary winding. The default value is `0.001`.

Wye secondary leakage resistance (pu)

Power loss in the wye secondary winding. The default value is `0.01`.

Wye secondary leakage reactance (pu)

Magnetic flux loss in the wye secondary winding. The default value is `0.001`.

Shunt magnetizing resistance (pu)

Magnetic losses in transformer core. The default value is `500`.

Shunt magnetizing reactance (pu)

Magnetic effects of the transformer core when operating in its linear region. The default value is `500`.

Initial Conditions Tab

Initial primary currents

Current through the primary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial delta secondary currents

Current through the delta secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial wye secondary currents

Current through the wye secondary leakage inductors at time zero. The default value is `[0, 0, 0] A`.

Initial fluxes

Magnetic fluxes in the limbs of the core at time zero. The default value is `[0, 0, 0] Wb`.

Ports

The block has the following ports:

`~1`

Expandable three-phase port for primary winding

`~2`

Expandable three-phase port for delta secondary winding

`~3`

Expandable three-phase port for wye secondary winding

n3

Electrical conserving port associated with the wye secondary winding neutral point

See Also

Delta-Delta Transformer | Delta1-Delta1-Wye Transformer | Delta11-Delta11-Wye Transformer | Wye-Delta1 Transformer | Wye-Delta1-Wye Transformer | Wye-Delta11 Transformer | Wye-Delta11-Wye Transformer | Wye-Wye Transformer | Zigzag-Delta1-Wye Transformer

Topics

"Expand and Collapse Three-Phase Ports on a Block"
Custom Zigzag Transformer

Introduced in R2015a

2

Functions — Alphabetical List

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:

$$M = \frac{harmonic\ magnitude}{\sqrt{2}},$$

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^{th} harmonic order.
- M is V_{RMS} or I_{RMS} as required.

You can use the `pe_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
```

- "Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block"
- "Choose a Simscape Power Systems Function for an Offline Harmonic Analysis"
- "Data Logging" (Simscape)
- "Harmonic Analysis of a Three-Phase Rectifier"

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^{th} 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`

See Also

Blocks
Spectrum Analyzer

Functions
`pe_getHarmonics` | `pe_plotHarmonics`

Using Objects
`simscape.logging.Node`

Topics
"Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block"
"Choose a Simscape Power Systems Function for an Offline Harmonic Analysis"
"Data Logging" (Simscape)
"Harmonic Analysis of a Three-Phase Rectifier"

Introduced in R2014a

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” (Simscape). 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 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 `pe_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_{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, for the MOSFET 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 variants. Thermal variants 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 variant, 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 variant 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.

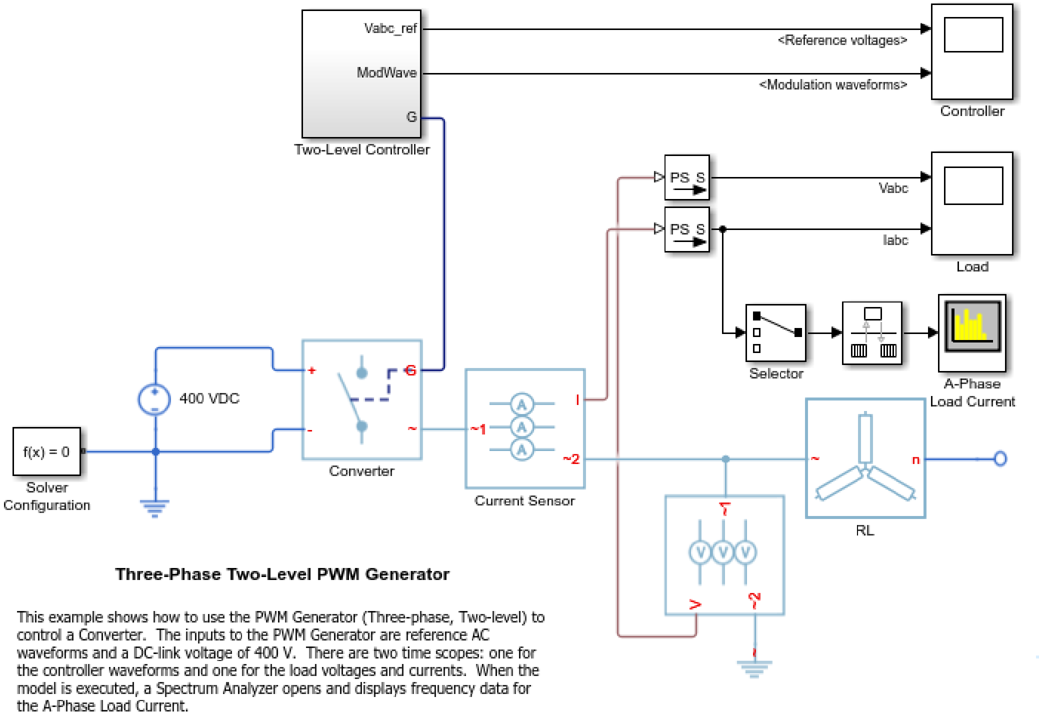
Examples

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 = 'pe_pwm_two_level';
open_system(model)
```



Three-Phase Two-Level PWM Generator

This example shows how to use the PWM Generator (Three-phase, Two-level) to control a Converter. The inputs to the PWM Generator are reference AC waveforms and a DC-link voltage of 400 V. There are two time scopes: one for the controller waveforms and one for the load voltages and currents. When the model is executed, a Spectrum Analyzer opens and displays frequency data for the A-Phase Load Current.

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_pe_pwm_two_level* is saved in the workspace.

Calculate the efficiency percentage.

```
efficiency = pe_getEfficiency('Load',simlog_pe_pwm_two_level)
```

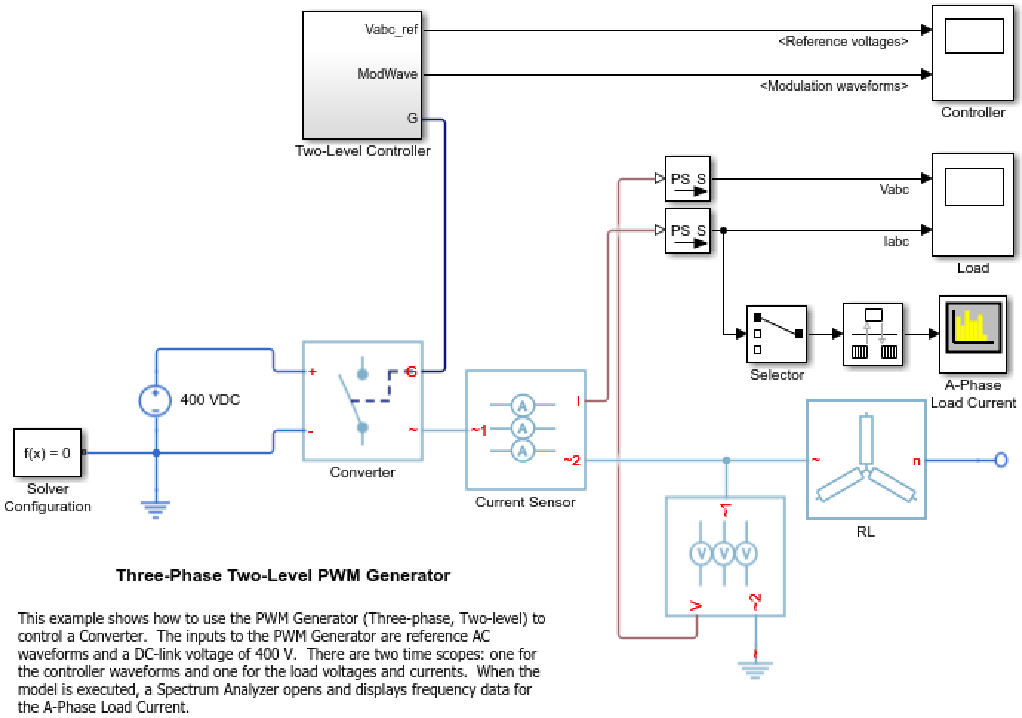
```
efficiency =

     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 = 'pe_pwm_two_level';
open_system(model)
```



Three-Phase Two-Level PWM Generator

This example shows how to use the PWM Generator (Three-phase, Two-level) to control a Converter. The inputs to the PWM Generator are reference AC waveforms and a DC-link voltage of 400 V. There are two time scopes: one for the controller waveforms and one for the load voltages and currents. When the model is executed, a Spectrum Analyzer opens and displays frequency data for the A-Phase Load Current.

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_pe_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_pe_pwm_two_level,0.000,0.005)
```

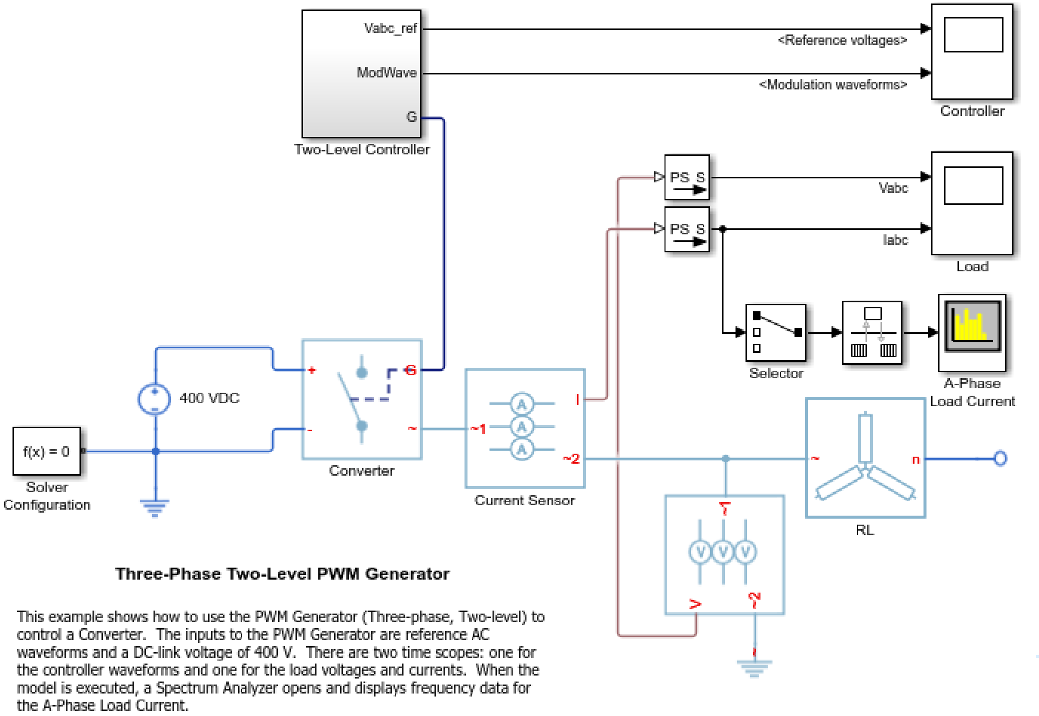
```
efficiency =

    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 = 'pe_pwm_two_level';
open_system(model)
```



Three-Phase Two-Level PWM Generator

This example shows how to use the PWM Generator (Three-phase, Two-level) to control a Converter. The inputs to the PWM Generator are reference AC waveforms and a DC-link voltage of 400 V. There are two time scopes: one for the controller waveforms and one for the load voltages and currents. When the model is executed, a Spectrum Analyzer opens and displays frequency data for the A-Phase Load Current.

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_pe_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_pe_pwm_two_level)
efficiency =

   99.1940


lossesTable =

  1×2 table

              LoggingNode                Power
    _______________________________    ______

    'pe_pwm_two_level.Converter'    268.73
```

- “Perform a Power-Loss Analysis”
- “Data Logging” (Simscape)
- “About the Simscape Results Explorer” (Simscape)

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
Block Type		RLC	RLC	Wye-Connected Load
Block Role in the Model		Source Impedance	Load Impedance	Load
`'loadIdentifier'`	`'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
Block Type		RLC	RLC	Wye-Connected Load
Block Role in the Model		Source Impedance	Load Impedance	Load
`'loadIdentifier'`	`'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_pe_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.

See Also

`pe_getPowerLossSummary` | `pe_getPowerLossTimeSeries` | `sscexplore`

Topics

“Perform a Power-Loss Analysis”
“Data Logging” (Simscape)
“About the Simscape Results Explorer” (Simscape)

Introduced in R2017a

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^{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.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_getHarmonics(simlog.Load.V)
```

Analyze 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_getHarmonics(simlog.Load.V,2,2.3,10,1,15)
```

Analyze 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 `12` periods of the signal, which has a bias of `1` V. The function analyzes the default number, `30`, of harmonics.

```
pe_getHarmonics(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. For information, see “Enable Data Logging for the Whole Model” (Simscape).

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 ith 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 2.3 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^{th} 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 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 **Type** to `Variable-step`	Enable the options to **Use local solver** and **Use fixed-cost runtime consistency iterations**
Global and local fixed-step	Set **Type** to `Fixed-step`	Enable the options to **Use local solver** and **Use fixed-cost runtime consistency iterations**
Global fixed-step	Set **Type** to `Fixed-step`	Clear the option to **Use local solver**

- 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.

See Also

Blocks
Spectrum Analyzer

Functions
`pe_calculateThdPercent` | `pe_plotHarmonics`

Using Objects
`simscape.logging.Node`

Topics
"Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block"
"Choose a Simscape Power Systems Function for an Offline Harmonic Analysis"
"Data Logging" (Simscape)

“Harmonic Analysis of a Three-Phase Rectifier”

Introduced in R2014a

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” (Simscape). 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_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 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 variants. Thermal variants 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 variant, 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 variant 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.

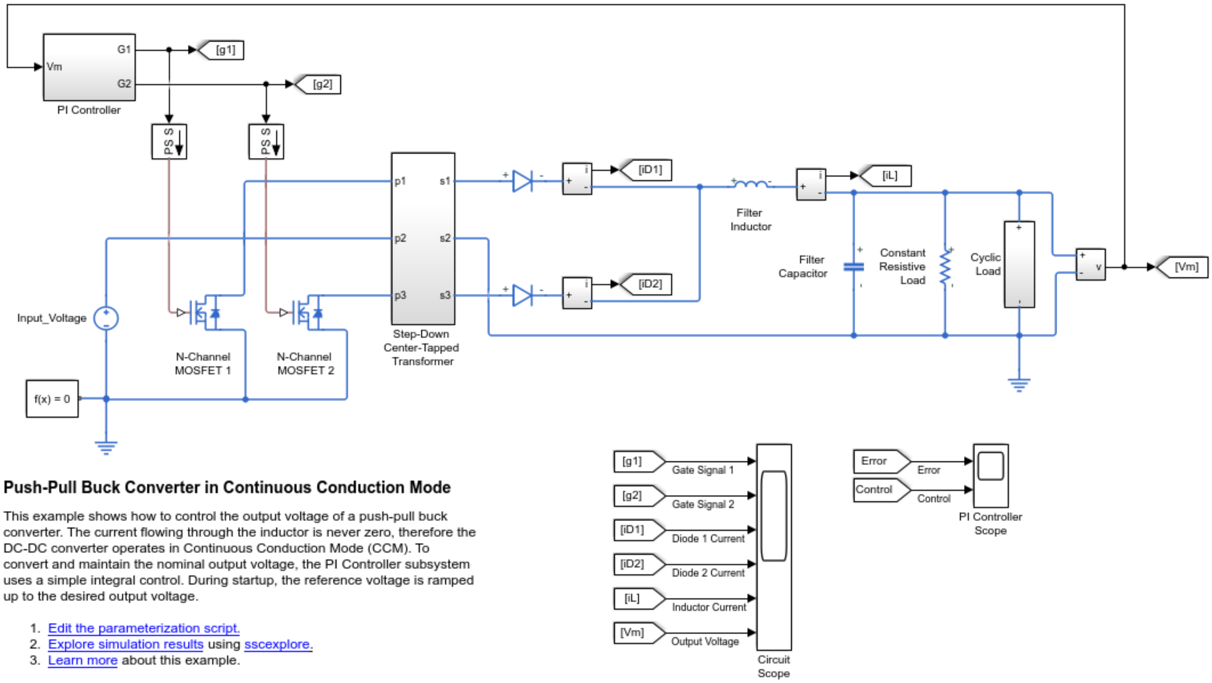
Examples

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 = 'pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

2. Run the simulation and create the simulation log variable.

```
sim(model);
```

The simulation log variable `simlog_pe_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_pe_push_pull_converter_ccm)
```

```
tabulatedLosses =

  4x2 table

                        LoggingNode                            Power  
    ________________________________________________________    _______
```

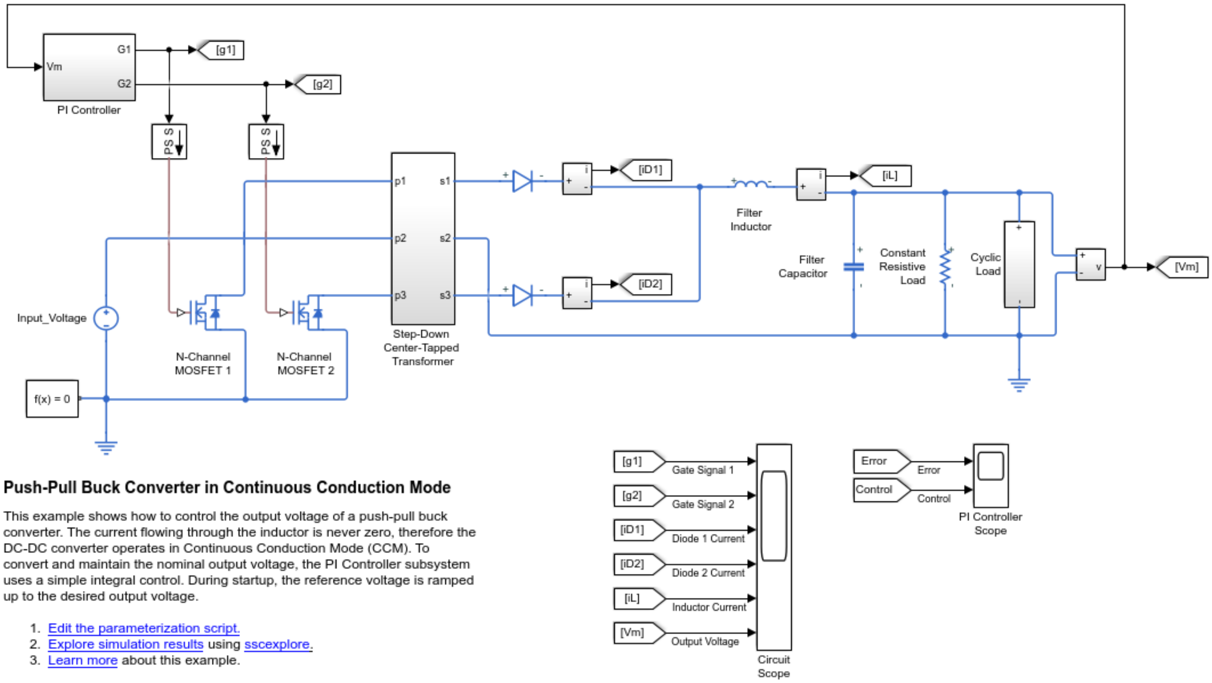
```
    'pe_push_pull_converter_ccm.Diode1'                              0.30937
    'pe_push_pull_converter_ccm.Diode'                                0.3092
    'pe_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation'  0.24497
    'pe_push_pull_converter_ccm.N_Channel_MOSFET_2.mosfet_equation'   0.2449
```

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 Single Block

1. Open the Push-Pull Buck Converter in Continuous Conduction Mode example model. At the MATLAB® command prompt, enter

```
model='pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

The model has data logging enabled.

2. Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable `simlog_pe_push_pull_converter_ccm` is saved in your workspace.

3. Calculate power losses for the N-Channel MOSFET 1 block and display the results in a table.

```
tabulatedLosses = pe_getPowerLossSummary(simlog_pe_push_pull_converter_ccm.N_Channel_MO
```

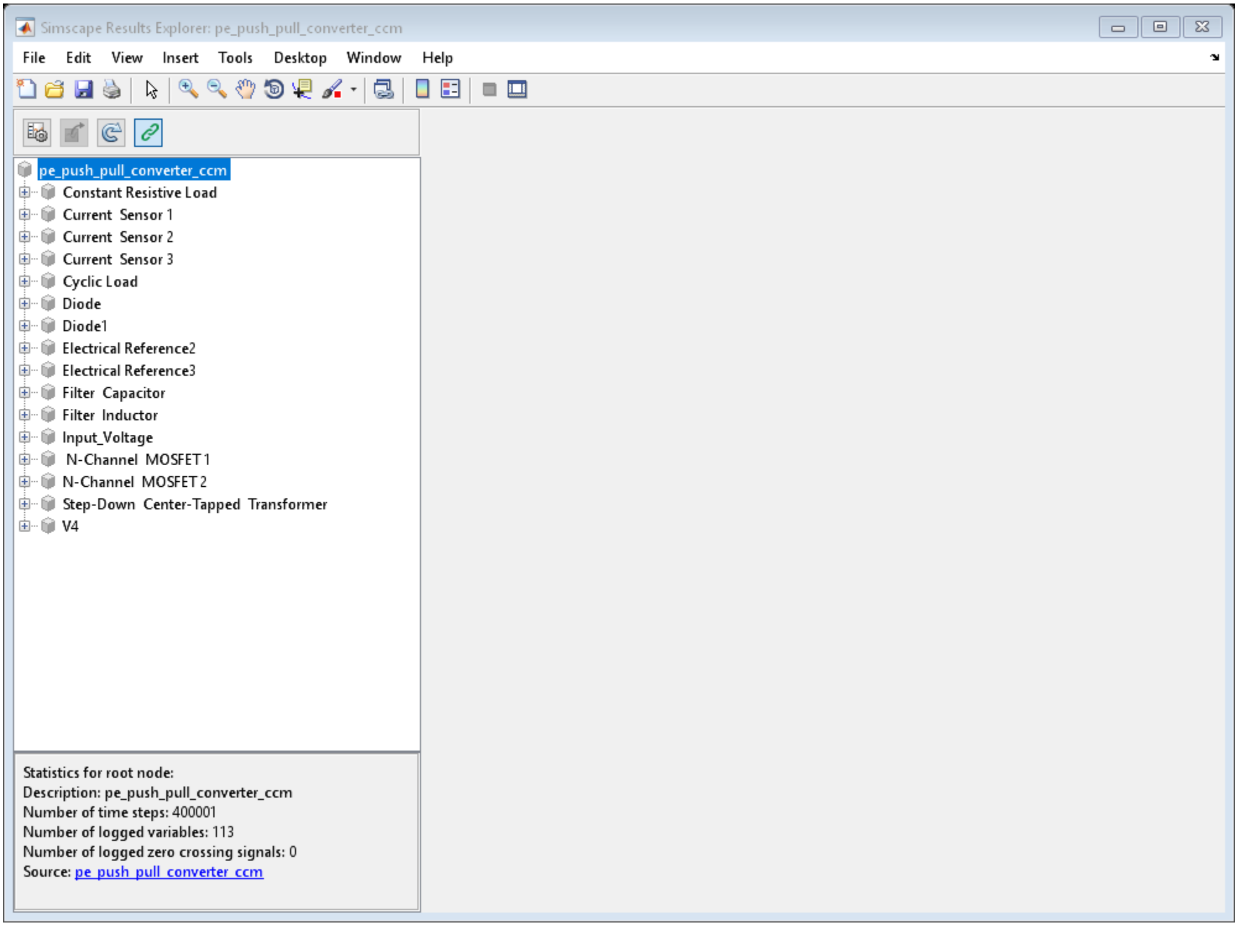
```
tabulatedLosses =

  1x2 table

              LoggingNode                    Power
    ____________________________________    _______

    'N_Channel_MOSFET_1.mosfet_equation'    0.24497
```

The table shows dissipated power losses for just the N-Channel MOSFET 1 block, averaged over the total simulation time.

4. Use the `sscexplore` function to explore the power loss data for the N-Channel MOSFET 1 block further.

`a. Open the Results Explorer.`

```
sscexplore(simlog_pe_push_pull_converter_ccm)
```

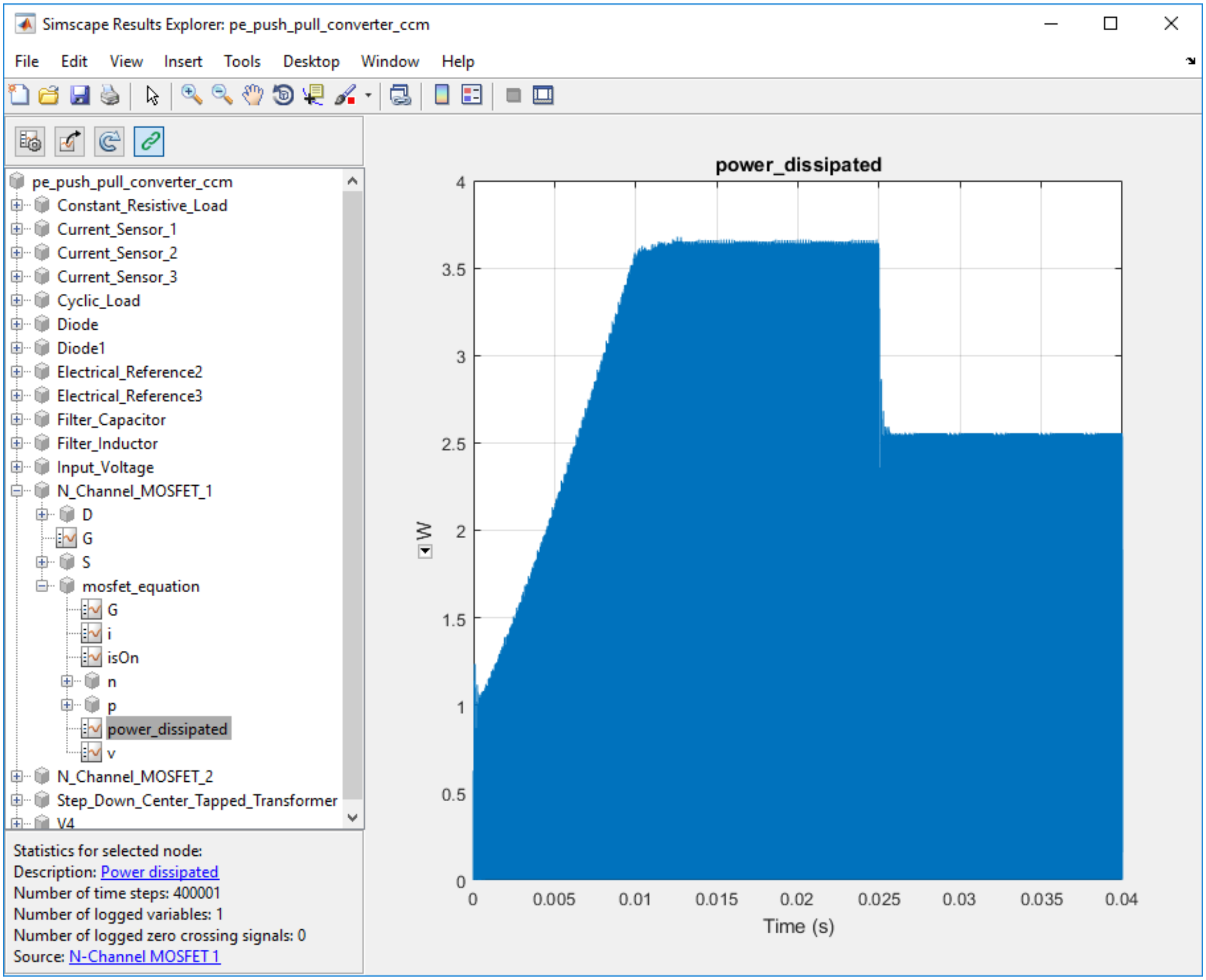


Statistics for root node:
Description: pe_push_pull_converter_ccm
Number of time steps: 400001
Number of logged variables: 113
Number of logged zero crossing signals: 0
Source: pe_push_pull_converter_ccm

`b. Expand these nodes:`

- `N_Channel_MOSFET_1`
- `mosfet_equation`

c. Click the `power_dissipated` node.



Statistics for selected node:
Description: Power dissipated
Number of time steps: 400001
Number of logged variables: 1
Number of logged zero crossing signals: 0
Source: N-Channel MOSFET 1

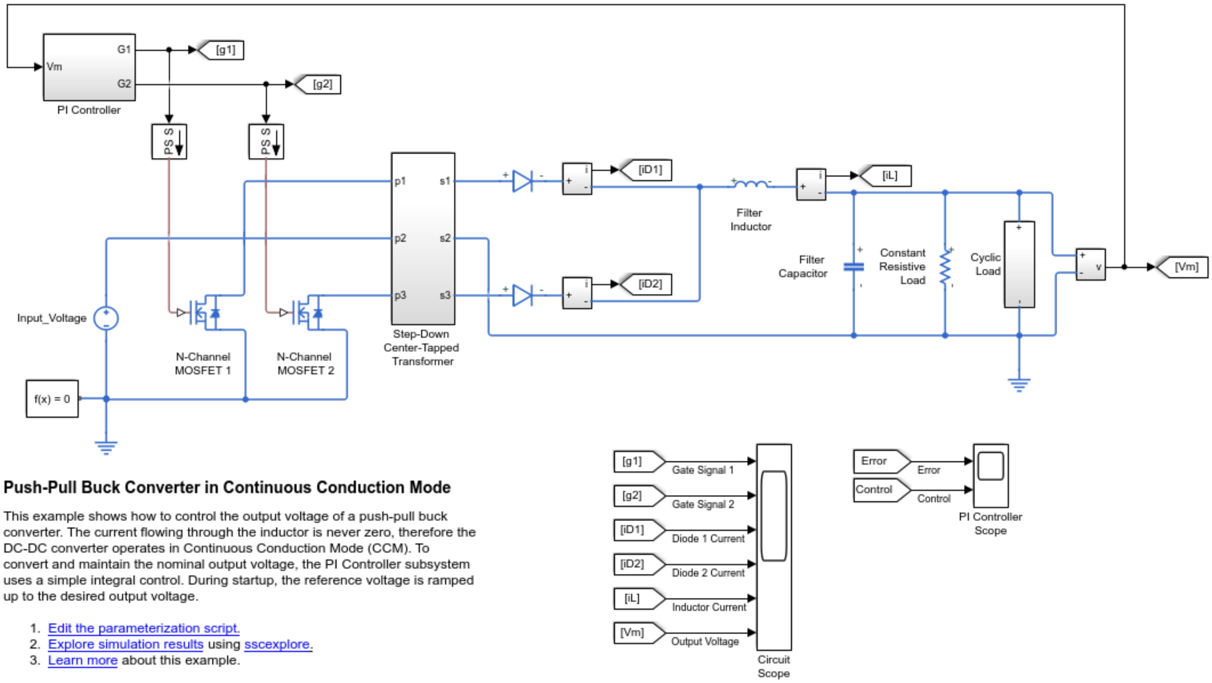
The N-Channel MOSFET 1 block has one `power_dissipated` variable for the `mosfet_equation` node. The power_dissipated figure shows the instantaneous power loss for the N-Channel MOSFET 1 block during the simulation.

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 = 'pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

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('pe_push_pull_converter_ccm/ N-Channel  MOSFET 1','diode_param','2')
```

Alternatively, you can add the component in the Simulink® Editor:

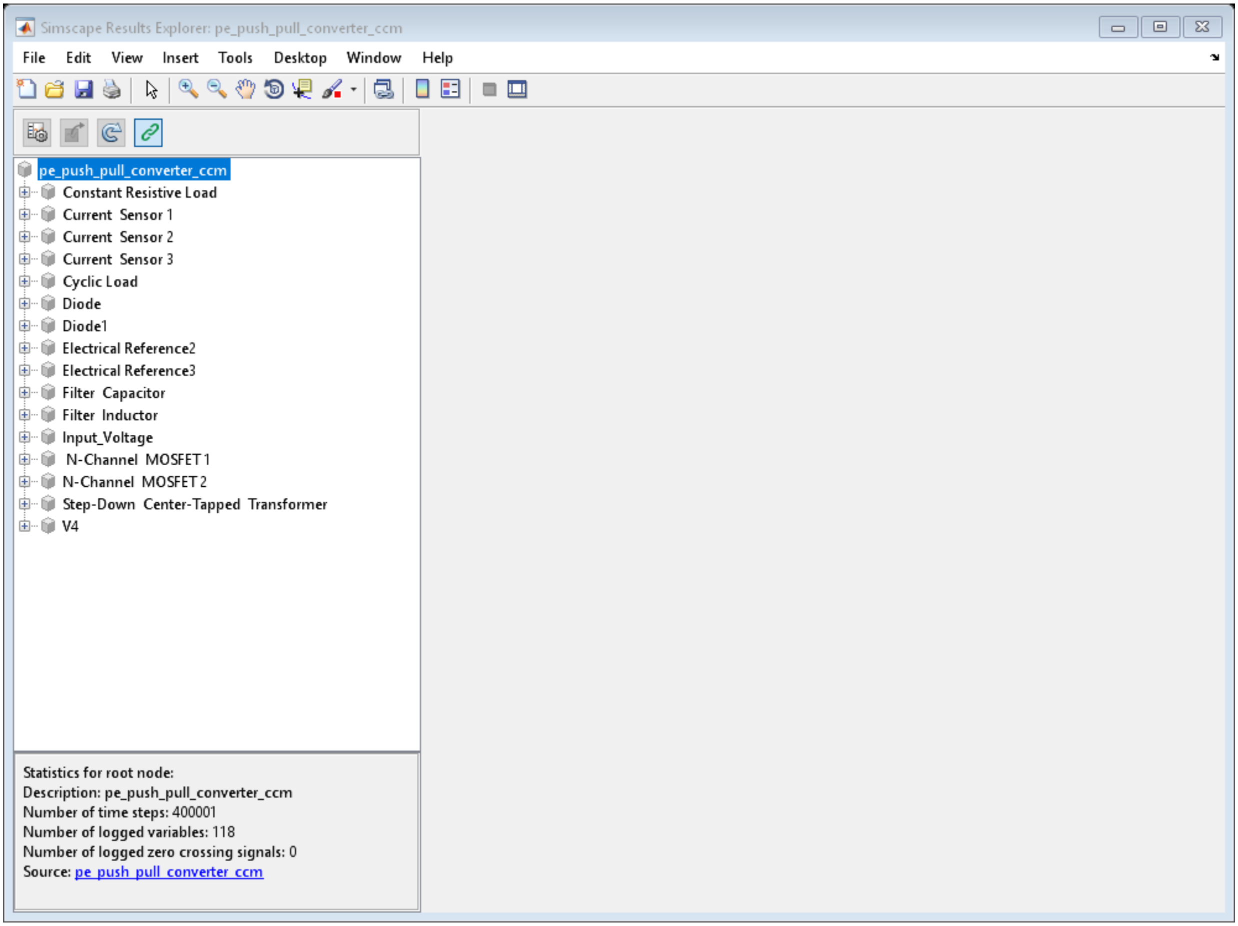
a. Open the Property Inspector pane. In the model window, in the menu bar, click **View > Property Inspector**

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.

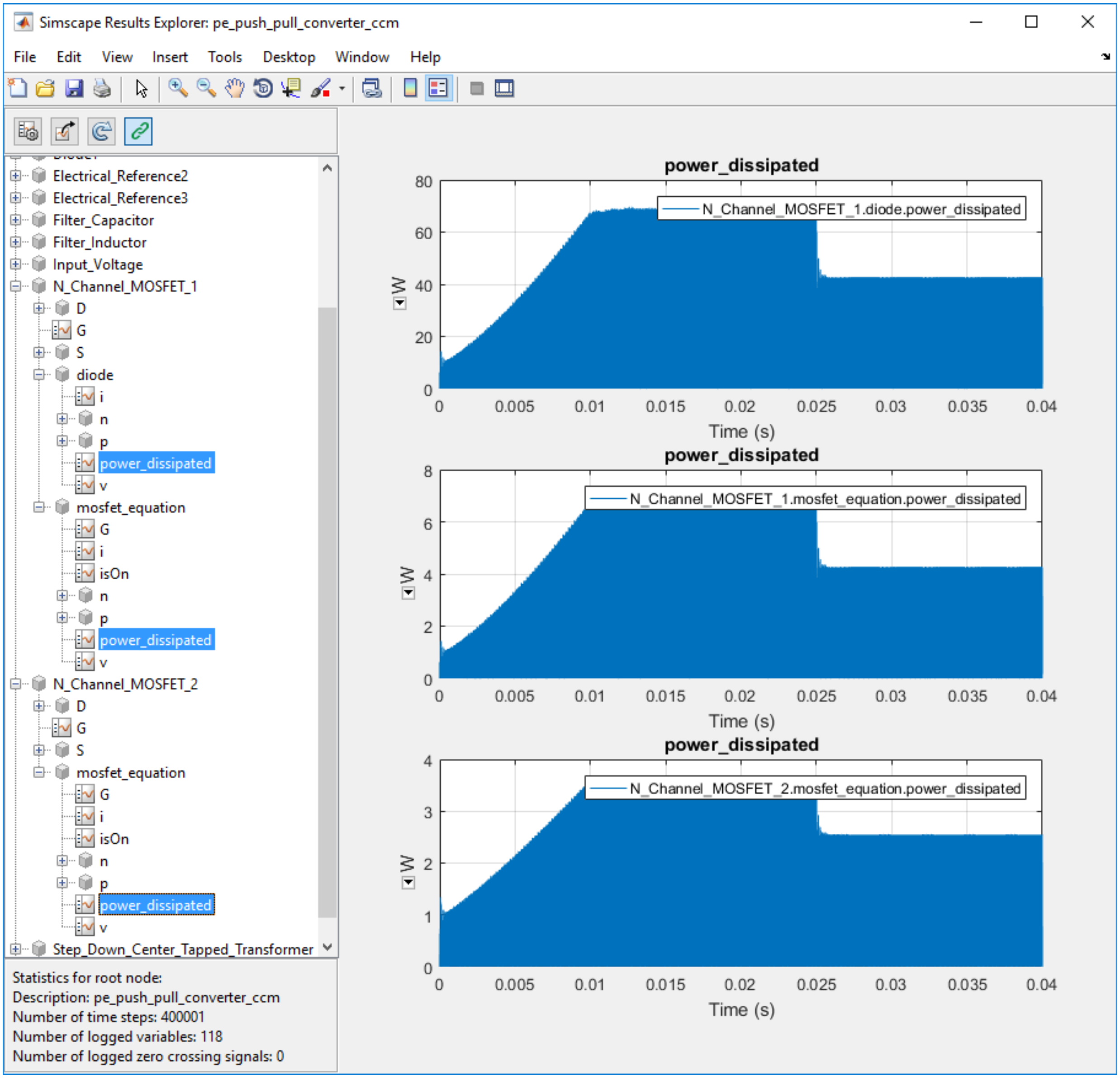
```
sim(model)
sscexplore(simlog_pe_push_pull_converter_ccm)
```



Statistics for root node:
Description: pe_push_pull_converter_ccm
Number of time steps: 400001
Number of logged variables: 118
Number of logged zero crossing signals: 0
Source: pe_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`



Statistics for root node:
Description: pe_push_pull_converter_ccm
Number of time steps: 400001
Number of logged variables: 118
Number of logged zero crossing signals: 0

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 = pe_getPowerLossSummary(simlog_pe_push_pull_converter_ccm.N_Channel_MO
```

```
tabulatedLosses =

  1x2 table

         LoggingNode          Power 
    ____________________    ______

    'N_Channel_MOSFET_1'    2.6075
```

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 = pe_getPowerLossSummary(simlog_pe_push_pull_converter_ccm.N_Channel_MO
```

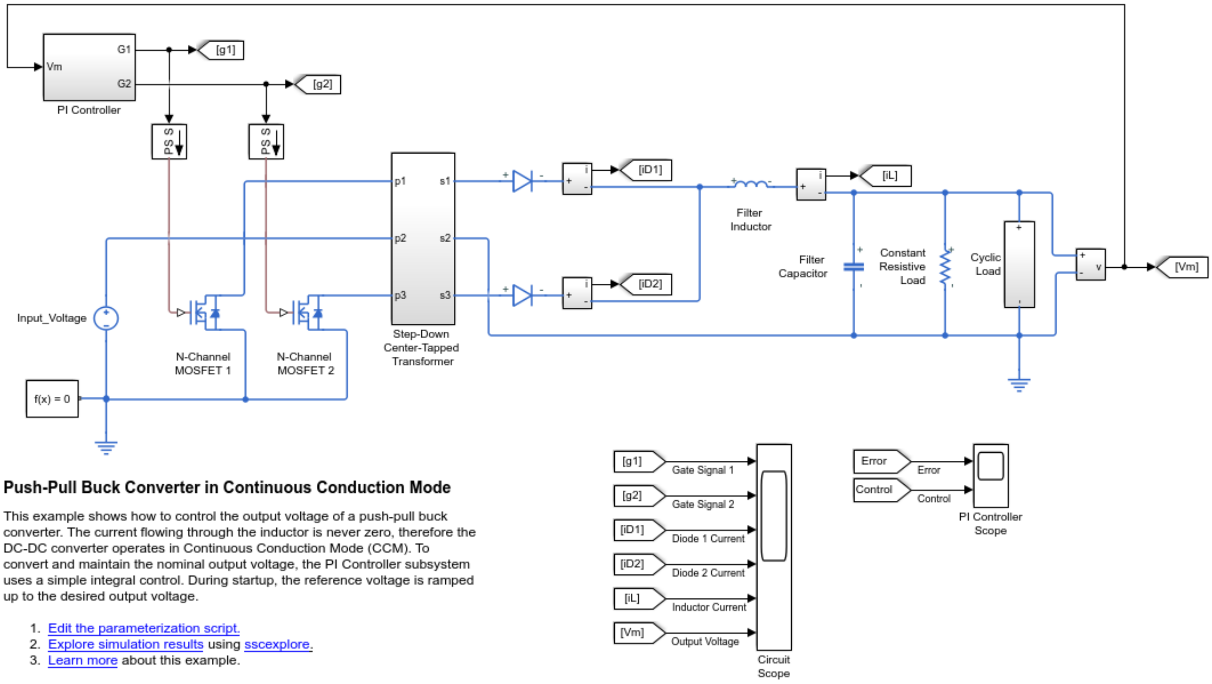
```
tabulatedLosses =

  1x2 table

    LoggingNode    Power 
    ___________    ______

    'diode'        2.3669
```

The table shows dissipated power losses only for the diode component of the block, averaged over the total simulation time.

Calculate Average Power Losses for a Specific Time Period

1. Open the Push-Pull Buck Converter in Continuous Conduction Mode example model. At the MATLAB® command prompt, enter

```
pe_push_pull_converter_ccm
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

The model has data logging enabled.

2. Run the simulation and create the simulation log variable.

```
sim('pe_push_pull_converter_ccm');
```

The simulation log variable `simlog_pe_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_pe_push_pull_converter_ccm,0.010,0.025)
```

```
tabulatedLosses = 

  4x2 table

                            LoggingNode                                   Power  
    _______________________________________________________________     _______

    'pe_push_pull_converter_ccm.Diode'                                   0.3703
    'pe_push_pull_converter_ccm.Diode1'                                 0.37022
    'pe_push_pull_converter_ccm.N_Channel_MOSFET_2.mosfet_equation'    0.26598
    'pe_push_pull_converter_ccm.N_Channel_MOSFET_1.mosfet_equation'    0.26595
```

The table shows dissipated power losses for each of the Diode and MOSFET blocks, averaged over the specified portion of simulation time.

- "Perform a Power-Loss Analysis"
- "Data Logging" (Simscape)
- "About the Simscape Results Explorer" (Simscape)

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.

See Also

`pe_getEfficiency` | `pe_getPowerLossTimeSeries` | `sscexplore`

Topics

“Perform a Power-Loss Analysis”
“Data Logging” (Simscape)
“About the Simscape Results Explorer” (Simscape)

Introduced in R2017a

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” (Simscape). 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 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 variants. Thermal variants 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 variant, 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 variant 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.

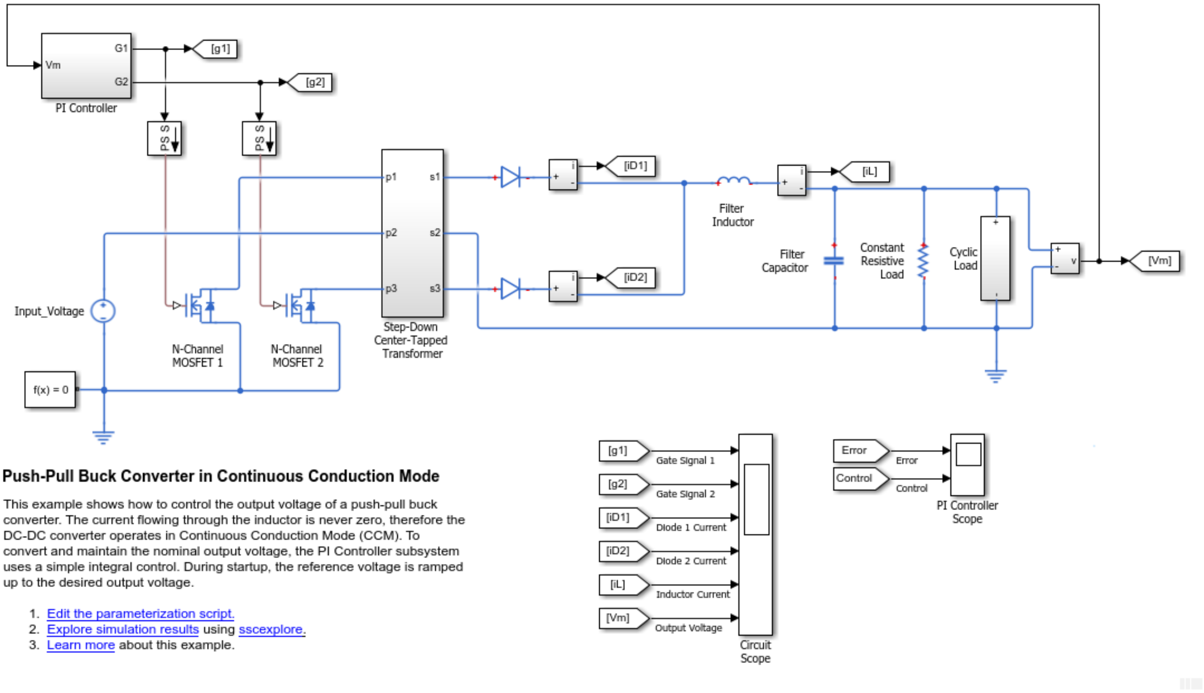
Examples

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 = 'pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable *simlog_pe_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_pe_push_pull_converter_ccm)
```

```
lossesCell =

  4×2 cell array

    'pe_push_pull_converter_ccm.N_Chann…'     [400001×3 double]
```

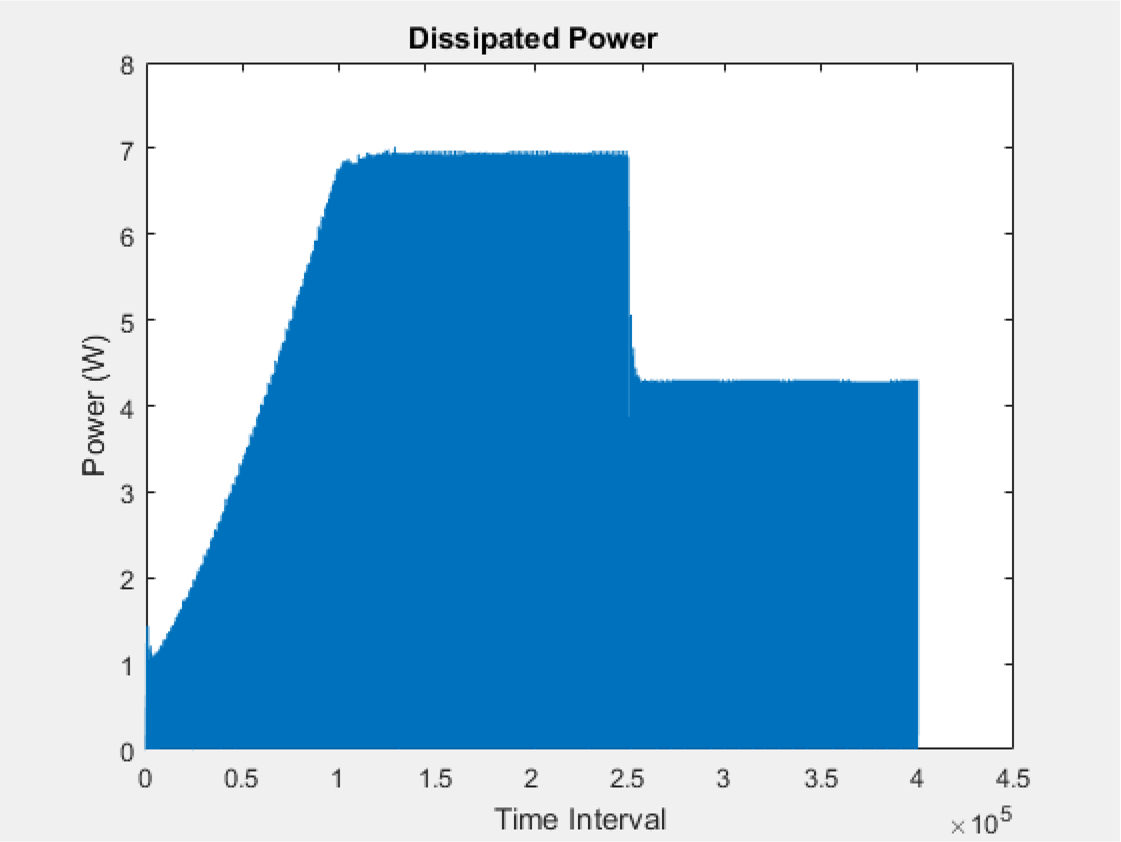
```
    'pe_push_pull_converter_ccm.N_Chann…'     [400001×3 double]
    'pe_push_pull_converter_ccm.Diode'        [400001×3 double]
    'pe_push_pull_converter_ccm.Diode1'       [400001×3 double]
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `400001x3 double` numeric array for the `pe_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)')
```

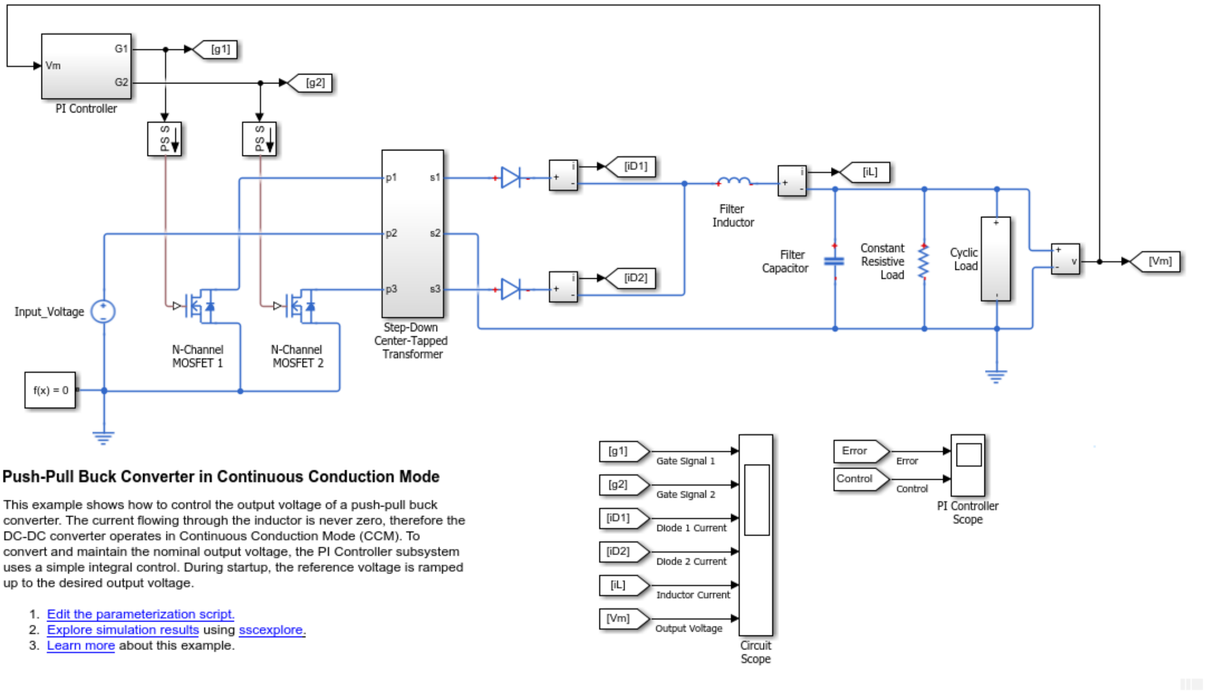


Calculate Dissipated Power Losses for a Specific Time Period

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 = 'pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable *simlog_pe_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_pe_push_pull_converter_ccm,0.010,0.025)
```

```
lossesCell =

  4×2 cell array

    'pe_push_pull_converter_ccm.N_Chann…'    [150002×3 double]
    'pe_push_pull_converter_ccm.N_Chann…'    [150002×3 double]
```

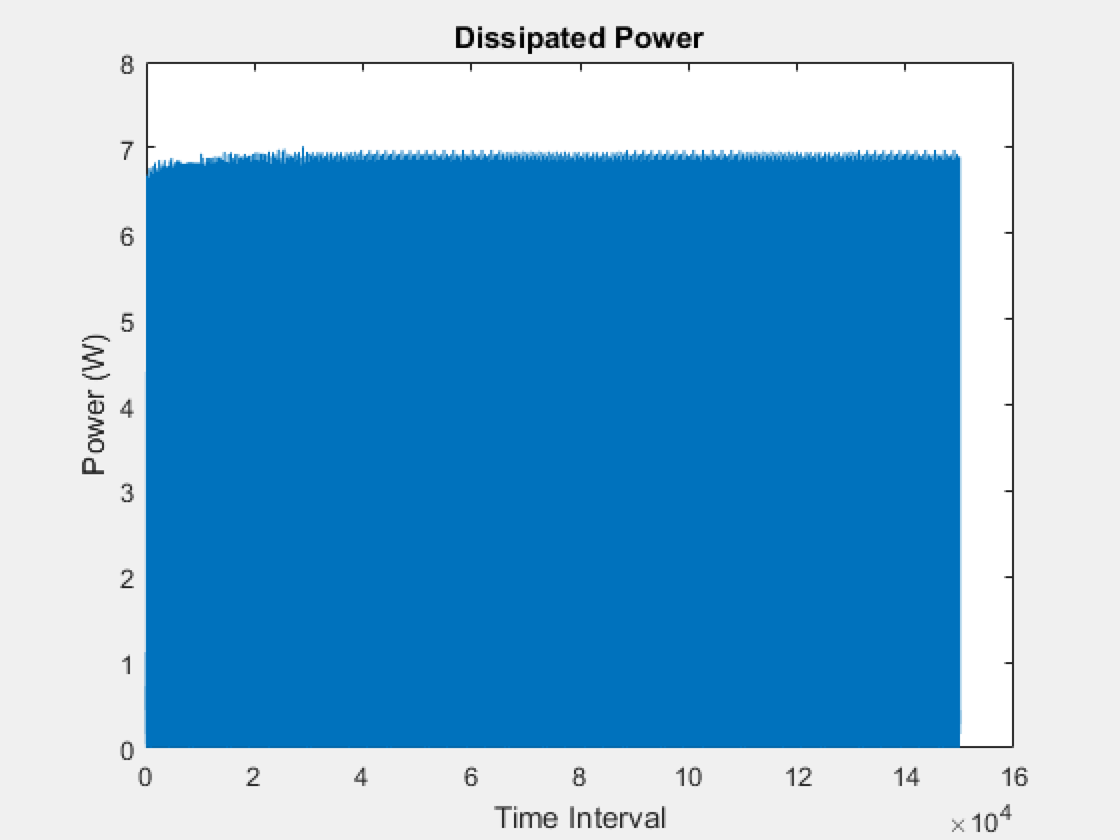
```
    'pe_push_pull_converter_ccm.Diode'        [150002×3 double]
    'pe_push_pull_converter_ccm.Diode1'       [150002×3 double]
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `150002x3 double` numeric array for the `pe_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)')
```

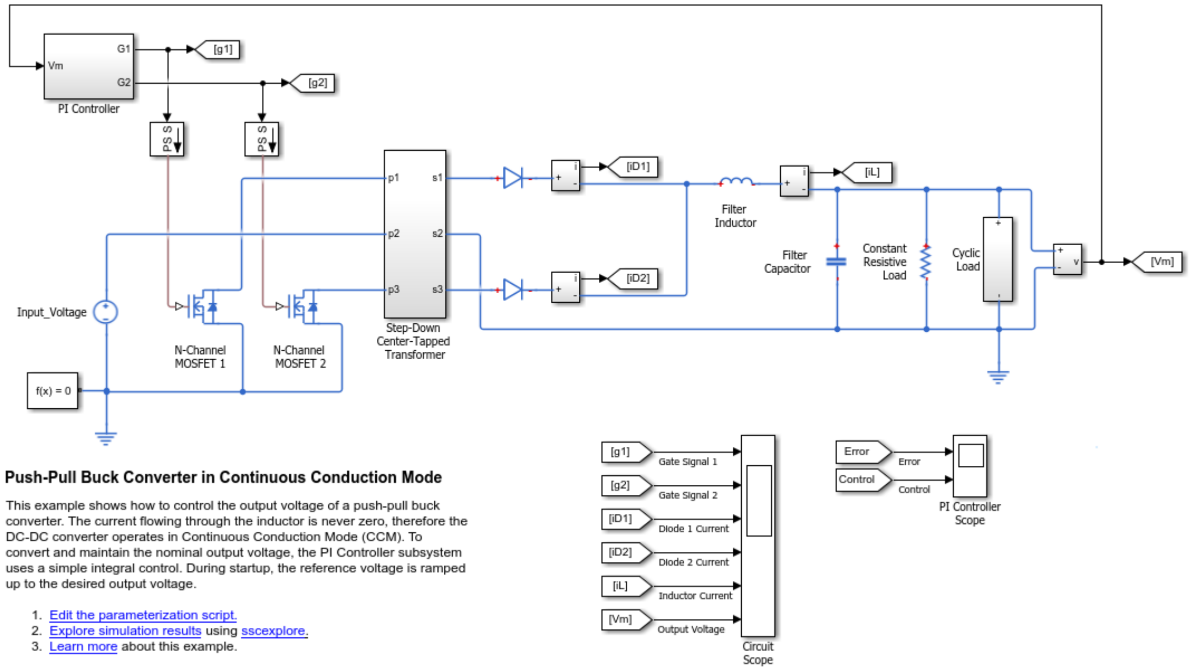


Calculate Dissipated Power Losses Using Specific Interval Widths

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 = 'pe_push_pull_converter_ccm';
open_system(model)
```



Push-Pull Buck Converter in Continuous Conduction Mode

This example shows how to control the output voltage of a push-pull buck converter. The current flowing through the inductor is never zero, therefore the DC-DC converter operates in Continuous Conduction Mode (CCM). To convert and maintain the nominal output voltage, the PI Controller subsystem uses a simple integral control. During startup, the reference voltage is ramped up to the desired output voltage.

1. Edit the parameterization script.
2. Explore simulation results using sscexplore.
3. Learn more about this example.

Run the simulation and create the simulation log variable.

```
sim(model)
```

The simulation log variable *simlog_pe_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_pe_push_pull_converter_ccm,0.010,0.025,1.
```

```
lossesCell =

  4×2 cell array

    'pe_push_pull_converter_ccm.N_…'    [136×3 double]
    'pe_push_pull_converter_ccm.N_…'    [136×3 double]
```

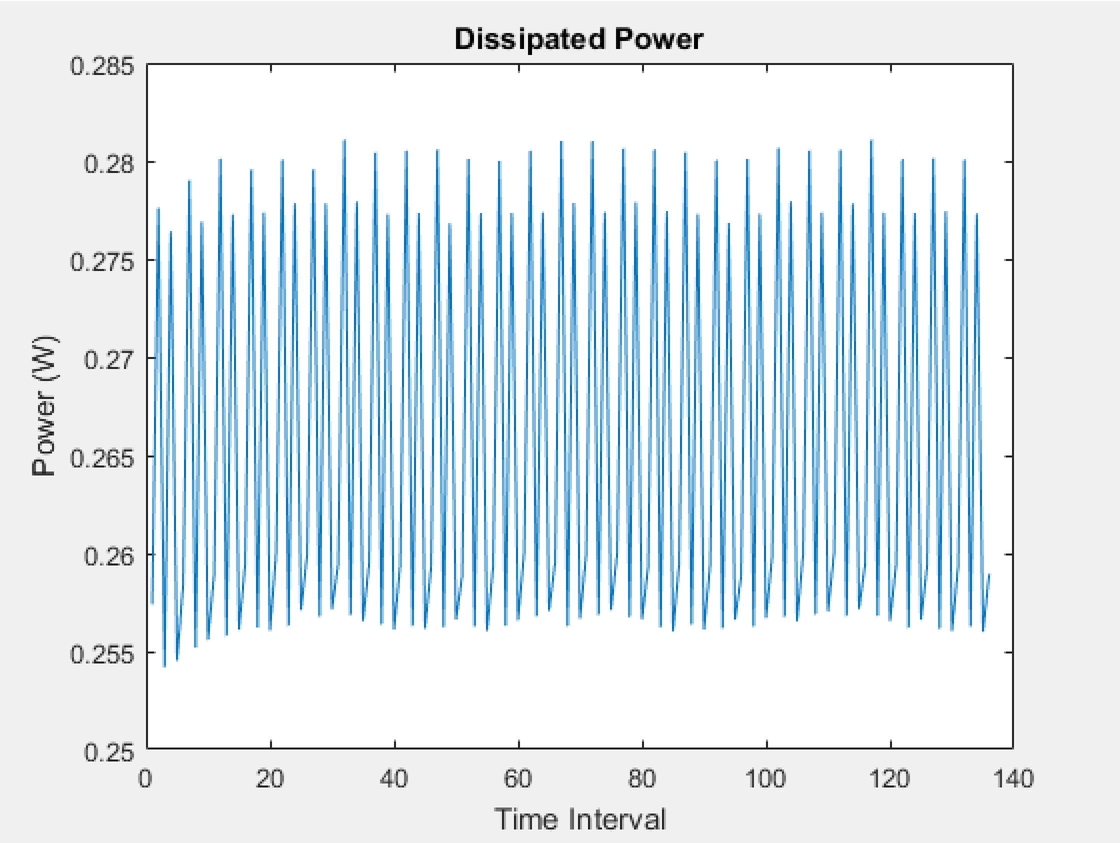
```
    'pe_push_pull_converter_ccm.Di…'    [136×3 double]
    'pe_push_pull_converter_ccm.Di…'    [136×3 double]
```

View the time series data. From the workspace, open the `lossesCell` cell array, then open the `136x3 double` numeric array for the `pe_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)')
```



- “Perform a Power-Loss Analysis”
- “Data Logging” (Simscape)
- “About the Simscape Results Explorer” (Simscape)

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_pe_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`

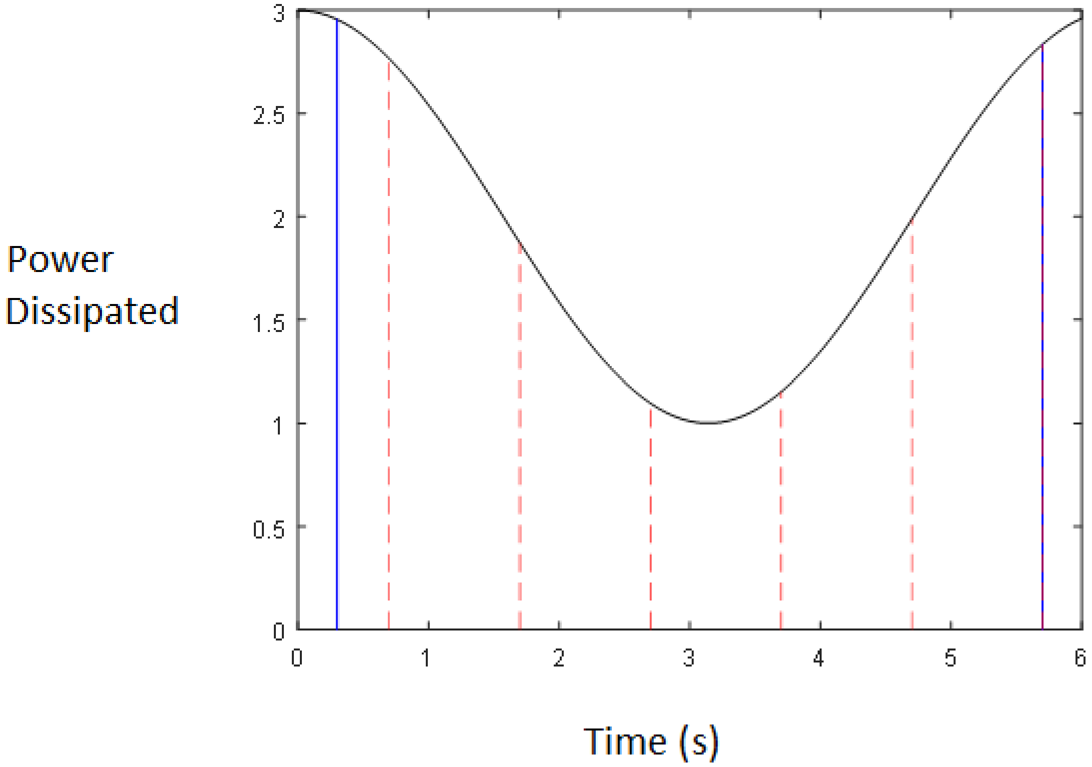
`endTime` — End of the time interval for calculating the data
simulation stop time (default) | real number

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.

See Also

`pe_getEfficiency` | `pe_getPowerLossSummary` | `sscexplore`

Topics

“Perform a Power-Loss Analysis”
“Data Logging” (Simscape)
“About the Simscape Results Explorer” (Simscape)

Introduced in R2017a

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^{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” (Simscape).

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 ith 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 `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`

See Also

Blocks
Spectrum Analyzer

Functions
`pe_calculateThdPercent` | `pe_getHarmonics`

Using Objects
`simscape.logging.Node`

Topics
"Perform an Online Harmonic Analysis Using the Simscape Spectrum Analyzer Block"
"Choose a Simscape Power Systems Function for an Offline Harmonic Analysis"
"Data Logging" (Simscape)
"Harmonic Analysis of a Three-Phase Rectifier"

Introduced in R2014a

Abbreviations and Naming Conventions in Simscape Components Libraries

B	Susceptance.
C	Capacitance.
composite three-phase port	Three-phase electrical conserving port, i.e., a port that represents three electrical conserving ports with a single connection. You can use composite three-phase ports to build models corresponding to single-line diagrams of three-phase electrical systems. Instead of explicitly connecting each phase of the three-phase system between blocks, you connect all three phases using a single port.
delta connection	Three-phase winding configuration. Each of the three windings is connected between phases. Physically, the connection resembles the Greek capital letter Δ. For a delta connection, phase shifts can be specified in terms of the hours of a clock. An 11 o’clock delta connection represents a 30 degree phase advance. A 1 o’clock delta connection represents a 30 degree phase delay.
expanded three-phase port	Three separate electrical conserving ports that represent the individual phases of a three-phase system. You individually connect each phase of the three-phase system between blocks.
F_{Rated}	Rated electrical frequency of three-phase machine.
G	Conductance.
i	Instantaneous current.
I	RMS current.
L	Inductance.
line voltage	RMS value of the voltage measured between phases. In a balanced three-phase system with no harmonics, peak line voltage equals peak phase voltage multiplied by $\sqrt{3}$. The RMS value equals peak line voltage divided by $\sqrt{2}$.

	Standard abbreviations are Vab, Vac, Vbc, etc. Line voltage is also known as rated voltage, rated RMS, name plate voltage, line-line voltage, and phase-phase voltage.
nPolePairs	Number of pole pairs for three-phase machine.
phase voltage	RMS value of the voltage measured between a phase and reference point. The reference point is usually a neutral or ground point. In a balanced three-phase system with no harmonics, peak phase voltage is equal to peak line voltage divided by $\sqrt{3}$. The RMS value equals peak phase voltage divided by $\sqrt{2}$. Standard abbreviations are Va, Vb, and Vc.
$P_{PerPhase}$	Real power per phase.
psi	Instantaneous peak magnetic flux linkage.
Psi	RMS magnetic flux linkage.
$Q_{PerPhase}$	Reactive power per phase.
R	Resistance.
S_{Rated}	Rated apparent power.
$S_{PerPhase}$	Apparent power per phase.
torque	Torque of three-phase machine.
v	Instantaneous voltage.
V	RMS voltage.
Va, Vb, Vc	Phase voltages.
Vab, Vac, Vbc,...	Line voltages.
V_{Rated}	Rated voltage of three-phase machine.
$\omega_{Electrical}$	Electrical angular speed of three-phase machine.

$\omega_{Mechanical}$	Mechanical angular speed of three-phase machine.
winding voltage	Voltage measured between both ends of a winding. For a wye connection, winding voltage equals phase voltage. For a delta connection, winding voltage equals line voltage.
wye connection	Three-phase winding configuration. Each of the three windings is connected between a phase and neutral point. Physically, the connection resembles the letter Y. A wye connection is also referred to as a star connection, or Y connection.
X	Reactance.
Y	Admittance.
z	Impedance.