SimPowerSystems

For Use with Simulink®

Hydro-Québec

TransÉnergie Technologies

- Modeling
- Simulation
- Implementation

User's Guide

Version 3

The MathWorks

How to Contact The MathWorks:

www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup
support@mathworks.com	Technical support
suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information
508-647-7000	Phone
508-647-7001	Fax
The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098	Mail

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

SimPowerSystems User's Guide

Printing History:			
	January 1998	First printing	Version 1 (Release 10)
	September 2000	Second printing	Revised for Version 2.1 (Release 12)
	June 2001	Online only	Revised for Version 2.2 (Release 12.1)
	July 2002	Online only	Revised for Version 2.3 (Release 13) (Renamed from *Power System Blockset User's Guide*)
	September 2003	3rd printing	Revised for Version 3 (Release 13SP1)

Contents

About This Guide

1 Modeling Simple Systems

2 Advanced Components and Techniques

3 Case Studies

4 Improving Simulation Performance

5 SimPowerSystems Block Reference

About This Guide

SimPowerSystems and Physical Modeling

Starting with MathWorks Release 13, SimPowerSystems and SimMechanics of the Physical Modeling product family work together with Simulink® to model electrical, mechanical, and control systems.

Electrical power systems are combinations of electrical circuits and electromechanical devices like motors and generators. Engineers working in this discipline are constantly improving the performance of the systems. Requirements for drastically increased efficiency have forced power system designers to use power electronic devices and sophisticated control system concepts that tax traditional analysis tools and techniques. Further complicating the analyst's role is the fact that the system is often so nonlinear that the only way to understand it is through simulation.

Land-based power generation from hydroelectric, steam, or other devices is not the only use of power systems. A common attribute of these systems is their use of power electronics and control systems to achieve their performance objectives.

SimPowerSystems was designed to provide a modern design tool that allows scientists and engineers to rapidly and easily build models that simulate power systems. SimPowerSystems uses the Simulink environment, allowing you to build a model using simple *click and drag* procedures. Not only can you draw the circuit topology rapidly, but your analysis of the circuit can include its interactions with mechanical, thermal, control, and other disciplines. This is possible because all the electrical parts of the simulation interact with the extensive Simulink modeling library. Since Simulink uses MATLAB® as its computational engine, designers can also use MATLAB toolboxes and Simulink blocksets. SimPowerSystems and SimMechanics share a special Physical Modeling block and connection line interface.

Users can rapidly put SimPowerSystems to work. The libraries contain models of typical power equipment such as transformers, lines, machines, and power electronics. These models are proven ones coming from textbooks, and their validity is based on the experience of the Power Systems Testing and Simulation Laboratory of Hydro-Québec, a large North American utility located in Canada. The capabilities of SimPowerSystems for modeling a typical electrical grid are illustrated in demonstration files. And for users who want to refresh their knowledge of power system theory, there are also self-learning case studies.

About the Authors

SimPowerSystems Version 3 was developed by the following people and organizations.

Gilbert Sybille

Hydro-Québec Research Institute (IREQ), Varennes, Québec. Original author of SimPowerSystems, technical coordinator, author of phasor simulation, discretization techniques, and documentation.

Patrice Brunelle

TransÉnergie Technologies Inc., Montréal, Québec. Main software engineer. Author of graphical user interfaces, model integration into Simulink and Physical Modeling, and documentation.

Pierre Giroux, Silvano Casoria, Richard Gagnon, Innocent Kamwa, Raymond Roussel

Hydro-Québec Research Institute (IREQ), Varennes, Québec. Key beta testers and developers of several SPS blocks, demos, and documentation.

Roger Champagne, Louis Dessaint

École de Technologie Supérieure (ETS), Montréal, Québec. Authors of machine models, revised state space formulation, and documentation.

Hoang Lehuy

Université Laval, Québec City. Validation tests and author of several functions and documentation.

Acknowledgments

The authors acknowledge the contributions of the following people involved in Versions 1 and 2.

Pierre Mercier, iOMEGAt. Project manager for Versions 1 and 2.

Also: *Kamal Al-Haddad, Mohamed Tou, Christian Dufour, Momcilo Gavrilovic, Christian Larose, David McCallum,* and *Bahram Khodabakhchian*.

Related Products

The MathWorks provides several products that are especially relevant to the kinds of tasks you can perform with SimPowerSystems.

Requirements for SimPowerSystems

You must have the following products installed to use SimPowerSystems.

- MATLAB 6.5
- Simulink 5.0

Other Related Products

The toolboxes listed in the following table include functions that extend the capabilities of MATLAB. The blocksets include blocks that extend the capabilities of Simulink. These products enhance your use of SimPowerSystems in various applications.

The Physical Modeling Product Family

In addition to SimPowerSystems, the Physical Modeling product family includes SimMechanics, for modeling and simulating mechanical systems. Use these products together to model physical systems in Simulink.

For more information about any of these products, see either

- The online documentation for that product, if it is installed or if you are reading the documentation from the CD
- The MathWorks Web site at `http://www.mathworks.com`; see the "Products" section

Product	Description
Control System Toolbox	Design and analyze feedback control systems
µ-Analysis and Synthesis Toolbox	Design multivariable feedback controllers for systems with model uncertainty
Optimization Toolbox	Solve standard and large-scale optimization problems

Product	Description
Real-Time Workshop®	Generate C code from Simulink models
Robust Control Toolbox	Design robust multivariable feedback control systems
SimMechanics	Model and simulate mechanical systems
Stateflow®	Design and simulate event-driven systems
System Identification Toolbox	Create linear dynamic models from measured input-output data
xPC Target	Perform real-time rapid prototyping using PC hardware

Using This Guide

If you are a new user, begin with the first two chapters to learn

- How to build and simulate electrical circuits using the **powerlib** library
- How to interface an electrical circuit with Simulink blocks
- How to analyze the steady-state and frequency response of an electrical circuit
- How to discretize your model in order to increase simulation speed, especially for power electronic circuits and large power systems
- How to use the phasor simulation method
- How to build your own nonlinear models

If you are an experienced blockset user, see these chapters:

- The Release Notes for details on the latest release
- "Modeling Simple Systems" to learn how to simulate discretized electrical circuits
- "Advanced Components and Techniques" to learn how to apply the phasor simulation to transient stability study of multimachine systems
- "Case Studies" for an overview of some applications of SimPowerSystems presented as case studies
- "Improving Simulation Performance" to learn how to increase simulation speed

All blockset users should use the "SimPowerSystems Block Reference" for reference information on blocks, simple demos, and GUI-based tools. For commands, refer to "SimPowerSystems Command Reference" for a synopsis of the command's syntax, as well as a complete explanation of options and operation.

Units

This manual uses the International System of Units (SI).

Quantity	Unit	Symbol
Time	second	s
Length	meter	m
Mass	kilogram	kg
Energy	joule	J
Current	ampere	A
Voltage	volt	V
Active power	watt	W
Apparent power	volt-ampere	VA
Reactive power	var	var
Impedance	ohm	Ω
Resistance	ohm	Ω
Inductance	henry	H
Capacitance	farad	F
Flux linkage	volt-second	V. s
Rotation speed	radians per second revolutions per minute	rad/s rpm
Torque	newton-meter	N.m
Inertia	kilogram-meter^2	kg.m^2
Friction factor	newton-meter-second	N.m.s

The manual also uses the per unit (p.u.) system on occasion to define the model parameters.

What Is the Per Unit System?

The per unit system is widely used in the power system industry to express values of voltages, currents, powers, and impedances of various power equipment. It is mainly used for transformers and AC machines.

For a given quantity (voltage, current, power, impedance, torque, etc.) the per unit value is the value related to a base quantity.

$$\text{base value in p.u.} = \frac{\text{quantity expressed in SI units}}{\text{base value}}$$

Generally the following two base values are chosen:

- The base power = nominal power of the equipment
- The base voltage = nominal voltage of the equipment

All other base quantities are derived from these two base quantities. Once the base power and the base voltage are chosen, the base current and the base impedance are determined by the natural laws of electrical circuits.

$$\text{base current} = \frac{\text{base power}}{\text{base voltage}}$$

$$\text{base impedance} = \frac{\text{base voltage}}{\text{base current}} = \frac{(\text{base voltage})^2}{\text{base power}}$$

For a transformer with multiple windings, each having a different nominal voltage, the same base power is used for all windings (nominal power of the transformer). However, according to the above definitions, there are as many base values as windings for voltages, currents, and impedances.

For AC machines, the torque and speed can be also expressed in p.u. The following base quantities are chosen:

- The base speed = synchronous speed
- The base torque = torque corresponding at base power and synchronous speed:

$$\text{base torque} = \frac{\text{base power (3 phases) in watts}}{\text{base speed in radians/second}}$$

Instead of specifying the rotor inertia in kg*m^2, you would generally give the inertia constant H defined as

$$H = \frac{\text{kinetic energy stored in the rotor at synchronous speed in joules}}{\text{machine nominal power in VA}}$$

$$H = \frac{\frac{1}{2} \times J \cdot w^2}{Pnom}$$

The inertia constant is expressed in seconds. For large machines, this constant is around 3 to 5 seconds. An inertia constant of 3 seconds means that the energy stored in the rotating part could supply the nominal load during 3 seconds. For small machines, H is lower. For example, for a 3 HP motor, it can be between 0.5 and 0.7 seconds.

Example 1: Three-Phase Transformer

Consider, for example, a three-phase two-winding transformer. The following typical parameters could be provided by the manufacturer:

- Nominal power = 300 kVA total for three phases
- Nominal frequency = 60 Hz
- Winding 1: connected in wye, nominal voltage = 25 kV RMS line-to-line resistance 0.01 p.u., leakage reactance = 0.02 p.u.
- Winding 2: connected in delta, nominal voltage = 600 V RMS line-to-line resistance 0.01 p.u., leakage reactance = 0.02 p.u.
- Magnetizing losses at nominal voltage in % of nominal current: Resistive 1%, Inductive 1%

The base values for each single phase transformer are first calculated:

- For winding 1:

Base power	300 kVA/3 = 100e3 VA/phase
Base voltage	25 kV/sqrt(3) = 14434 V RMS
Base current	100e3/14434 = 6.928 A RMS
Base impedance	14434/6.928 = 2083 Ω

Base resistance	14434/6.928 = 2083 Ω
Base inductance	2083/(2π*60)= 5.525 H

- For winding 2:

Base power	300 kVA/3 = 100e3 VA
Base voltage	600 V RMS
Base current	100e3/600 = 166.7 A RMS
Base impedance	600/166.7 = 3.60 Ω
Base resistance	600/166.7 = 3.60 Ω
Base inductance	3.60/(2π*60) = 0.009549 H

The values of the winding resistances and leakage inductances expressed in SI units are therefore

- For winding 1: R1= 0.01 * 2083 = 20.83 Ω; L1= 0.02*5.525 = 0.1105 H
- For winding 2: R2= 0.01 * 3.60 = 0.0360 Ω; L2= 0.02*0.009549 = 0.191 mH

For the magnetizing branch, magnetizing losses of 1% resistive and 1% inductive mean a magnetizing resistance Rm of 100 p.u. and a magnetizing inductance Lm of 100 p.u. Therefore, the values expressed in SI units referred to winding 1 are

- Rm = 100*2083 = 208.3 kΩ
- Lm = 100*5.525 = 552.5 H

Example 2: Asynchronous Machine

Now consider the three-phase four-pole Asynchronous Machine in SI units provided in the Machines library of **powerlib**. It is rated 3 HP, 220 V RMS line-to-line, 60 Hz.

The stator and rotor resistance and inductance referred to stator are

- Rs = 0.435 Ω; Ls = 2 mH
- Rr = 0.816 Ω; Lr = 2 mH

The mutual inductance is Lm = 69.31 mH. The rotor inertia is J = 0.089 kg.m^2.

The base quantities for one phase are calculated as follows:

Base power	3 HP*746VA/3 = 746 VA/phase
Base voltage	220 V/sqrt(3) = 127.0 V RMS
Base current	746/127.0 = 5.874 A RMS
Base impedance	127.0/5.874 = 21.62 Ω
Base resistance	127.0/5.874 = 21.62 Ω
Base inductance	21.62/(2π*60)= 0.05735 H = 57.35 mH
Base speed	1800 rpm = 1800*(2π)/60 = 188.5 radians/second
Base torque (3-phase)	746*3/188.5 = 11.87 newton-meters

Using the above base values, you can compute the values in per units.

Rs= 0.435 / 21.62 = 0.0201 p.u. Ls= 2 / 57.35 = 0.0349 p.u.
Rr= 0.816 / 21.62 = 0.0377 p.u. Lr= 2 / 57.35 = 0.0349 p.u.
Lm = 69.31/57.35 = 1.208 p.u.

The inertia is calculated from inertia J, synchronous speed, and nominal power.

$$H = \frac{\frac{1}{2} \times J \cdot w^2}{Pnom} = \frac{\frac{1}{2} \times 0.089 \times (188.5)^2}{3 \times 746} = 0.7065 \text{ seconds}$$

If you open the dialog box of the Asynchronous Machine block in p.u. units provided in the Machines library of **powerlib**, you find that the parameters in p.u. are the ones calculated above.

Base Values for Instantaneous Voltage and Current Waveforms

When displaying instantaneous voltage and current waveforms on graphs or oscilloscopes, you normally consider the peak value of the nominal sinusoidal voltage as 1 p.u. In other words, the base values used for voltage and currents are the RMS values given above multiplied by $\sqrt{2}$.

Why Use the Per Unit System Instead of the Standard SI Units?

Here are the main reasons for using the per unit system:

- When values are expressed in p.u., the comparison of electrical quantities with their “normal” values is straightforward.

  For example, a transient voltage reaching a maximum of 1.42 p.u. indicates immediately that this voltage exceeds the nominal value by 42%.

- The values of impedances expressed in p.u. stay fairly constant whatever the power and voltage ratings.

  For example, for all transformers in the 3 kVA to 300 kVA power range, the leakage reactance varies approximately between 0.01 p.u. and 0.03 p.u., whereas the winding resistances vary between 0.01 p.u. and 0.005 p.u., whatever the nominal voltage. For transformers in the 300 kVA to 300 MVA range, the leakage reactance varies approximately between 0.03 p.u. and 0.12 p.u., whereas the winding resistances vary between 0.005 p.u. and 0.002 p.u.

  Similarly, for salient pole synchronous machines, the synchronous reactance X_{d} is generally between 0.60 and 1.50 p.u., whereas the sub transient reactance X'_{d} is generally between 0.20 and 0.50 p.u.

  It means that if you do not know the parameters for a 10 kVA transformer, you are not making a major error by assuming an average value of 0.02 p.u. for leakage reactances and 0.0075 p.u. for winding resistances.

- The calculations using the per unit system are simplified. When all impedances in a multivoltage power system are expressed on a common power base and on the nominal voltages of the different subnetworks, the total impedance in p.u. seen at one bus is obtained by simply adding all impedances in p.u., without taking into consideration the transformer ratios.

Typographical Conventions

This manual uses some or all of these conventions.

Item	Convention	Example
Example code	`Monospace` font	To assign the value 5 to `A`, enter `A = 5`
Function names, syntax, filenames, directory/folder names, user input, items in drop-down lists	`Monospace` font	The `cos` function finds the cosine of each array element. Syntax line example is `MLGetVar ML_var_name`
Buttons and keys	**Boldface** with book title caps	Press the **Enter** key.
Literal strings (in syntax descriptions in reference chapters)	**`Monospace bold`** for literals	`f = freqspace(n,'`**`whole`**`')`
Mathematical expressions	*Italics* for variables Standard text font for functions, operators, and constants	This vector represents the polynomial $p = x^2 + 2x + 3$.

MATLAB output	`Monospace font`	MATLAB responds with
		`A =`
		`5`

Menu and dialog box titles	**Boldface** with book title caps	Choose the **File Options** menu.
New terms and for emphasis	*Italics*	An *array* is an ordered collection of information.
Omitted input arguments	(...) ellipsis denotes all of the input/output arguments from preceding syntaxes.	`[c,ia,ib] = union(...)`
String variables (from a finite list)	*`Monospace italics`*	`sysc = d2c(sysd,'`*`method`*`')`

1

Modeling Simple Systems

SimPowerSystems operates in the Simulink environment. Therefore, before starting this user's guide, you should be familiar with Simulink. For help with Simulink, see the Using Simulink guide.

To master SimPowerSystems, you must learn how to model and simulate electrical circuits. This chapter is organized into four tutorials, all based on a simple power system, that demonstrate basic circuit modeling, analysis, and simulation.

Simulating a Simple Circuit

SimPowerSystems allows you to build and simulate electrical circuits containing linear and nonlinear elements.

In this section you

- Explore the **powerlib** library of SimPowerSystems
- Learn how to build a simple circuit from the **powerlib** library
- Interconnect Simulink blocks with your circuit

The circuit below represents an equivalent power system feeding a 300 km transmission line. The line is compensated by a shunt inductor at its receiving end. A circuit breaker allows energizing and deenergizing of the line. In order to simplify matters, only one of the three phases is represented. The parameters shown in the figure are typical of a 735 kV power system.

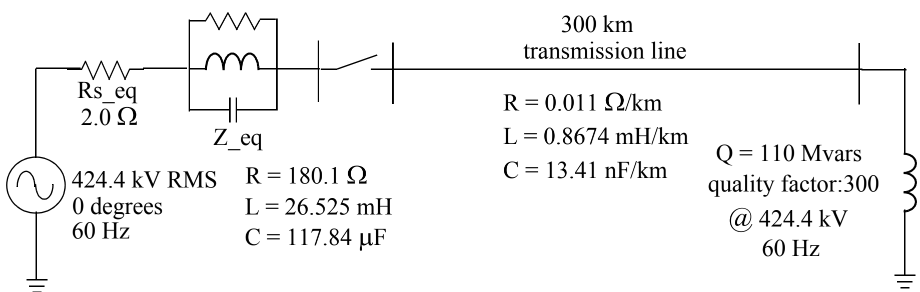


Figure 1-1: Circuit to Be Modeled with SimPowerSystems

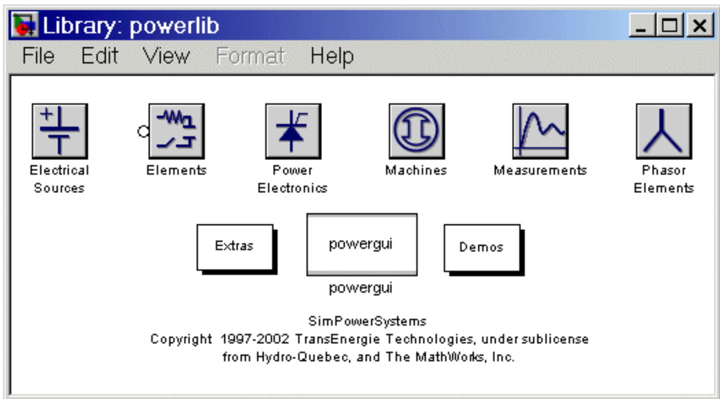
Building the Electrical Circuit with powerlib Library

The graphical user interface makes use of the Simulink functionality to interconnect various electrical components. The electrical components are grouped in a special library called **powerlib**.

Open the SimPowerSystems library by entering the following command at the MATLAB prompt:

```
powerlib
```

This command displays a Simulink window showing icons of different block libraries.



You can open these libraries to produce the windows containing the blocks to be copied into your circuit. Each component is represented by a special icon having one or several inputs and outputs corresponding to the different terminals of the component:

1 From the **File** menu of the **powerlib** window, open a new window to contain your first circuit and save it as `circuit1`.

2 Open the Electrical Sources library and copy the AC Voltage Source block into the **circuit1** window.

3 Open the **AC Voltage Source** dialog box by double-clicking the icon and enter the Amplitude, Phase, and Frequency parameters according to the values shown in Figure 1-1.

   Note that the amplitude to be specified for a sinusoidal source is its peak value (424.4e3*sqrt(2) volts in this case).

4 Change the name of this block from Voltage Source to Vs.

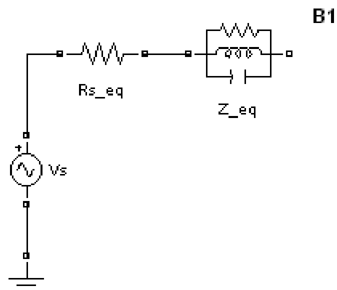
5 Copy the Parallel RLC Branch block, which can be found in the Elements library of **powerlib**, set its parameters as shown in Figure 1-1, and name it Z_eq.

6 The resistance Rs_eq of the circuit can be obtained from the Parallel RLC Branch block. Duplicate the Parallel RLC Branch block, which is already in your **circuit1** window, set the R parameter according to Figure 1-1, and set the L and C parameters respectively to infinity (`inf`) and zero (`0`).

   Once the dialog box is closed, notice that the L and C components have disappeared so that the icon now shows a single resistor. The same result would have been obtained with the Series RLC Branch block by setting L and C respectively at zero and `inf`.

7 Name this block Rs_eq.

Resize the various components and interconnect blocks by dragging lines from outputs to inputs of appropriate blocks.



In order to complete the circuit of Figure 1-1, you need to add a transmission line and a shunt reactor. You add the circuit breaker later in "Simulating Transients" on page 1-19.

The model of a line with uniformly distributed R, L, and C parameters normally consists of a delay equal to the wave propagation time along the line. This model cannot be simulated as a linear system because a delay corresponds to an infinite number of states. However, a good approximation of the line with a finite number of states can be obtained by cascading several PI circuits, each representing a small section of the line.

A PI section consists of a series R-L branch and two shunt C branches. The model accuracy depends on the number of PI sections used for the model. Copy the PI Section Line block from the Elements library into the **circuit1** window, set its parameters as shown in Figure 1-1, and specify one line section.

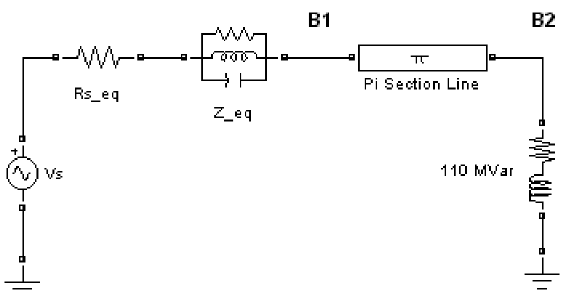
The shunt reactor is modeled by a resistor in series with an inductor. You could use a Series RLC Branch block to model the shunt reactor. Set the R and L values corresponding to the active and reactive power specified in Figure 1-1 (**Q** = 110 Mvar; **P** = 110/300 = 0.37 MW at **V** = 424.4 kV rms and **f** = 60 Hz).

You might find it more convenient to use a Series RLC Load block that allows you to specify directly the active and reactive powers absorbed by the shunt reactor.

Copy the Series RLC Load block, which can be found in the Elements library of **powerlib**. Name this block 110 Mvar. Set its parameters as follows:

Vn	424.4e3 V
fn	60 Hz
P	110e6/300 W (quality factor = 300)
QL	110e6 vars
Qc	0

Note that, as no reactive capacitive power is specified, the capacitor disappears on the block icon when the dialog box is closed. Interconnect the new blocks as shown.



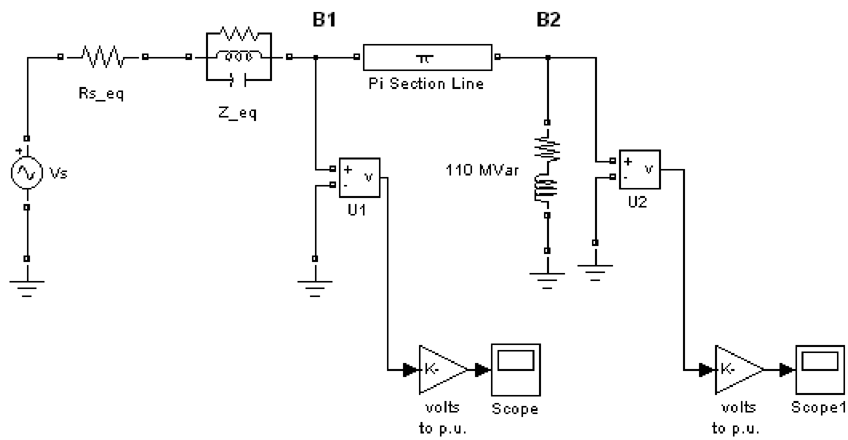
You need a Voltage Measurement block to measure the voltage at node B1. This block is found in the Measurements library of **powerlib**. Copy it and name it U1. Connect its positive input to the node B1 and its negative input to a new Ground block.

In order to observe the voltage measured by the Voltage Measurement block named U1, a display system is needed. This can be any device found in the Sinks library of Simulink.

Open the Sinks library of Simulink and copy the Scope block into your **circuit1** window. If the scope were connected directly at the output of the voltage measurement, it would display the voltage in volts. However, electrical engineers in power systems are used to working with normalized quantities (per unit system). The voltage is normalized by dividing the value in volts by a base voltage corresponding to the peak value of the system nominal voltage. In this case the scaling factor K is

$$K = \frac{1}{424.4 \times 10^3 \times \sqrt{2}}$$

Copy a Gain block from the Simulink library and set its gain as above. Connect its output to the Scope block and connect the output of the Voltage Measurement block to the Gain block. Duplicate this voltage measurement system at the node B2, as shown below.



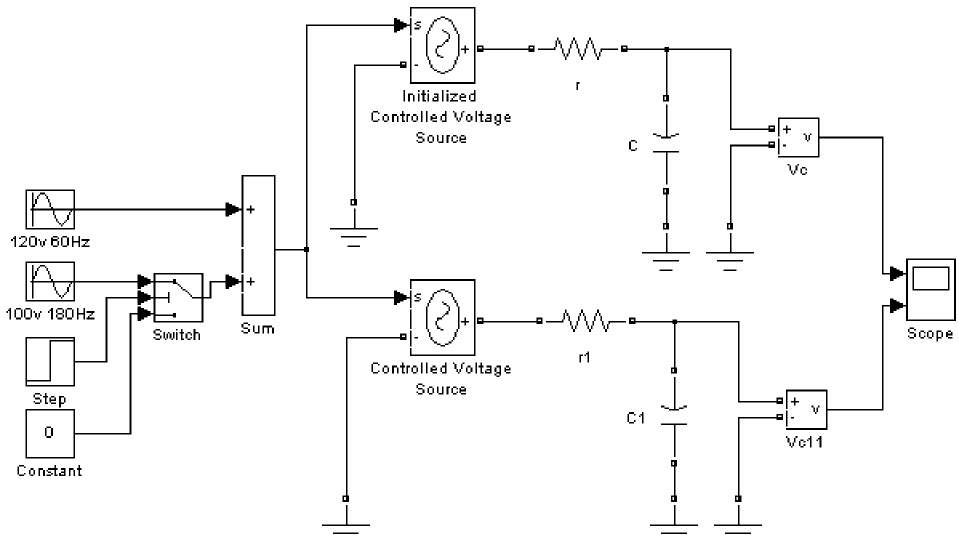
Interfacing the Electrical Circuit with Simulink

The Voltage Measurement block acts as an interface between the SimPowerSystems blocks and the Simulink blocks. For the system shown above, you implemented such an interface from the electrical system to the

Simulink system. The Voltage Measurement blocks converts the measured voltages into Simulink signals.

Note that the Current Measurement block from the Measurements library of **powerlib** can also be used to convert any measured current into a Simulink signal.

You can also interface from Simulink blocks to the electrical system. For example, you can use the Controlled Voltage Source block to inject a voltage in an electrical circuit. The voltage is then controlled by a Simulink signal.



Electrical Terminal Ports and Connection Lines SimPowerSystems is part of the Physical Modeling environment. Its blocks often feature both normal Simulink input and output ports > and special electrical terminal ports □.

- Lines that connect normal Simulink ports > are directional signal lines.
- Lines that connect terminal ports □ are special electrical connection lines. These lines are nondirectional and can be branched. But you cannot connect them to Simulink ports > or to normal Simulink signal lines.

- You can connect Simulink ports > only to other Simulink ports and terminal ports □ only to other terminal ports.
- Converting Simulink signals to electrical connections or vice versa requires using a SimPowerSystems block that features both Simulink and terminal ports.

Some SimPowerSystems blocks feature only one type of port.

Simulating Your Circuit

Now you can start the simulation from the **Simulation** menu. As expected, voltage is sinusoidal with peak value of 1 p.u.

While the simulation is running, open the Vs block dialog box and modify the amplitude. Observe the effect on the two scopes. You can also modify the frequency and the phase. You can zoom in on the waveforms in the scope windows by drawing a box around the region of interest with the left mouse button.

Note To simulate this circuit, the default integration algorithm (`ode45`) was used. However, for most applications of SimPowerSystems your circuits contain switches and other nonlinear models. In such a case, you must specify a different integration algorithm. This is discussed in “Simulating Transients” on page 1-19, where a circuit breaker is added to your circuit.

Analyzing a Simple Circuit

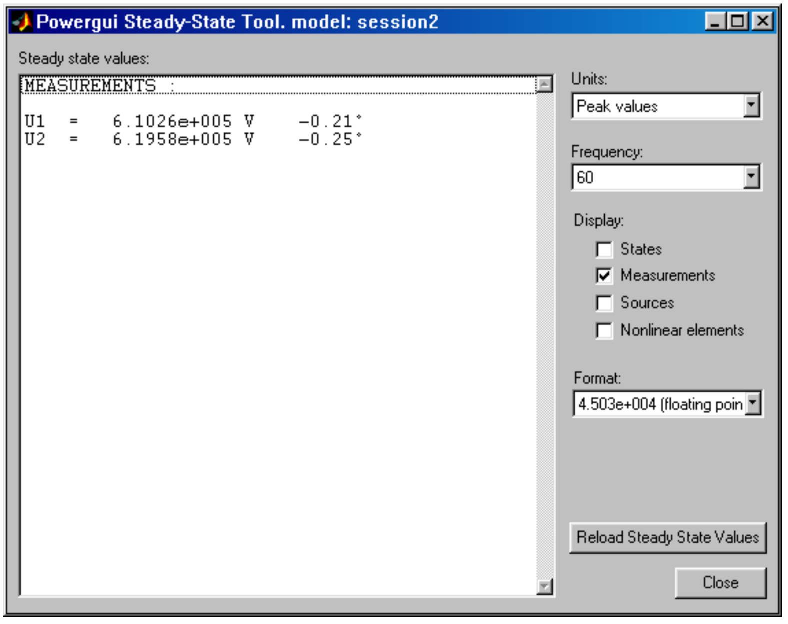
In this section you

- Use the Powergui (graphical user interface) block
- Obtain the steady-state outputs of the system
- Analyze your circuit with `power_analyze` command
- Analyze an electrical circuit in the frequency domain

Steady-State Analysis

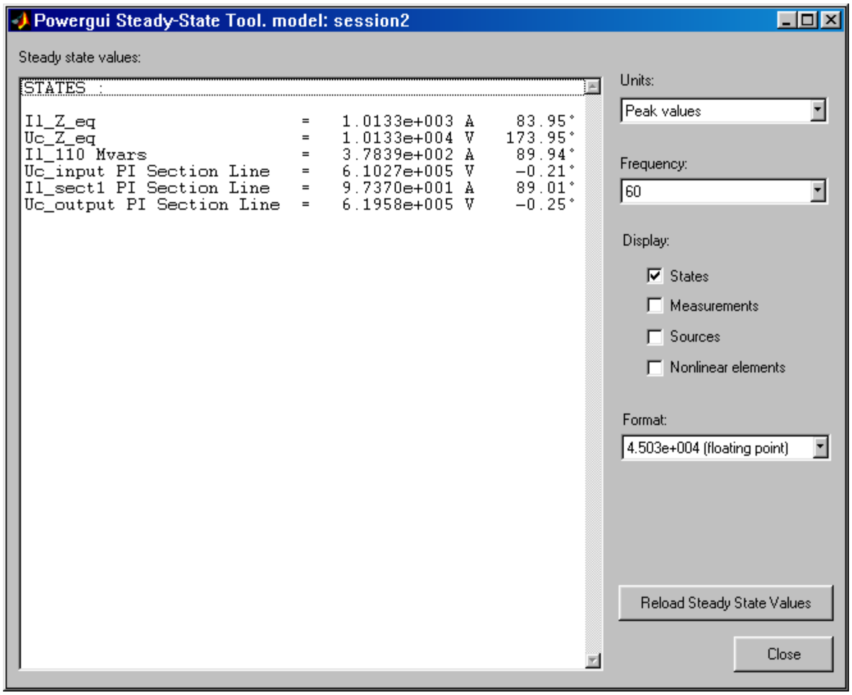
In order to facilitate the steady-state analysis of your circuit, a graphical user interface (GUI) is provided in the **powerlib** library. Copy the Powergui block into your **circuit1** window and double-click the block icon to open it.

From the **Analysis tools** menu of Powergui, select **Steady-State Voltages and Currents**. This opens the **Steady State Tool** window where the steady-state phasors measured by the two measurement blocks are displayed in polar form.



Each measurement output is identified by a string corresponding to the measurement block name. The magnitudes of the phasors U1 and U2 correspond to the peak value of the sinusoidal voltages.

From the **Steady State** window you can also choose to display the steady-state values of the source voltage or the steady-state values of states by selecting either the **Sources** or the **States** check box.



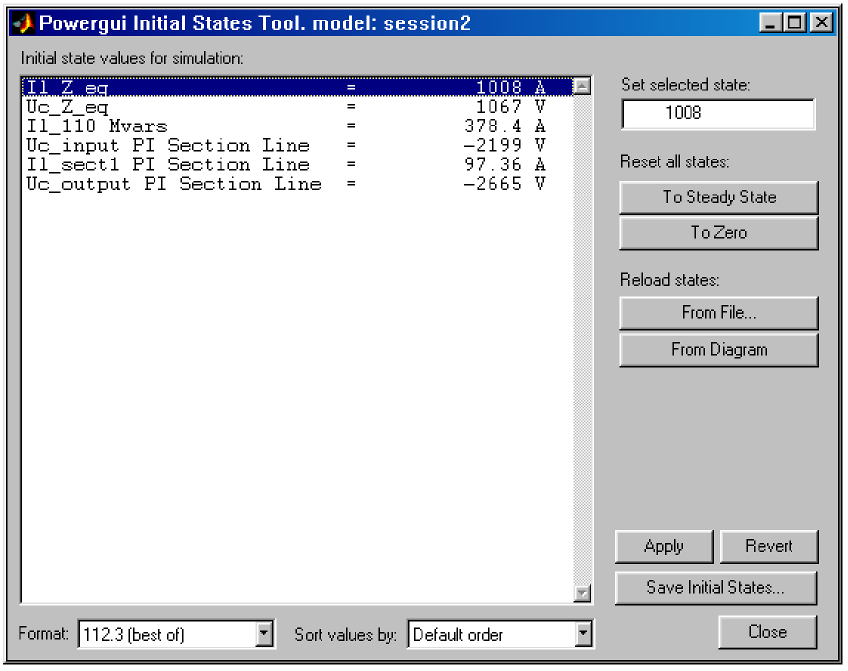
The state variable names contain the name of the block where the inductor or capacitor is found, preceded by the `Il_` prefix for inductor currents or the `Uc_` prefix for capacitor voltages.

The sign conventions used for the voltages and currents of sources and state variables are determined by the orientation of the blocks:

- Inductor currents flowing in the arrow direction are considered positive.
- Capacitor voltages are $V_{block\ output} - V_{block\ input}$.

Note Depending on the exact position of the various blocks in your `circuit1` diagram, the state variables might not be ordered in the same way as in the preceding figure.

Now, from the **Tools** menu of Powergui, select **Initial State Settings**. The initial values of the six state variables (three inductor currents and three capacitor voltages) are displayed. These initial values are set in order to start the simulation in steady state.



Frequency Analysis

The Measurements library of **powerlib** contains an Impedance Measurement block that measures the impedance between any two nodes of a circuit. In the following two sections you measure the impedance of your circuit between node B2 and ground by using two methods:

- Calculation from the state-space model
- Automatic measurement using the Impedance Measurement block and the Powergui block

Obtaining the Impedance vs Frequency Relation from the State-Space Model

To measure the impedance versus frequency at node B2, you need a current source at node B2 providing a second input to the state-space model. Open the Electrical Sources library and copy the AC Current Source block into your model. Connect this source at node B2, as shown below. Set the current source magnitude to zero and keep its frequency at `60 Hz`. Rearrange the blocks as follows:

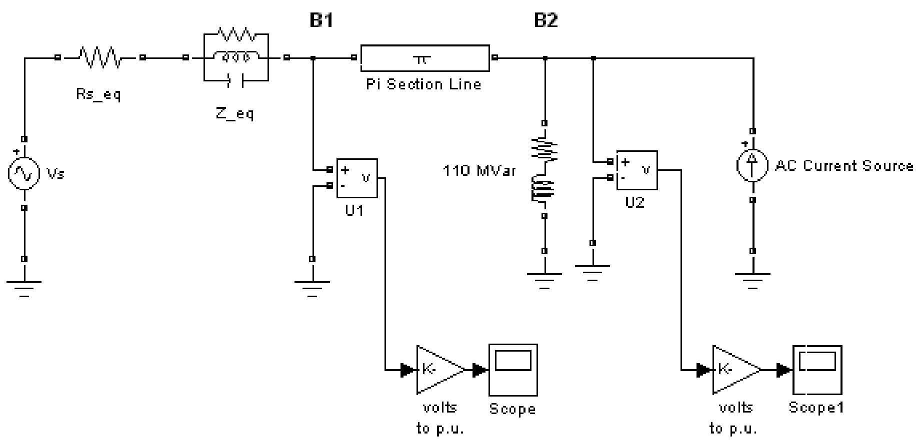


Figure 1-2: AC Current Source at the B2 Node

Now compute the state-space representation of the model `circuit1` with the `power_analyze` command. Enter the following command at the MATLAB prompt.

```
[A,B,C,D,x0,states,inputs,outputs]=power_analyze('circuit1');
```

The `power_analyze` command returns the state-space model of your circuit in the four matrices A, B, C, and D. x0 is the vector of initial conditions that you just displayed with the Powergui block. The names of the state variables, inputs, and outputs are returned in three string matrices.

```
states =

Il_110 Mvars
Uc_input PI Section Line
Il_ sect1 PI Section Line
Uc_output PI Section Line
Il_Z_eq
Uc_Z_eq

inputs =

U_Vs
I_AC Current Source

outputs =

U_U1
U_U2
```

Note that you could have obtained the names and ordering of the states, inputs, and outputs directly from the Powergui block.

Once the state-space model of the system is known, it can be analyzed in the frequency domain. For example, the modes of this circuit can be found from the eigenvalues of matrix A (use the MATLAB `eig` command).

```
eig(A)
ans =

1.0e+05 *
-2.4972
-0.0001 + 0.0144i <– 229 Hz
-0.0001 - 0.0144i
-0.0002 + 0.0056i <– 89 Hz
-0.0002 - 0.0056i
-0.0000
```

This system has two oscillatory modes at 89 Hz and 229 Hz. The 89 Hz mode is due to the equivalent source, which is modeled by a single pole equivalent. The 229 Hz mode is the first mode of the line modeled by a single PI section.

Note If you have Control System Toolbox installed, you can compute the impedance of the network as a function of frequency by using the `bode` function.

In the Laplace domain, the impedance Z2 at node B2 is defined as the transfer function between the current injected at node B2 (input 2 of the system) and the voltage measured at node B2 (output 2 of the system):

$$Z2(s) = \frac{U2(s)}{I2(s)}$$

The impedance at node B2 for the 0 to 1500 Hz range can be calculated and displayed as follows:

```
freq=0:1500;
w=2*pi*freq;
[mag1,phase1]=bode(A,B,C,D,2,w);
semilogy(freq,mag1(:,2));
```

Repeat the same process to get the frequency response with a 10 section line model. Open the **PI Section Line** dialog box and change the number of sections from `1` to `10`. To calculate the new frequency response and superimpose it upon the one obtained with a single line section, enter the following commands:

```
[A,B,C,D]=power_analyze('circuit1');
[mag10,phase10]=bode(A,B,C,D,2,w);
semilogy(freq,mag1(:,2),freq,mag10(:,2));
```

This is the resulting plot.

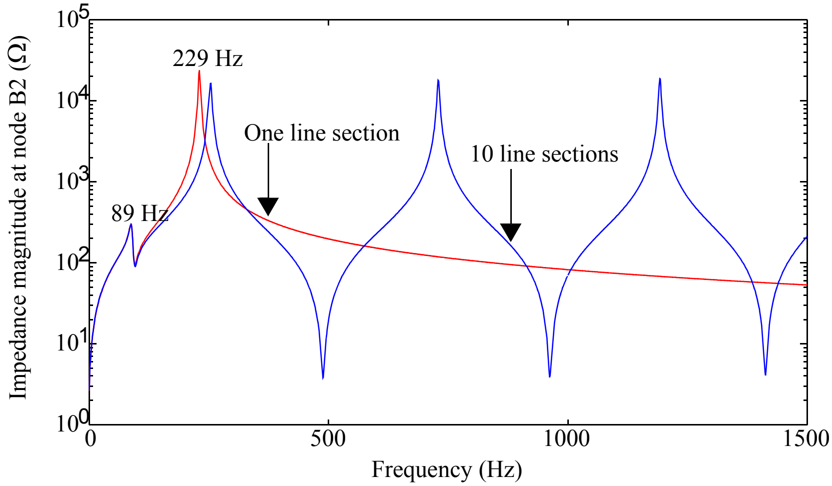


Figure 1-3: Impedance at Node B2 as Function of Frequency

This graph indicates that the frequency range represented by the single line section model is limited to approximately 150 Hz. For higher frequencies, the 10 line section model is a better approximation.

For a distributed parameter line model the propagation speed is

$$v = \frac{1}{\sqrt{L \cdot C}} = 293{,}208 \text{ km/s}$$

The propagation time for 300 km is therefore T = 300/293,208 = 1.023 ms and the frequency of the first line mode is f1 = 1/4T = 244 Hz. A distributed parameter line would have an infinite number of modes every 244 + n*488 Hz (n = 1, 2, 3...). The 10 section line model simulates the first 10 modes. The first three line modes can be seen in Figure 1-3 (244 Hz, 732 Hz, and 1220 Hz).

Obtaining the Impedance vs Frequency Relation from the Impedance Measurement and Powergui blocks

The process described above to measure a circuit impedance has been automated in SimPowerSystems. Open the Measurements library of **powerlib**, copy the Impedance Measurement block into your model, and rename it ZB2. Connect the two inputs of this block between node B2 and ground as shown.

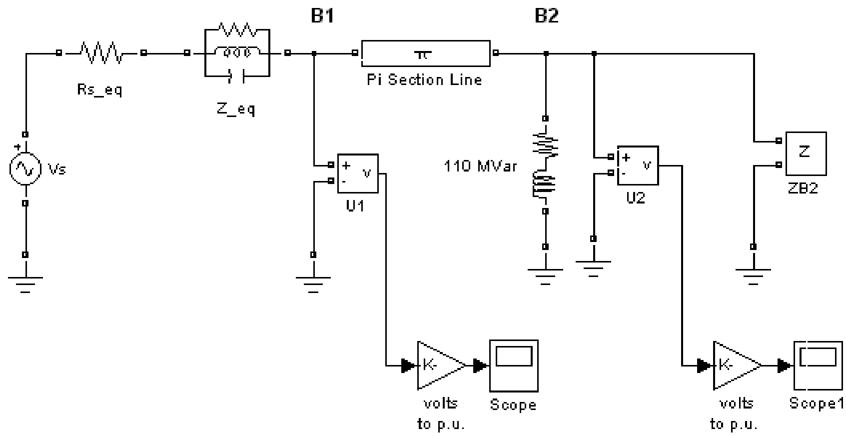
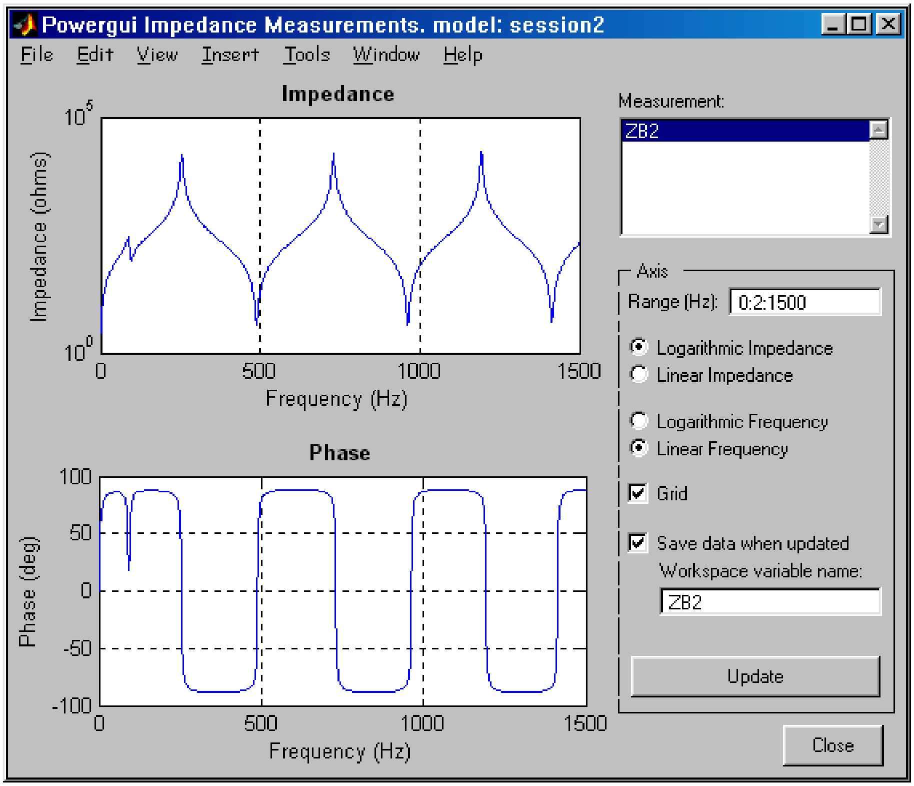


Figure 1-4: Measuring Impedance vs Frequency with the Impedance Measurement Block

Now open the Powergui. In the **Tools** menu, select **Impedance vs Frequency Measurement.** A new window opens, showing the list of Impedance Measurement blocks available in your circuit.

In your case, only one impedance is measured, and it is identified by ZB2 (the name of the ZB2 block) in the window. Fill in the frequency range by entering `0:2:1500` (zero to 1500 Hz by steps of 2 Hz). Select the logarithmic scale to display Z magnitude. Select the **Save data when updated** check box and enter `ZB2` as the variable name to contain the impedance vs. frequency. Click the **Display/Save** button.



When the calculation is finished, the window displays the magnitude and phase as functions of frequency. The magnitude should be identical to the plot (for one line section) shown in Figure 1-3. If you look in your workspace, you should have a variable named `ZB2`. It is a two-column matrix containing frequency in column 1 and complex impedance in column 2.

Now open the **Simulation —> Simulation parameters** dialog of your `circuit1` model. On the **Solver** pane, select the `ode23tb` integration algorithm. Set the relative tolerance to 1e-4 and keep `auto` for the other parameters. Set the stop time to `0.05`. Open the scopes and start the simulation.

Look at the waveforms of the sending and receiving end voltages on ScopeU1 and ScopeU2. As the state variables are automatically initialized, the system starts in steady state and sinusoidal waveforms are observed.

Finally open the Powergui. In the **Tools** menu, select **Initial State Settings** and reset all the states to zero by selecting the **To zero** button and then the **Apply** button. Restart the simulation and observe the transient when the line is energized from zero.

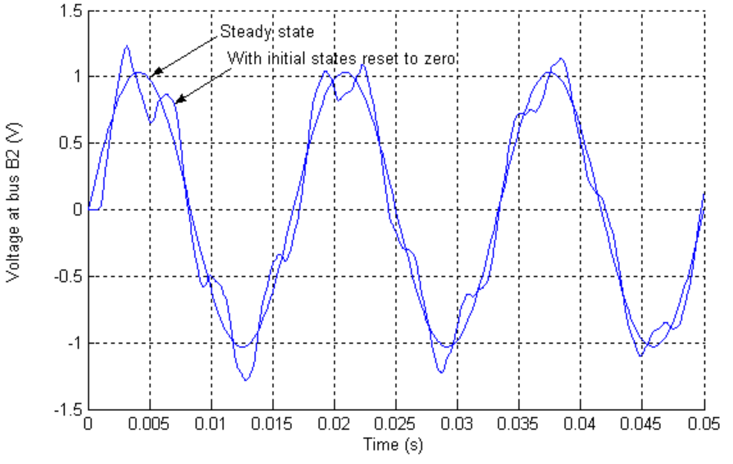


Figure 1-5: Receiving End Voltage U2 with 10 PI Section Line

Simulating Transients

In this section you

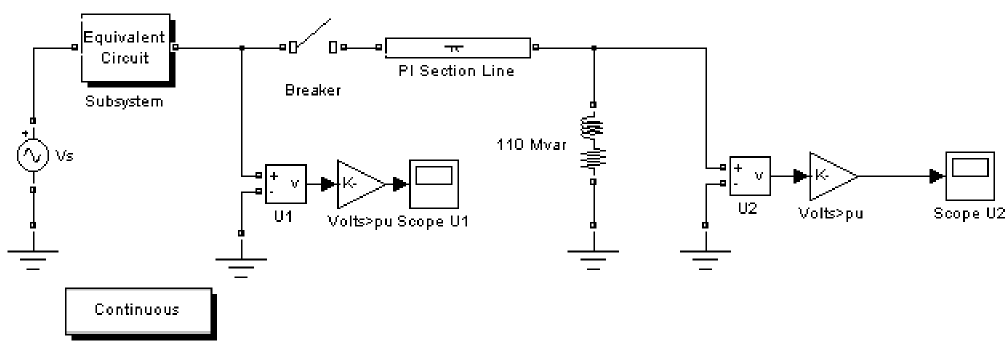
- Learn how to create an electrical subsystem
- Simulate transients with a circuit breaker
- Compare time domain simulation results with different line models
- Learn how to discretize a circuit and compare results thus obtained with results from a continuous, variable time step algorithm

One of the main uses of SimPowerSystems is to simulate transients in electrical circuits. This can be done with either mechanical switches (circuit breakers) or switches using power electronic devices.

First open your `circuit1` system and delete the current source connected at node B2. Save this new system as `circuit2`. Before connecting a circuit breaker, you need to modify the schematic diagram of `circuit2`. As with Simulink, SimPowerSystems allows you to group several components into a subsystem. This feature is useful to simplify complex schematic diagrams.

Use this feature to transform the source impedance into a subsystem:

1 Select the two blocks identified as Rs_eq and Z_eq by surrounding them by a box with the left mouse button and use the **Edit —> Create subsystem** menu item. The two blocks now form a new block called Subsystem.



2 Using the **Edit —> Mask subsystem** menu item, change the icon of that subsystem. In the **Icon** section of the mask editor, enter the following drawing command:

```
disp('Equivalent\nCircuit')
```

The icon is now “Equivalent Circuit,” as shown in the figure above.

3 Use the **Format —> Show drop shadow** menu item to get the appearance shown in the figure. You can now double-click the Subsystem block and look at its content.

4 Insert a circuit breaker into your circuit in order to simulate a line energization by opening the Elements library of **powerlib**. Copy the Breaker block into your **circuit2** window.

The circuit breaker is a nonlinear element modeled by an ideal switch in series with a resistance. Because of modeling constraints, this resistance cannot be set to zero. However, it can be set to a very small value, say 0.001 Ω, that does not affect the performance of the circuit.

1 Open the Breaker block dialog box and set its parameters as follows:

Ron	`0.001 Ω`
Initial state	`0 (open)`
Rs	`inf`
Cs	`0`
Switching times	`[(1/60)/4]`

2 Insert the circuit breaker in series with the sending end of the line, then rearrange the circuit as shown in the previous figure.

3 Finally connect a Scope block, from the Sinks library of Simulink, at the output of the Gain block measuring U2. Click the Scope **Parameters** icon and select the **Data history** tab. Click the **Save data to workspace** button and specify a variable name `U2` to save the simulation results; then change the **Format** option for U2 to be **Array**. Also, clear **Limit rows to last** to display the entire waveform for long simulation times.

You are now ready to simulate your system.

Continuous, Variable Time Step Integration Algorithms

Open the **PI section Line** dialog box and make sure the number of sections is set to 1. Open the **Simulation —> Simulation parameters** dialog. As you now have a system containing switches, you need a stiff integration algorithm to simulate the circuit. In the **Solver** pane, select the variable-step stiff integration algorithm `ode23tb`.

Keep the default parameters (relative tolerance set at `1e-3`) and set the stop time to `0.02` seconds. Open the scopes and start the simulation. Look at the waveforms of the sending and receiving end voltages on ScopeU1 and ScopeU2.

Once the simulation is complete, copy the variable `U2` into `U2_1` by entering the following command in the MATLAB window:

```
U2_1 = U2;
```

These two variables now contain the waveform obtained with a single PI section line model.

Open the **PI section Line** dialog box and change the number of sections from `1` to `10`. Start the simulation. Once the simulation is complete, copy the variable `U2` into `U2_10`.

Before modifying your circuit to use a distributed parameter line model, save your system as `circuit2_10pi`, which you can reuse later.

Delete the PI section line model and replace it with a single-phase Distributed Parameter Line block. Set the number of phases to `1` and use the same R, L, C, and length parameters as for the PI section line (see Figure 1-1). Save this system as `circuit2_dist`.

Restart the simulation and save the `U2` voltage in the `U2_d` variable.

You can now compare the three waveforms obtained with the three line models. Each variable `U2_1`, `U2_10`, and `U2_d` is a two-column matrix where the time is in column 1 and the voltage is in column 2. Plot the three waveforms on the same graph by entering the following command:

```
plot(U2_1(:,1), U2_1(:,2), U2_10(:,1),U2_10(:,2),
U2_d(:,1),U2_d(:,2));
```

These waveforms are shown in the next figure. As expected from the frequency analysis performed during "Analyzing a Simple Circuit" on page 1-9, the single

PI model does not respond to frequencies higher than 229 Hz. The 10 PI section model gives a better accuracy, although high-frequency oscillations are introduced by the discretization of the line. You can clearly see in the figure the propagation time delay of 1.03 ms associated with the distributed parameter line.

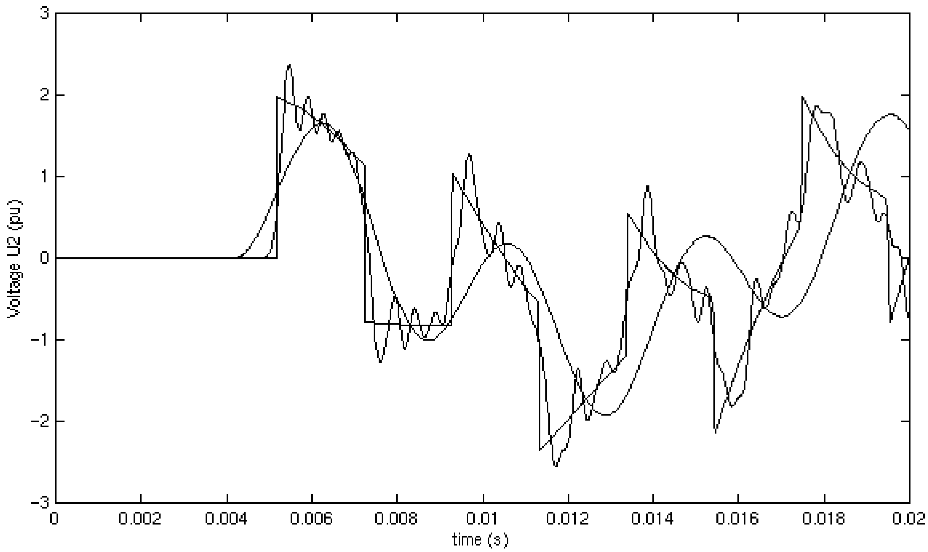


Figure 1-6: Receiving End Voltage Obtained with Three Different Line Models

Discretizing the Electrical System

An important feature of SimPowerSystems is its ability to simulate either with continuous, variable step integration algorithms or with discrete solvers. For small systems, variable time step algorithms are usually faster than fixed step methods, because the number of integration steps is lower. For large systems that contain many states or many nonlinear blocks such as power electronic switches, however, it is advantageous to discretize the electrical system.

When you discretize your system, the precision of the simulation is controlled by the time step. If you use too large a time step, the precision might not be sufficient. The only way to know if it is acceptable is to repeat the simulation with different time steps and find a compromise for the largest acceptable time step. Usually time steps of 20 μs to 50 μs give good results for simulation of switching transients on 50 Hz or 60 Hz power systems or on systems using line-commutated power electronic devices such as diodes and thyristors. You must reduce the time step for systems using forced-commutated power electronic switches. These devices, the insulated-gate bipolar transistor (IGBT), the field-effect transistor (FET), and the gate-turn-off thyristor (GTO) are operating at high switching frequencies.

For example, simulating a pulse-width-modulated (PWM) inverter operating at 8 kHz would require a time step of at least 1 μs.

You now learn how to discretize your system and compare simulation results obtained with continuous and discrete systems. Open the `circuit2_10pi` system that you saved from a previous simulation. This system contains 24 states and one switch. Open the Powergui and select **Discretize electrical model**. Set the sample time to `25e-6 s`. When you restart the simulation, the power system is discretized using the Tustin method (corresponding to trapezoidal integration) using a 25 μs sample time.

Open the **Simulation —> Simulation parameters —> Solver** dialog and set the simulation time to `0.2 s`. Start the simulation.

Note Once the system is discretized, there are no more continuous states in the electrical system. So you do not need a variable-step integration method to simulate. In the **Simulation —> Simulation parameters —> Solver** dialog, you could have selected the `Fixed-step` and `discrete (no continuous states)` options and specified a fixed step of 25 μs.

In order to measure the simulation time, you can restart the simulation by entering the following commands:

```
tic; sim(gcs); toc
```

When the simulation is finished the elapsed time in seconds is displayed in the MATLAB window.

To return to the continuous simulation, open the Powergui block and select **Continuous**. If you compare the simulation times, you find that the discrete system simulates approximately 3.5 times faster than the continuous system.

In order to compare the precision of the two methods, perform the following three simulations:

1 Simulate a continuous system, with Ts = 0.

2 Simulate a discrete system, with Ts = 25 μs.

3 Simulate a discrete system, with Ts = 50 μs.

For each simulation, save the voltage U2 in a different variable. Use respectively `U2c`, `U2d25`, and `U2d50`. Plot the U2 waveforms on the same graph by entering the following command:

```
plot(U2c(:,1), U2c(:,2), U2d25(:,1),U2d25(:,2),
U2d50(:,1),U2d(50:,2))
```

Using the zoom button of the graphics window, zoom in on the 4 to 12 ms region. You see differences on the high-frequency transients. The 25 μs compares reasonably well with the continuous simulation. However, increasing the time step to 50 μs produces appreciable errors. The 25 μs time step would therefore be acceptable for this circuit, while obtaining a gain of 3.5 on simulation speed.

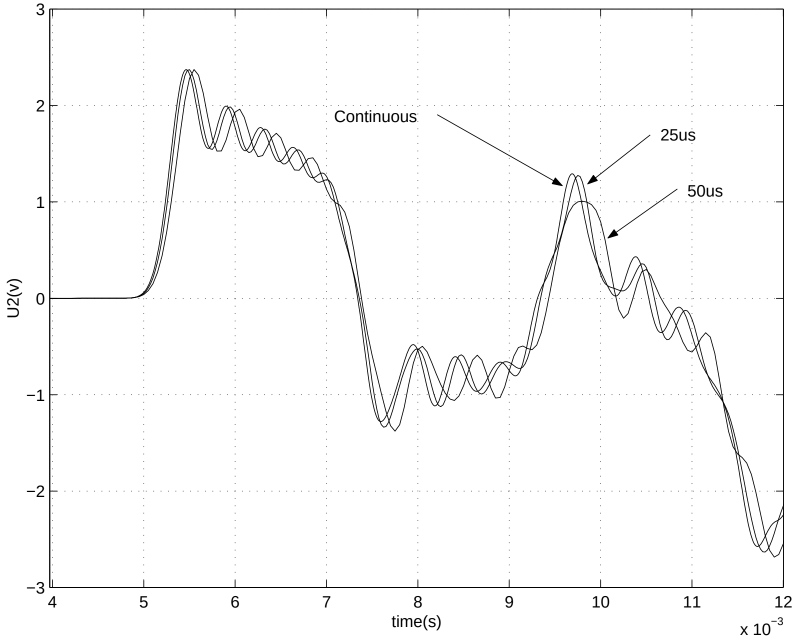


Figure 1-7: Comparison of Simulation Results for Continuous and Discrete Systems

Introducing the Phasor Simulation Method

In this section, you

- Apply the phasor simulation method to a simple linear circuit
- Learn advantages and limitations of this method

Up to now you have used two methods to simulate electrical circuits:

- Simulation with variable time steps using the continuous Simulink solvers
- Simulation with fixed time steps using a discretized system

This section explains how to use a third simulation method: the phasor solution method. This technique was introduced in Version 2.3.

When to Use the Phasor Solution

The phasor solution method is mainly used to study electromechanical oscillations of power systems consisting of large generators and motors. An example of this method is the simulation of a multimachine system in "Three-Phase Systems and Machines" on page 2-24. However, this technique is not restricted to the study of transient stability of machines. It can be applied to any linear system.

If, in a linear circuit, you are interested only in the changes in magnitude and phase of all voltages and currents when switches are closed or opened, you do not need to solve all differential equations (state-space model) resulting from the interaction of R, L, and C elements. You can instead solve a much simpler set of algebraic equations relating the voltage and current phasors. This is what the phasor solution method does. As its name implies, this method computes voltages and currents as phasors. Phasors are complex numbers representing sinusoidal voltages and currents at a particular frequency. They can be expressed either in Cartesian coordinates (real and imaginary) or in polar coordinates (amplitude and phase). As the electrical states are ignored, the phasor solution method does not require a particular solver to solve the electrical part of your system. The simulation is therefore much faster to execute. You must keep in mind, however, that this faster solution technique gives the solution only at one particular frequency.

Phasor Simulation of a Circuit Transient

You now apply the phasor solution method to a simple linear circuit. Open the Demos library of **powerlib.** Open the Simple Demos library and select the demo named “Transient Analysis.” A system named `power_transient` opens.

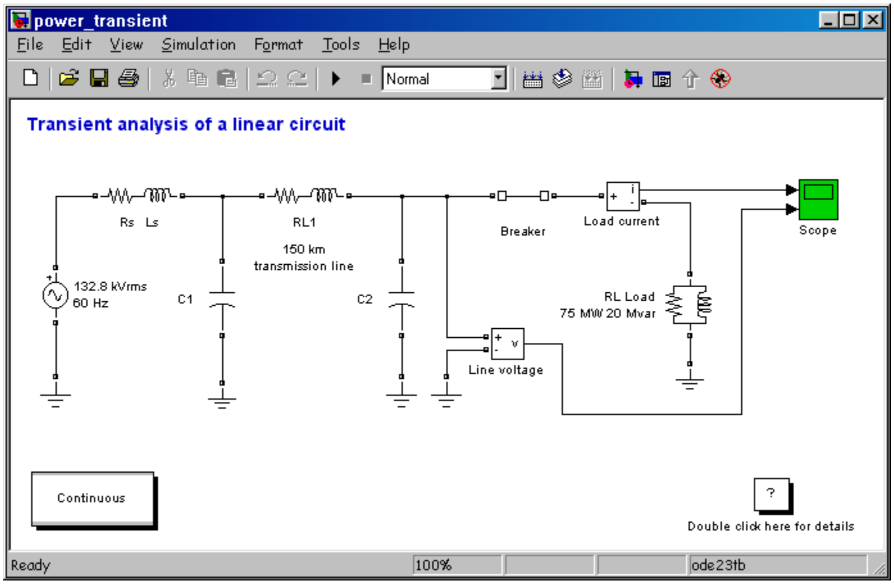


Figure 1-8: Simple Linear Circuit Built in SimPowerSystems

This circuit is a simplified model of a 60 Hz, 230 kV three-phase power system where only one phase is represented. The equivalent source is modeled by a voltage source (230 kV RMS / sqrt(3) or 132.8 kV RMS, 60 Hz) in series with its internal impedance (Rs Ls). The source feeds an RL load through a 150 km transmission line modeled by a single PI section (RL1 branch and two shunt capacitances, C1 and C2). A circuit breaker is used to switch the load (75 MW, 20 Mvar) at the receiving end of the transmission line. Two measurement blocks are used to monitor the load voltage and current.

The Powergui block at the lower left corner indicates that the model is continuous. Start the simulation and observe transients in voltage and current waveforms when the load is first switched off at t = 0.0333 s (2 cycles) and switched on again at t = 0.1167 s (7 cycles).

Invoking the Phasor Solution in the Powergui Block

You now simulate the same circuit using the phasor simulation method. This option is accessible through the Powergui block. Open this block and select **Phasor simulation.** You must also specify the frequency used to solve the algebraic network equations. A default value of 60 Hz should already be entered in the **Frequency** menu. Close the Powergui and notice that the word Phasors now appears on the Powergui icon, indicating that the Powergui now applies this method to simulate your circuit.

Restart the simulation. The magnitudes of the 60 Hz voltage and current are now displayed on the scope. Waveforms obtained from the continuous simulation and the phasor simulation are superimposed in this plot.

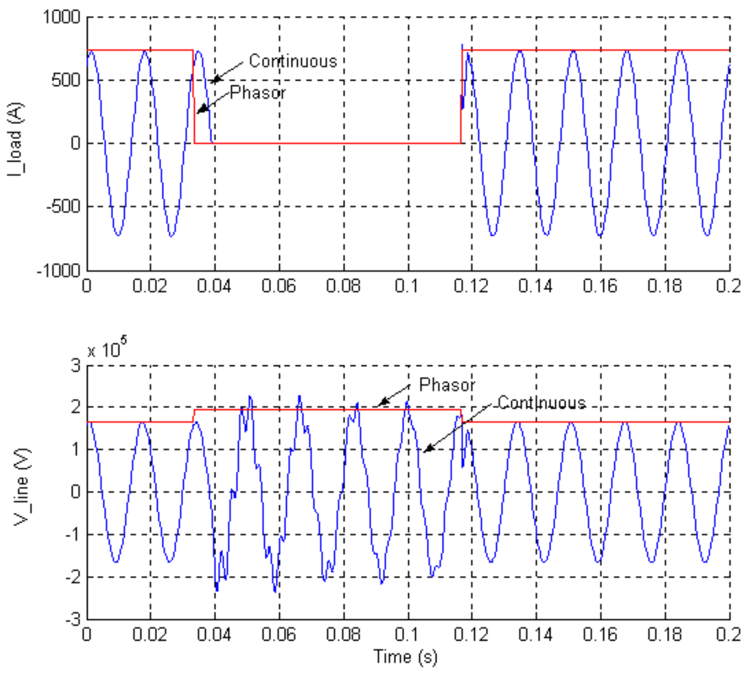


Figure 1-9: Waveforms Obtained with the Continuous and Phasor Simulation Methods

Note that with continuous simulation, the opening of the circuit breaker occurs at the next zero crossing of current following the opening order; whereas for the phasor simulation, this opening is instantaneous. This is because there is no concept of zero crossing in the phasor simulation.

Selecting Phasor Signal Measurement Formats

If you now double-click the Voltage Measurement block or the Current Measurement block, you see that a menu allows you to output phasor signals in four different formats: `Complex`, `Real-Imag`, `Magnitude-Angle`, or just `Magnitude` (default choice). If you select `Magnitude-Angle`, both magnitude and angle (in degrees) are multiplexed at the output of the measurement block. You might need to demultiplex these two signals to send them on separate traces of the scope. Note that the oscilloscope does not accept complex signals. You should instead use the `Real-Imag` format.

The `Complex` format allows the use of complex operations and processing of phasors without separating real and imaginary parts. Suppose for example that you need to compute the power consumption of the load (active power P and reactive power Q). The complex power $\bar{S}$ is obtained from the voltage and current phasors as

$$\bar{S} = P + jQ = \frac{1}{2} \cdot \bar{V} \cdot \bar{I}^*$$

where $\bar{I}^*$ is the conjugate of the current phasor. The 1/2 factor is required to convert magnitudes of voltage and current from peak values to RMS values.

Select the `Complex` format for both current and voltage and, using blocks from the Simulink Math library, implement the power measurement as shown.

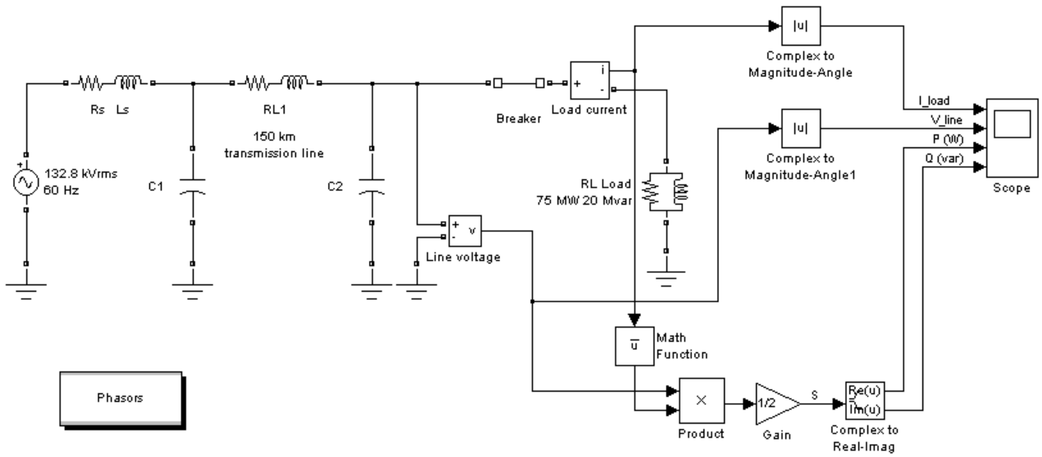


Figure 1-10: Power Computation Using Complex Voltage and Current

The Complex to Magnitude-Angle blocks are now required to convert complex phasors to magnitudes before sending them to the scope.

The power computation system you just implemented is already built into SimPowerSystems: the Active & Reactive Power (Phasor Type) block is available in the Extras library under the Phasor collection of blocks.

2

Advanced Components and Techniques

This chapter introduces methods and devices that enhance your power system simulations and make them more realistic.

The first two tutorials illustrate power electronics, simple motors, and Fourier analysis. The third tutorial demonstrates three-phase power systems, electrical machinery, load flow, and use of the phasor solution method for transient stability studies of electromechanical systems. The fourth explains how you can create and customize your own nonlinear blocks.

Introducing Power Electronics

In this section you

- Learn how to use power electronics components
- Learn how to use transformers
- Change initial conditions of a circuit

SimPowerSystems is designed to simulate power electronic devices. This section uses a simple circuit based on thyristors as the main example.

Consider the circuit shown below. It represents one phase of a static var compensator (SVC) used on a 735 kV transmission network. On the secondary of the 735 kV /16 kV transformer, two variable susceptance branches are connected in parallel: one thyristor-controlled reactor (TCR) branch and one thyristor-switched capacitor (TSC) branch.

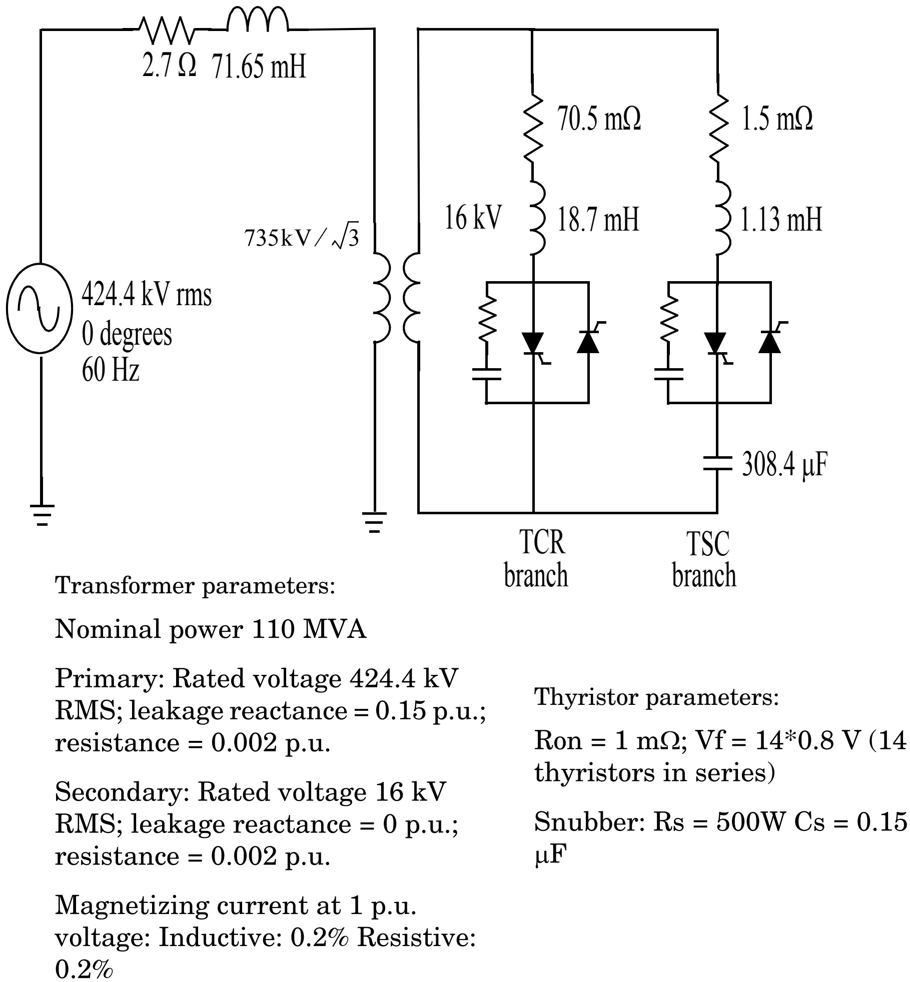


Figure 2-1: One Phase of a TCR/TSC Static Var Compensator

The TCR and TSC branches are both controlled by a valve consisting of two thyristor strings connected in antiparallel. An RC snubber circuit is connected across each valve. The TSC branch is switched on/off, thus providing discrete step variation of the SVC capacitive current. The TCR branch is phase controlled in order to obtain a continuous variation of the net SVC reactive current.

Now build two circuits illustrating the operation of the TCR and the TSC branches.

Simulation of the TCR Branch

1 Open a new window and save it as `circuit3`.

2 Open the Power Electronics library and copy the Thyristor block into your `circuit3` model.

3 Open the **Thyristor** menu and set the parameters as follows:

Ron	`1e-3`
Lon	`0`
Vf	`14*0.8`
Rs	`500`
Cs	`0.15e-6`

   Notice that the snubber circuit is integral to the **Thyristor** dialog box.

4 Rename this block Th1 and duplicate it.

5 Connect this new thyristor Th2 in antiparallel with Th1, as shown in Figure 2-2.

   As the snubber circuit has already been specified with Th1, the snubber of Th2 must be eliminated.

6 Open the **Th2** dialog box and set the snubber parameters to

Rs	`Inf`
Cs	`0`

   Notice that the snubber disappears on the Th2 icon.

The linear transformer is located in the Elements library. Copy it, rename it to TrA, and open its dialog box. Set its nominal power, frequency, and winding parameters (**winding 1** = `primary`; **winding 2** = `secondary`) as shown in Figure 2-1.

Note that the leakage reactance and resistance of each winding have to be specified directly in per unit quantities. As there is no tertiary winding, deselect **Three windings transformer**. Notice that winding 3 disappears on the Tr*A* block.

Finally, set the magnetizing branch parameters **Rm** and **Xm** at `[500, 500]`. These values correspond to 0.2% resistive and inductive currents.

Add a voltage source, series RL elements, and a Ground block. Set the parameters as shown in Figure 2-1. Add a current measurement to measure the primary current. Interconnect the circuit as shown in Figure 2-2.

Notice that the Thyristor blocks have an output identified by the letter `m`. This output returns a Simulink vectorized signal containing the thyristor current (Iak) and voltage (Vak). Connect a Demux block with two outputs at the m output of Th1. Then connect the two demultiplexer outputs to a dual trace scope that you rename Scope_Th1. (To create a second input to your scope, in the **Scope properties —> General** menu item, set the number of axes to `2`.) Label the two connection lines `Ith1` and `Vth1`. These labels are automatically displayed on the top of each trace.

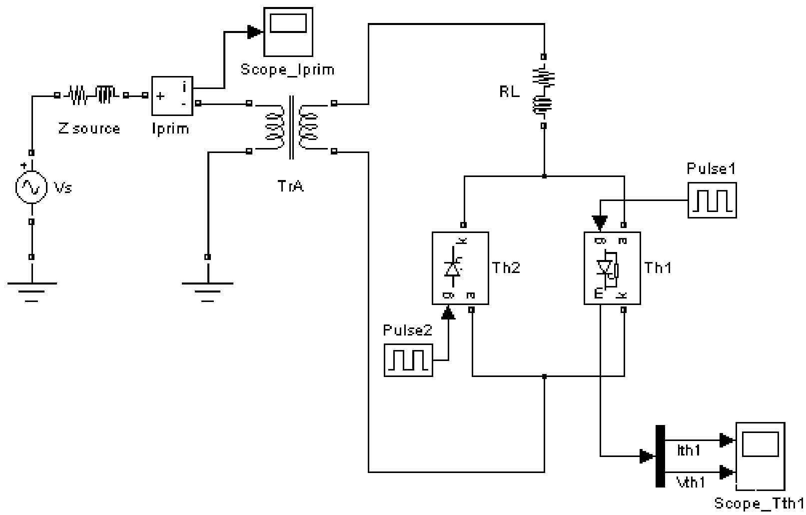


Figure 2-2: Simulation of the TCR Branch

You can now model the synchronized pulse generators firing thyristors Th1 and Th2. Copy two Simulink pulse generators into your system, name them Pulse1 and Pulse2, and connect them to the gates of Th1 and Th2.

Now you have to define the timing of the Th1 and Th2 pulses. At every cycle a pulse has to be sent to each thyristor α degrees after the zero crossing of the thyristor commutation voltage. Set the Pulse1 and Pulse2 parameters as follows:

Amplitude	`1`
Period	`1/60 s`
Pulse width (% of period)	`1% (3.6 degrees pulses)`
Phase Delay	`1/60+T` for `Pulse1` `1/60+1/120+T` for `Pulse2`

The pulses sent to Th2 are delayed by 180 degrees with respect to pulses sent to Th1. The delay T is used to specify the firing angle α. In order to get a 120 degree firing angle, specify `T` in the workspace by entering

```
T = 1/60/3;
```

Now open the **Simulation —> Simulation parameters** dialog. Select the `ode23tb` integration algorithm. Keep the default parameters but set the relative tolerance to `1e-4` and the stop time to `0.1`. Start the simulation. The results are shown in Figure 2-3.

Note You could also choose to discretize your system. Try for example 50 µs sample time. The simulation results should compare well with the continuous system.

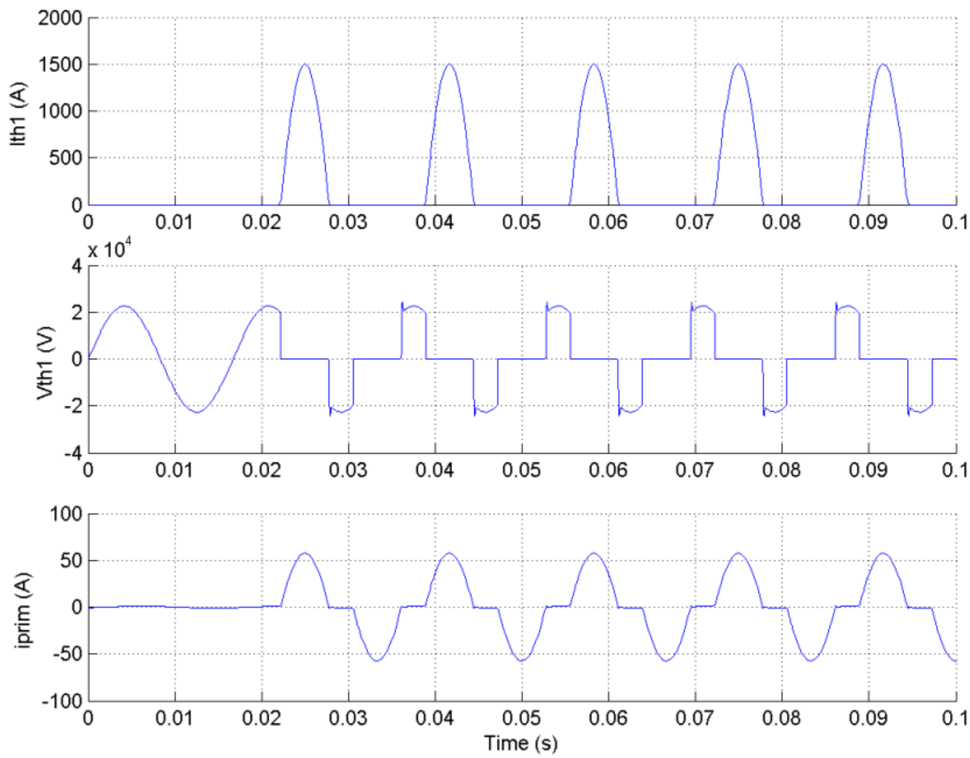


Figure 2-3: TCR Simulation Results

Simulation of the TSC Branch

You can now modify your `circuit3` system and change the TCR branch to a TSC branch. Save `circuit3` as a new system and name it `circuit4`.

Connect a capacitor in series with the RL inductor and Th1/Th2 valve as shown in Figure 2-4. Change the **R**, **L**, and **C** parameters as shown in Figure 2-1. Connect a voltmeter and scope to monitor the voltage across the capacitor.

Contrary to the TCR branch, which was fired by a synchronous pulse generator, a continuous firing signal is now applied to the two thyristors. Delete the two pulse generators. Copy a Step block from the Simulink library and connect its output at both gates of Th1 and Th2. Set its step time at 1/60/4 (energizing at the first positive peak of the source voltage). Your circuit should now be similar to the one shown here.

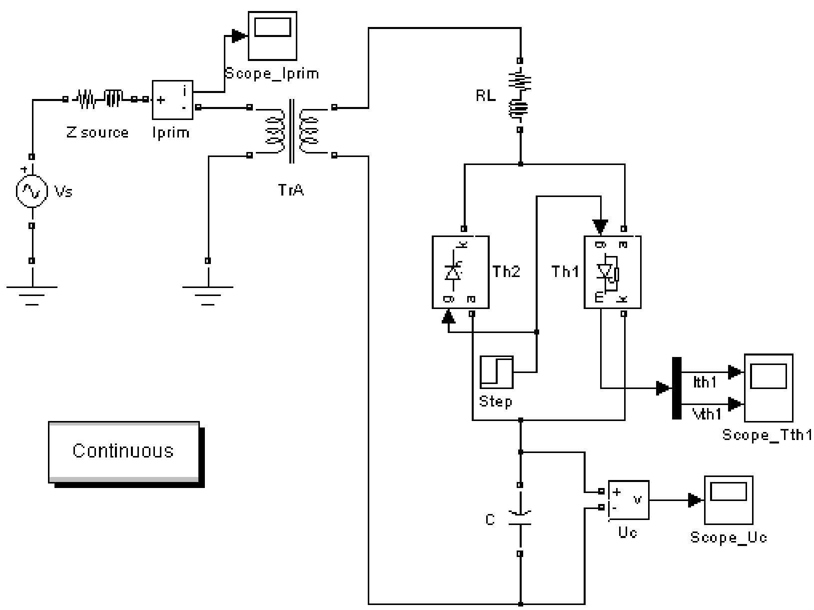


Figure 2-4: Simulation of the TSC Branch

Open the three scopes and start the simulation.

As the capacitor is energized from zero, you can observe a low damping transient at 200 Hz, superimposed with the 60 Hz component in the capacitor voltage and primary current. During normal TSC operation, the capacitor has an initial voltage left since the last valve opening. In order to minimize the closing transient with a charged capacitor, the thyristors of the TSC branch must be fired when the source voltage is at maximum value and with the correct polarity. The initial capacitor voltage corresponds to the steady-state voltage obtained when the thyristor switch is closed. The capacitor voltage is 17.67 kVrms when the valve is conducting. At the closing time, the capacitor must be charged at the peak voltage.

$$Uc = 17670 \times \sqrt{2} = 24989\ V$$

You can now use the Powergui block to change the capacitor initial voltage. Open the Powergui and select **Initial States Setting**. A list of all the state variables with their default initial values appears. The value of the initial voltage across the capacitor C (variable `Uc_C`) should be –0.3141 V. This voltage is not exactly zero because the snubber allows circulation of a small current when both thyristors are blocked. Now select the `Uc_C` state variable and enter `24989` in the upper right field. Then click the **Apply** button in order to make this change effective.

Start the simulation. As expected the transient component of capacitor voltage and current has disappeared. The voltages obtained with and without initial voltage are compared in this plot.

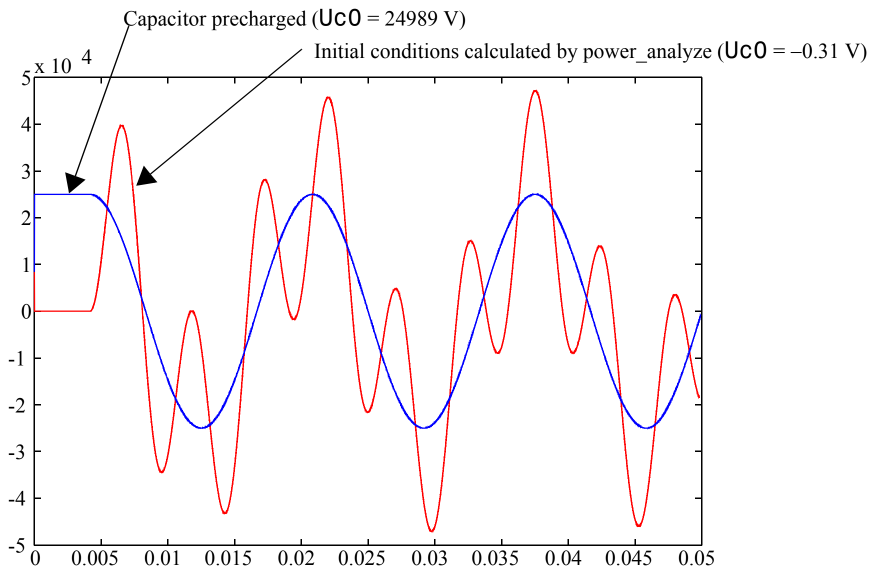


Figure 2-5: Transient Capacitor Voltage With and Without Initial Charge

Simulating Motor Drives

In this section you

- Use electrical machines and power electronics to simulate a simple motor drive
- Learn how to use the Universal Bridge block
- Discretize your model and compare variable-step and fixed-step simulation methods
- Learn how to use the Multimeter block
- Learn how to use the FFT tool

Variable speed control of AC electrical machines makes use of forced-commutated electronic switches such as IGBTs, MOSFETs, and GTOs. Asynchronous machines fed by *pulse width modulation* (PWM) voltage sourced converters (VSC) are nowadays gradually replacing the DC motors and thyristor bridges. With PWM, combined with modern control techniques such as field-oriented control or direct torque control, you can obtain the same flexibility in speed and torque control as with DC machines. This section shows how to build a simple open loop DC drive controlling an asynchronous machine. A more elaborate example of a PWM drive is presented in Chapter 3, "Case Studies." The SimPowerSystems circuit to simulate is shown in Figure 2-6. It uses blocks from the Machines and Power Electronics libraries.

The Machines library contains four of the most commonly used three-phase machines: simplified and complete synchronous machines, asynchronous machine, and permanent magnet synchronous machine. Each machine can be used either in generator or motor mode. Combined with linear and nonlinear elements such as transformers, lines, loads, breakers, etc., they can be used to simulate electromechanical transients in an electrical network. They can also be combined with power electronic devices to simulate drives.

The Power Electronics library contains blocks allowing you to simulate diodes, thyristors, GTO thyristors, MOSFETs, and IGBT devices. You could interconnect several blocks together to build a three-phase bridge. For example, an IGBT inverter bridge would require six IGBTs and six antiparallel diodes.

In order to facilitate implementation of bridges, the Universal Bridge block automatically performs these interconnections for you.

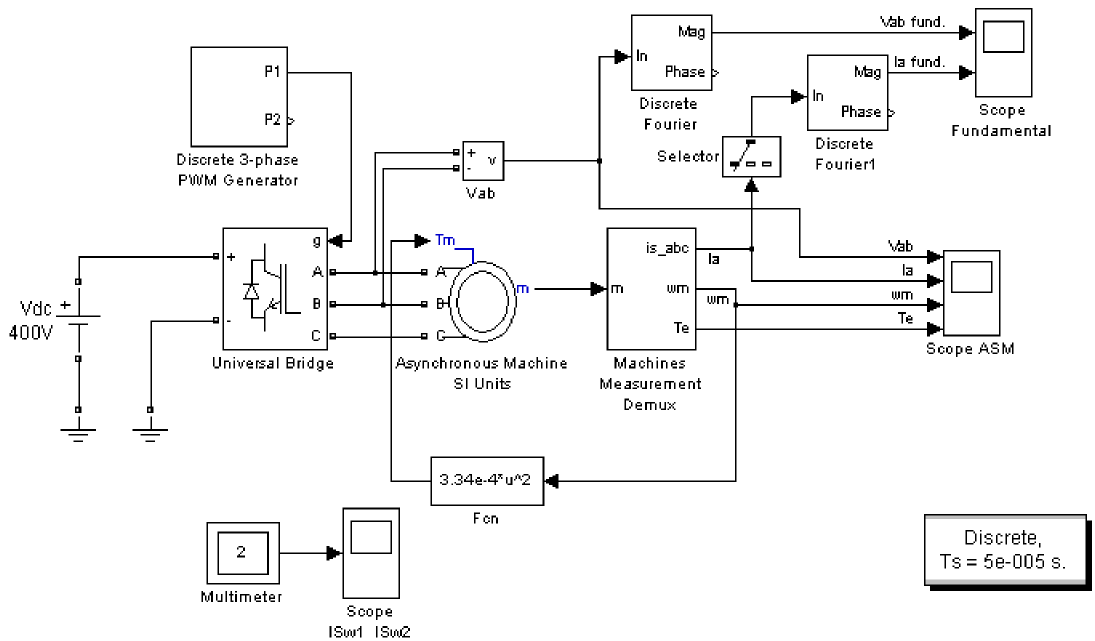


Figure 2-6: Circuit 5: PWM Control of an Induction Motor

Building and Simulating the PWM Motor Drive

Follow these steps to build a PWM-controlled motor.

Assembling and Configuring the Motor Blocks

In the first steps, you copy and set up the motor blocks.

1 Open a new window and save it as `circuit5`.

2 Open the Power Electronics library and copy the Universal Bridge block into your `circuit5` model.

3 Open the Universal Bridge dialog and set its parameters as follows:

Power electronic device	`IGBT/Diodes`
Snubber	
Rs	`1e5` Ω
Cs	`inf`
Ron	`1e-3` Ω
Forward voltages	
Vf	0 V
Vfd	`0 V`
Tail	
Tf	`1e-6 s`
Tt	`1e-6 s`

Notice that the snubber circuit is integral to the Universal Bridge dialog box. As the Cs capacitor value of the snubber is set to `Inf` (short-circuit), we are using a purely resistive snubber. Generally, IGBT bridges do not use snubbers; however, because each nonlinear element in SimPowerSystems is modeled as a current source, you have to provide a parallel path across each IGBT in order to allow connection to an inductive circuit (stator of the asynchronous machine). The high resistance value of the snubber does not affect the circuit performance.

4 Open the Machines library. Copy the Asynchronous Machine SI Units block as well as the Machine Measurement Demux block into your `circuit5` model.

5 Open the Asynchronous Machine menu and look at its parameters. They are set for a 3 HP, 60 Hz machine with two pairs of poles. Its nominal speed is therefore slightly lower than the synchronous speed of 1800 rpm, or w_s= 188.5 rad/s.

6 Notice that the three rotor terminals `a`, `b`, and `c` are made accessible. During normal motor operation these terminals should be short-circuited together.

In the Asynchronous Machine menu change the rotor type to **Squirrel cage**. Notice that after this change the rotor connections are no longer accessible.

7 Open the Machine Measurement Demux block menu. When this block is connected to a machine measurement output, it allows you to access specific internal signals of the machine. First select the **Asynchronous** machine type. Deselect all signals except the following three signals: `is_abc` (three stator currents), `wm` (rotor speed), and `Te` (electromagnetic torque).

Loading and Driving the Motor

You now implement the torque-speed characteristic of the motor load. Assume a quadratic torque-speed characteristic (fan or pump type load). The torque T is then proportional to the square of the speed ω.

$$T = k \times \omega^2$$

The nominal torque of the motor is

$$T_n = \frac{3 \times 746}{188.5} = 11.87Nm$$

Therefore, the constant k should be

$$k = \frac{T_n}{\omega_s^2} = \frac{11.87}{188.5^2} = 3.34 \times 10^{-4}$$

1 Open the Math Operations library of Simulink and copy the Math Function block into your `circuit5` model. Open the block menu and enter the expression of torque as a function of speed: `3.34e-4*u^2`.

2 Connect the input of the Math Function block to the speed output of the Machines Measurement Demux block, labeled `wm`, and its output to the torque input of the motor, labeled `Tm`.

3 Open the Electrical Sources library and copy the DC Voltage Source block into your `circuit5` model. Open the block menu and set the voltage to 400 V.

4 Open the Measurements library and copy a Voltage Measurement block into your `circuit5` model. Change the block name to Vab.

5 Using Ground blocks from the Elements library, complete the power elements and voltage sensor interconnections as shown in Figure 2-6.

Controlling the Inverter Bridge with a Pulse Generator

In order to control your inverter bridge, you need a pulse generator. Such a generator is available in the Extras library of **powerlib**.

1 Open the Extras/Discrete Control blocks library and copy the Discrete 3-Phase PWM Generator block into your `circuit5` model. This block can be used to generate pulses for a two-level or a three-level bridge. In addition the block generates two sets of pulses (outputs P1 and P2) that can be sent to two different three-arm bridges when the converter uses a twin bridge configuration. In this case, use it as a two-level single-bridge PWM generator. The converter operates in an open loop, and the three PWM modulating signals are generated internally. Connect the P1 output to the pulses input of the Universal Bridge block

2 Open the Discrete Three-Phase PWM Generator block dialog and set the parameters as follows.

Type	`2 level`
Mode of operation	`Un-synchronized`
Carrier frequency	`18*60Hz (1080 Hz)`
Internal generation of modulating signals	Selected
Modulation index m	`0.9`
Output voltage frequency	`60 Hz`
Output voltage phase	`0 degrees`
Sample time	`10e-6 s`

3 Use the **Edit —> Look Under Mask** menu item of your model window to see how the PWM is implemented. This control system is made entirely with Simulink blocks. The block has been discretized so that the pulses change at

multiples of the specified time step. A time step of 10 µs corresponds to +/– 0.54% of the switching period at 1080 Hz.

One common method of generating the PWM pulses uses comparison of the output voltage to synthesize (60 Hz in this case) with a triangular wave at the switching frequency (1080 Hz in this case). This is the method that is implemented in the Discrete 3-Phase PWM Pulse Generator block. The line-to-line RMS output voltage is a function of the DC input voltage and of the modulation index *m* as given by the following equation:

$$V_{LLrms} = \frac{m}{2} \times \frac{\sqrt{3}}{\sqrt{2}} Vdc = m \times 0.612 \times VDC$$

Therefore, a DC voltage of 400 V and a modulation factor of 0.90 yield the 220 Vrms output line-to-line voltage, which is the nominal voltage of the asynchronous motor.

Displaying Signals and Measuring Fundamental Voltage and Current

1 You now add blocks measuring the fundamental component (60 Hz) embedded in the chopped Vab voltage and in the phase A current. Open the Extras/Discrete Measurements library of **powerlib** and copy the Discrete Fourier block into your `circuit5` model.

   Open the Discrete Fourier block dialog and check that the parameters are set as follows:

Fundamental frequency f1	`60 Hz`
Harmonic number	`1`
Initial input	`[0 0]`
Sample time	`10e-6 s`

   Connect this block to the output of the Vab voltage sensor.

2 Duplicate the Discrete Fourier block. In order to measure the phase A current, you need to select the first element of the `is_abc` output of the ASM Measurement Demux block.

Copy a Selector block from the Signals & Systems library of Simulink.

Open its menu and set **Element** to `1`. Connect the Selector output to the second Discrete Fourier block and its input to the `is_abc` output of the Machines Measurement Demux block as shown in Figure 2-6.

3 Finally, add scopes to your model. Copy one Scope block into your circuit. This scope is used to display the instantaneous motor voltage, currents, speed, and electromagnetic torque. In the **Scope Properties —> General** menu of the scope, set the following parameters:

Number of axes	`4`
Time range	`0.05 s`
Tick labels	`bottom axis only`

Connect the four inputs and label the four connection lines as shown in Figure 2-6. When you start the simulation, these labels are displayed on top of each trace.

In order to allow further processing of the signals displayed on the oscilloscope, you have to store them in a variable. In the **Scope Parameters/Data history** menu of the scope, set the following parameters:

Limit data point to last	`deselected`
Save data to workspace	`selected`
variable name	`ASM`
Format	`Structure with time`

After simulation, the four signals displayed on the scope are available in a structure array named ASM.

4 Duplicate the four-input Scope and change its number of inputs to 2. This scope is used to display the fundamental component of Vab voltage and Ia current. Connect the two inputs to the outputs of the Fourier blocks. Label the two connection lines as shown in Figure 2-6.

You are now ready to simulate the motor starting.

Simulating the PWM Motor Drive with Continuous Integration Algorithm

Open the **Simulation —> Simulation parameters** menu. Select the `ode23tb` integration algorithm. Set the relative tolerance to `1e-4`, the absolute tolerance and the Max step size to auto, and the stop time to `1 s`. Start the simulation. The simulation results are shown in Figure 2-7.

The motor starts and reaches its steady-state speed of 181 rad/s (1728 rpm) after 0.5 s. At starting, the magnitude of the 60 Hz current reaches 90 A peak (64 A RMS) whereas its steady-state value is 10.5 A (7.4 A RMS). As expected, the magnitude of the 60 Hz voltage contained in the chopped wave stays at

$$220 \times \sqrt{2} = 311V$$

Also notice strong oscillations of the electromagnetic torque at starting. If you zoom in on the torque in steady state, you should observe a noisy signal with a mean value of 11.9 N.m, corresponding to the load torque at nominal speed.

If you zoom in on the three motor currents, you can see that all the harmonics (multiples of the 1080 Hz switching frequency) are filtered by the stator inductance, so that the 60 Hz component is dominant.

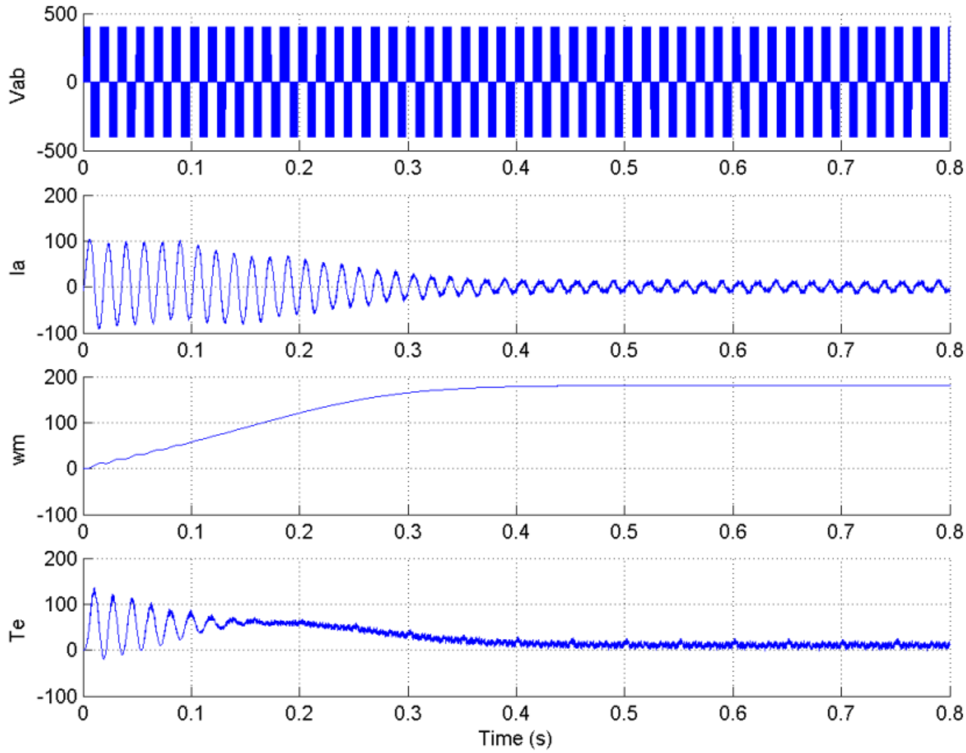


Figure 2-7: PWM Motor Drive; Simulation Results for Motor Starting at Full Voltage

Using the Multimeter Block

The Universal Bridge block is not a conventional subsystem where all the six individual switches are accessible. If you want to measure the switch voltages and currents, you must use the Multimeter block, which gives access to the bridge internal signals.

1 Open the **Universal Bridge** dialog and set the **Measurement** parameter to `Device currents`.

2 Copy the Multimeter block from the Measurements library into your `circuit5` circuit. Double-click the Multimeter block. A window showing the six switch currents appears.

3 Select the two currents of the bridge arm connected to phase A. They are identified as

iSw1	`Universal Bridge`
iSw2	`Universal Bridge`

4 Click **OK**. The number of signals (2) is displayed in the multimeter icon.

5 Using a Demux block, send the two multimeter output signals to a two-trace scope and label the two connection lines (Trace 1: `iSw1` Trace 2: `iSw2`).

6 Restart the simulation. The waveforms obtained for the first 20 ms are shown in this plot.

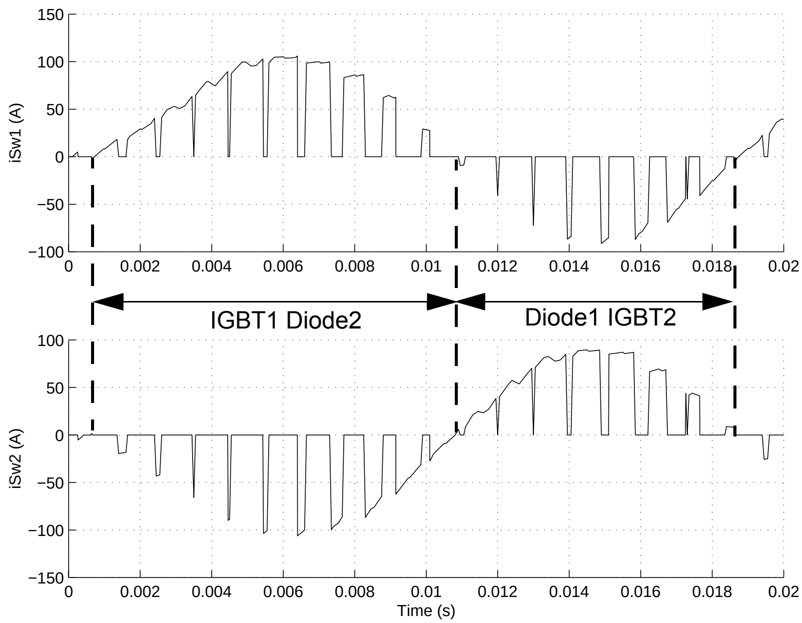


Figure 2-8: Currents in IGBT/Diode Switches 1 and 2

As expected, the currents in switches 1 and 2 are complementary. A positive current indicates a current flowing in the IGBT, whereas a negative current indicates a current in the antiparallel diode.

Note Multimeter block use is not limited to the Universal Bridge block. All the elements of the Electrical Sources and Elements libraries have a Measurement parameter where you can select voltages, currents, and saturable transformer fluxes. A judicious use of the Multimeter block reduces the number of current and voltage sensors in your circuit, making it easier to follow.

Discretizing the PWM Motor Drive

You might have noticed that the simulation using a variable-step integration algorithm is relatively long. Depending on your computer, it might take some minutes to simulate one second. In order to shorten the simulation time, you can discretize your circuit and simulate at fixed simulation time steps.

Open the Powergui and select **Discretize electrical model**. Set the **Sample Time** to `10e-6 s`. When you restart the simulation, the power system, including the asynchronous machine, is discretized at a 10 μs sample time.

As there are no more continuous states in the electrical system, you do not need a variable-step integration method to solve this system. In the **Simulation —> Simulation parameters —> Solver** dialog pane, select the `Fixed-step` and `discrete (no continuous states)` options.

Start the simulation. Observe that the simulation is now approximately three times faster than with the continuous system. Results compare well with the continuous system.

Performing Harmonic Analysis Using the FFT Tool

The two Discrete Fourier blocks allow computation of the fundamental component of voltage and current while simulation is running. If you would like to observe harmonic components also you would need a Discrete Fourier block for each harmonic. This approach is not convenient.

Now use the FFT tool of Powergui to display the frequency spectrum of voltage and current waveforms. These signals are stored in your workspace in the ASM structure array.

Open Powergui and select **FFT Analysis.** A new window opens. Set the parameters specifying the analyzed signal, the time window, and the frequency range as follows:

Structure	`ASM`
Input	`Vab`
Signal number	`1`
Start time	`0.7 s`
Number of cycles	`2`
(pull-down menu)	`Display FFT window`
Fundamental frequency	`60 Hz`
Max Frequency	`5000 Hz`
Frequency axis	`Harmonic order`
Display style	`Bar (relative to Fund or DC)`

The analyzed signal is displayed in the upper window. Click **Display**. The frequency spectrum is displayed in the bottom window. See Figure 2-9.

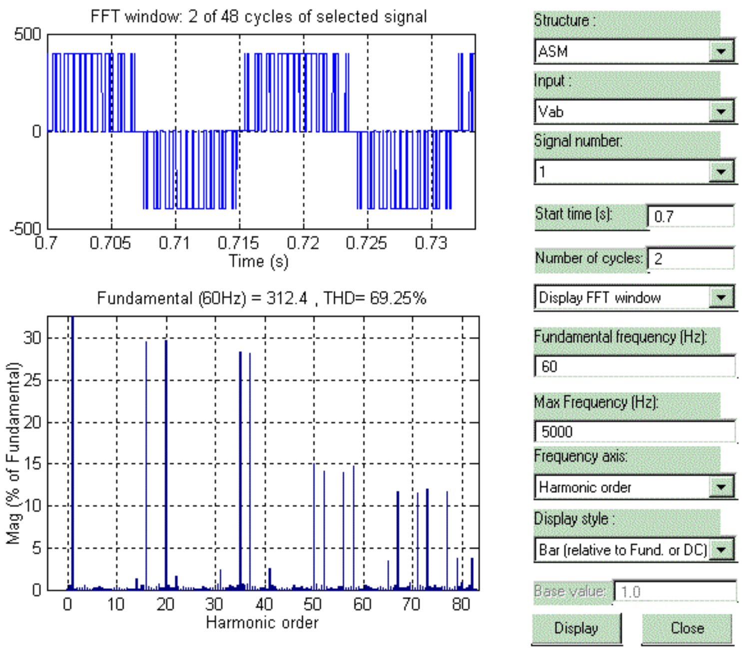


Figure 2-9: FFT Analysis of the Motor Line-to-Line Voltage

The fundamental component and *total harmonic distortion* (THD) of the Vab voltage are displayed above the spectrum window. The magnitude of the fundamental of the inverter voltage (312 V) compares well with the theoretical value (311 V for m=0.9).

Harmonics are displayed in percent of the fundamental component. As expected, harmonics occur around multiples of carrier frequency (n*18 +– k). Highest harmonics (30%) appear at 16th harmonic (18 – 2) and 20th harmonic (18 + 2). Note that the THD value (69%) has been computed for the specified 0 to 5000 Hz frequency range. If you recompute the FFT with a maximum frequency range of 10000 Hz, you should see the THD increasing to 74% (5% contribution in THD for the 5000 to 10000 Hz frequencies).

Finally, select input Ia instead of Vab and display its current spectrum.

Three-Phase Systems and Machines

In this section you

- Learn how to simulate a three-phase power system containing electrical machines and other three-phase models
- Perform a load flow study and initialize machines to start simulation in steady state by using the **Load Flow and Machine Initialization** option of the Powergui
- Simulate the power system and observe its dynamic performance by using both the standard solution technique using a continuous solver and the **Phasor Solution** method

You now use three types of machines of the Electrical Machines library: simplified synchronous machine, detailed synchronous machine, and asynchronous machine. You interconnect these machines with linear and nonlinear elements such as transformers, loads, and breakers to study the transient stability of an uninterruptible power supply using a diesel generator.

Three-Phase Network with Electrical Machines

The two-machine system shown in this single line diagram is this section's main example:

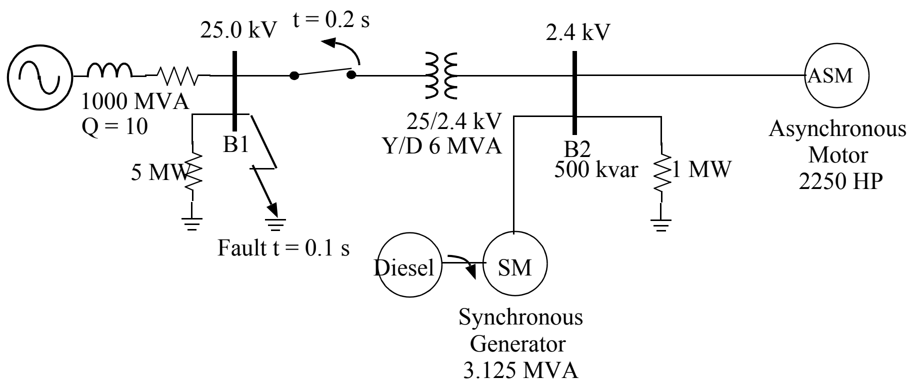


Figure 2-10: Diesel Generator and Asynchronous Motor on Distribution Network

This system consists of a plant (bus B2), simulated by a 1 MW resistive load and a motor load (ASM) fed at 2400 V from a distribution 25 kV network through a 6 MVA, 25/2.4 kV transformer, and from an emergency synchronous generator/diesel engine unit (SM).

The 25 kV network is modeled by a simple R-L equivalent source (short-circuit level 1000 MVA, quality factor X/R = 10) and a 5 MW load. The asynchronous motor is rated 2250 HP, 2.4 kV, and the synchronous machine is rated 3.125 MVA, 2.4 kV.

Initially, the motor develops a mechanical power of 2000 HP and the diesel generator is in standby, delivering no active power. The synchronous machine therefore operates as a synchronous condenser generating only the reactive power required to regulate the 2400 V bus B2 voltage at 1.0 p.u. At t = 0.1 s, a three-phase to ground fault occurs on the 25 kV system, causing the opening of the 25 kV circuit breaker at t = 0.2 s, and a sudden increase of the generator loading. During the transient period following the fault and islanding of the motor-generator system, the synchronous machine excitation system and the diesel speed governor react to maintain the voltage and speed at a constant value.

This system is modeled in a SimPowerSystems demo. Open the Demos library of **powerlib** and double-click the demo called "Three-Phase Machines and Load Flow." A system named `power_machines` opens.

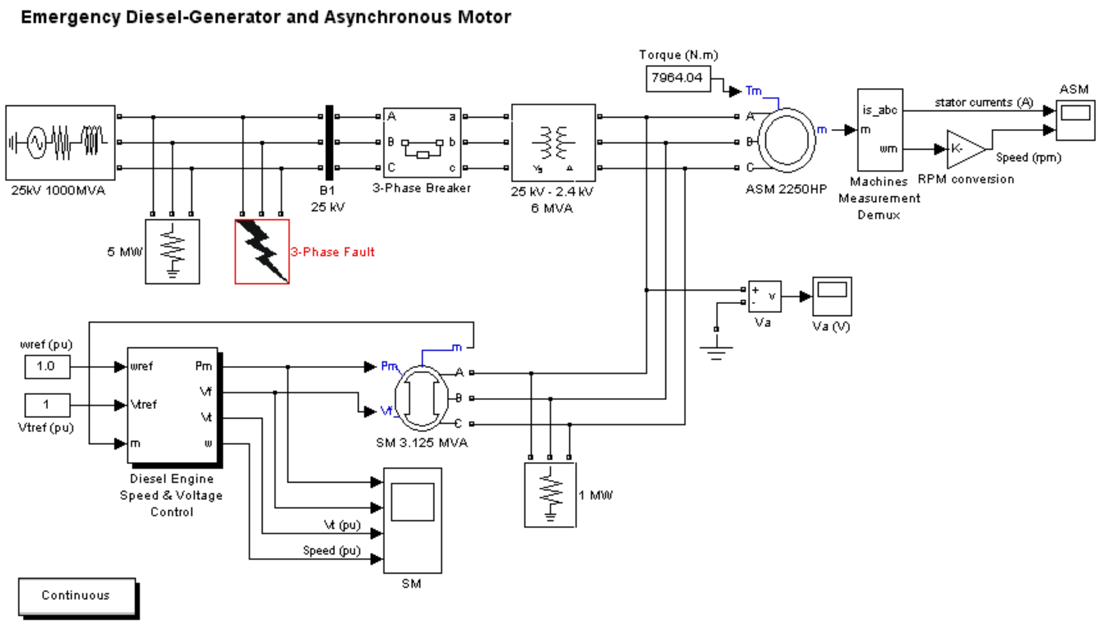


Figure 2-11: Power System of Figure 2-10 Built with SimPowerSystems

The Synchronous Machine (SM) block uses standard parameters, whereas the Asynchronous Machine (ASM) block uses S.I. parameters.

The other three-phase elements such as the inductive voltage source, the Y grounded/Delta transformer, and the loads are standard blocks from the Electrical Source and Elements libraries of **powerlib**. If you open the dialog box of the Three-Phase Fault and Three-Phase Breaker blocks, you see how the switching times are specified. The Machine Measurement Demux block provided in the Machines library is used to demux the output signals of the SM and ASM machines.

The SM voltage and speed outputs are used as feedback inputs to a Simulink control system that contains the diesel engine and governor block as well as an excitation block. The excitation system is the standard block provided in the Machines library. The SM parameters as well as the diesel engine and governor models were taken from reference [1].

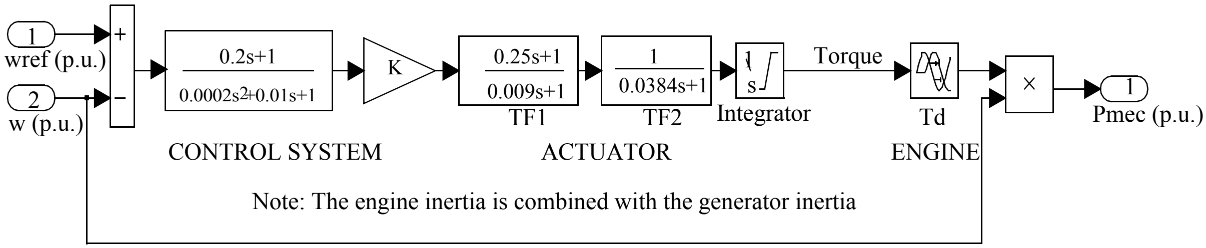


Figure 2-12: Diesel Engine and Governor System

If you simulate this system for the first time, you normally do not know what the initial conditions are for the SM and ASM to start in steady state.

These initial conditions are

- SM block: Initial values of speed deviation (usually 0%), rotor angle, magnitudes and phases of currents in stator windings, and initial field voltage required to obtain the desired terminal voltage under the specified load flow
- ASM block: Initial values of slip, rotor angle, magnitudes and phases of currents in stator windings

Open the dialog box of the Synchronous Machine and Asynchronous Machine blocks. All initial conditions should be set at `0`, except for the initial SM field voltage and ASM slip, which are set at `1 p.u.` Open the three scopes monitoring the SM and ASM signals as well as the bus B2 voltage. Start the simulation and observe the first 100 ms before fault is applied.

As the simulation starts, note that the three ASM currents start from zero and contain a slowly decaying DC component. The machine speeds take a much longer time to stabilize because of the inertia of the motor/load and diesel/generator systems. In our example, the ASM even starts to rotate in the wrong direction because the motor starting torque is lower than the applied load torque. Stop the simulation.

Load Flow and Machine Initialization

In order to start the simulation in steady state with sinusoidal currents and constant speeds, all the machine states must be initialized properly. This is a difficult task to perform manually, even for a simple system. In the next section you learn how to use the **Load Flow and Machine Initialization** option of the Powergui to perform a load flow and initialize the machines.

Double-click the Powergui**.** In the **Tools** menu, click the **Load Flow and Machine Initialization** button. A new window appears. In the upper right window you have a list of the machines appearing in your system. Select the SM 3.125 MVA machine. Note that for the **Bus Type**, you have a menu allowing you to choose either PV Generator, PQ Generator, or Swing Generator.

For synchronous machines you normally specify the desired terminal voltage and the active power that you want to generate (positive power for generator mode) or absorb (negative power for motor mode). This is possible as long as you have a *swing* (or *slack)* bus that generates or absorbs the excess power required to balance the active powers throughout the network.

The swing bus can be either a voltage source or any other synchronous machine. If you do not have any voltage source in your system, you must declare one of the machines as a swing machine. In the next section, you perform a load flow with the 25 kV voltage source connected to bus B1, which is used as a swing bus.

Load Flow Without a Swing Machine

In the **Load Flow** window, your SM **Bus Type** should already be initialized as `P & V generator`, indicating that the load flow is performed with the machine controlling its active power and terminal voltage. By default, the desired **Terminal Voltage UAB** is initialized at the nominal machine voltage (2400 Vrms). Keep it unchanged and set the **Active Power** to zero. The synchronous machine therefore absorbs or generates reactive power only in order to keep terminal voltage at 1 p.u. Now select the ASM 2250 HP machine in the upper right window. The only parameter that is needed is the **Mechanical power** developed by the motor. Enter 2000*746 (2000 HP). You now perform the load flow with the following parameters.

SM

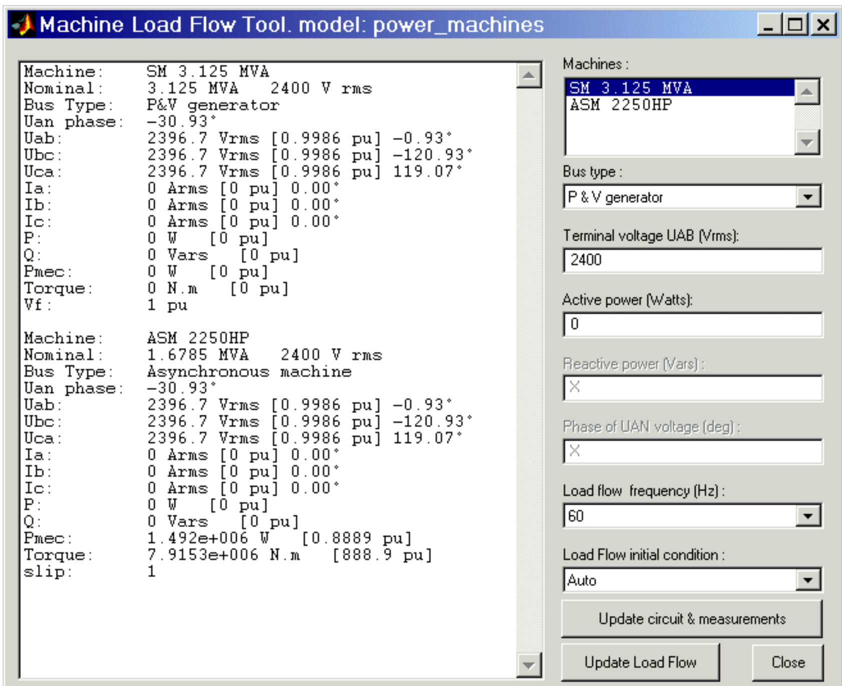
Terminal Voltage `2400 Vrms`

Active Power `0 kW`

ASM

Mechanical Power `2000*746 W (2000 HP)`

Click the **Update Load Flow** button. Once the load flow is solved, the three phasors of line-to-line machine voltages as well as currents are updated as shown on the next figure. Values are displayed both in SI units (volts RMS or amperes RMS) and in p.u.



The SM active and reactive powers, mechanical power, and field voltage are displayed.

`P`	`0 W`
`Q`	`856 kvar` or `856/3125=0.2739 p.u.`

`Pmec`	`844.2 W` or `0.00027 p.u.`, representing internal machine losses in stator windings
`Ef (field voltage)`	`1.427 p.u.`

The ASM active and reactive powers absorbed by the motor, slip, and torque are also displayed.

`P`	`1.515 MW (0.9024 p.u.)`
`Q`	`615 kvar (0.3662 p.u.)`
`Pmec`	`1.492 MW (2000 HP)`
`Slip`	`0.006119`
`Torque`	`7964 N.m (0.8944 p.u.)`

Close the **Load Flow** window.

The ASM torque value (7964 N.m) should already be entered in the Constant block connected at the ASM torque input. If you now open the **SM** and **ASM** dialog boxes you can see the updated initial conditions. If you open the Powergui, you can see updated values of the measurement outputs. You can also click the **Nonlinear** button to obtain voltages and currents of the nonlinear blocks. For example, you should find that the magnitude of the Phase A voltage across the fault breaker (named `Uc_3-Phase Fault/Breaker1`) is 14.42 kV RMS, corresponding to a 24.98 kV RMS phase-to-phase voltage.

In order to start the simulation in steady state, the states of the Governor & Diesel Engine and the Excitation blocks should also be initialized according to the values calculated by the load flow. Open the Governor & Diesel Engine subsystem, which is inside the Diesel Engine Speed and Voltage Control subsystem. Notice that the initial mechanical power has been automatically set to `0.0002701 p.u.` Open the Excitation block and notice that the initial terminal voltage and field voltage have been set respectively to `1.0` and `1.427 p.u.`

Note that the load flow also initializes the Constant blocks connected at the reference inputs (wref and vref) of the Governor and Excitation blocks as well as the Constant block connected at the load torque input (Tm) of the Asynchronous Machine block.

Open the three scopes displaying the internal signals of synchronous and asynchronous machines and phase A voltage. Start the simulation. The simulation results are shown in the following figure.

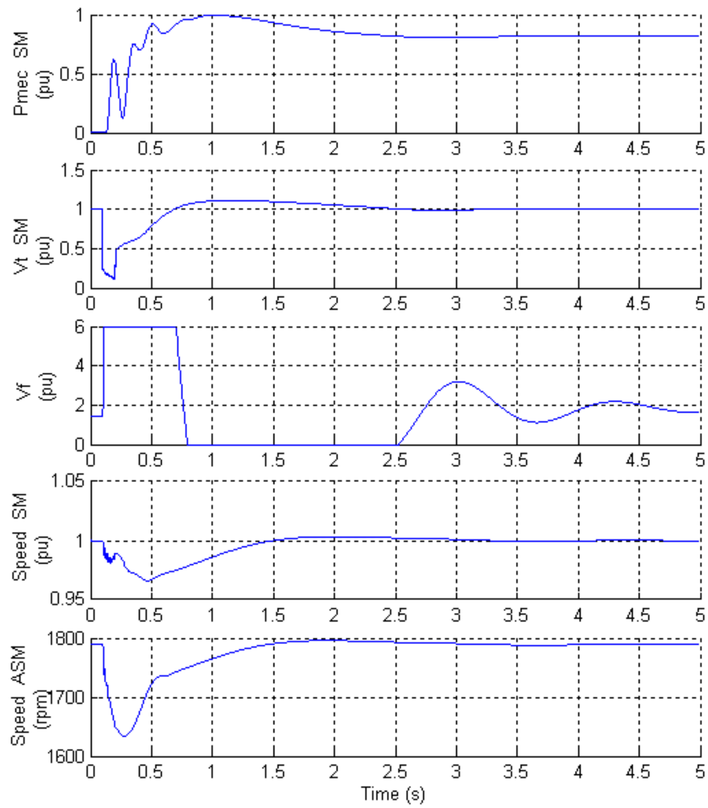


Figure 2-13: Simulation Results

Observe that during the fault, the terminal voltage drops to about 0.2 p.u., and the excitation voltage hits the limit of 6 p.u. After fault clearing and islanding, the SM mechanical power quickly increases from its initial value of 0 p.u. to 1 p.u. and stabilizes at the final value of 0.82 p.u. required by the resistive and motor load (1.0 MW resistive load + 1.51 MW motor load = 2.51 MW = 2.51/3.125 = 0.80 p.u.). After 3 seconds the terminal voltage stabilizes close to its reference value of 1.0 p.u. The motor speed temporarily decreases from 1789 rpm to 1635 rpm, then recovers close to its normal value after 2 seconds.

If you increase the fault duration to 12 cycles by changing the breaker opening time to 0.3 s, notice that the system collapses. The ASM speed slows down to zero after 2 seconds.

Load Flow With a Swing Machine

In this section you make a load flow with two synchronous machine types: a *PV generator* and a *swing generator*. In your **power_machines** window, delete the inductive source and replace it with the Simplified Synchronous Machine block in p.u. that you find in the Machines library. Rename it SSM 1000MVA and save this new system in your working directory as `power_machines2.mdl`. Open the SSM 1000 MVA dialog box and enter the following parameters.

Connection type	`3-wire Y`
Pn(VA), Vn(Vrms), fn(Hz)	`[1000e6 25e3 60]`
H(s), **Kd()**, **p ()**	`[inf 0 2]`
R(p.u.), X(p.u.)	`[0.1 1.0]`
Init. cond.	Leave all initial conditions at zero.

As you specify an infinite inertia, the speed and therefore the frequency of the machine are kept constant. Notice how easily you can specify an inductive short-circuit level of 1000 MVA and a quality factor of 10 with the per unit system.

Also, connect at inputs 1 and 2 of the SSM block two Constant blocks specifying respectively the required mechanical power (Pmec) and its internal voltage (E). These two constants are updated automatically according to the load flow solution.

When there is no voltage source imposing a reference angle for voltages, you must choose one of the synchronous machines as a reference. In a load flow program, this reference is called the *swing bus.* The swing bus absorbs or generates the power needed to balance the active power generated by the other machines and the power dissipated in loads as well as losses in all elements.

Open the Powergui. In the **Tools** menu, select **Load Flow and Machine Initialization**. Change the SSM **Bus Type** to `Swing Generator`. Specify the load flow by entering the following parameters for the SM and ASM machines:

`SM 1000 MVA:`

Terminal Voltage UAB	`2400 Vrms`
Active Power	`0 W`

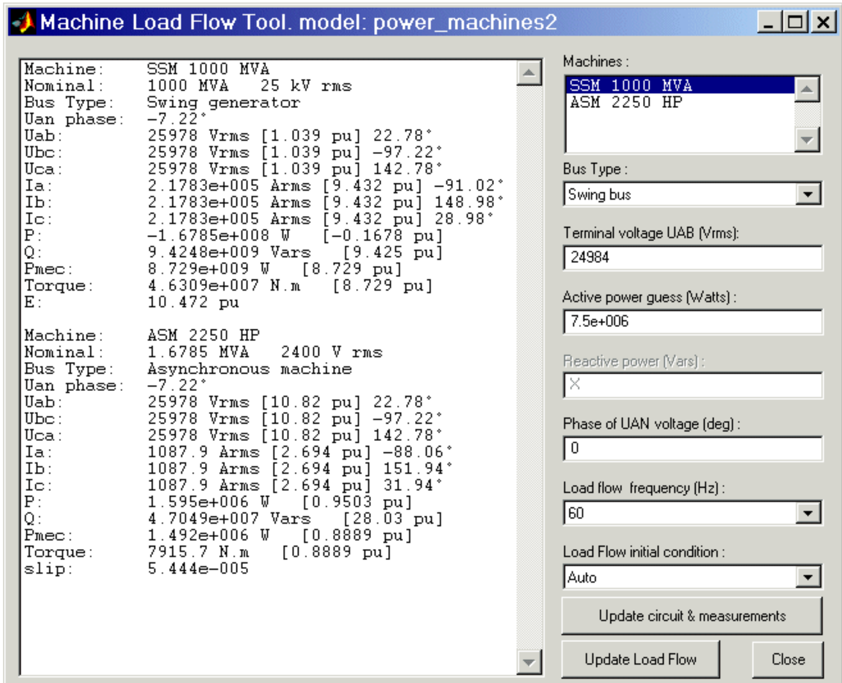
`ASM 2250 HP:`

Mechanical power	`1.492e+06 W (2000 HP)`

For the SSM swing machine you only have to specify the requested terminal voltage (magnitude and phase). The active power is unknown. However, you can specify an active power that is used as an initial guess and help load flow convergence. Respecify the following SSM parameters:

Terminal Voltage	`24984 Vrms` (this voltage obtained at bus B1 from the previous load flow)
Phase of UAN voltage	`0 degrees`
Active Power guess	`7.5e6 W` (estimated power = 6 MW (resistive load) + 1.5 MW motor load)

Click the **Update Load Flow** button. Once the load flow is solved the following solution is displayed. Use the scroll bar of the left window to look at the solution for each of the three machines.



The active and reactive electrical powers, mechanical power, and internal voltage are displayed for the SSM block.

`P=7.542 MW; Q=-147 kvar;`
`Pmec=7.547 MW` (or `7.547/1000=0.007547 p.u.`);
`Internal voltage E=1.0 p.u.`

The active and reactive electrical powers, mechanical power, and field voltage of the SM block are

```
P=0 W; Q=856 kvar;
Pmec=844 W; Vf=1.428 p.u.
```

The active and reactive powers absorbed by the motor, slip, and torque of the ASM block are also displayed.

```
P=1.515MW; Q=615 kvar; Pmec=1.492 MW (2000 HP)
Slip=0.006119; Torque=7964 N.m
```

As expected, the solution obtained is exactly the same as the one obtained with the R-L voltage source. The active power delivered by the swing bus is 7.54 MW (6.0 MW resistive load + 1.51 MW motor load = 7.51 MW, the difference (0.03 MW) corresponding to losses in the transformer).

Restart the simulation. You should get the same waveforms as those of Figure 2-13.

Reference

[1] Yeager, K.E., and J.R.Willis, "Modeling of Emergency Diesel Generators in an 800 Megawatt Nuclear Power Plant," *IEEE Transactions on Energy Conversion*, Vol. 8, No. 3, September, 1993.

Using the Phasor Solution Method for Stability Studies

Up to now, you have simulated a relatively simple power system consisting of a maximum of three machines. If you increase complexity of your network by adding extra lines, loads, transformers, and machines, the required simulation time becomes longer and longer. Moreover, if you are interested in slow electromechanical oscillation modes (typically between 0.02 Hz and 2 Hz on large systems) you might have to simulate for several tens of seconds, implying simulation times of minutes and even hours. The conventional continuous or discrete solution method is therefore not practical for stability studies involving low-frequency oscillation modes. In order to allow such studies, you have to use the phasor technique (see "Introducing the Phasor Simulation Method" on page 1-26).

For a stability study, we are not interested in the fast oscillation modes resulting from the interaction of linear R, L, C elements and distributed parameter lines. These oscillation modes, which are usually located above the fundamental frequency of 50 Hz or 60 Hz, do not interfere with the slow machine modes and regulator time constants. In the phasor solution method, these fast modes are ignored by replacing the network's differential equations by a set of algebraic equations. The state-space model of the network is therefore replaced by a transfer function evaluated at the fundamental frequency and relating inputs (current injected by machines into the network) and outputs (voltages at machine terminals). The phasor solution method uses a reduced state-space model consisting of slow states of machines, turbines, and regulators, thus dramatically reducing the required simulation time. Continuous variable-step solvers are very efficient in solving this type of problem. Recommended solvers are ode15s or ode23tb with a maximum time step of one cycle of the fundamental frequency (1/60 s or 1/50 s).

Now apply the phasor solution method to the two-machine system you have just simulated with the conventional method. Open the `power_machines` demo.

Double-click the Powergui. Select the **Phasor simulation** option. You must also specify the fundamental frequency used to solve the algebraic network equations. A default value of 60 Hz should already be entered in the **Frequency** menu. Close the Powergui and notice that **Phasors** appears on thee Powergui icon, indicating that this new method can be used to simulate your circuit. In order to start the simulation in steady state, you must first repeat the load flow and machine initialization procedure explained in the previous section, "Load Flow and Machine Initialization" on page 2-28.

In the **Simulation Parameters** dialog, specify a **Max step size** of `1/60 s` (one cycle) and start the simulation.

Observe that simulation is now much faster. The results compare well with those obtained in the previous simulation. A comparison of synchronous machine and asynchronous machine signals is shown in Figure 2-14.

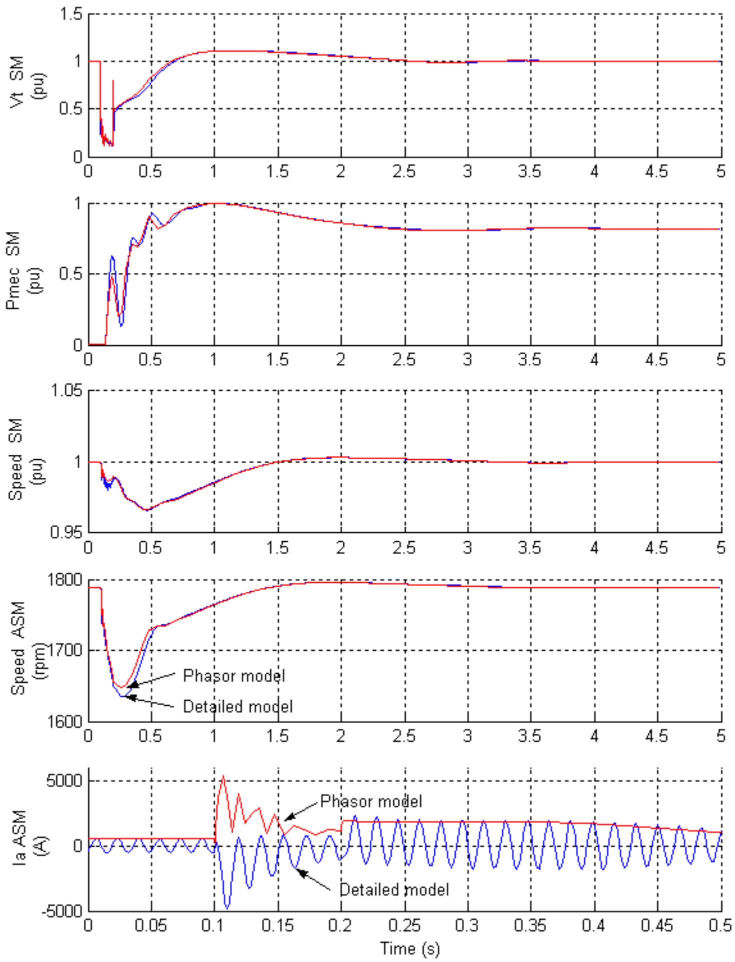


Figure 2-14: Comparison of Results for Continuous and Phasor Simulation Methods

The phasor solution method is illustrated on more complex networks presented in the Demos library. These demos are identified as

- Transient stability of two machines with PSS and SVC
- Performance of three PSS for interarea oscillations

The first demo illustrates the impact of power system stabilizers (PSS) and use of a static var compensator (SVC) to stabilize a two-machine system. The second demo compares the performance of three different types of power system stabilizers on a four-machine, two-area system.

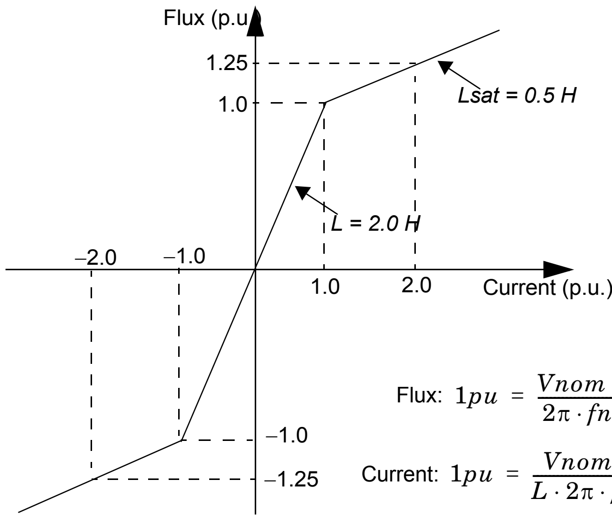
Building and Customizing Nonlinear Models

SimPowerSystems provides a wide collection of nonlinear models. It can happen, however, that you need to interface your own nonlinear model with the standard models provided in the **powerlib** library. This model could be a simple nonlinear resistance simulating an arc or a varistor, a saturable inductor, a new type of motor, etc.

In the following section you learn how to build such a nonlinear model. A simple saturable inductance and a nonlinear resistance serve as examples.

Modeling a Nonlinear Inductance

Consider an inductor of 2 henries designed to operate at a nominal voltage, Vnom = 120 V RMS, and a nominal frequency, fnom = 60 Hz. From zero to 120 V RMS the inductor has a constant inductance, L = 2 H. When voltage exceeds its nominal voltage, the inductor saturates and its inductance is reduced to Lsat = 0.5 H. The nonlinear flux-current characteristic is plotted in the next figure. Flux and current scales are in per units. The nominal voltage and nominal current are chosen as base values for the per-unit system.



Flux: $$1pu = \frac{Vnom \cdot \sqrt{2}}{2\pi \cdot fnom} = \frac{120 \cdot \sqrt{2}}{2\pi \cdot 60} = 0.450\ V \cdot s$$

Current: $$1pu = \frac{Vnom \cdot \sqrt{2}}{L \cdot 2\pi \cdot fnom} = \frac{120 \cdot \sqrt{2}}{4\pi \cdot 60} = 0.225\ A$$

Figure 2-15: Flux-Current Characteristic of the Nonlinear Inductance

The current i flowing in the inductor is a nonlinear function of flux linkage ψ that, in turn, is a function of v appearing across its terminals. These relations are given by the following equations:

$$v = L \cdot \frac{di}{dt} = \frac{d\psi}{dt} \qquad \text{or} \qquad \psi = \int v \cdot dt$$

$$i = \frac{\psi}{L(\psi)}$$

The model of the nonlinear inductance can therefore be implemented as a controlled current source, where current i is a nonlinear function of voltage v, as shown.

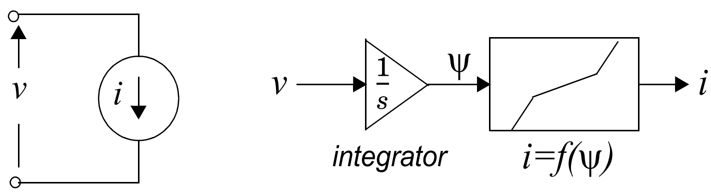


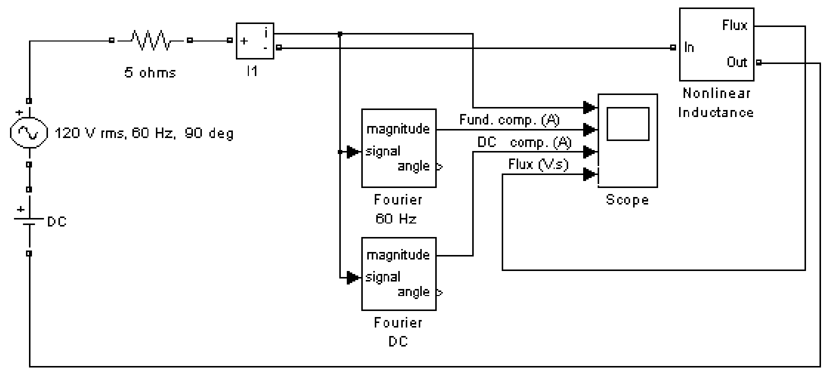
Figure 2-16: Model of a Nonlinear Inductance

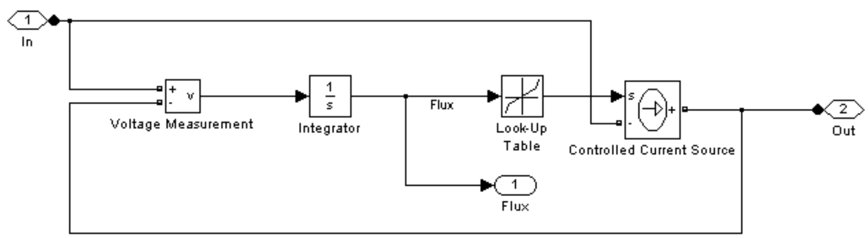
Figure 2-17 shows a circuit using a 2 H nonlinear inductance. The nonlinear inductance is connected in series with two voltage sources (an AC Voltage Source block of 120 volts RMS, 60 Hz, and a DC Voltage Source block) and a 5 ohm resistor.

All the elements used to build the nonlinear model have been grouped in a subsystem named Nonlinear Inductance. The inductor terminals are labeled `In` and `Out`. Notice that a second output returning the flux has been added to the subsystem. You can use this Simulink output to observe the flux by connecting it to a Simulink Scope block.

The nonlinear model uses two **powerlib** blocks and two Simulink blocks. The two **powerlib** blocks are a Voltage Measurement block to read the voltage at the inductance terminals and a Controlled Current Source block. The direction of the arrow of the current source is oriented from input to output according to the model shown in Figure 2-16.

The two Simulink blocks are an Integrator block computing the flux from the voltage input and a Look-Up Table block implementing the saturation characteristic $i = f(\psi)$ described by Figure 2-15.





Nonlinear Inductance subsystem

Figure 2-17: Implementation of a Nonlinear Inductance

Two Fourier blocks from the Measurements library of **powerlib_extras** are used to analyze the fundamental component and the DC component of the current.

Using blocks of the **powerlib** and Simulink libraries, build the circuit of Figure 2-17. To implement the $i = f(\psi)$ relation, specify the following vectors in the Look-Up Table block.

Vector of input values (flux)	`[-1.25 -1 1 1.25 ] *(120*sqrt(2)/(2π*60))`
Vector of output values (current)	`[-2 -1 1 2]*(120*sqrt(2)/(4π*60))`

Save your circuit as `circuit7`.

Set the following parameters for the two sources.

AC source

Peak amplitude	`120*sqrt(2)`
Phase	`90 degrees`
Frequency	`60 Hz`

DC source

Amplitude	`0 V`

Adjust the simulation time to `1.5 s` and select the `ode33tb` integration algorithm with default parameters. Start the simulation.

As expected, the current and the flux are sinusoidal. Their peak values correspond to the nominal values.

$$Peak \cdot Current = \frac{120 \cdot \sqrt{2}}{2 \cdot 2\pi \cdot 60} = 0.225\ A$$

$$Peak \cdot Flux = \frac{120 \cdot \sqrt{2}}{2\pi \cdot 60} = 0.450\ V \cdot s$$

Current and flux waveforms are shown.

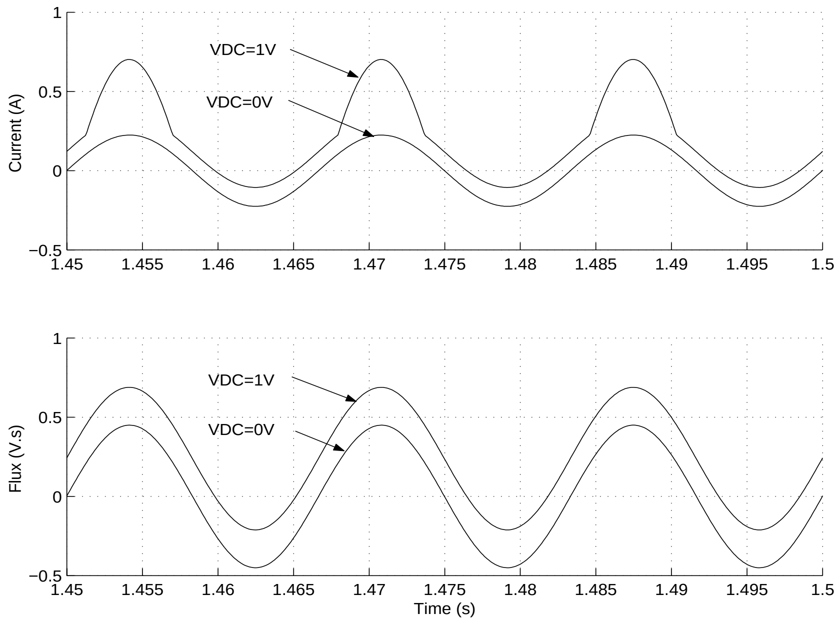


Figure 2-18: Current and Flux Waveforms Obtained with VDC = 0 V and VDC = 1 V

Now change the DC voltage to 1 V and restart the simulation. Observe that the current is distorted. The 1 V DC voltage is now integrated, causing a flux offset, which makes the flux enter into the nonlinear region of the flux-current characteristic ($\psi > 0.450$ V.s). As a result of this flux saturation, the current contains harmonics. Zoom in on the last three cycles of the simulation. The peak value of the current now reaches 0.70 A and the fundamental component has increased to 0.368 A. As expected, the DC component of the current is 1 V/ 0.5 Ω = 0.2. The current and flux waveforms obtained with and without saturation are superimposed in Figure 2-18.

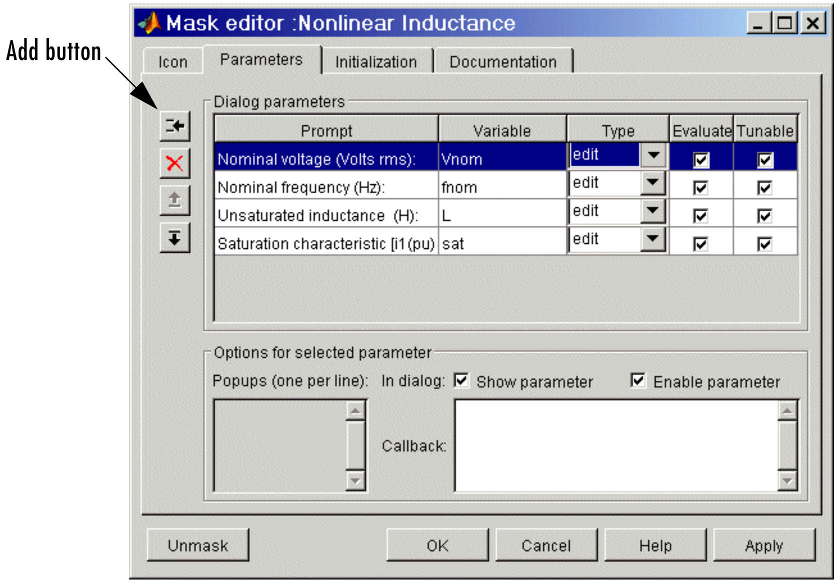
Customizing Your Nonlinear Model

Up to now, you have used a nonlinear model with fixed parameters. If you plan to use this block in other circuits with different parameters (for example, an inductance with a different voltage rating or a saturation characteristic defined with more than two segments), it proves more convenient to enter the block parameters in a dialog box, rather than modifying individual blocks of your subsystem.

In the following section, you learn how to use the Simulink masking facility to create a dialog box, an icon, and documentation for your model. For more details, refer to the Using Simulink guide.

Block Initialization

Select the Nonlinear Inductance subsystem and in the **Edit** menu, select **Edit mask**. The **Mask editor** window appears. Select the **Parameters** tab.



In the **Mask type** field, enter `Nonlinear Inductance` and click **Apply**.

The parameters that you have to specify are the nominal voltage, the inductance in the linear region, and the flux-current characteristic (flux and current vectors in p.u.).

Click **Add**. In the **Prompt** field, enter

```
Nominal voltage (Volts rms):
```

In the **Variable** field, enter the variable name associated with that field: `Vnom`.

Repeat the preceding steps to define the dialog boxes and associated variables listed below.

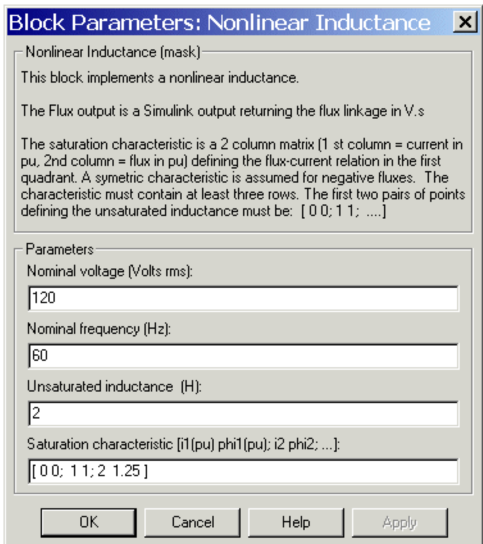
```
Nominal frequency (Hz):
fnom

Unsaturated inductance (H):
L

Saturation characteristic [i1(p.u.) phi1(p.u.); i2 phi2; ...]:
sat
```

Select the **Initialization** tab. Enter the following MATLAB commands. This code prepares the two vectors `Current_vect` and `Flux_vect` to be used in the Look-Up Table block of the model.

```
% Define base current and Flux for p.u. system
I_base=Vnom*sqrt(2)/(L*2π*fnom);
Phi_base=Vnom*sqrt(2)/(2π*fnom);

% Check first two points of the saturation characteristic
if ~all(all(sat(1:2,:)==[0 0; 1 1])),
    h=errordlg('The first two points of the characteristic must
be [0 0; 1 1]','Error');
    uiwait(h);
end

% Complete negative part of saturation characteristic
[npoints,ncol]=size(sat);
sat1=[sat ; -sat(2:npoints,:)];
sat1=sort(sat1);

% Current vector (A)  and flux vector (V.s)
Current_vect=sat1(:,1)*I_base;
Flux_vect=sat1(:,2)*Phi_base;
```

As the saturation characteristic is specified only in the first quadrant, three lines of code are added to complete the negative part of the saturation characteristic. Notice also how the validity of the first segment of the saturation characteristic is verified. This segment must be defined by two points [0 0; 1 1] specifying a 1 p.u. inductance (nominal value) for the first segment.

Click the **OK** button to close the **Mask Editor** window. Double-click the icon of your masked block. Its dialog opens with all fields empty. Enter the values as shown here.



Before you can use the masked block, you must apply the two internal variables defined in the initialization section of the Look-Up Table block. Select your block and, in the **Edit** menu, select **Look under mask**.

The Nonlinear Inductance subsystem opens. Open the Look-Up Table block dialog box and enter the following variable names in the two fields:

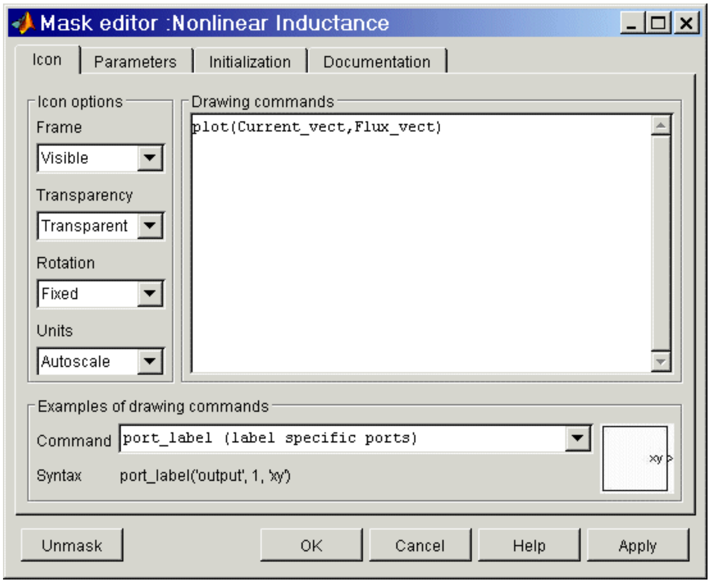
Vector of input values (flux)	`Flux_vect`
Vector of output values (current)	`Current_vect`

Close the Nonlinear Inductance subsystem and start the simulation. You should get the same waveforms as shown in Figure 2-18.

Block Icon

In this section you learn how to customize your block's icon and make it more attractive.

Select your block and, in the **Edit** menu, select **Edit mask**. The **Mask editor** window opens. Select the **Icon** tab.



In the **Drawing commands** section, you can specify any drawing that appears in your block icon by using the `plot` function. You can, for example, plot the flux-current characteristic of your inductance.

Remember that the currents and fluxes of the nonlinear characteristic are stored respectively in the `Current_vect` and `Flux_vect` internal variables of the masked block. Enter the following command in the **Drawing commands** section.

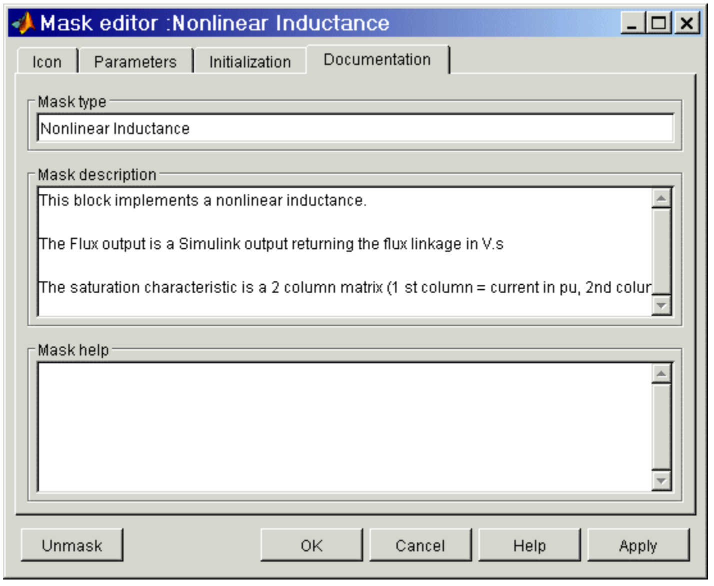
```
plot(Current_vect,Flux_vect)
```

Click **Apply** and notice that the saturation characteristic is displayed on the icon. Notice also that the input and output names have disappeared.

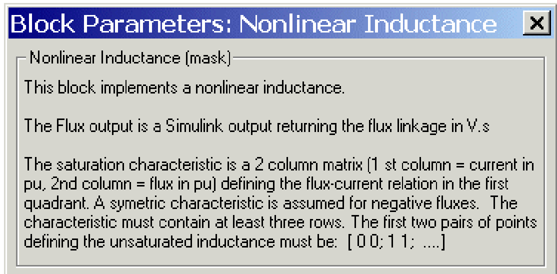
To make them visible, in the **Icon transparency** pop-up menu, select **Transparent**. Click **OK** to close the **Mask Editor** window.

Block Documentation

In this section, you add documentation to your block dialog box. Select your block and, in the **Edit** menu, select **Edit Mask**. The **Mask Editor** window opens. Select the **Documentation** tab.



Enter in the **Mask description** the text shown in the dialog box of the next figure. Then, click **OK** to close the **Mask Editor** window. The next time you double-click your block, this description appears on the dialog box of the block.



Modeling a Nonlinear Resistance

The technique for modeling a nonlinear resistance is similar to the one used for the nonlinear inductance.

A good example is a metal-oxide varistor (MOV) having the following V-I characteristic:

$$i = I_o \cdot \left(\frac{v}{V_o}\right)^{\alpha}$$

where

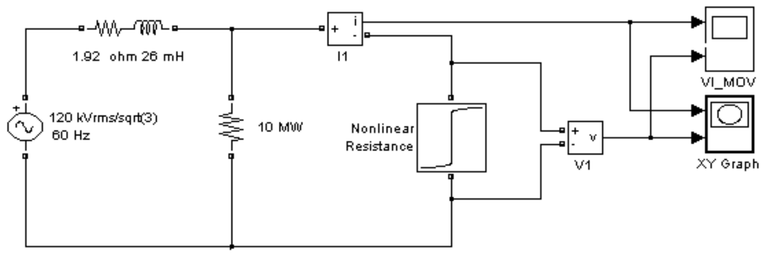
v, i = Instantaneous voltage and current

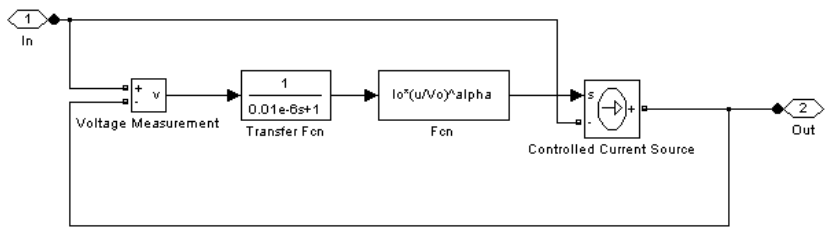
Vo = Protection voltage

Io = Reference current used to specify the protection voltage

α = Exponent defining the nonlinear characteristic (typically between 10 and 50)

The following figure shows an application of such a nonlinear resistance to simulate a MOV used to protect equipment on a 120 kV network. In order to keep the circuit simple, only one phase of the circuit is represented.





Nonlinear Resistance Subsystem

Figure 2-19: Nonlinear Resistance Applied on a 120 kV Network

Using blocks of the **powerlib** and Simulink libraries, build this circuit. Group all components used to model the nonlinear model in a subsystem named Nonlinear Resistance. Use an X-Y Graph block to plot the V-I characteristic of the Nonlinear Resistance subsystem.

Notice that the model does not use a Look-Up Table block as in the case of the nonlinear inductance model. As the analytical expression of current as a function of voltage is known, the nonlinear *I(V)* characteristic is implemented directly with a Math Function block from the Math Operations library of Simulink.

This purely resistive model contains no states. It produces an algebraic loop in the state-space representation of the circuit, as shown in the next figure. See Chapter 5, "SimPowerSystems Block Reference," for more details on how SimPowerSystems works.

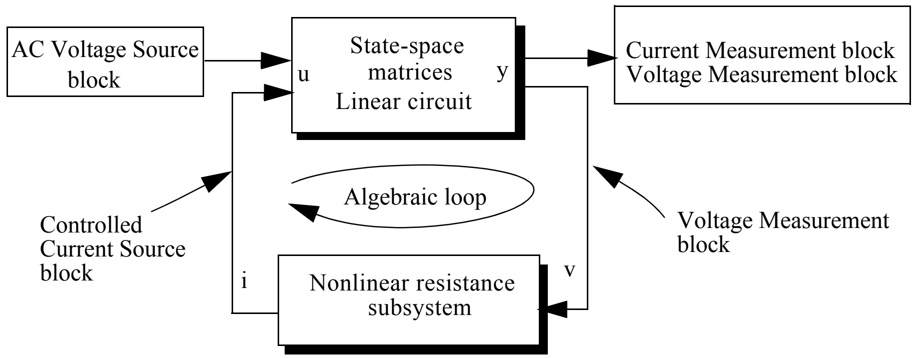


Figure 2-20: Algebraic Loop Introduced by the Nonlinear Resistance Model

Although Simulink can solve algebraic loops, they often lead to slow simulation times. You should break the loop with a block that does not change the nonlinear characteristic. Here a first-order transfer function $H(s) = 1/(1+Ts)$ is introduced into the system, using a fast time constant (T = 0.01 μs).

Use the technique explained for the nonlinear inductance block to mask and customize your nonlinear resistance block as shown.

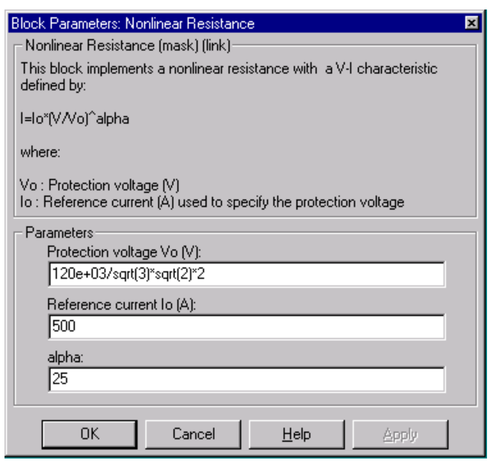


Figure 2-21: Dialog Box of the Nonlinear Resistance Block

Open the dialog box of your new masked block and enter the parameters shown in Figure 2-21. Notice that the protection voltage Vo is set at 2 p.u. of the nominal system voltage. Adjust the source voltage at 2.3 p.u. by entering the following peak amplitude:

```
120e3/sqrt(3)*sqrt(2)*2.3
```

Save your circuit as `circuit8`.

Using the `ode23tb` integration algorithm, simulate your `circuit8` system for 0.1 s. The results are shown below.

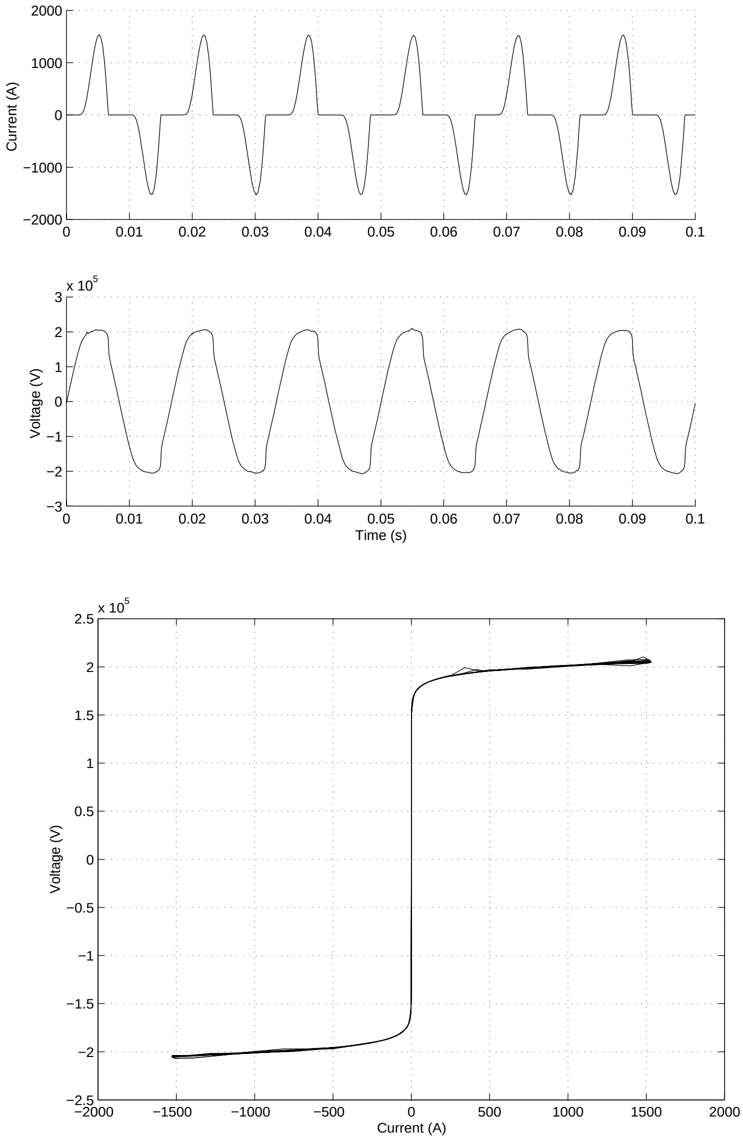


Figure 2-22: Current and Voltage Waveforms and V-I Characteristic Plotted by the X-Y Graph Block

Creating Your Own Library

Simulink lets you create your own libraries of SimPowerSystems blocks. To create a library, in the **File** menu choose **New Library**. A new Simulink window named **Library: untitled** opens. Now copy the Nonlinear Inductance block of your `circuit7` system and the Nonlinear Resistance block of your `circuit8` system into that library. Save this library as **my_powerlib**. Next time you develop a new model, you can add it to your personal library. You can also organize your library in different sublibraries according to their functions, as is done in the **powerlib** library.

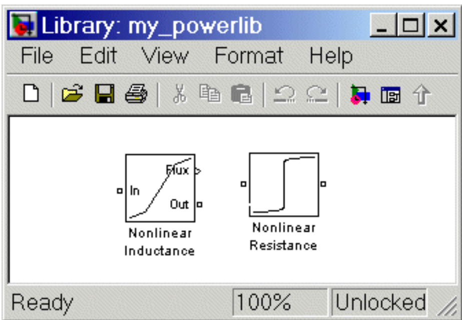


Figure 2-23: Nonlinear Inductance and Resistance Blocks in my_powerlib

One advantage of using a library is that all blocks that you copy from that library are referenced to the library. In other words, if you make a correction in your library block, the correction is automatically applied to all circuits using that block.

Connecting Your Model with Other Nonlinear Blocks

You now learn how to avoid error messages that can appear with nonlinear blocks when they are simulated by a current source. Obviously, a current source cannot be connected in series with an inductor, another current source, or an open circuit. Such circuit topologies are forbidden in SimPowerSystems.

Similarly, if your nonlinear model uses a Controlled Voltage Source block, this model could not be short-circuited or connected across a capacitor.

Suppose, for example, that you want to study the inrush current in a nonlinear inductance when it is energized on a voltage source. Using blocks from

powerlib library and **my_powerlibrary**, you can build the circuit shown here. Change the Breaker block parameters as follows.

Snubber resistance Rs	`inf` (no snubber)
Snubber capacitance Cs	`0`
External control	Not selected
Switching times	`[1/60]`

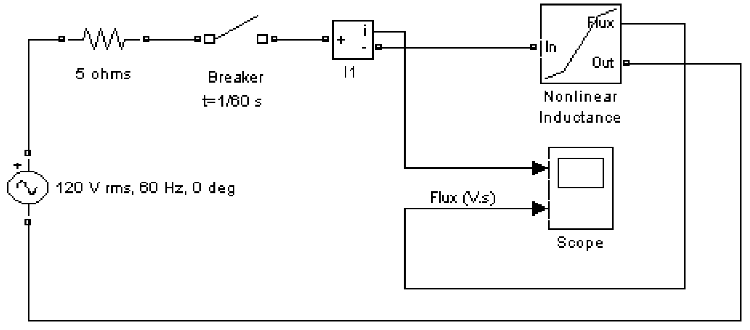
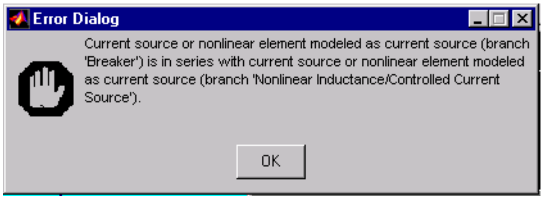


Figure 2-24: Circuit Topology Causing an Error

If you try to simulate this circuit, you get the following error message.



This topology is forbidden because two nonlinear elements simulated by current sources are connected in series: the Breaker block and the Nonlinear Inductance block. To be able to simulate this circuit you must provide a current path around one of the two nonlinear blocks. You could, for example, connect a large resistance, say 1 MΩ, across the Breaker block or the Inductance block.

In this case, it is more convenient to choose the Breaker block because a series RC snubber circuit is provided with the model. Open the Breaker block dialog box and specify the following snubber parameters:

Snubber resistance Rs (ohms)	`1e6`
Snubber capacitance Cs (F)	`inf`

Notice that in order to get a purely resistive snubber you have to use an infinite capacitance.

Note Using an inductive source impedance (R-L series) instead of a purely resistive impedance would have produced another error message, because the current source modeling the nonlinear inductance would have been in series with an inductance, even with a resistive snubber connected across the breaker. In such a case, you could add either a parallel resistance across the source impedance or a large shunt resistance connected between one breaker terminal and the source neutral terminal.

Make sure that the phase angle of the voltage source is zero. Use the `ode23tb` integration algorithm and simulate the circuit for 1 second. Voltage and current waveforms are shown here.

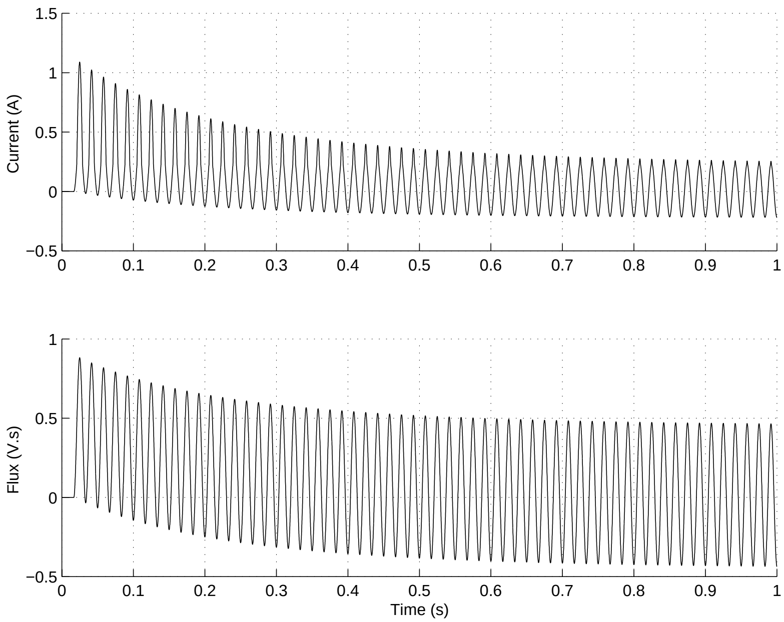


Figure 2-25: Current and Flux Waveforms When Energizing the Nonlinear Inductance with Maximum Flux Offset

Figure 2-25 shows that energizing the inductor at a zero crossing of voltage results in a maximum flux offset and saturation.

3

Case Studies

These case studies provide detailed, realistic examples of how to use SimPowerSystems.

Series-Compensated Transmission Network

The example described in this section illustrates phenomena related to subsynchronous resonance in a series-compensated AC transmission network.

Description of the Transmission Network

The single diagram shown here represents a three-phase, 60 Hz, 735 kV power system transmitting power from a power plant consisting of six 350 MVA generators to an equivalent network through a 600 km transmission line. The transmission line is split into two 300 km lines connected between buses B1, B2, and B3.

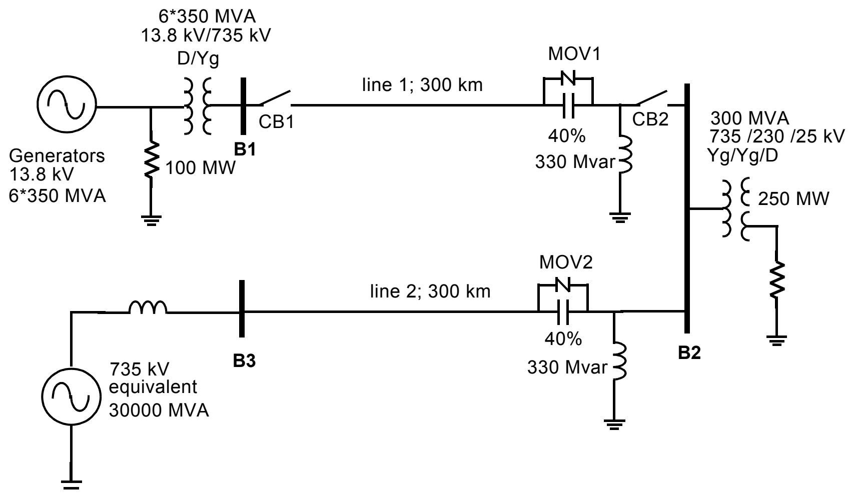


Figure 3-1: Series and Shunt Compensated Network

In order to increase the transmission capacity, each line is series compensated by capacitors representing 40% of the line reactance. Both lines are also shunt compensated by a 330 Mvar shunt reactance. The shunt and series compensation equipment is located at the B2 substation where a 300 MVA-735/230 kV transformer feeds a 230 kV-250 MW load.

Each series compensation bank is protected by metal-oxide varistors (MOV1 and MOV2). The two circuit breakers of line 1 are shown as CB1 and CB2.

This network is available in the power_3phseriescomp model. Load this model and save it in your working directory as case1.mdl in order to allow further modifications to the original system.

Compare the circuit modeled in SimPowerSystems (Figure 3-2) with the schematic diagram of Figure 3-1. The generators are simulated with a Simplified Synchronous Machine block. A Three-Phase Transformer (Two Windings) block and a Three-Phase Transformer (Three Windings) block are used to model the two transformers. Saturation is implemented on the transformer connected at bus B2.

The B1, B2, and B3 blocks are Three-Phase V-I Measurement blocks taken from the Measurements library. These blocks are reformatted and given a black background color to give them the appearance of bus bars. They output the three line-to-ground voltages and the three line currents. Open the dialog boxes of B1 and B2. Note how the blocks are programmed to output voltages in p.u. and currents in p.u./100 MVA. Notice also that the voltage and current signals are sent to internal Goto blocks by specifying signal labels. The signals are picked up by the From blocks in the Data Acquisition subsystem.

The fault is applied on line 1, on the line side of the capacitor bank. Open the dialog boxes of the Three-Phase Fault block and of the Three-Phase Breaker blocks CB1 and CB2. See how the initial breaker status and switching times are specified. A line-to-ground fault is applied on phase A at t = 1 cycle. The two circuit breakers that are initially closed are then open at t = 5 cycles, simulating a fault detection and opening time of 4 cycles. The fault is eliminated at t = 6 cycles, one cycle after the line opening.

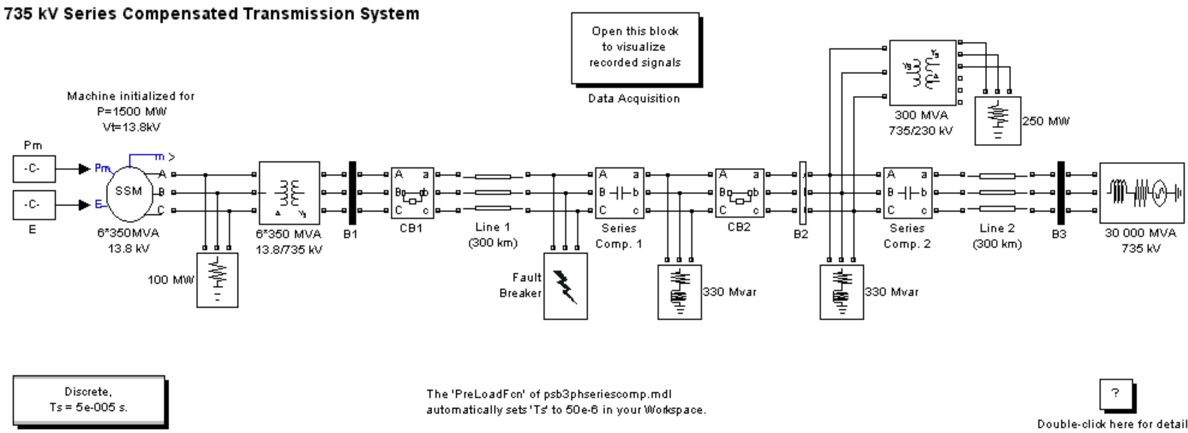


Figure 3-2: Series-Compensated Network (power_3phseriescomp)

Series Compensation1 Subsystem

Now, open the Series Compensation1 subsystem of the `power_3phseriescomp` model. The three-phase module consists of three identical subsystems, one for each phase. A note indicates how the capacitance value and the MOV protection level are calculated. Open the Series Compensation1/Phase A subsystem. You can see the details of the connections of the series capacitor and the Surge Arrester block (renamed MOV). The transmission line is 40% series compensated by a 62.8 μF capacitor. The capacitor is protected by the MOV block. If you open the dialog box of the MOV block, notice that it consists of 60 columns and that its protection level (specified at a reference current of 500 A/column or 30 kA total) is set at 298.7 kV. This voltage corresponds to 2.5 times the nominal capacitor voltage obtained at a nominal current of 2 kA RMS.

A gap is also connected in parallel with the MOV block. The gap is fired when the energy absorbed by the surge arrester exceeds a critical value of 30 MJ. In order to limit the rate of rise of capacitor current when the gap is fired, a damping RL circuit is connected in series. Open the Energy & Gap firing subsystem. It shows how you calculate the energy dissipated in the MOV by integrating the power (product of the MOV voltage and current).

When the energy exceeds the 30 MJ threshold, a closing order is sent to the Breaker block simulating the gap.

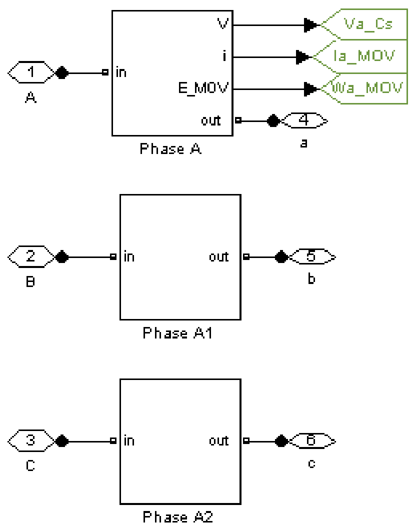
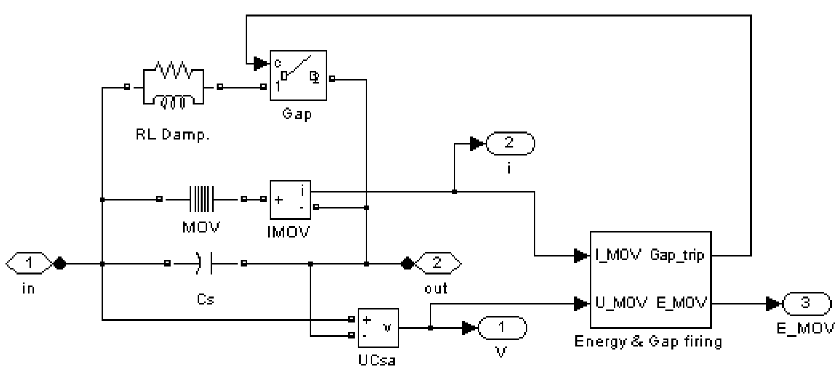
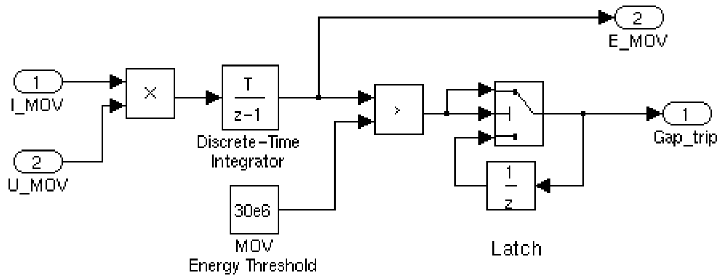


Figure 3-3: Series Compensation Module

Series Compensation1/PhaseA Subsystem:



Series Compensation1/PhaseA Subsystem/Energy & Gap firing:

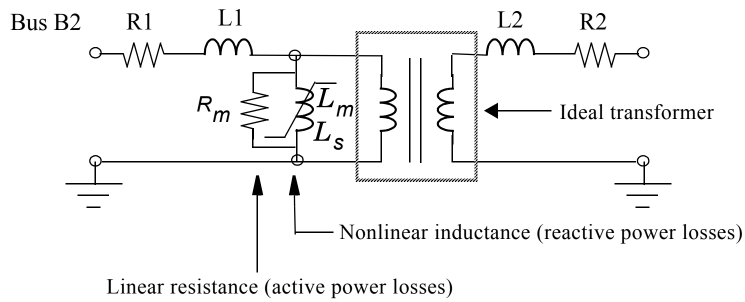


Three-Phase Saturable Transformer Model

Open the 300 MVA 735/230 kV Transformer dialog box and notice that the current-flux saturation characteristic is set at

`[0 0 ; 0.0012 1.2; 1 1.45]` in p.u.

These data are the current and flux values at points 1, 2, and 3 of the piecewise linear approximation to the flux linkage curve shown here:



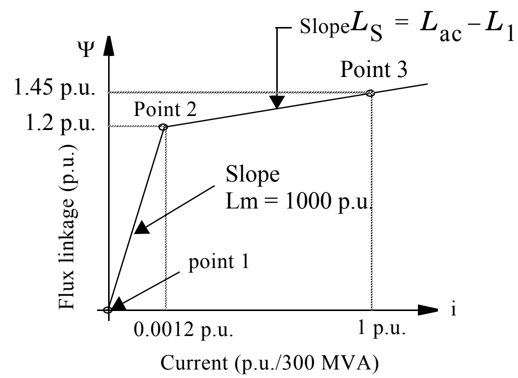


Figure 3-4: Saturable Transformer Model

The flux-current characteristic is approximated by the two segments shown in the graph here. The saturation knee point is 1.2 p.u. The first segment corresponds to the magnetizing characteristic in the linear region (for fluxes below 1.2 p.u.). At 1 p.u. voltage, the inductive magnetizing current is 0.0010/1.0 = 0.001 p.u., corresponding to 0.1% reactive power losses.

The iron core losses (active power losses) are specified by the magnetization resistance Rm = 1000 p.u., corresponding to 0.1% losses at nominal voltage.

The slope of the saturation characteristic in the saturated region is 0.25 p.u. Therefore, taking into account the primary leakage reactance (L1 = 0.15 p.u.), the air core reactance of the transformer seen from the primary winding is 0.4 p.u./300 MVA.

Setting the Initial Load Flow and Obtaining Steady State

Before performing transient tests you must initialize your model for the desired load flow. Use the load flow utility of the Powergui to obtain an active power flow of 1500 MW out of the machine with a terminal voltage of 1 p.u. (13.8 kV).

Open the Powergui and select **Load Flow and Machine Initialization**. A new window appears. In the upper right window you have the name of the only machine present in your system. Its **Bus Type** should be **PV Generator** and the desired **Terminal Voltage** should already be set to the nominal voltage of 13800 V. In the **Active Power** field, enter 1500e6 as the desired output power. Click the **Execute load flow** button. Once the load flow is solved, the phasors of AB and BC machine voltages as well as currents flowing in phases A and B are updated in the left window. The required mechanical power to drive the machine is displayed in watts and in p.u., and the required excitation voltage E is displayed in p.u.

Pmec	`1.5159e9 W [0.72184 p.u.]`
E/Vf	`1.0075 p.u.`

Notice that Constant blocks containing these two values are already connected to the `Pm` and `E` inputs of the machine block. If you open the machine dialog box, you see that the machine initial conditions (initial speed deviation dw = 0; internal angle theta, current magnitudes, and phase angles) are automatically transferred in the last line.

Once the load flow is performed, you can obtain the corresponding voltage and current measurements at the different buses. In the Powergui, select **Steady State Voltages and Currents**. You can observe, for example, the phasors for phase A voltages at buses `B1`, `B2`, and `B3` and the current entering line 1 at bus `B1`.

B1/Va	`6.088e5 V; 18.22 degrees`
B2/Va	`6.223e5 V; 9.26 degrees`
B3/Va	`6.064e5 V; 2.04 degrees`
B1/Ia	`1560 A; 30.50 degrees`

The active power flow for phase A entering line 1 is therefore

$$P_a = V_a \cdot I_a \cdot \cos(\varphi_a) = \frac{608.8\ kV}{\sqrt{2}} \cdot \frac{1.56\ kA}{\sqrt{2}} \cos(30.50 - 18.22) = 464\ MW$$

corresponding to a total of 464 * 3 = 1392 MW for the three phases.

Transient Performance for a Line Fault

In order to speed up the simulation, you need to discretize the network. The sample time is specified in the Powergui block as a variable `Ts`. This sample time `Ts` is also used in the Integrator block of the MOV energy calculator controlling the gap.

In the MATLAB window, define the variable

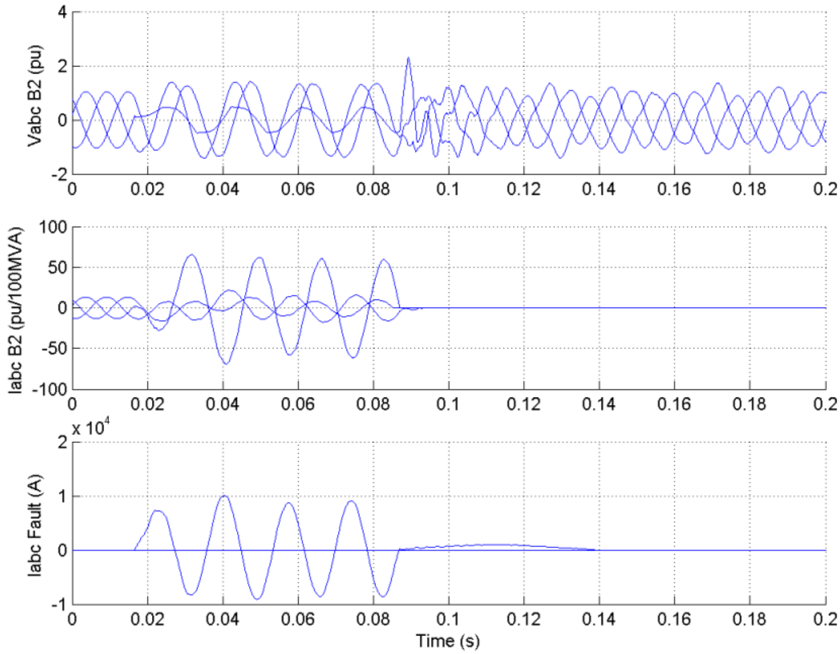
```
Ts = 50e-6
```

Ensure that the simulation parameters are set as follows:

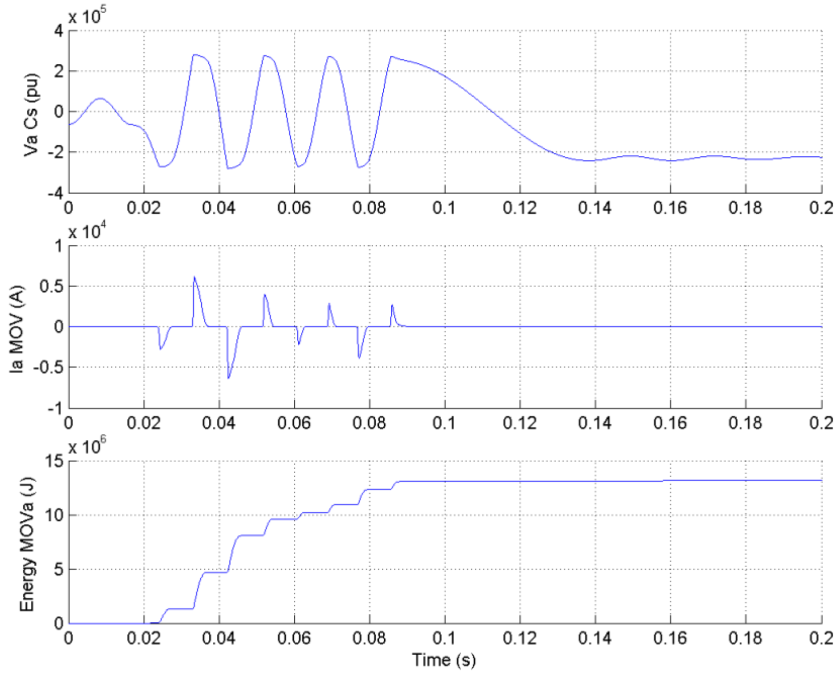
Stop time	`0.2`
Solver options Type	`Fixed-step; discrete (no continuous state)`
Fixed step size	`Ts`

Line-to-Ground Fault Applied on Line 1

Ensure that the fault breaker is programmed for a line-to-ground fault on phase A. Start the simulation and observe the waveforms on the three scopes. These waveforms are shown here:



(a)



(b)

Figure 3-5: Simulation Results for a Four-Cycle Line-to-Ground Fault at the End of Line 1

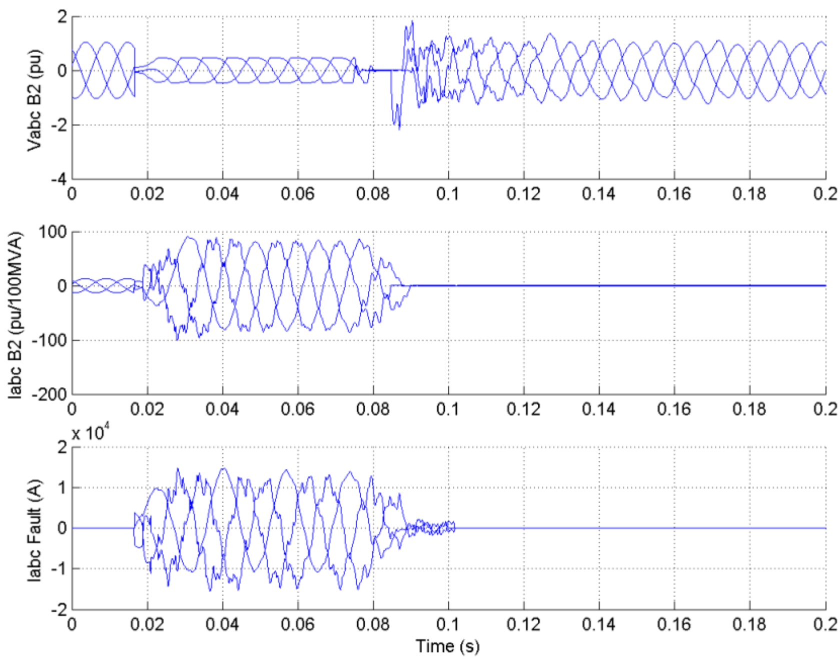
The simulation starts in steady state. At the t = 1 cycle, a line-to-ground fault is applied and the fault current reaches 10 kA (a: trace 3). During the fault, the MOV conducts at every half cycle (b: trace 5) and the energy dissipated in the MOV (b: trace 6) builds up to 13 MJ. At t = 5 cycles the line protection relays open breakers CB1 and CB2 (see the three line currents on trace 2) and the energy stays constant at 13 MJ. As the maximum energy does not exceed the 30 MJ threshold level, the gap is not fired. At the breaker opening, the fault current drops to a small value and the line and series capacitance starts to discharge through the fault and the shunt reactance. The fault current extinguishes at the first zero crossing after the opening order given to the fault breaker (t = 6 cycles). Then the series capacitor stops discharging and its voltage oscillates around 220 kV (b: trace 4).

Three-Phase-to-Ground Fault Applied on Line 1

Open the Three-Phase Fault block dialog box. Select the Phase B Fault and Phase C Fault, so that you now have a three-phase-to-ground fault.

Restart the simulation. The resulting waveforms are shown.

(a)



(b)

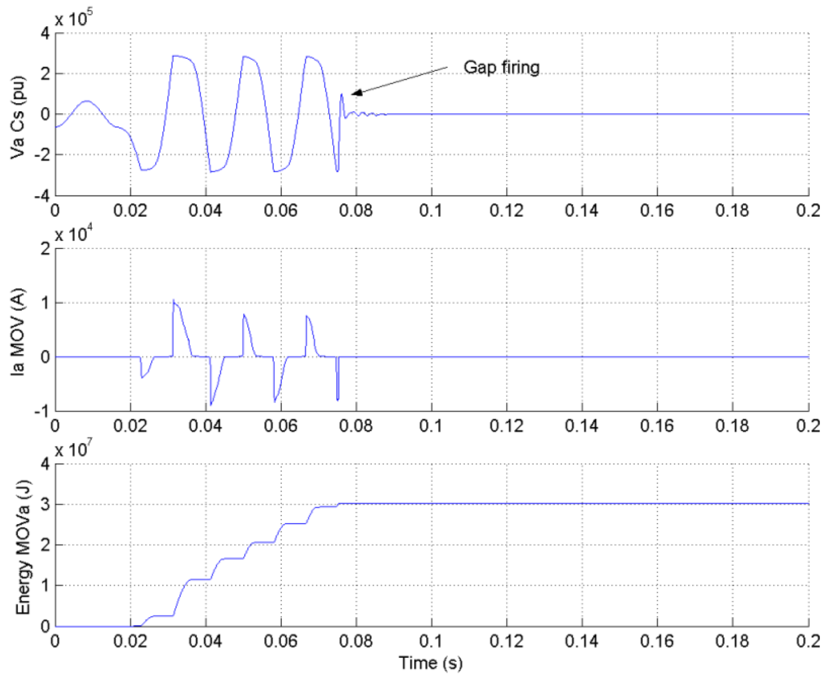


Figure 3-6: Simulation Results for a Four-Cycle Three-Phase-to-Ground Fault at the End of Line 1

Note that during the fault the energy dissipated in the MOV (trace 6) builds up faster than in the case of a line-to-ground fault. The energy reaches the 30 MJ threshold level after three cycles, one cycle before the opening of the line breakers. As a result, the gap is fired and the capacitor voltage (trace 4) quickly discharges to zero through the damping circuit.

Frequency Analysis

One particular characteristic of series-compensated systems is the existence of subsynchronous modes (poles and zeros of the system impedance below the fundamental frequency). Dangerous resonances can occur if the mechanical torsion modes of turbine/generator shafts are in the vicinity of the zeros of the system impedance. Also, high subsynchronous voltages due to impedance poles at subsynchronous frequencies drive transformers into saturation. The transformer saturation due to subsynchronous voltages is illustrated at the end of this case study. The torque amplification on a thermal machine is illustrated in another demonstration (see the `power_thermal` model).

Now measure the positive-sequence impedance versus frequency seen from bus B2.

The section "Analyzing a Simple Circuit" on page 1-9 in the "Modeling Simple Systems" chapter explains how the Impedance Measurement block allows you to compute the impedance of a linear system from its state-space model. However, your `case1` model contains several nonlinear blocks (machine and saturation of transformers). If you connect the Impedance Measurement block to your system, all nonlinear blocks are ignored. This is correct for the transformer, but you get the impedance of the system with the machine disconnected. Before measuring the impedance, you must therefore replace the machine block with an equivalent linear block having the same impedance.

Delete the Simplified Synchronous Machine block from your `case1` model and replace it with the Three-Phase Source block from the Electrical Sources library. Open the block dialog box and set the parameters as follows in order to get the same impedance value (L = 0.22 p.u./ (6 * 350 MVA) Quality factor = 15).

Phase-to-phase rms voltage	`13.8e3`
Phase angle of phase A	`0`
Frequency (Hz)	`60`
Internal connection Yg	`Specify impedance using short-circuit level`
3-phase short-circuit level	`6*350e6`
Base voltage	`13.8e3`
X/R ratio	`15`

Save your modified model as `case1Zf.mdl`.

Open the Measurements library of **powerlib** and copy the Impedance Measurement block into your model. This block is used to perform the impedance measurement. Connect the two inputs of this block between phase A and phase B of the B2 bus. Measuring the impedance between two phases gives two times the positive-sequence impedance. Therefore you must apply a factor of 1/2 to the impedance in order to obtain the correct impedance value. Open the dialog box and set the multiplication factor to 0.5.

In the Powergui, select **Impedance vs Frequency Measurement**. A new window opens, showing your Impedance Measurement block name. Fill in the frequency range by entering `0:500`. Select the linear scales to display Z magnitude vs. frequency plot. Click the **Save data to workspace** button and enter `Zcase1` as the variable name to contain the impedance vs. frequency. Click the **Display** button.

When the calculation is finished, the magnitude and phase as a function of frequency are displayed in the two graphs on the window. If you look in your workspace, you should have a variable named `Zcase1`. It is a two-column matrix containing frequency in column 1 and complex impedance in column 2.

The impedance as a function of frequency (magnitude and phase) is shown here:

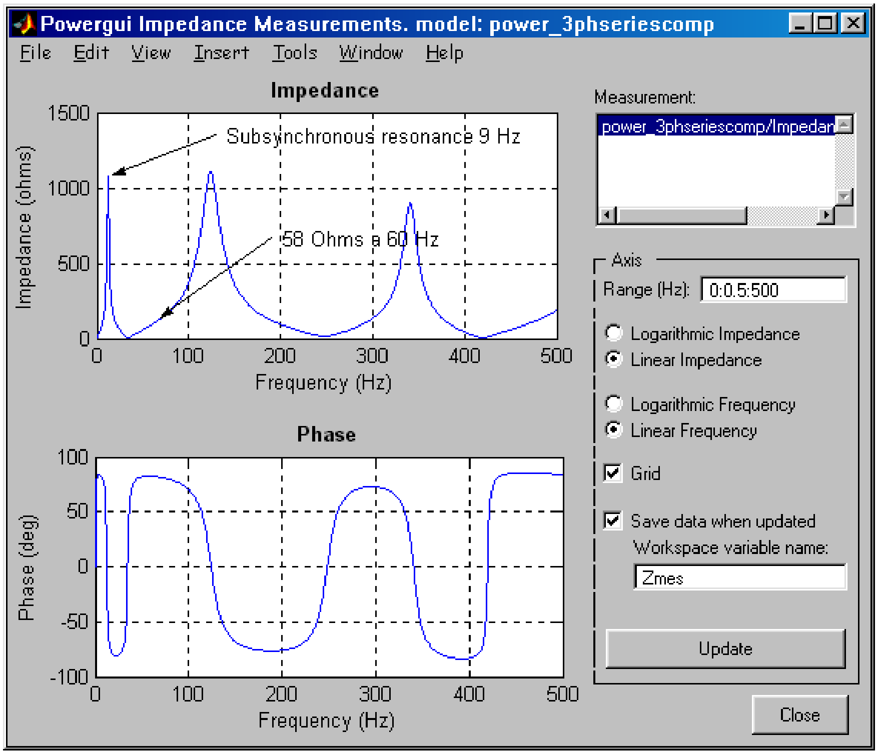


Figure 3-7: Impedance vs. Frequency Seen from Bus B2

You can observe three main modes: 9 Hz, 175 Hz, and 370 Hz. The 9 Hz mode is mainly due to a parallel resonance of the series capacitor with the shunt inductors. The 175 Hz and 370 Hz modes are due to the 600 km distributed parameter line. These three modes are likely to be excited at fault clearing.

If you zoom in on the impedance in the 60 Hz region, you can find the system's short-circuit level at bus B2. You should find a value of 58 Ω at 60 Hz, corresponding to a three-phase short-circuit power of $(735 \text{ kV})^2 / 58 = 9314$ MVA.

Transient Performance for a Fault at Bus B2

The configuration of the substation circuit breakers normally allows clearing a fault at the bus without losing the lines or the transformers. You now modify your `case1` model in order to perform a three-cycle, three-phase-to-ground fault at bus B2:

1 Disconnect the Three-Phase Fault block and reconnect it so that the fault is now applied on bus B2.

2 Open the Three-Phase Fault dialog box and make the following modifications:

Phase A, **Phase B**, **Phase C**, **Ground Faults**	All selected
Transition times	`[2/60 5/60]`
Transition status [1, 0, 1...]	`(0/1)`

You have now programmed a three-phase-to-ground fault applied at the *t* = 2 cycles.

3 Open the dialog boxes of circuit breakers CB1 and CB2 and make the following modifications:

Switching of Phase A	Not selected
Switching of Phase B	Not selected
Switching of Phase C	Not selected

The circuit breakers are not switched anymore. They stay at their initial state (closed).

4 In the Data Acquisition subsystem, insert a Selector block (from the Simulink Signals & Systems library) in the Vabc_B2 output of bus B2 connected to the scope. Set the **Elements** parameter to 1. This allows you to see the phase A voltage clearly on the scope.

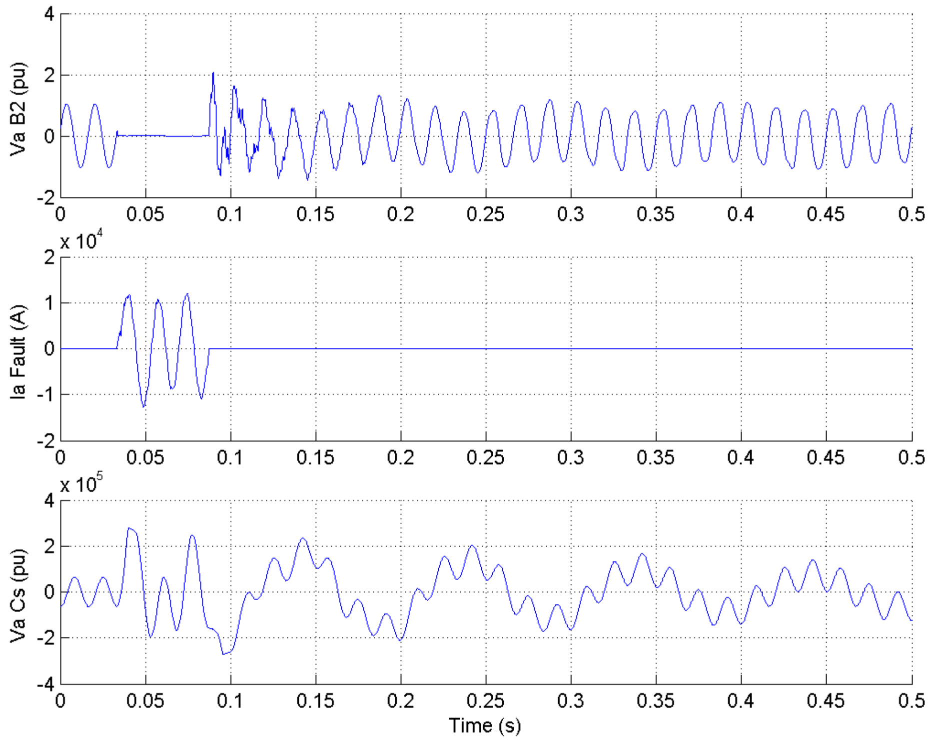
5 You now add blocks to read the flux and the magnetization current of the saturable transformer connected at bus B2.

   Copy the Multimeter block from the Measurements library into your `case1` model. Open the **Transformer** dialog box. In the **Measurements** pop-up menu, select **Flux and magnetization Current**. Open the Multimeter block. Verify that you have six signals available. Select flux and magnetization current on phase A, and click **OK**.

6 You now have two signals available at the output of the Multimeter block. Use a Demux block to send these two signals on a two-trace scope.

7 In the **Simulation —> Simulation parameters** dialog, change the stop time to 0.5. This longer simulation time allows you to observe the expected low-frequency modes (9 Hz). Start the simulation.

The resulting waveforms are plotted here:


Va B2 (pu)
Ia Fault (A)
x 10^4
Va Cs (pu)
x 10^5
Time (s)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

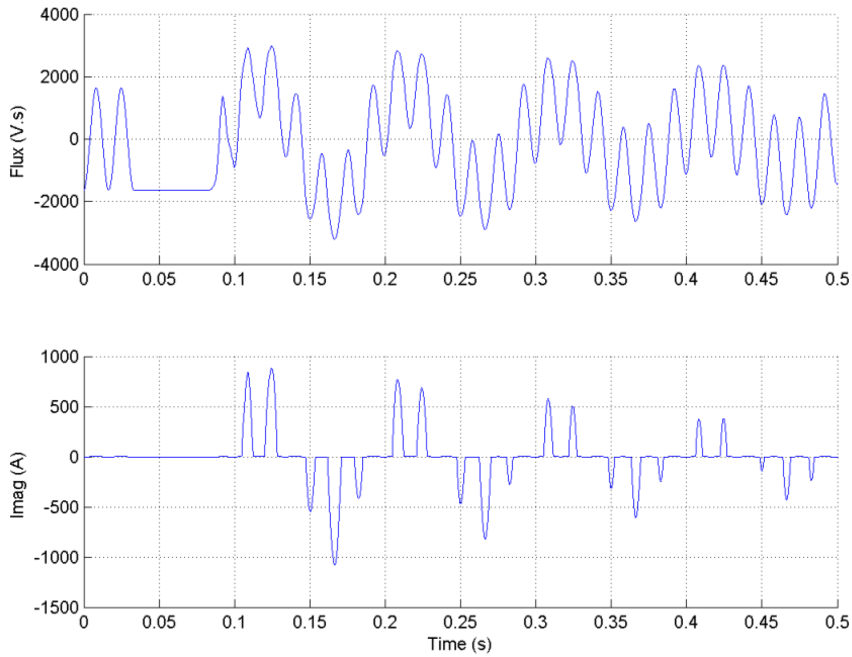


Figure 3-8: Simulation Results for a Three-Cycle Three-Phase-to-Ground Fault at Bus B2

The 9 Hz subsynchronous mode excited at fault clearing is clearly seen on the phase A voltage at bus B2 (trace 1) and capacitor voltage (trace 3). The 9 Hz voltage component appearing at bus B2 drives the transformer into saturation, as shown on the transformer magnetizing current (trace 5). The flux in phase A of the transformer is plotted on trace 4. At fault application the voltage at transformer terminals drops to zero and the flux stays constant during the fault.

At fault clearing, when the voltage recovers, the transformer is driven into saturation as a result of the flux offset created by the 60 Hz and 9 Hz voltage components. The pulses of the transformer magnetizing current appear when the flux exceeds its saturation level. This current contains a 60 Hz reactive component modulated at 9 Hz.

Chopper-Fed DC Motor Drive

The example described in this section illustrates application of SimPowerSystems to the operation of a DC motor drive in which the armature voltage is controlled by a GTO thyristor chopper.

The objective of this example is to demonstrate the use of electrical blocks, in combination with Simulink blocks, in the simulation of an electromechanical system with a control system. The electrical part of the DC motor drive, including the DC source, the DC motor, and the chopper, is built using blocks from the Elements, Machines, and Power Electronics libraries. The DC Machine block models both electrical and mechanical dynamics. The load torque-speed characteristic and the control system are built using Simulink blocks.

Description of the Drive System

A simplified diagram of the drive system is shown in the next figure. The DC motor is fed by the DC source through a chopper that consists of the GTO thyristor, Th1, and the free-wheeling diode D1. The DC motor drives a mechanical load that is characterized by the inertia J, friction coefficient B, and load torque T_L (which can be a function of the motor speed).

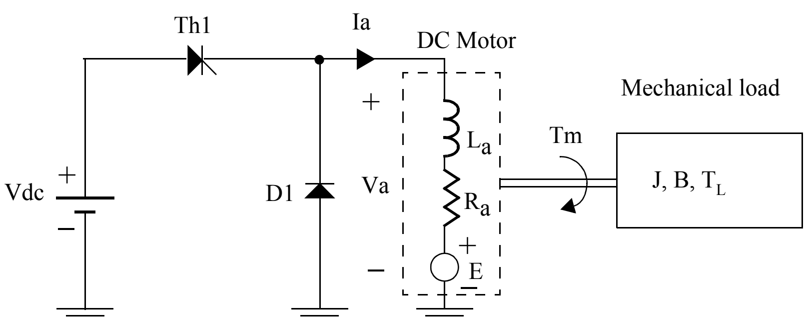


Figure 3-9: Chopper-Fed DC Motor Drive

In this diagram, the DC motor is represented by its equivalent circuit consisting of inductor L_a and resistor R_a in series with the counter electromotive force (emf) E.

The back reaction EMF is proportional to the motor speed

$$E = K_E \omega$$

where K_E is the motor voltage constant and ω is the motor speed.

In a separately excited DC machine, the motor voltage constant KE is proportional to the field current I_f

$$K_E = L_{af} I_f$$

where L_{af} is the field-armature mutual inductance.

The torque developed by the DC motor is proportional to the armature current I_a:

$$T_m = K_T I_a$$

where K_T is the motor torque constant.

The DC motor torque constant is equal to the voltage constant.

$$K_T = K_E$$

Thyristor Th1 is triggered by a pulse-width-modulated (PWM) signal to control the average motor voltage. (See "Variable-Frequency Induction Motor Drive" on page 3-33 for more details on pulse-width modulation.) Theoretical waveforms illustrating the chopper operation are shown here:

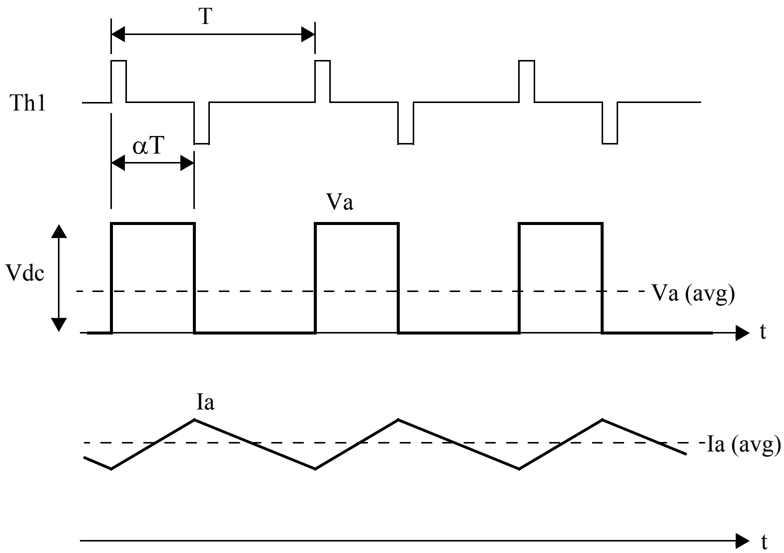


Figure 3-10: Waveforms Illustrating the Chopper Operation

The average armature voltage is a direct function of the chopper duty cycle α.

$$V_a(avg) = \alpha V_{dc}$$

Note that this relation is valid only when the armature current is continuous. In steady state, the armature average current is equal to

$$I_a(avg) = \frac{V_a(avg) - E}{R_a}$$

The peak-to-peak current ripple is

$$\Delta i = \frac{Vdc}{Ra}\frac{(1 - e^{-\alpha r} + e^{-r} - e^{-(1-\alpha)r})}{1 - e^{-r}}$$

where α is the duty cycle and r is the ratio between the chopper period and the DC motor electrical time constant.

$$r = \frac{T}{(L_a/R_a)}$$

In this case study, a variable-speed DC motor drive using a cascade control configuration is considered. Here is a block diagram of this drive:

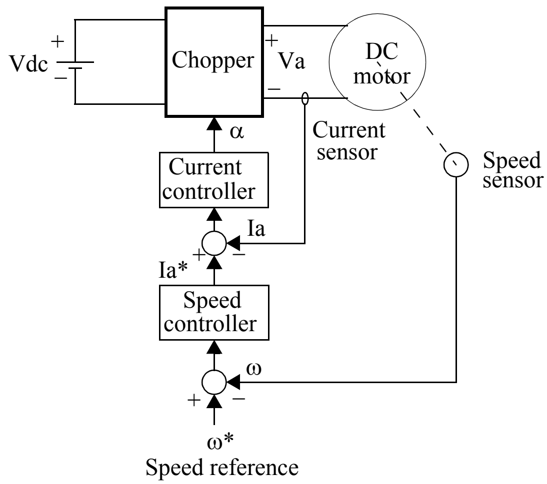


Figure 3-11: Variable-Speed DC Motor Drive

The motor torque is controlled by the armature current I_a, which is regulated by a current control loop. The motor speed is controlled by an external loop, which provides the current reference I_a* for the current control loop.

Modeling the DC Drive

Open the `power_dcdrive` model and save this model as `case2.mdl` in your working directory so that you can make further modifications without altering the original file.

The drive system diagram is built with blocks from the **powerlib** library combined with Simulink blocks. The system diagram is shown here:

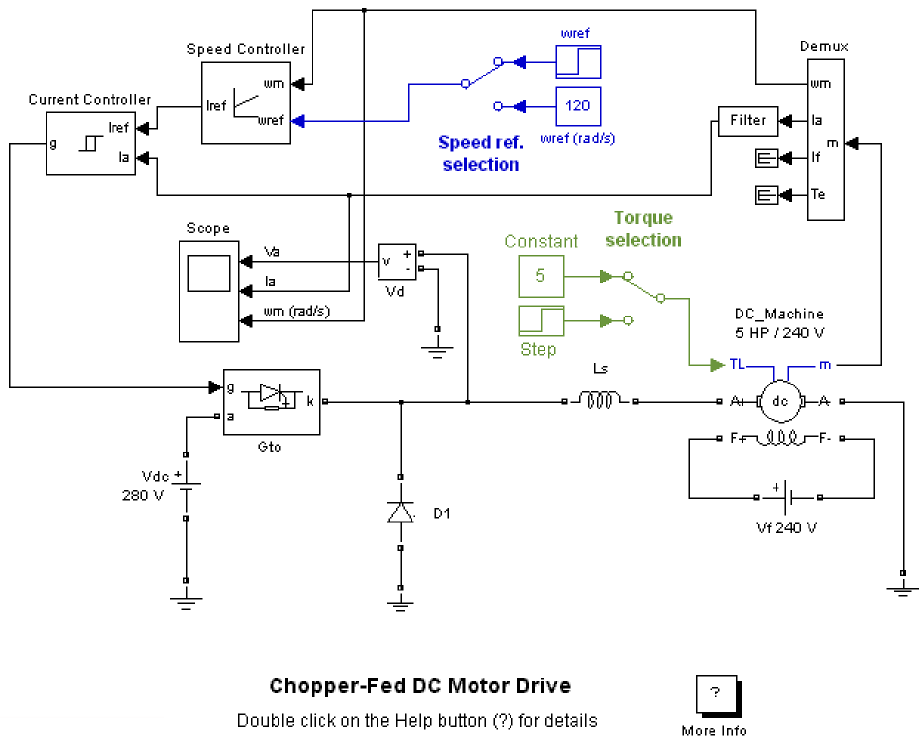
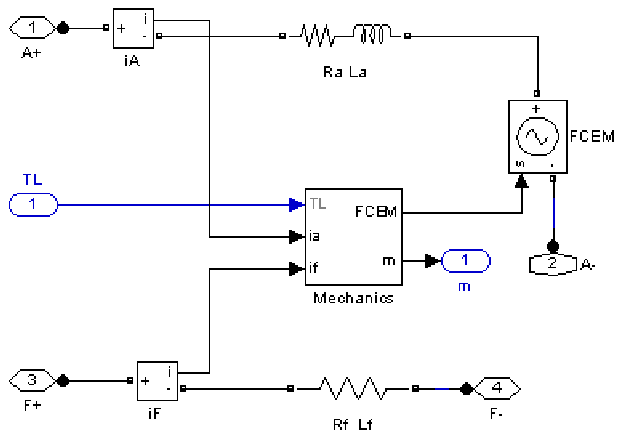


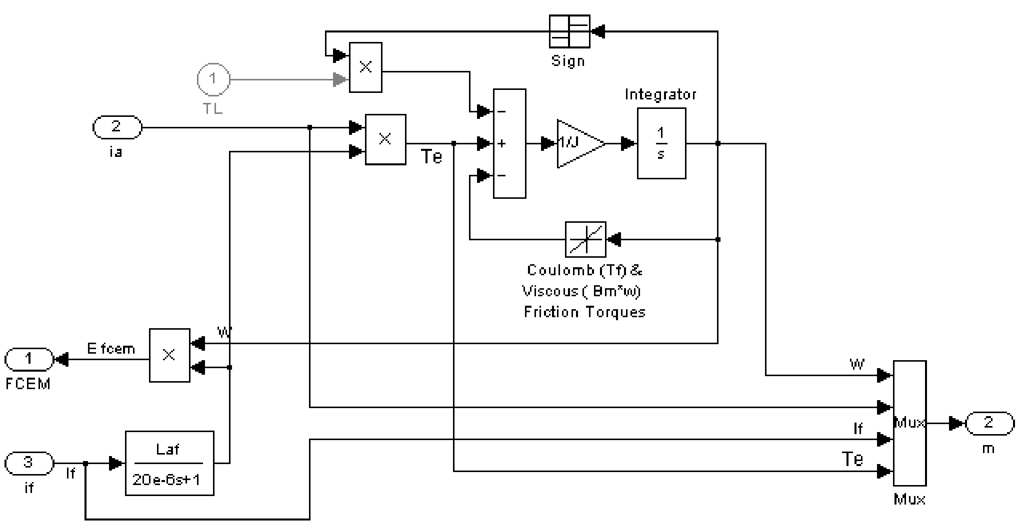
Figure 3-12: DC Motor Drive Using SimPowerSystems (power_dcdrive)

With a Manual Switch block, you can select both the reference speed and the load torque applied to the motor shaft in order to use either a constant value or a step function. Initially the reference speed is set to a constant value of 120 rad/s and the load torque is also maintained constant at 5 N.m.

The DC motor represented by the DC Machine block is modeled in two separate parts: electrical and mechanical. To view the Simulink model of the DC motor, click the DC Machine block and use the **Look under mask** item in the model **Edit** menu.



The mechanical subsystem is



The armature circuit is represented by an RL circuit in series with a controlled voltage source, the value of which is $K_E\omega$.

The field circuit is represented by an RL circuit.

The mechanical part is represented by Simulink blocks, which implement the following equation:

$$T_m = J\frac{d\omega}{dt} + B\omega + \mathrm{sgn}(\omega)T_L$$

Set the DC machine parameters to the desired values by using the dialog box of the DC Machine block.

You implement the load torque-speed characteristic using a Simulink Math Function block.

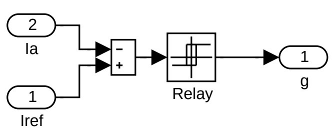
The motor used in this case study is a separately excited, 5 HP/240 V DC motor with the following parameters:

Ra	`0.5 Ω`
La	`10 mH`
KE	`1.23 V/(rad/s)`
KT	`1.23 N.m/A`

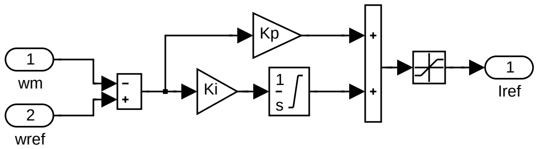
A 10 mH inductor (Ls) is connected in series with the DC motor to smooth out the armature current. The constant excitation is implemented by the connection of a DC Voltage Source block to the field winding.

The required trigger signal for the GTO thyristor is generated by a hysteresis current controller, which forces the motor current to follow the reference within +h/2 and –h/2 limits (h is the hysteresis band).

The current controller is a masked block that contains



The speed control loop uses a proportional-integral controller, which is implemented by Simulink blocks.



Simulation of the DC Drive

Run the simulation by selecting **Start** from the **Simulation** menu in Simulink. Set the simulation parameters in the **Simulation parameters** dialog as follows.

Simulation time	**Start Time**: `0`, **Stop time**: `1.2`
Solver Type	`Variable-step ode23tb (stiff/TR-BDF2)`
Max Step Size	`auto`
Initial Step Size	`auto`
Relative Tolerance	`1e-3`
Absolute Tolerance	`1e-3`

The motor voltage, current waveforms, and motor speed are displayed on three axes of the scope connected to the variables `Va`, `Ia`, and ω.

Starting the Drive

This test simulates the starting transient of the DC drive. The inertia of the mechanical load is small in order to bring out the details of the chopper commutation. The speed reference is 120 rad/s s, and you can observe the DC motor speed and current.

The transient responses for the starting of the DC motor drive are shown in Figure 3-13.

Note that you can save the final system state vector `xFinal` by selecting the **Workspace I/O —> Save to workspace —> Final state** check box in the **Simulation parameters** dialog. It can be used as the initial state in a subsequent simulation so that the simulation can start under steady-state conditions.

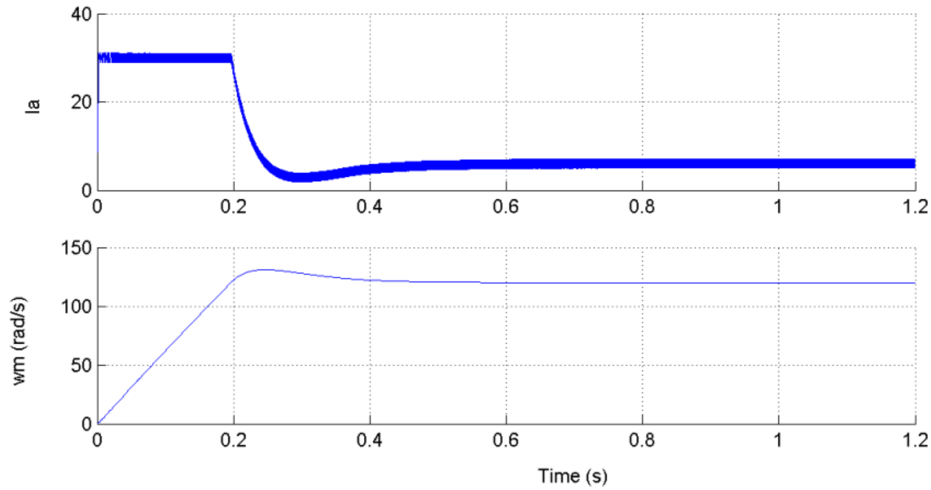


Figure 3-13: Starting the DC Motor Drive

Steady-State Voltage and Current Waveforms

When the steady state is attained, you can stop the simulation. The DC motor current and voltage waveforms obtained at the end of the starting test are shown here:

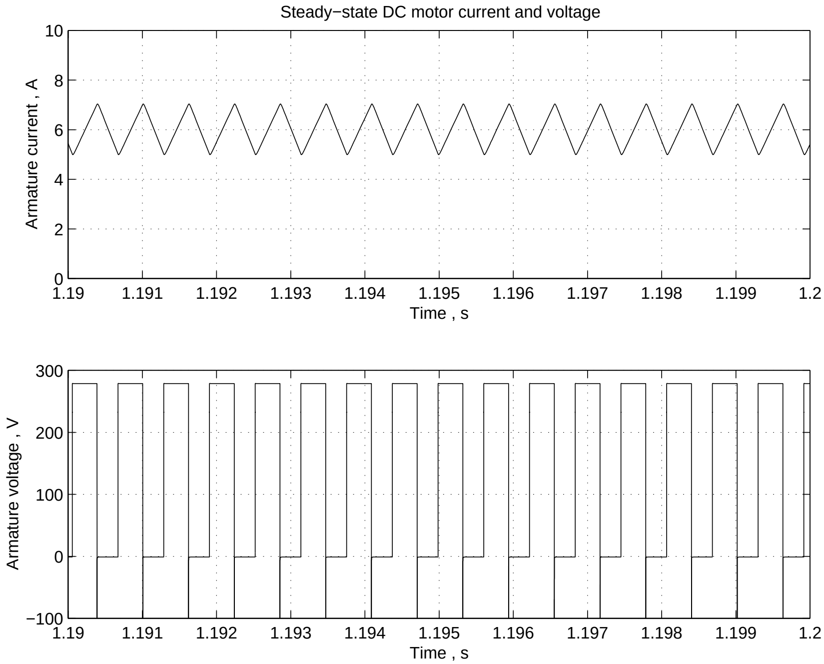


Figure 3-14: Steady-State Motor Current and Voltage Waveforms

Speed Regulation Dynamic Performance

You can study the drive dynamic performance (speed regulation performance versus reference and load torque changes) by applying two successive changes of operating conditions to the DC drive: a step change in speed reference and a step change in load torque.

Click the Torque Step block to step the load torque from 5 N.m to 25 N.m at t = 1.2 s. Then click also the Speed Step block to step the reference speed from120 rad/s to 160 rad/s at t = 0.4 s. In order to start the simulation in steady state, the final state vector obtained with the previous simulation can be used as the initial condition. Copy the `xFinal` variable (state vector saved at the end of the last simulation) into the `xInitial` variable. Select the **Workspace I/O —> Load from workspace O—> Initial state** check box in the **Simulation parameters** window and restart the simulation.

The response of the DC motor drive to successive changes in speed reference and load torque is plotted here:

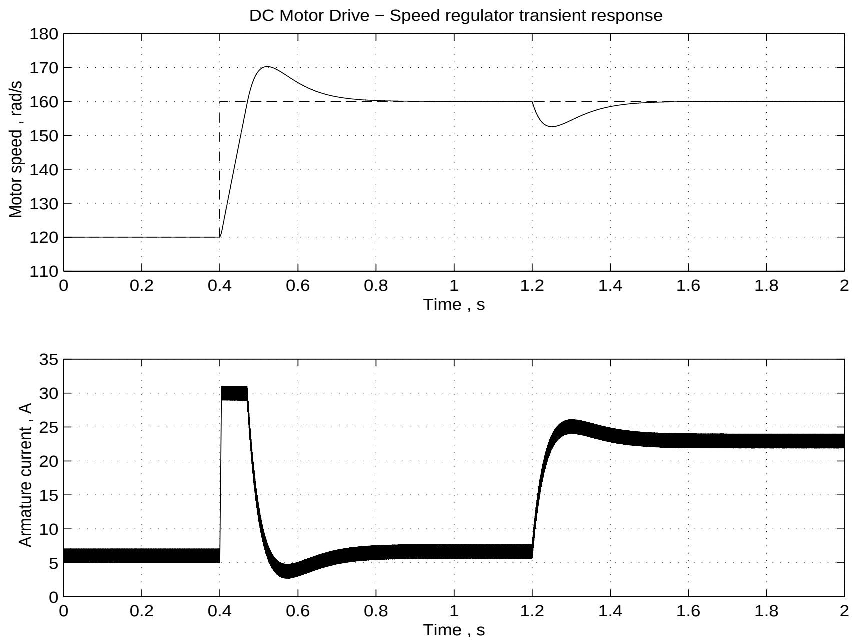


Figure 3-15: Dynamic Transient of the DC Motor Drive

Simulating with a Discretized System

Up to now you have performed all simulations using a continuous model of the DC drive and a variable-step solver. You can also perform the same simulations at fixed steps using a discretized system. Discretizing electronic converters is advantageous because simulation is much faster than with a continuous system.

As explained in the “Improving Simulation Performance” chapter, you cannot discretize power electronic converters using forced-commutated devices such as GTOs when these devices are simulated by individual blocks. However, discretization is possible when the Universal Bridge block or the Three-Level Bridge block is used to model the converter.

The discrete version of the DC drive has been saved in the `power_dcdrive_disc` demo. Open this model. Note that the GTO block and the Diode block have been replaced by the Universal Bridge block. Open the Universal Bridge block menu and note that the number of arms has been set to 1. The specified type of power electronic device is GTO/Diodes. It means that the converter consists of two GTO/Diode pairs (upper pair and lower pair) connected in series. For the type of buck converter used in our case study, only the upper GTO and the lower diode are used. Therefore no pulse is sent to the lower GTO.

The system is discretized by means of the Powergui block, where a sample time `Ts` has been specified. Define a variable `Ts = 10e-6` in your MATLAB Command Window. The control system and the DC machine is also discretized using the same 10 μs sample time. Open the **Simulation/Parameters** menu and notice that in the **Solver** section, `Discrete (no continuous states)` is selected.

Start the simulation and observe motor starting. Note that simulation now runs faster than with the continuous model. Results should compare well with those presented on Figure 3-13 for the continuous model.

References

[1] Leonhard,W., *Control of Electrical Drives*, Springer-Verlag, Berlin 1996.

Variable-Frequency Induction Motor Drive

This case study examines a variable-frequency AC motor drive model. A pulse-width-modulated (PWM) inverter is used as a variable-voltage, variable-frequency source to drive an induction motor in variable-speed operation.

You model the drive, including the motor, the power converter, and the speed control system, by using SimPowerSystems and Simulink blocks. The drive operation is studied for different operating conditions: starting, steady-state, and transients.

The objective of this example is to demonstrate the use of Machines library and Power Electronics library blocks in combination with Simulink blocks in the simulation of a complex electromechanical system operating at high frequency. The electrical part of the AC motor drive, including the PWM inverter, is built using the Universal Bridge block. The induction motor is represented by the Asynchronous Machine block, which models both electric and mechanical dynamics. The control system, including current and speed regulators, is built using Simulink blocks.

Description of the Induction Motor Drive

The induction motor requires a variable-frequency three-phase source for variable-speed operation. You can realize this source by using a power converter system consisting of a rectifier connected to an inverter through a DC link.

The next figure shows a block diagram of the power circuit of a typical variable-frequency induction motor drive.

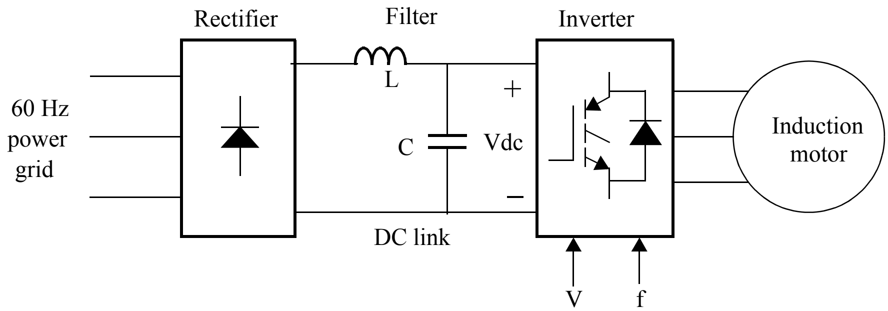


Figure 3-16: Variable-Frequency Induction Motor Drive

The power grid AC voltage is converted into a fixed DC voltage by the rectifier. The harmonics are filtered out by an LC filter to provide a smooth DC voltage, which is then applied to the inverter input.

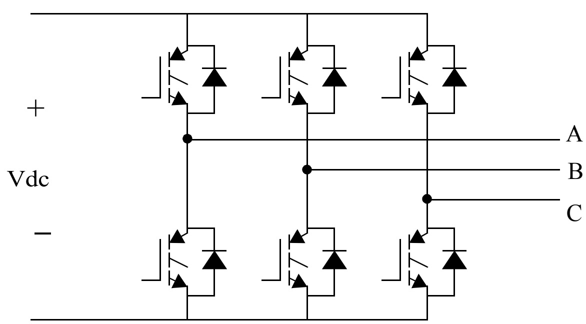


Figure 3-17: Three-Phase IGBT Inverter

The inverter consists essentially of six power switches that can be metal-oxide semiconductor field-effect transistors (MOSFET), gate turn-off thyristors (GTO), or insulated gate bipolar transistors (IGBT), depending on the drive power capacity and the inverter switching frequency (Hz). The preceding figure shows a simplified diagram of a three-phase IGBT inverter.

The inverter converts the DC link voltage into an adjustable three-phase AC voltage. Different control schemes can be used to control the inverter output voltage and frequency. One of the most utilized schemes is pulse width modulation (PWM) in which you obtain three-phase variable sinusoidal voltage waveforms by modulating the on and off times of the power switches.

In industrial drive applications, the PWM inverter operates as a three-phase variable-frequency, variable-voltage source with fundamental frequency varying from zero to three times the motor nominal frequency.

In some control schemes where a three-phase, variable-frequency current source is required, current control loops are added to force the motor currents to follow an input reference (usually sinusoidal).

You can control the inverter-fed induction motor drive with various schemes depending on the application, desired performance, and controller design complexity. The most utilized schemes are

- Stator V/Hz control
- Stator currents and open loop flux control
- Vector control (field-oriented control)
- Direct torque control (DTC)

A Field-Oriented Variable-Speed Induction Motor Drive

This case study illustrates a variable-speed induction motor drive using field-oriented control. In this control scheme, a dq coordinates reference frame locked to the rotor flux space vector is used to achieve decoupling between the motor flux and torque. They can thus be controlled separately by stator direct-axis current and quadrature-axis current respectively, as in a DC motor. This figure shows a block diagram of a field-oriented induction motor drive:

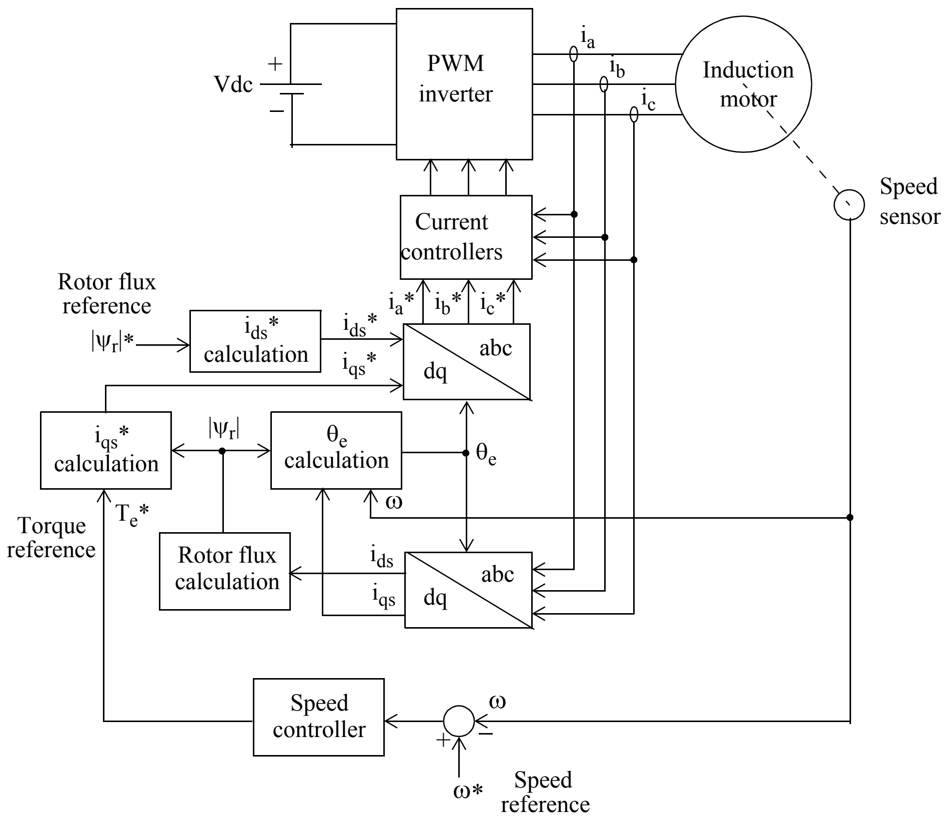


Figure 3-18: Field-Oriented Variable-Frequency Induction Motor Drive

The induction motor is fed by a current-controlled PWM inverter, which operates as a three-phase sinusoidal current source. The motor speed ω is compared to the reference ω* and the error is processed by the speed controller to produce a torque command Te*.

As shown below, the rotor flux and torque can be separately controlled by the stator direct-axis current i_{ds} and quadrature-axis current i_{qs}, respectively.

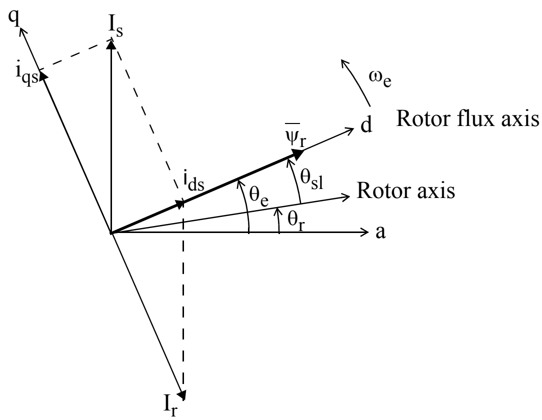


Figure 3-19: Field-Oriented Control Principle

The stator quadrature-axis current reference i_{qs}* is calculated from torque reference T_e* as

$$i_{qs}^* = \frac{2}{3} \cdot \frac{2}{p} \cdot \frac{L_r}{L_m} \cdot \frac{T_e^*}{|\psi_r|_{est}}$$

where L_r is the rotor inductance, L_m is the mutual inductance, and $|\psi_r|_{est}$ is the estimated rotor flux linkage given by

$$|\psi_r|_{est} = \frac{L_m i_{ds}}{1 + \tau_r s}$$

where $\tau_r = L_r / R_r$ is the rotor time constant.

The stator direct-axis current reference i_{ds}* is obtained from rotor flux reference input $|\psi_r|^*$:

$$i_{ds}^* = \frac{|\psi_r|^*}{L_m}$$

The rotor flux position θ_e required for coordinates transformation is generated from the rotor speed ω_m and slip frequency ω_{sl}:

$$\theta_e = \int(\omega_m + \omega_{sl})dt$$

The slip frequency is calculated from the stator reference current i_{qs}* and the motor parameters.

$$\omega_{sl} = \frac{L_m}{|\psi_r|_{est}} \cdot \frac{R_r}{L_r} \cdot i_{qs}^*$$

The i_{qs}* and i_{ds}* current references are converted into phase current references i_a*, i_b*, i_c* for the current regulators. The regulators process the measured and reference currents to produce the inverter gating signals.

The role of the speed controller is to keep the motor speed equal to the speed reference input in steady state and to provide a good dynamic during transients. It can be of proportional-integral type.

Modeling the Induction Motor Drive

Open the `power_acdrive` model and save it as `case3.mdl` in your working directory so that you can make further modifications without altering the original file.

The next figure shows the `power_acdrive` model in which blocks from SimPowerSystems and Simulink are used to model the induction motor drive.

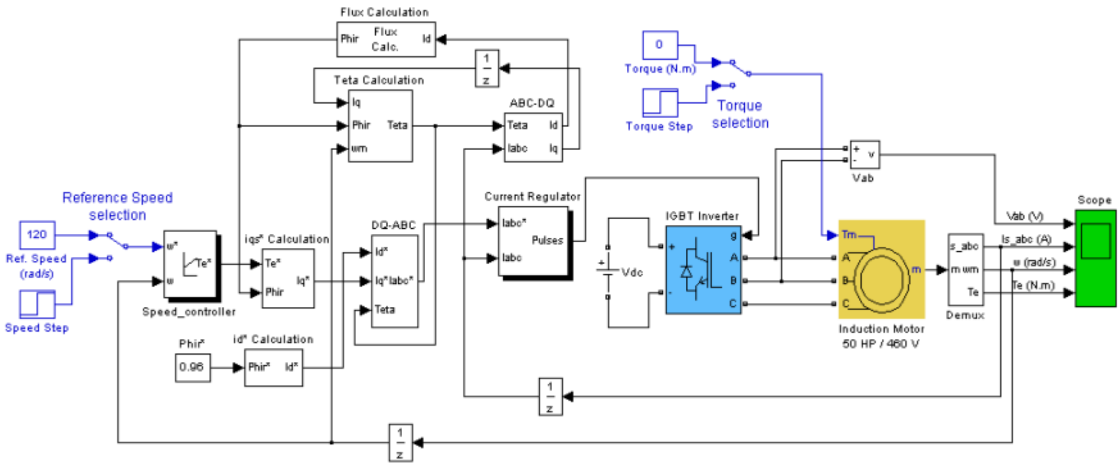


Figure 3-20: Variable-Speed Field-Oriented Induction Motor Drive (power_acdrive)

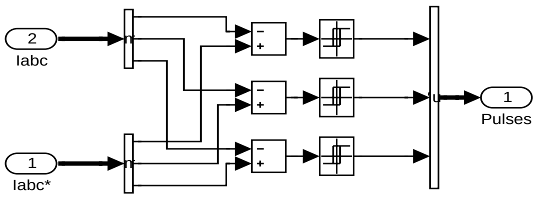
The induction motor is modeled by an Asynchronous Machine block. The motor used in this case study is a 50 HP, 460 V, four-pole, 60 Hz motor having the following parameters:

Rs	`0.087 Ω`
Lls	`0.8 mH`
Lm	`34.7 mH`
Rr	`0.228 Ω`
Llr	`0.8 mH`

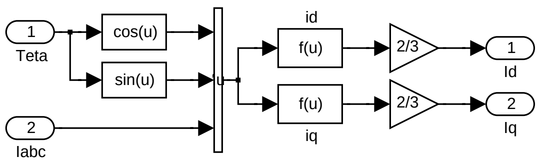
The reference speed and the load torque applied to the motor shaft can be both selected by a Manual Switch block in order to use either a constant value or a step function. Initially the reference speed is set a constant value of 120 rad/s and the load torque is also maintained constant at 0 N.m

The current-controlled PWM inverter circuit is shown in Figure 3-20. The IGBT inverter is modeled by a Universal Bridge block in which the **Power Electronic device** and **Port configuration** options are selected as `IGBT/Diode` and ABC as output terminals respectively. The DC link input voltage is represented by a 780 V DC voltage source.

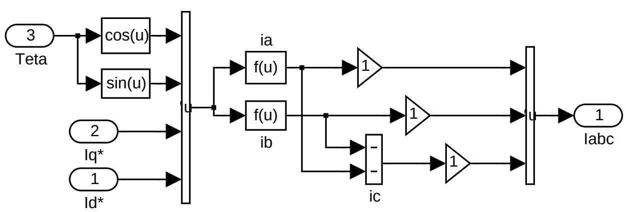
The current regulator, which consists of three hysteresis controllers, is built with Simulink blocks. The motor currents are provided by the measurement output of the Asynchronous Machine block.



The conversions between abc and dq reference frames are executed by the abc_to_dq0 Transformation and dq0_to_abc Transformation blocks of Figure 3-20.

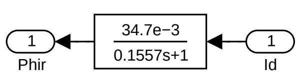


abc_dq

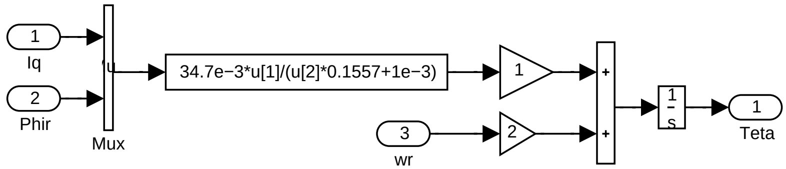


dq_abc

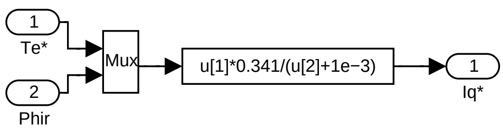
The rotor flux is calculated by the Flux_Calculation block of Figure 3-20.



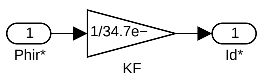
The rotor flux position (θe) is calculated by the Teta Calculation block of Figure 3-20.



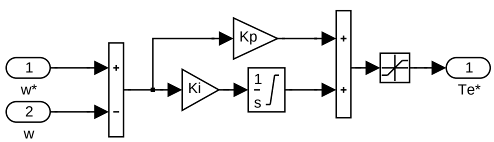
The stator quadrature-axis current reference (iqs*) is calculated by the iqs*_Calculation block of Figure 3-20.



The stator direct-axis current reference (ids*) is calculated by the id*_Calculation block of Figure 3-20.



The speed controller is of proportional-integral type and is implemented using Simulink blocks.



Simulating the Induction Motor Drive

In order to increase simulation speed, this model is discretized using a sample time of 2 μs. The variable `Ts = 2e-6` automatically loads into your workspace when you open this model. This sample time Ts is used both for the power circuit (`Ts` specified in the Powergui) and the control system.

Run the simulation by selecting **Start** from the **Simulation** menu in Simulink.

The motor voltage and current waveforms as well as the motor speed and torque are displayed on four axes of the scope connected to the variables Vab, Is, Te, and ω.

Starting the Drive

You can start the drive from a standstill by specifying `[1,0,0,0,0,0,0,0]` as the initial conditions for the Asynchronous Machine block (initial slip = 1 and no currents flowing in the three phases). The speed reference is 120 rad/s.

The motor speed, electromechanical torque, and currents observed during the starting of the induction motor drive are shown in Figure 3-21.

Note that you can save the final system state vector `xFinal` by selecting the **Workspace I/O —> Save to workspace —> Final state** check box in the **Simulation parameters** dialog. It can be used as the initial state in a subsequent simulation so that the simulation can start under steady-state conditions.

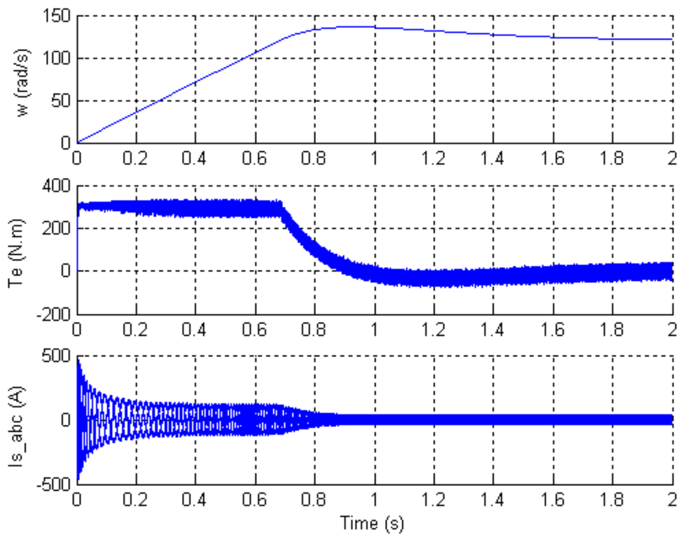


Figure 3-21: Starting the Induction Motor Drive

Steady-State Voltage and Current Waveforms

When the steady state is attained, you can stop the simulation and zoom on the scope signals.

This figure shows the motor voltage, current, and torque waveforms obtained when the motor is running at no load (torque = 0 N.m) at a speed of 120 rad/s.

The 20 A band imposed by the hysteresis current regulator is clearly seen on the three motor currents.

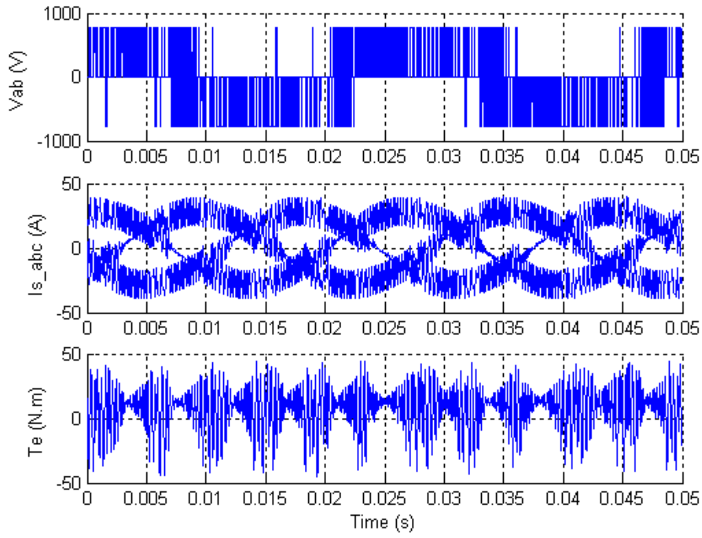


Figure 3-22: Steady-State Motor Current, Voltage, and Torque Waveforms

Speed Regulation Dynamic Performance

You can study the drive dynamic performance (speed regulation performance versus reference and load torque changes) by applying two changing operating conditions to the drive: a step change in speed reference and a step change in load torque.

Use the Reference Speed selection switch and the Torque selection switch to set speed reference steps from 120 rad/s to 160 rad/s at t = 0.2 s and the load torque steps from 0 N.m to 200 N.m at t = 1.8 s. The final state vector obtained with the previous simulation can be used as the initial condition so that the simulation starts from steady state. Load the `power_acdrive_init.mat` file, which creates the `xInitial` variable. Select the **Workspace I/O —> Load from workspace —> Initial state** check box in the **Simulation parameters** dialog and restart the simulation.

The response of the induction motor drive to successive changes in speed reference and load torque is shown here:

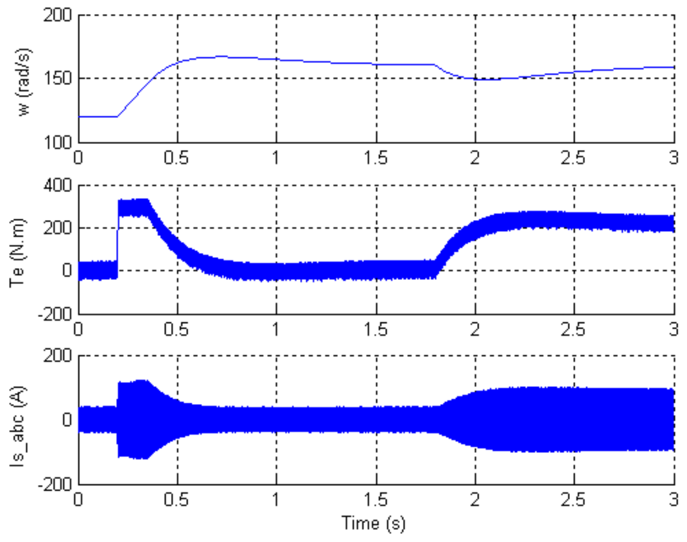


Figure 3-23: Dynamic Performance of the Induction Motor Drive

References

[1] Leonhard, W., *Control of Electrical Drives*, Springer-Verlag, Berlin, 1996.

[2] Murphy, J. M. D., and Turnbull, F. G., *Power Electronic Control of AC Motors*, Pergamon Press, Oxford, 1985.

[3] Bose, B. K., *Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, N.J., 1986.

HVDC System

The final example described in this section illustrates modeling of a high-voltage direct current (HVDC) transmission link [1]. Perturbations are applied in order to examine the system performance [2]. The objectives of this example are to demonstrate the use of the Universal Bridge block and the Three-Phase Transformer (Three Windings) block in combination with Simulink blocks in the simulation of a complete pole of a 12-pulse HVDC transmission system. The electrical part representing the AC network is built using three-phase blocks. The Discrete 12-Pulse HVDC control system is a generic control available in the Discrete Control Blocks library of **powerlib_extras**.

Description of the HVDC Transmission System

Open the `power_hvdc12pulse` model and save it as `case4.mdl` in order to allow further modifications to the original system. This system is shown in Figure 3-24.

A 1000 MW (500 kV, 2 kA) DC interconnection is used to transmit power from a 500 kV, 5000 MVA, 60 Hz network to a 345 kV, 10000 MVA, 50 Hz network. The AC networks are represented by damped L-R equivalents with an angle of 80 degrees at fundamental frequency (60 Hz or 50 Hz) and at the third harmonic.

The rectifier and the inverter are 12-pulse converters using two Universal Bridge blocks connected in series. Open the two converter subsystems to see how they are built. The converters are interconnected through a 300 km line and 0.5 H smoothing reactors. The converter transformers (Wye grounded /Wye/Delta) are modeled with Three-Phase Transformer (Three-Windings) blocks. The transformer tap changers are not simulated. The tap position is rather at a fixed position determined by a multiplication factor applied to the primary nominal voltage of the converter transformers (0.90 on the rectifier side; 0.96 on the inverter side).

From the AC point of view, an HVDC converter acts as a source of harmonic currents. From the DC point of view, it is a source of harmonic voltages.

The order n of these characteristic harmonics is related to the pulse number p of the converter configuration: $n = kp \pm 1$ for the AC current and $n = kp$ for the direct voltage, k being any integer. In the example, p = 12, so that injected harmonics on the AC side are 11, 13, 23, 25, and on the DC side are 12, 24.

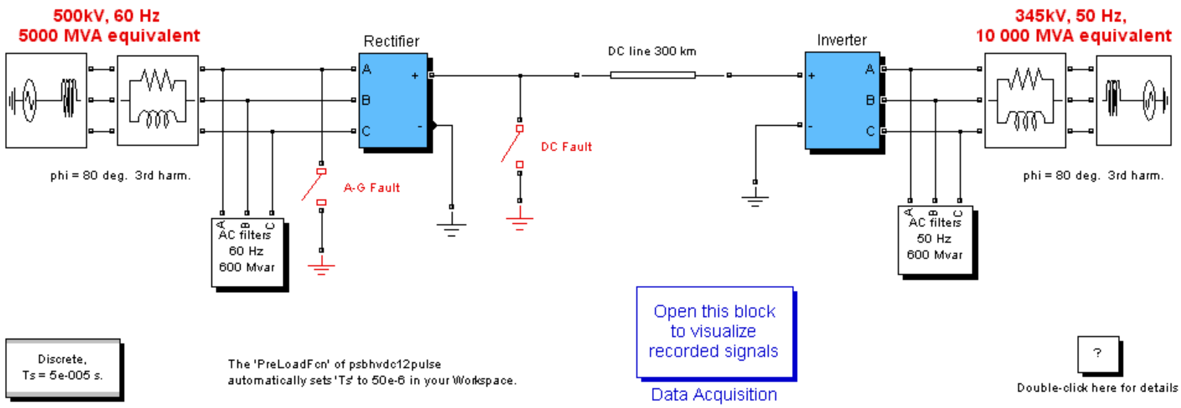


Figure 3-24: HVDC System

AC filters are used to prevent the odd harmonic currents from spreading out on the network. The filters are grouped in two subsystems. These filters also appear as large capacitors at fundamental frequency, thus providing reactive power compensation for the rectifier consumption due to the firing angle α. For α = 30 degrees, the converter reactive power demand is approximately 60% of the power transmitted at full load. Look under the AC filters subsystem mask to see the high Q (100) tuned filters at the 11th and 13th harmonics and the low Q (3), or damped filter, used to eliminate the higher order harmonics, e.g., 23rd and up. Extra reactive power is also provided by capacitor banks.

Two circuit breakers are used to apply faults on the rectifier AC and DC sides.

The rectifier and inverter control systems use the Discrete 12-Pulse HVDC Control block of the Discrete Control Blocks library of **powerlib_extras**.

The power system and the control system are both discretized with the same sample time Ts.

Define parameter `Ts = 50e-6` in your workspace before starting the simulation.

Frequency Response of the AC and DC Systems

You now measure the frequency response of the AC systems (rectifier and inverter sides) and of the DC line.

The section “Analyzing a Simple Circuit” on page 1-9 in the “Modeling Simple Systems” chapter explains how the Impedance Measurement block allows you to compute the impedance of a linear system from its state-space model. As the thyristor valves of the converters are nonlinear blocks, they are ignored in the impedance calculation and you get the impedances with the valves open.

Open the Measurements library, copy three Impedance Measurement blocks into your model, and rename them Zrec, Zinv, and ZDC. Connect the two inputs of Zrec and Zinv between phase A and phase B of the AC system on the rectifier and inverter sides. Measuring the impedance between two phases gives two times the positive-sequence impedance. Therefore you must apply a factor of 1/2 to the impedance in order to obtain the correct impedance value. Open the two Impedance Measurement blocks and set the **Multiplication factor** to `0.5`. Finally, connect input 1 of the ZDC block between the DC line terminal and the rectifier smoothing reactor, and connect input 2 to ground. Save your modified model as `case4Zf.mdl`.

In the Powergui, select **Impedance vs Frequency Measurement**. A new window opens, showing the three Impedance Measurement block names. Fill in the **Frequency range** by entering `10:2:1500`. Select the `lin` scale to display the Z magnitude and `lin` scale for the frequency axis. Click the **Save data to workspace** button and enter `Zcase4` as the variable name to contain the impedance vs. frequency. Click the **Display** button.

When the calculation is finished, the magnitude and phase as functions of frequency measured by the three Impedance Measurement blocks are displayed in the window. Your workspace should have a variable named `Zcase5`. It is a four-column matrix containing frequency in column 1 and the three complex impedances in columns 2, 3, and 4 with the same order as in the window displaying the block names.

The magnitudes of the three impedances as a function of frequency are shown here.

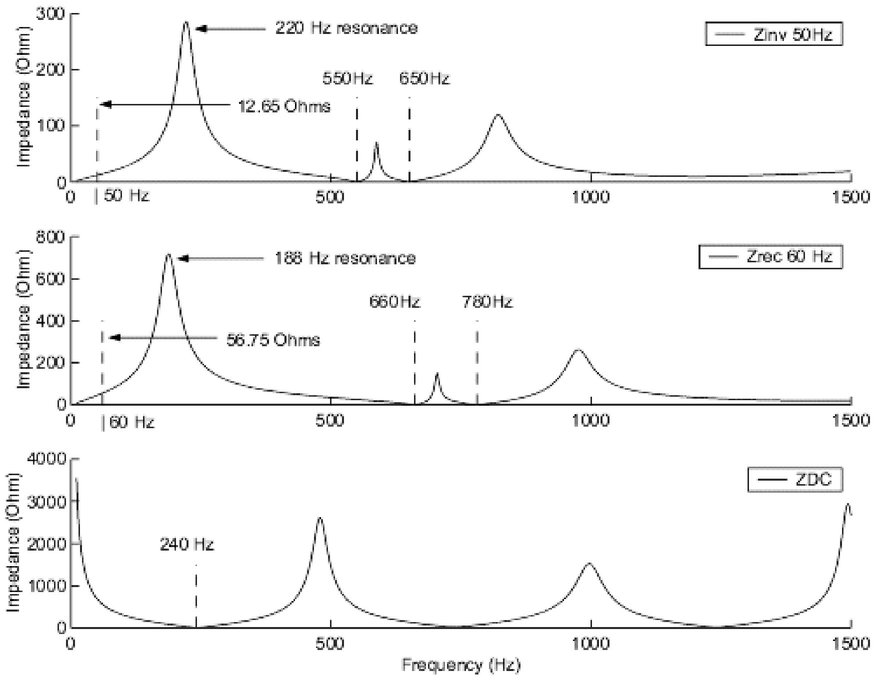


Figure 3-25: Positive-Sequence Impedances of the Two AC Networks and of the DC Line

Note the two minimum impedances on the Z magnitudes of the AC systems. These series resonances are created by the 11th and 13th harmonic filters. They occur at 660 Hz and 780 Hz on the 60 Hz system. Note also that the addition of 600 Mvar capacitive filters on the inductive systems creates resonances (around 188 Hz on the rectifier side and 220 Hz on the inverter side). Zoom in on the impedance magnitude in the 60 Hz region. You should find a magnitude of 56.75 Ω for the 60 Hz system, corresponding to an effective short-circuit level of $500^2/56.75 = 4405$ MVA on the rectifier side (5000 MVA - 600 Mvar of filters).

For the DC line, note the series resonance at 240 Hz, which corresponds to the main mode likely to be excited on the DC side, under large disturbances.

Description of the Control System

The control systems of the rectifier and of the inverter use the same 12-Pulse HVDC Control block from the Discrete Control Blocks library of **powerlib_extras**. The block can operate either in rectifier or inverter mode. Use **Look under mask** to see how this block is built.

Inputs and Outputs

Input 1 (`Vabc`) is a vectorized signal of the three line-to-ground voltages measured at the primary of the converter transformer. These three voltages are used to synchronize the pulse generation on the line voltages. Inputs 2 and 3 are the DC line voltage (`VdL`) and current (`Id`). Note that the measured DC currents (`IdR` and `IdI` in A) and DC voltages (`VdLR` and `VdLI` in V) are scaled to p.u. (1 p.u. current = 2 kA; 1 p.u. voltage = 500 kV) before they are used in the controllers.

Inputs 4 and 5 (`Id_ref` and `Vd_ref`) are the Vd and Id reference values in p.u. The `VdL` and `Id` inputs are filtered before being processed by the regulators. A first-order filter is used on the `Id` input and a second-order filter is used on the `VdL` input. The filter parameters are shown in the dialog box of Figure 3-27.

Input 6 (`Block`) accepts a logical signal (0 or 1) used to block the converter when `Block` = 1. Input 7 is also a logical signal that can be used for protection purposes. If this signal is high (1), the firing angle is forced at the value defined in the block dialog box.

The first two block outputs (`PulseY` and `PulseD`) contain the vectorized signals of the six pulses to be sent to each of the six-pulse converters connected to the wye and delta windings of the converter transformer. The third output (`alpha`) is the firing delay angle in degrees ordered by the regulator. The fourth output (`Id_ref_lim`) is the actual reference current value (value of `Id_ref` limited by the VDCOL function as explained below). The fifth output (`Mode`) is an indication of the actual state of the converter control mode or firing pulses. The state is given by a number (from 0 to 5) as follows:

0	Blocked pulses
1	Current control
2	Voltage control
3	Alpha minimum limitation

4 Alpha maximum limitation

5 Forced or constant alpha

Synchronization System

The Discrete 12-Pulse HVDC Control block uses the primary voltages (input 1) to synchronize and generate the pulses according to `Vd_ref` and `Id_ref` set points (inputs 4 and 5). The synchronizing voltages are measured at the primary side of the converter transformer because the waveforms are less distorted. The firing command pulse generator is synchronized to the fundamental frequency of the AC source. At the zero crossings of the commutating voltages (AB, BC, CA), a ramp is reset. A firing pulse is generated whenever the ramp value becomes equal to the desired delay angle provided by the regulator. In order to improve the commutating voltages used by the pulse generator, the primary voltages (Vabc) are filtered by a low Q second-order band-pass filter centered at the fundamental system frequency. The base system frequency and the filter bandwidth are defined in the block dialog box.

Steady-State V-I Characteristic

The Discrete 12-Pulse HVDC Control block implements this steady-state characteristic:

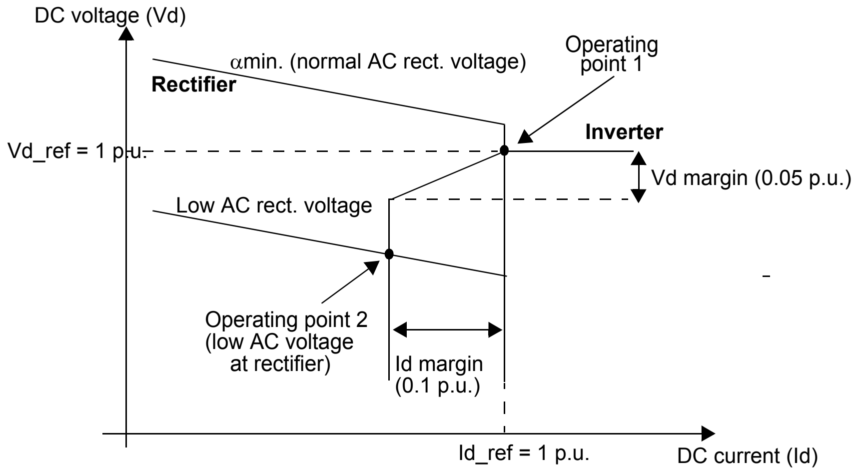


Figure 3-26: Rectifier and Inverter Steady-State Characteristics and VDCOL Function

In normal operation, the rectifier controls the current at the `Id_ref` reference value, whereas the inverter controls the voltage at the `Vd_ref` reference value. The `Id_margin` and `Vd_margin` parameters are defined in the inverter dialog box. They are set respectively at 0.1 p.u. and 0.05 p.u. The system normally operates at point 1 as shown in the figure. However, during a severe contingency producing a voltage drop on the AC network 1 feeding the rectifier, the operating point moves to point 2. The rectifier therefore is forced to a minimum α mode and the inverter is in current control mode.

Note In industrial controllers, the α angle at the inverter is normally limited in order to keep a minimum γ angle, where

- γ = extinction angle = $180° - \alpha - \mu$
- μ =commutation or overlap angle

The γ control required to avoid commutation failures is not implemented in this version of the HVDC control. Nevertheless, a block taken from the Discrete Control Blocks library of **powerlib_extras** is used to monitor γ. Such a block could be used to control minimum γ.

VDCOL Function

Another important control function is implemented to change the reference current according to the value of the DC voltage. This control, named Voltage Dependent Current Order Limiter (VDCOL), automatically reduces the reference current (`Id_ref`) set point when `VdL` decreases (as, for example, during a DC line fault or a severe AC fault). Reducing the `Id` reference currents also reduces the reactive power demand on the AC network, helping to recover from fault. The VDCOL parameters of the Discrete 12-Pulse HVDC Control block dialog box are explained by this diagram:

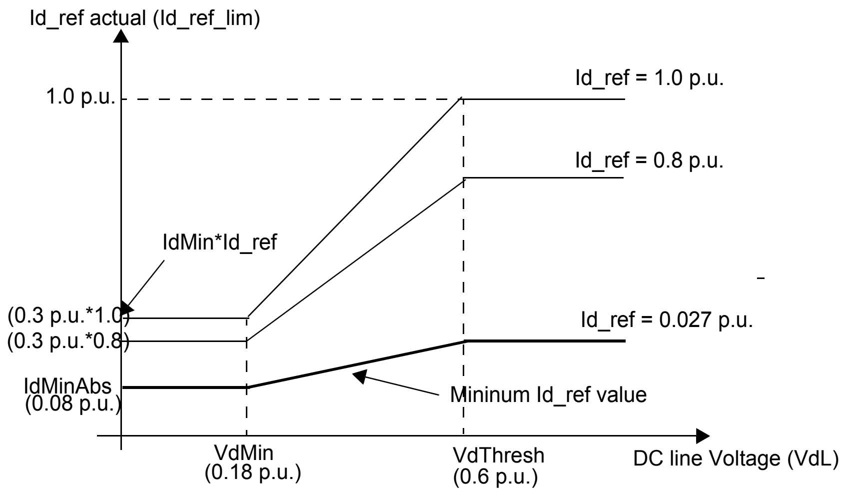


Figure 3-27: VDCOL Characteristic; Id_ref = f(VdL)

The `Id_ref` value starts to decrease when the Vd line voltage falls below a threshold value `VdThresh` (0.6 p.u.). The actual reference current used by the controllers is available at the fourth controller output, named `Id_ref_lim`. `IdMinAbs` is the absolute minimum `Id_ref` value, set at 0.08 p.u. When the DC line voltage falls below the `VdThresh` value, the VDCOL drops instantaneously to `Id_ref`. However, when the DC voltage recovers, VDCOL limits the `Id_ref` rise time with a time constant defined by parameter `Tup` (80 ms in the example).

Current and Voltage Regulators

The rectifier and the inverter controls both have a voltage and a current regulator operating in parallel calculating firing angles α_v and α_i. The effective α angle is the minimum value of α_v and α_i. This angle is available at the third block output, named `alpha (deg)`. Both regulators are of the proportional-integral type. They should have high enough gains for low frequencies (<10 Hz) to maintain the current or voltage response equal to the reference current (`Id_ref_lim`) or reference voltage (`Vd_ref`), as long as α is within the minimum and maximum limits ($5° < \alpha < 165°$ for rectifier, $92° < \alpha < 165°$ for inverter). The regulator gains Kp and KI are adjusted during small perturbations in the current reference. The following gains are used:

Current regulator	**Kp** = `92 deg/p.u.`	**Ki** = `4500 deg/p.u./s`
Voltage regulator	**Kp** = `35 deg/p.u.`	**Ki** = `2250 deg/p.u./s`

Another particularity of the regulator is the linearization of the proportional gain. As the `Vd` voltage generated by the rectifier and the inverter is proportional to $\cos(\alpha)$, the ΔVd variation due to a $\Delta\alpha$ change is proportional to $\sin(\alpha)$. With a constant Kp value, the effective gain is therefore proportional to $\sin(\alpha)$. In order to keep a constant proportional gain, independent of the α value, the gain is linearized by multiplying the Kp constant by $1/\sin(\alpha)$. This linearization is applied for a range of α defined by two limits specified in the dialog box (third line).

System Startup and Steady State

Notice that the system is discretized, using sample time `Ts` (you should already have `Ts = 50e-6` defined in your workspace).

The system is programmed to start and reach a steady state. Then a step is applied to the reference current so you can observe the dynamic response of the regulators.

Start the simulation and observe the signals on the rectifier and inverter scopes. The waveforms are reproduced here:

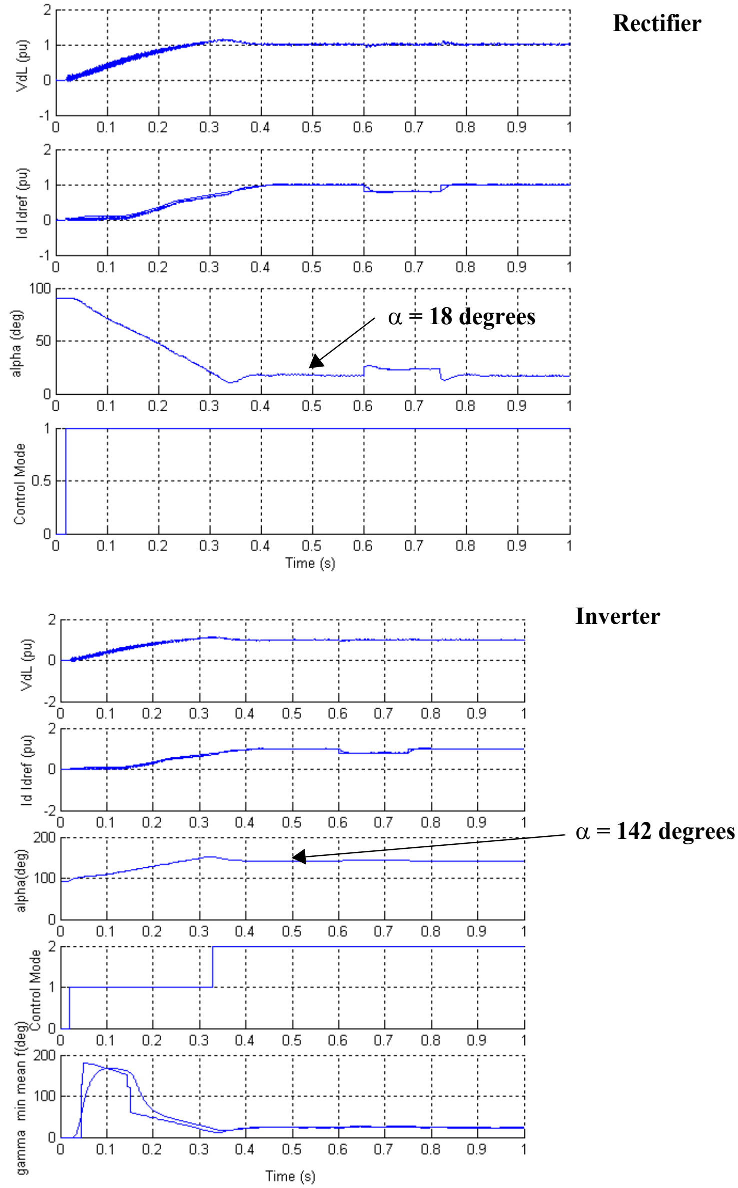


Figure 3-28: Startup of the DC System and Step Applied on the Reference

The reference current follows a ramp from zero to 1 p.u. (2 kA) in 0.4 s. Observe that the DC current starts to build up at t = 20 ms, time at which the controller and the pulse generators are deblocked. The DC current and voltages start from zero and reach steady state in approximately 0.5 s. The rectifier controls the current and the inverter controls the voltage. Trace 1 of both rectifier and inverter scopes shows the DC line voltage (1 p.u. = 500 kV). Trace 2 shows the reference current and the measured `Id` current (1 p.u. = 2 kA). During the ramp the inverter is actually controlling the current (Trace 4: Mode = 1) to the value of `Id_ref_lim` less the Current Margin (0.1 p.u.) and the rectifier tries to control the current at `Id_ref_lim`. At the inverter, the control mode changes to voltage control (Mode = 2) at t = 0.33 s and the rectifier becomes effectively in control of the current. Once steady state is attained, the α firing angles are 18 degrees and 142 degrees respectively on the rectifier and inverter side. At the inverter, the Gamma Measurement block monitors the extinction angle γ for each thyristor of a six-pulse bridge (e.g., the bridge connected to the Wye windings) by determining the elapsed time expressed in electrical degrees from the end of current conduction to the zero crossing of the commutating voltage. The minimum and filtered means of six γ values are shown in trace 5. In steady state, the filtered mean γ is around 24.5 degrees. Then, at t = 0.6 s, a step is applied to the reference current so that you can observe the dynamic response of the regulators.

Comparison of Theoretical and Simulation Results in Steady-State

The main equations governing the steady-state operation of the DC system are given here so that you can compare the theoretical values to the simulation results.

The following expression relates the mean direct voltage Vd of a 12-pulse bridge to the direct current Id and firing angle α:

$$Vd = 2 \times (Vdo \times \cos(\alpha) - Rc \times Id)$$

where Vdo is the ideal no-load direct voltage for a six-pulse bridge:

$$Vdo = (3\sqrt{2}/\pi) \times Vc$$

Vc is the line-to-line RMS commutating voltage that is dependent on the AC system voltage and the transformer ratio.

Rc is the equivalent commutating resistance

$$Rc = (3/\pi) \times Xc$$

Xc is the commutating reactance or transformer reactance referred to the valve side.

The following rectifier parameters were used in the simulation.

The *Vc* voltage must take into account the effective value of the voltage on the 500 kV bus and the transformer ratio. If you look at the waveforms displayed on the AC_RECTIFIER scope, you find 0.96 p.u. when the direct current *Id* has reached its steady state (1 p.u.).

If you open the rectifier transformer dialog box, you find a multiplication factor of 0.90 applied to the primary nominal voltage. The voltage applied to the inverter is therefore boosted by a factor of 1/0.90.

Vc = 0.96 * 200 kV/0.90 = 213.3 kV

Id = 2 kA

α = 18°

Xc = 0.24 p.u., based on 1200 MVA and 222.2 kV = 9.874 Ω

Therefore

$$Vdo = (3\sqrt{2}/\pi) \times 213.3 = 288.1\ kV$$

$$Rc = (3/\pi) \times 9.874 = 9.429\ \Omega$$

$$Vd = 2 \times (288.1\ kV \times \cos(18°) - 9.429 \times 2) = 510\ kV$$

This theoretical voltage corresponds well with the expected rectifier voltage calculated from the inverter voltage and the voltage drop in the DC line.

$$Vd = VdL_{inverter} + (R_{DCline} + R_{\text{inductance}}) \times Id$$

$$Vd = 500\ kV + (4.5\ \Omega + 1\ \Omega) \times 2 = 511\ kV$$

The μ commutation or overlap angle can also be calculated. Its theoretical value depends on α, the DC current *Id,* and the commutation reactance *Xc*.

$$\mu = \mathrm{acos}\left[\cos(\alpha) - \frac{Xc \cdot Id \cdot \sqrt{2}}{Vc}\right] - \alpha$$

$$\mu = \mathrm{acos}\left[\cos(18°) - \frac{9.874 \cdot 2 \cdot \sqrt{2}}{213.3}\right] - 18° = 16.9°$$

Now verify the commutation angle by plotting the currents in two valves, showing for example current extinction in valve 1 and current buildup in valve 3 of one six-pulse bridge of the rectifier.

Open the rectifier subsystem. Then open the upper bridge dialog box and select **All voltages and currents** for the **Measurement** parameter. Now copy the Multimeter block from the Measurements library into your `case4` model. Double-click the Multimeter block. A window showing all the bridge voltages and currents appears. Select the following signals:

uSw1 `Rectifier/Universal Bridge`

iSw1 `Rectifier/Universal Bridge`

iSw3 `Rectifier/Universal Bridge`

The number of signals (3) is displayed in the multimeter icon. Using a Demux block, send the three multimeter output signals to a two-trace scope (Trace 1: `uSw1` Trace 2: `iSw1` and `iSw3`). Restart the simulation. The waveforms illustrating two cycles are shown in the following figure. The measured commutation angle is 14 steps of 50 μs or 15.1° of a 60 Hz period. The resolution with a 50 μs time step is 1.1°; this angle compares reasonably well with the theoretical value.

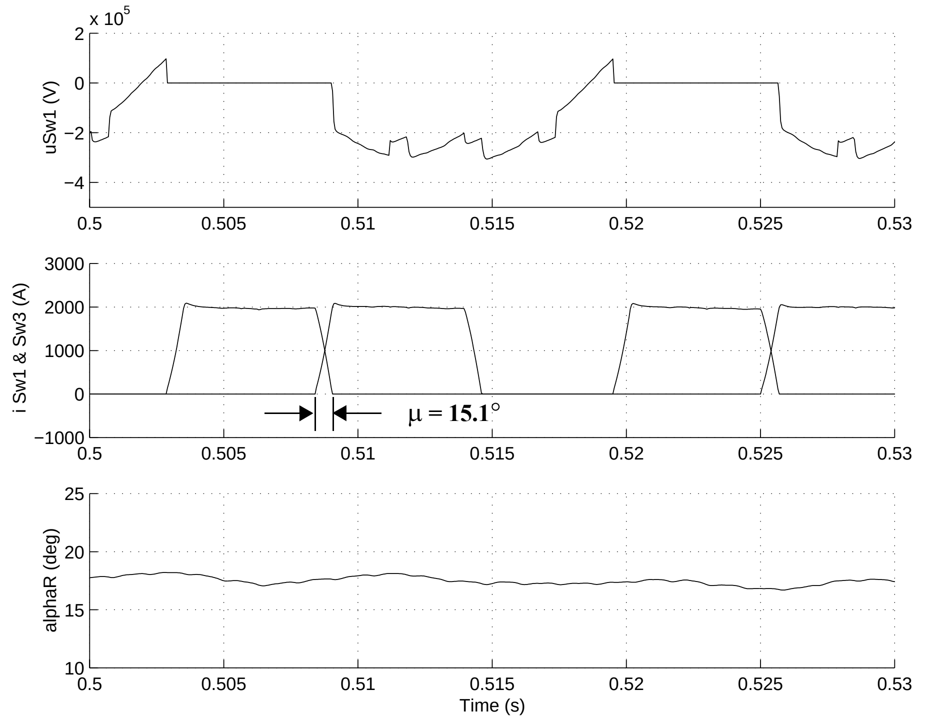


Figure 3-29: Valve Voltage and Currents (Commutation from Valve 1 to Valve 3)

Finally, to validate the γ measurement at the inverter, plot the valve 1 voltage and current. Also plot the commutating voltage corresponding to the outgoing valve 1 to be extinguished and the filtered mean value of γ as shown in Figure 3-30. (The filter is low-pass with a time constant of 20 ms.) Verify also that the values of α, μ, and γ add up to 180°.

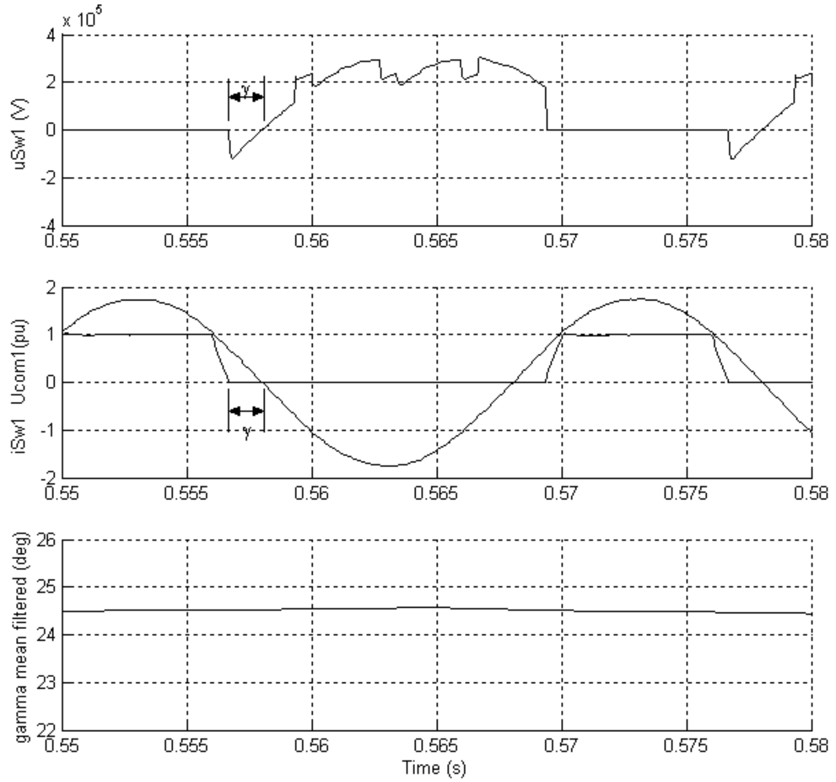


Figure 3-30: Current and Commutation Voltage of Valve 1 Showing γ

Response to a Step of Reference Current

At t = 0.6 s, a 0.2 p.u. step is applied to the reference current (decrease from 1 p.u. to 0.8 p.u.). At t = 0.75 s, another step is applied to set the reference back to 1 p.u. Observe the response of the current regulator. It stabilizes in approximately 0.1 s.

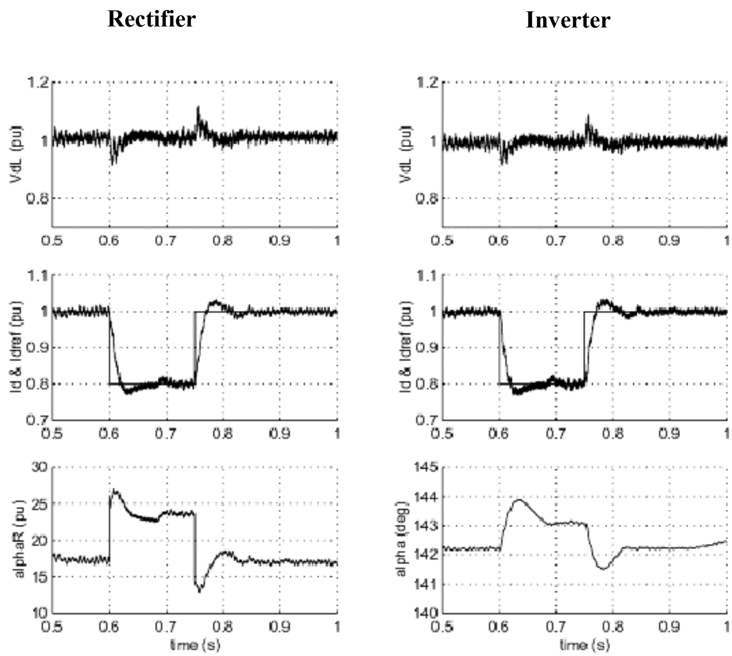


Figure 3-31: Response to a 0.2 p.u. Step of the Reference Current

DC Line Fault

Disconnect the Step Up & Down block in order to eliminate the step disturbance applied to the reference current. In the DC Fault and Forced Delay blocks of the `power_hvdc12pulse` model, change the multiplication factor of 100 to 1, so that a fault is now applied at t = 0.6 s. Open the FAULT scope to observe the fault current. Restart the simulation.

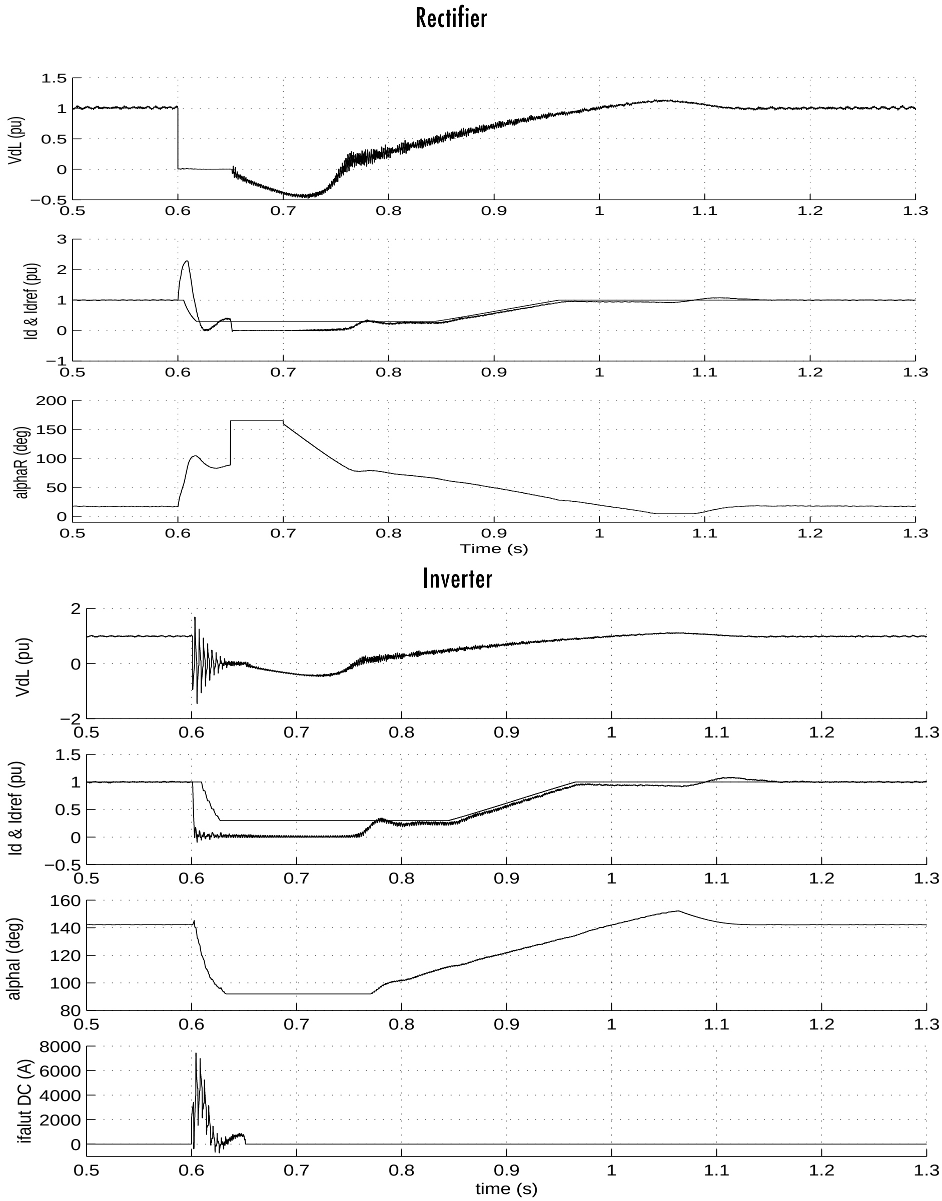


Figure 3-32: DC Line Fault on the Rectifier Side

At fault application (t = 0.6 s), the DC current increases to 2.3 p.u. and the DC voltage falls to zero at the rectifier. This DC voltage drop is seen by the Voltage Dependent Current Order Limiter (VDCOL), which reduces the reference current to 0.3 p.u. at the rectifier. A DC current still continues to circulate in the fault. Then, at t = 0.65 s, the rectifier α firing angle is forced to 165 degrees when the signal applied to the `ForcedAlpha` input goes high. This signal would normally be provided by the protection system not simulated here. The rectifier now operates in inverter mode. The DC line voltage becomes negative and the energy stored in the line is returned to the AC network, causing rapid extinction of the fault current at its next zero crossing. Then α is released at t = 0.7 s and the normal DC voltage and current recover in approximately 0.5 s.

AC Line-to-Ground Fault at the Rectifier

Now you modify the fault timings in order to apply a line-to-ground fault. In the DC Fault and Forced Delay blocks of `power_hvdc12pulse`, change the multiplication factor of 1 to 100, so that the DC fault is now eliminated. In the A-G Fault block, change the multiplication factor in the switching times to 1, so that a six-cycle line-to-ground fault is now applied at the rectifier. Restart the simulation.

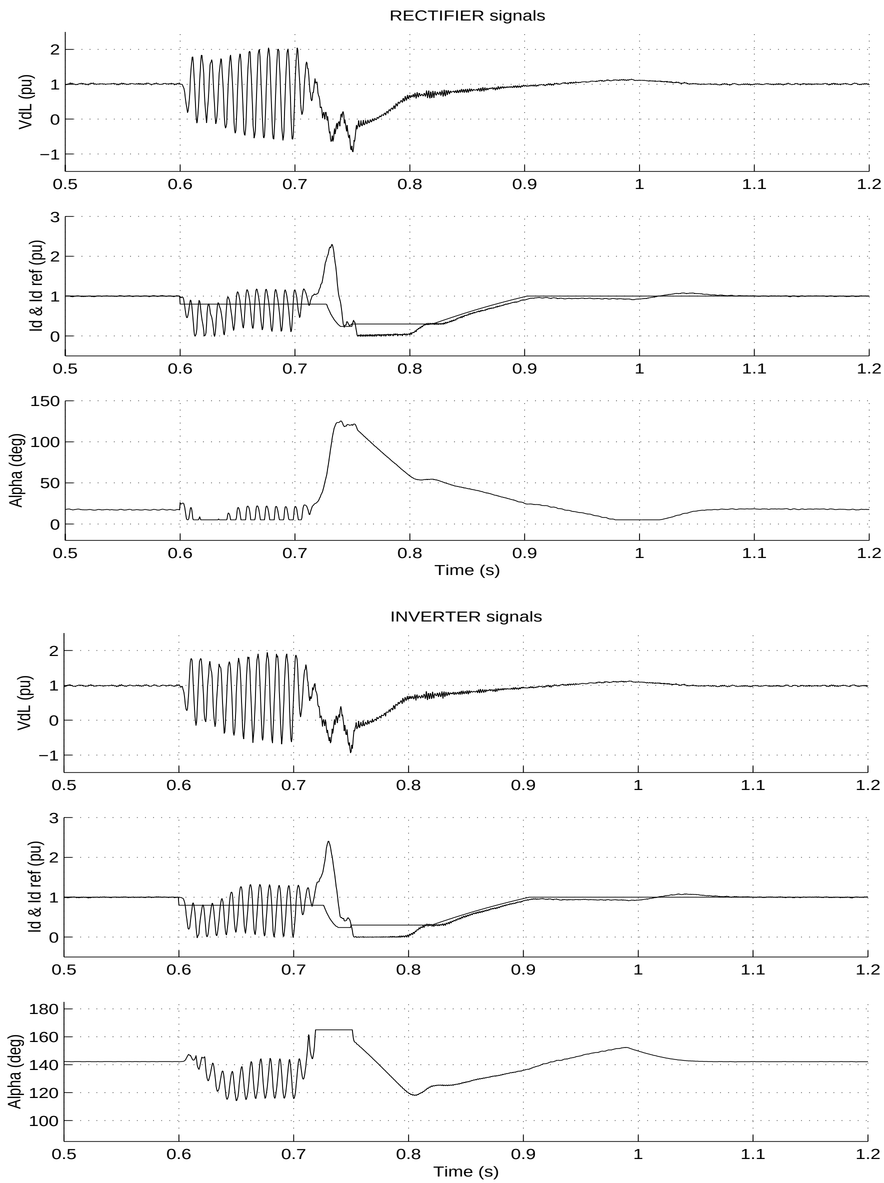


Figure 3-33: Rectifier, Inverter Signals for an AC Line Fault on Rectifier Side

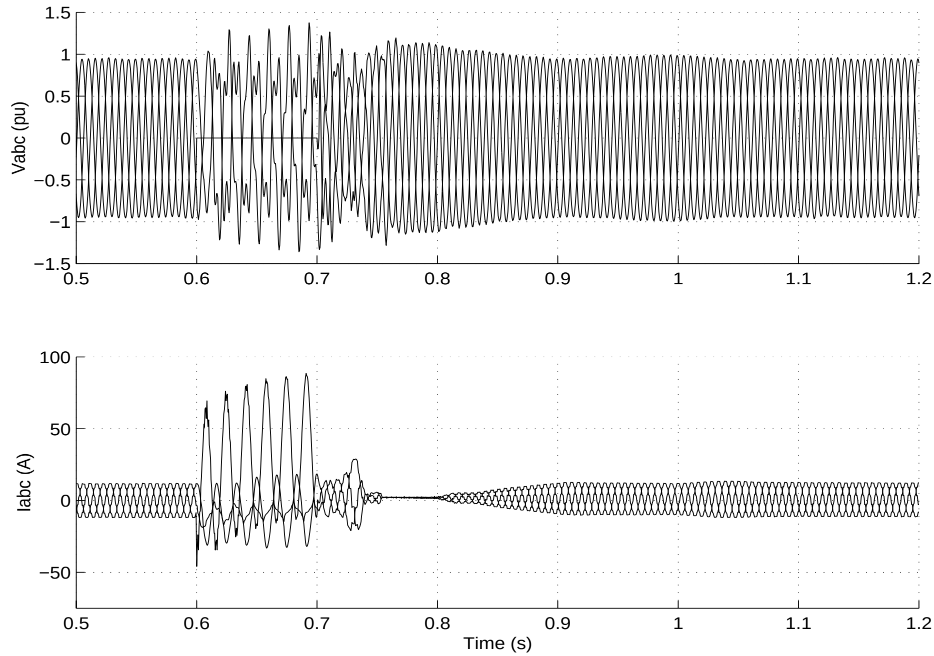


Figure 3-34: Voltages and Currents on the 60 Hz Side for an AC Line Fault on the Rectifier Side

Notice the 120 Hz oscillations in the DC voltage and currents during the fault. When the fault is cleared at t = 0.7 s, the VDCOL operates and reduces the reference current to 0.3 p.u. The system recovers in approximately 0.4 s after fault clearing.

References

[1] Arrilaga, J., *High Voltage Direct Current Transmission,* IEEE Power Engineering Series 6, Peter Peregrinus, Ltd., 1983.

[2] *Electromagnetic Transients Program (EMTP), Workbook IV (TACS),* EL-4651, Volume 4, Electric Power Research Institute, 1989.

4

Improving Simulation Performance

SimPowerSystems gives you many tools to speed up your power system simulations. Depending on your model, you can choose among continuous, discrete, and phasor integration methods. Simulink and related products provide additional ways to enhance model performance, including code generation, creating your own model libraries, and tuning block parameters.

How SimPowerSystems Works

Once you have built your circuit with the blocks of **powerlib**, you can start the simulation just like any other Simulink model. Each time you start the simulation, a special initialization mechanism is called. This initialization process computes the state-space model of your electric circuit and builds the equivalent system that can be simulated by Simulink.

The `power_analyze` command is part of that process. It obtains the state-space model and builds the Simulink model of your circuit. You can also call `power_analyze` from the command line to obtain the state-space model of the linear part of the circuit. When called by the initialization process, `power_analyze` performs the following five steps as shown in Figure 4-1:

1. Sorts all SimPowerSystems blocks, gets the block parameters and evaluates the network topology. The blocks are separated into linear and nonlinear blocks, and each electrical node is automatically given a node number.

2. Once the network topology has been obtained, the state-space model (A, B, C, D matrices) of the linear part of the circuit is computed by the `power_statespace` command. All steady-state calculations and initializations are performed at this stage.

   If you have chosen to discretize your circuit, the discrete state-space model is computed from the continuous state-space model, using the Tustin method.

   If you are using the phasor solution method, the state-space model is replaced with the complex transfer matrix $H(j\omega)$ relating inputs and outputs (voltage and current phasors) at the specified frequency. This matrix defines the network algebraic equations.

3. Builds the Simulink model of your circuit and stores it inside one of the measurement blocks. This means that you need at least one measurement block (Current Measurement block, Voltage Measurement block, Three-Phase V-I Measurement block, or Multimeter block) in your model. The connections between the equivalent circuit and measurements blocks are performed by invisible links using the Goto and From blocks.

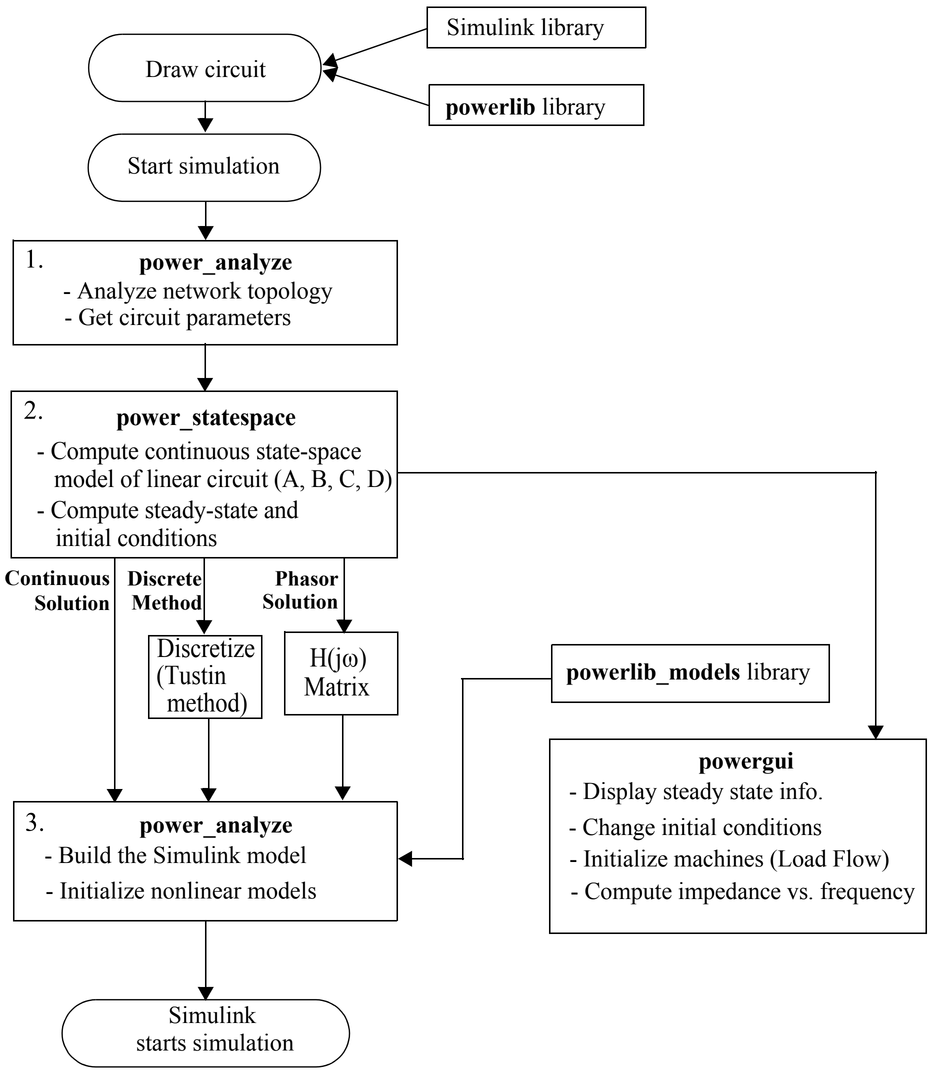


Figure 4-1: SimPowerSystems Flowchart

The Simulink model uses a State-Space block or an S-Function block to model the linear part of the circuit. Predefined Simulink models are used to simulate nonlinear elements. These models can be found in the **powerlib_models** library available with SimPowerSystems. Simulink Source blocks connected at the input of the State-Space block are used to simulate the electrical source blocks.

The next figure represents the interconnections between the different parts of the complete Simulink model. The nonlinear models are connected in feedback between voltage outputs and current inputs of the linear model.

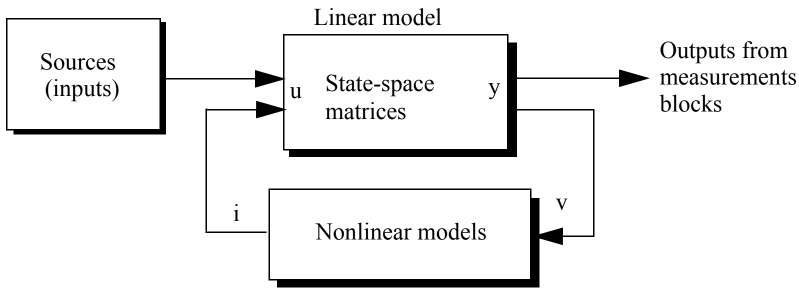


Figure 4-2: Interconnection of Linear Circuit and Nonlinear Models

Once `power_analyze` has completed the initialization process, Simulink starts the simulation. You can observe waveforms on scopes connected at the outputs of your measurement blocks. Through the Powergui, you can access the LTI viewer and obtain transfer functions of your system between any pair of input and output. The Powergui also allows you to perform a FFT analysis of recorded signals in order to obtain their frequency spectrum.

If you stop the simulation and drag a copy of the Powergui block into your circuit window, you have access to the steady-state values of inputs, outputs, and state variables displayed as phasors. You can also use the interface to modify the initial conditions. The Powergui block interface allows you to perform a load flow with circuits involving three-phase machinery and initialize the machine models so that the simulation starts in steady state. This feature avoids long transients due to mechanical time constants of machines. The Powergui block allows you to specify the desired frequency range, visualize impedance curves, and store results in your workspace for Impedance Measurement blocks connected in your circuit.

Choosing an Integration Method

Three solution methods are available through the Powergui block. These are:

- Continuous solution method using Simulink variable-step solvers
- Discretization for solution at fixed time steps
- Phasor solution method using Simulink variable-step solvers

Continuous versus Discrete Solution

One important feature of SimPowerSystems is its ability to simulate electrical systems either with continuous variable-step integration algorithms or with a fixed-step using a discretized system. For small size systems, the continuous method is usually more accurate. Variable-step algorithms are also faster because the number of steps is fewer than with a fixed-step method giving comparable accuracy. When using line-commutated power electronics, the variable-step, event-sensitive algorithms detect the zero crossings of currents in diodes and thyristors with a high accuracy so that you do not observe any current chopping. However, for large systems (containing either a large number of states or nonlinear blocks), the drawback of the continuous method is that its extreme accuracy slows down the simulation. In such cases, it is advantageous to discretize your system. In the following two sections, we explain these two methods, their advantages, and their limitations.

What do we mean by “small size” and “large size”? Although the distinction is not sharp, you can consider small size a system that contains fewer than 30 electrical states and fewer than 6 electronic switches. Circuit breakers do not affect the speed much, because unlike power electronic switches, which are commutated at every cycle, these devices are operated only a couple of times during a test.

Phasor Solution Method

If you are interested only in the changes in magnitude and phase of all voltages and currents when switches are closed or opened, you don’t need to solve all differential equations (state-space model) resulting from the interaction of R, L, C elements. You can instead solve a much simpler set of algebraic equations relating the voltage and current phasors. This is what the phasor solution method does. As its name implies, this method computes voltages and currents as phasors. The phasor solution method is particularly useful for studying transient stability of networks containing large generators and motors. In this type of problem, we are interested in electromechanical oscillations resulting from interactions of machine inertias and regulators. These oscillations produce a modulation of the magnitude and phase of fundamental voltages and currents at low frequencies (typically between 0.02 Hz and 2 Hz). Long simulation times are therefore required (several tens of seconds). The continuous or discrete solution methods are not appropriate for this type of problem.

In the phasor solution method, the fast modes are ignored by replacing the network differential equations by a set of algebraic equations. The state-space model of the network is replaced by a complex matrix evaluated at the fundamental frequency and relating inputs (currents injected by machines into the network) and outputs (voltages at machine terminals). As the phasor solution method uses a reduced state-space model consisting of slow states of machines, turbines and regulators, it dramatically reduces the required simulation time.

Continuous variable-step solvers are very efficient in solving this type of problem. Recommended solvers are `ode15s` or `ode23tb` with a maximum time step of one cycle of the fundamental frequency (1/60 s or 1/50 s).You must keep in mind however that this faster solution technique gives the solution only in the vicinity of the fundamental frequency.

Simulating with Continuous Integration Algorithms

Simulink provides a variety of solvers. Most of the variable-step solvers work well with linear circuits. However circuits containing nonlinear models, especially circuits with circuit breakers and power electronics, require stiff solvers.

Choosing an Integration Algorithm

Fastest simulation speed is usually achieved with `ode23tb` or `ode15s` with default parameters.

Solver	`ode23tb` or `ode15s`
Relative tolerance	`1e-3`
Absolute tolerance	`auto`
Maximum step siz	`auto`
Initial step size	`auto`
Maximum order	(for `ode15s`) = 5

Normally, you can choose `auto` for the absolute tolerance and the maximum step size. In some occasions you might have to limit the maximum step size and the absolute tolerance. Selecting too small a tolerance can slow down the simulation considerably. The choice of the absolute tolerance depends on the maximum expected magnitudes of the state variables (inductor currents and capacitor voltages). For example, if you work with high-power converters where expected voltage and currents are thousands of volts and amperes, an absolute tolerance of 0.1 or even 1.0 should be sufficient. If you are working with low-power circuits involving maximum values of 100 V and 10 A, you should use a smaller absolute tolerance, such as 0.001 or 0.01.

Simulating Switches and Power Electronic Devices

Two methods are used for simulation of switches and power electronic devices:

- If the switch is purely resistive the switch model is considered as part of the linear circuit. The state-space model of the circuit, including open and closed switches, is therefore recalculated at each switch opening or closing, producing a change in the circuit topology. This method is always used with the Breaker block and the Ideal Switch block because these elements do not have internal inductance. It is also applied for the Diode block and the Thyristor block, with Ron > 0 and Lon = 0, and for the Universal Bridge with forced commutated devices.

- If the switch contains a series inductance (Diode and Thyristor with Lon > 0, IGBT, MOSFET, or GTO), the switch is simulated as a current source driven by voltage across its terminals. The nonlinear element (with a voltage input and a current output) is then connected in feedback on the linear circuit, as shown in Figure 4-2.

You have therefore the choice to simulate diodes and thyristors with or without Lon internal inductance. In most applications, it is not necessary to specify an inductance Lon. However, for circuit topologies resulting in zero commutation or overlap angle, you have to specify a switch inductance Lon in order to help commutation.

Consider for example the circuit shown in the following figure. This circuit is available in the `power_rectifier_ideal` model. The thyristor bridge is fed from an infinite source (zero impedance) so that the commutation between thyristors is quasi instantaneous.

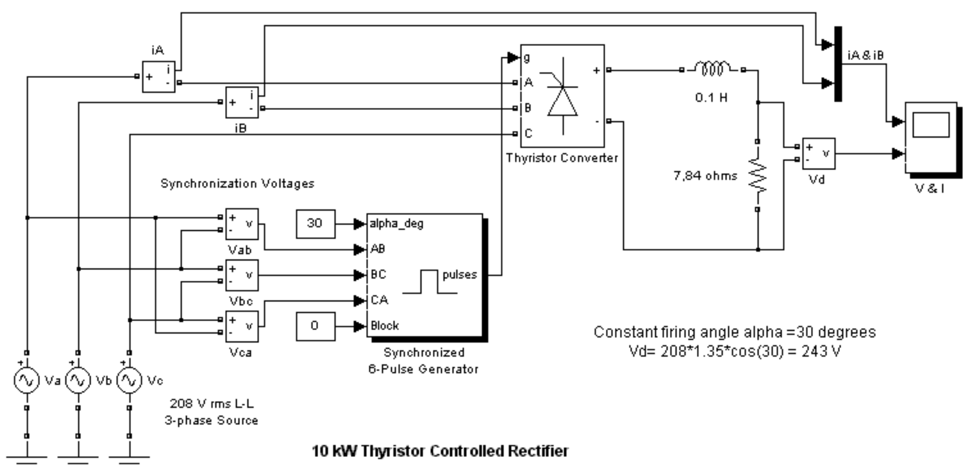


Figure 4-3: Three-Phase Thyristor Rectifier on Infinite Source

If you simulate this circuit without internal thyristor inductances (Lon = 0), observe high current spikes flowing in the three lines. This happens because during commutation two thyristors connected to the same positive or negative terminal of the bridge are in conduction for a short period of time, applying a line-to-line short circuit on the source (see Figure 4-4 following). During commutation, the current is limited only by the internal resistance of thyristors (with Ron = 0.01 ohms, the current reaches 7.35 kA ($208*\sqrt{2}$ $*\sin(30^o)$ / (2*0.01) or 245 times the normal DC current of 30 A). These short circuits can be avoided by using a small Lon = 1 μH in the thyristor model. If you repeat the simulation, you get square current waveforms with a peak value of 30 A.

If you zoom in on the line current during a commutation, you discover that the commutation is not instantaneous. The commutation time depends on the Lon value and the DC current.

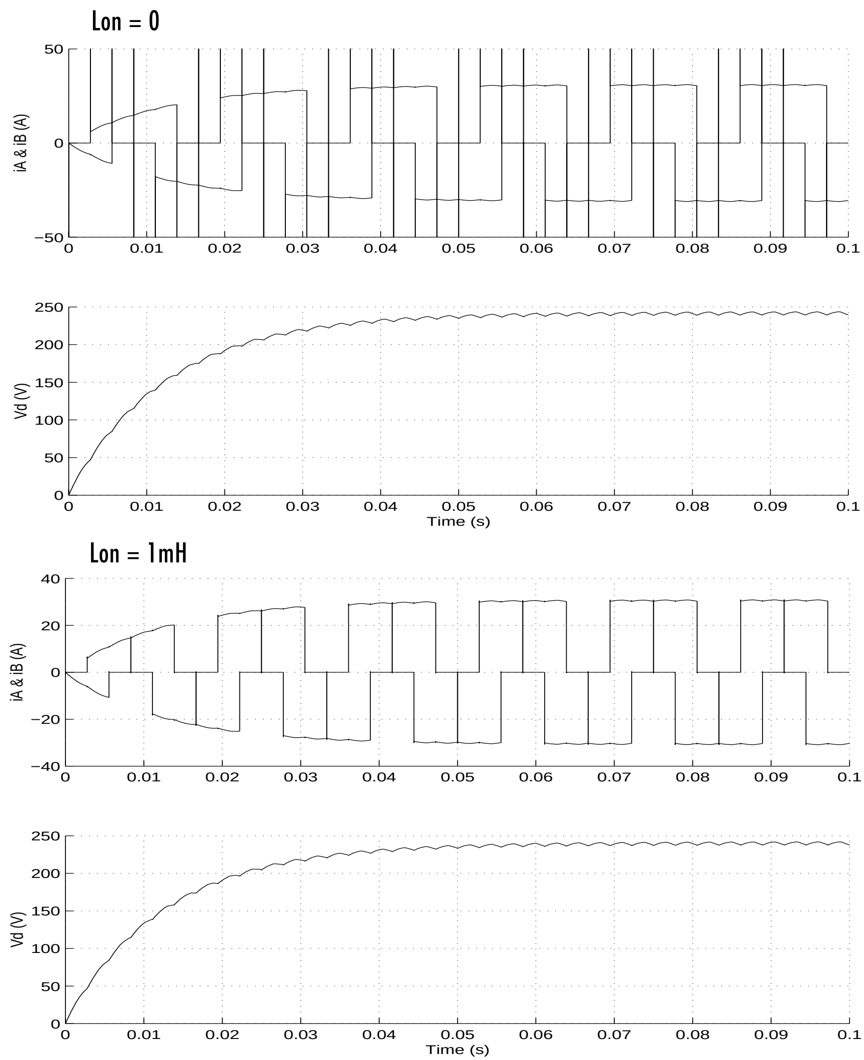


Figure 4-4: Source Currents and DC Load Voltage with Lon = 0 and Lon = 1 μH

Simulating Discretized Electrical Systems

You implement discretization by dragging the Powergui block into your system. The sample time is specified in the block dialog box. The electrical system is discretized using the Tustin method, which is equivalent to a fixed-step trapezoidal integration. In order to avoid algebraic loops, the electrical machines are discretized using the Forward Euler method.

The precision of the simulation is controlled by the time step you choose for the discretization. If you use too large a sample time, the precision might not be sufficient. The only way to know if it is acceptable is to repeat the simulation with different sample times or to compare with a continuous method and to find a compromise for the largest acceptable sample time. Usually sample times of 20 μs to 50 μs give good results for simulation of switching transients on 50 Hz or 60 Hz power systems or on systems using line-commutated power electronic devices such as diodes and thyristors. However, for systems using forced-commutated power electronic switches, you must reduce the time step. These devices, the insulated-gate-bipolar transistor (IGBT), the field-effect transistor (FET), and the gate-turn-off thyristor (GTO) are usually operating at high switching frequencies. For example, simulating a pulse-width-modulated (PWM) inverter operating at 8 kHz requires a time step of 1 μs or less.

Note that even if you discretize your electric circuit, you can still use a continuous control system. However, the simulation speed is improved by use of a discrete control system.

Limitations of Discretization with Nonlinear Models

There are a few limitations to discretizing nonlinear models.

Discretization of individual forced-commutated electronic devices is not allowed

Discretization of circuits containing forced-commutated power electronic devices (IGBT, GTO, or MOSFET) is permitted only with the Universal Bridge block. Discretization of circuits containing individual forced-commutated devices is not allowed. For example, an attempt to discretize the buck DC chopper circuit saved in the `power_buckconv` model produces a warning message:

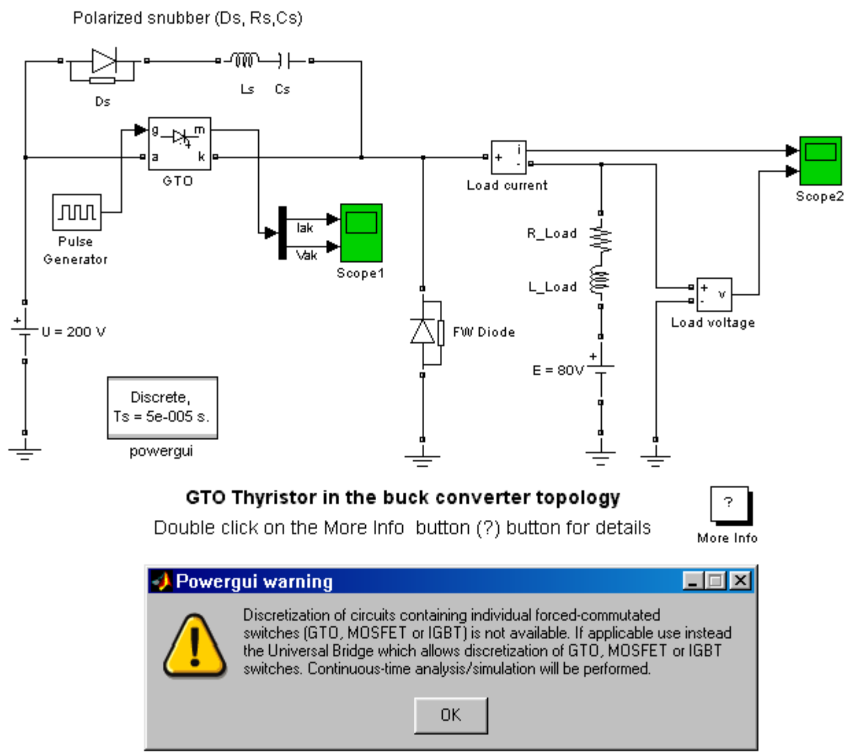


Figure 4-5: A Circuit Containing Individual Forced Commutated Electronic Switches Cannot be Discretized

In this circuit, the opening of the GTO forces a quasi instantaneous conduction of the freewheeling diode. If the circuit was discretized, the diode would be fired with one step delay, and the inductive current chopping would produce large overvoltages. However, for conventional converter topologies as in the case of the Universal Bridge, the switch interactions are known in advance. For example, in a six-switch IGBT/Diode inverter (Figure 4-6 following), opening of IGBT1 causes instantaneous conduction of diode D2 in the same arm. As the circuit topology is predetermined, it is possible to force firing of the diode in the same step that the IGBT opens. You should use a continuous method if you prefer to use individual IGBT and Diode blocks to simulate a complete inverter.

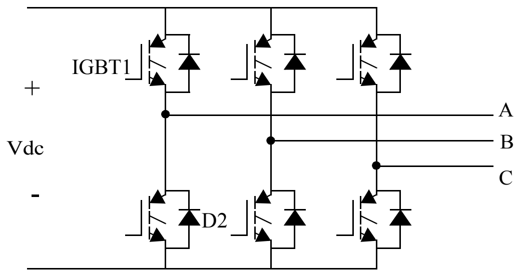


Figure 4-6: IGBT Inverter Simulated by the Universal Bridge

Minimal load is required at machine terminals

When using electrical machines in discrete systems, you might have to use a small parasitic resistive load, connected at the machine terminals, in order to avoid numerical oscillations. Large sample times require larger loads. The minimum resistive load is proportional to the sample time. As a rule of thumb, remember that with a 25 μs time step on a 60 Hz system, the minimum load is approximately 2.5% of the machine nominal power. For example, a 200 MVA synchronous machine in a power system discretized with a 50 μs sample time requires approximately 5% of resistive load or 10 MW. If the sample time is reduced to 20 μs, a resistive load of 4 MW should be sufficient.

Lon = 0 is used for diodes and thyristors in discrete circuits

Diodes and thyristors used in a discretized circuit must have a zero internal inductance. If you discretize a circuit containing diodes or thyristors with Lon > 0, SimPowerSystems prompts you with a warning indicating that Lon will be reset to zero.

Increasing Simulation Speed

Once the proper method (continuous, discrete, or phasor), solver type, and parameters have been selected, there are additional steps you can take to optimize your simulation speed:

- Discretize your electric circuit and your control system. You can even use a larger sample time for the control system, provided that it is a multiple of the smallest sample time.
- Simulating large systems or complex power electronic converters can be time consuming. If you have to repeat several simulations from a particular operating point, you can save time by specifying a vector of initial states in the **Simulation —> Simulation parameters —> Workspace IO** dialog pane. This vector of initial conditions must have been saved from a previous simulation run.
- Reducing the number of open scopes and the number of points saved in the scope also helps in reducing the simulation time.
- If you have the Simulink Performance Tools option installed, you can use the Accelerator. The performance gain obtained with the Accelerator varies with the size and complexity of your model. Typically you can expect performance improvements by factors of two to 10.

Using Accelerator Mode and Real-Time Workshop

The Simulink Accelerator mode is explained in the Simulink user's guide.

The Simulink Accelerator speeds up the execution of Simulink models by replacing the interpreted M code running beneath the Simulink blocks with compiled code as your model executes. The Simulink Accelerator uses portions of Real-Time Workshop (RTW) to generate this code on the fly. Although the Simulink Accelerator uses RTW technology, Real-Time Workshop is not required to run it. Also, if you do not have your own C compiler installed, you can use the lcc C compiler provided with MATLAB.

To activate the Simulink Accelerator, select **Accelerator** instead of **Normal** in the **Simulation** menu of your model window. Alternatively, select `Accelerator` in the pull-down menu to the right and below the **Simulation** menu.

The following table shows typical performance gains obtained with discretization and Simulink Accelerator applied on the following two demos: a

DC drive using a chopper and the AC-DC converter using a three-phase, three-level voltage-sourced converter. Two versions of the DC drive model are provided in the Demos library: a continuous version, `power_dcdrive`, and a discrete version, `power_dcdrive_disc`. The AC-DC converter is available as the `power_3levelVSC` demo.

	Simulation Time in Seconds*	
Simulation Method	DC drive (Stop time = 1 s)	AC-DC converter (Stop time = 0.15 s)
Continuous: `ode23tb` default parameters	175	—
Discrete	23 (Ts = 10 μs)	25 (Ts = 5 μs)
Discrete + Accelerator	10 (Ts = 10 μs)	8.4 (Ts = 5 μs)

* Simulation times obtained on a Pentium II 500 MHz processor, with 128MB of RAM

The table shows how discretizing your circuit boosts the simulation speed by a factor of 7.6 for the DC drive. Using the Accelerator mode, an additional factor of 2.3 performance gain is obtained. For complex power electronic converter models, the Accelerator provides performance gains up to factors of 10.

To take full advantage of the performance enhancements made possible by converting your models to code, you must use Real-Time Workshop to generate stand-alone C code. You can then compile and run this code and, with xPC Target, also run it on a target PC operating the xPC Target real-time kernel.

The Nonlinear Model Library

The building blocks used to assemble the Simulink model of the nonlinear circuit are stored in a library named **powerlib_models**. You do not normally need to work with the **powerlib_models** library. However, you might have to look inside the models or modify them for particular applications. You can access that library by entering `powerlib_models` in the MATLAB Command Window.

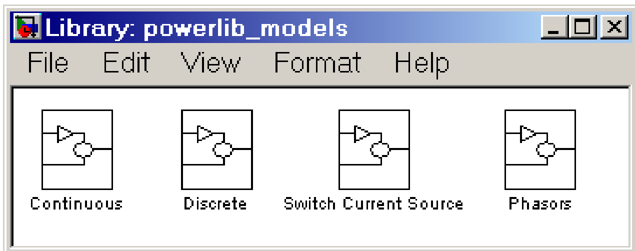
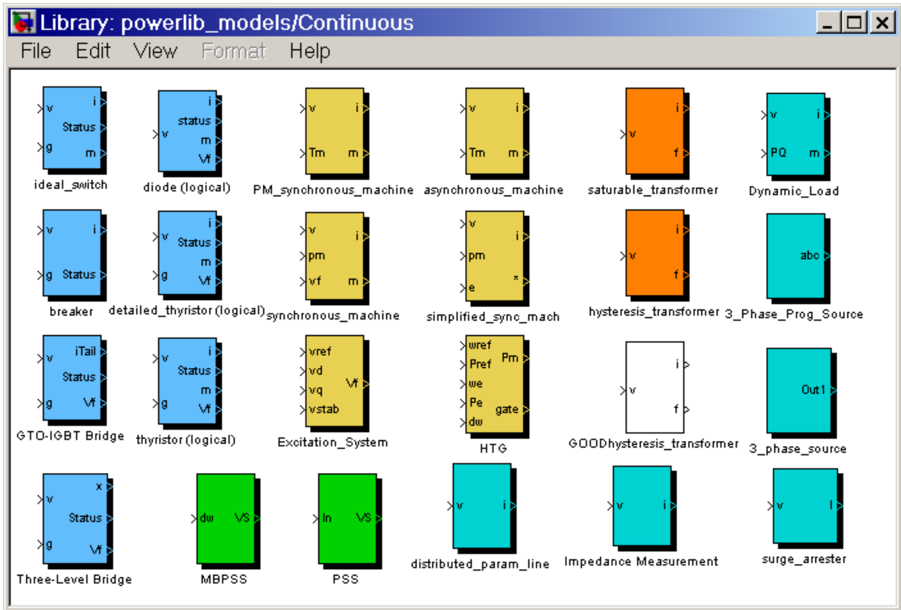


Figure 4-7: The powerlib_models Library

The Continuous Library

The Continuous library contains two types of blocks:

- Current sources simulating continuous machine models, surge arrester, saturable transformer, and distributed parameter lines
- Switching logics used for purely resistive power electronic devices: breaker, diode, three-level bridge, thyristor, universal bridge, and individual forced-commutated devices



Nonlinear Blocks Simulated by Current Sources

These blocks use a voltage input (output of the state-space model of the linear circuit) and their current output is fed into the state-space model. For complex models, such as electrical machines requiring several inputs and outputs, vectorized signals are used. Useful internal signals are also returned by most of the models in a measurement output vector `m`.

For example, the Asynchronous Machine model is stored in the block named asynchronous_machine. The model uses as inputs a vector of four voltages, two rotor voltages and two stator voltages, respectively: (VabR, VbcR, VabS, VbcS). It returns a vector of four currents, two rotor currents and two stator currents, respectively: (IaR, IbR, IaS, IbS). The model also returns a measurement output vector of 20 signals. When the Asynchronous Machine block is used from **powerlib** this measurement output vector is accessible through the `m` output of the machine icon. You can get details on the model inputs and outputs from the documentation of **powerlib** and **powerlib_models** block icons.

Logics for Switches and Power Electronic Devices

For switches and power electronic devices, the blocks contain only the logic returning the status (open or closed) of the switch. The switch status is passed to an S-function, which recomputes the state-space model of the linear circuit each time that a switch status is changed. The `m` output is a vector returning the switch current and voltage. The `i` output returns the tail current of forced-commutated devices such as IGBTs and GTOs. All the switch logics are vectorized. This means that a single model is used by `power_analyze` to simulate all the devices having the same type.

The Discrete Library

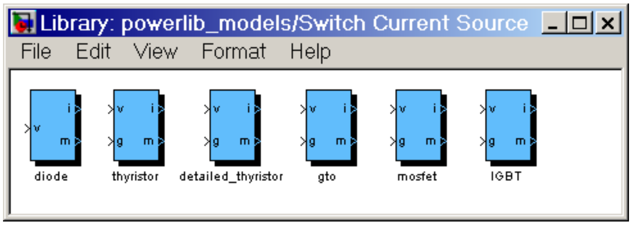
The Discrete library contains the discrete versions of the continuous models described above.

The Phasors Library

The Phasors library contains the phasor versions of some of the continuous models described above. See the "Modeling Simple Systems" chapter for more details on the phasor simulation.

The Switch Current Source Library

This library contains models of power electronic devices, which are simulated by a current source external to the linear circuit.



These devices are the diode and the thyristor with Lon > 0, and the three forced-commutated devices: gate-turn-off thyristor (GTO), metal-oxide-semiconductor field-effect transistor (MOSFET), and the insulated-gate-bipolar transistor (IGBT). All these models are continuous and contain an internal inductance, allowing you to handle fast transitions of forced-commutated converters. As for electrical machines, these models use a voltage input (output of the state-space model of the linear circuit) and their current output is fed into the state-space model. All these models are vectorized.

Limitations of the Nonlinear Models

Because nonlinear models are simulated as current sources, they cannot be connected in series with inductors and their terminals cannot be left open.

If you feed a machine through an inductive source, `power_analyze` prompts you with an error message. You can avoid this by connecting large resistances in parallel with the source inductances or across the machine terminals.

A series RC snubber circuit is included in the model of the Breaker block and power electronics blocks. You should not have any problems if you keep these snubber circuits in service. The snubber can be changed to a single resistance by setting **Cs** to `Inf`, or to a single capacitor by setting **Rs** = `0`. To eliminate the snubber, specify **Rs** = `Inf` or **Cs** = `0`.

Modifying the Nonlinear Models of the powerlib_models Library

To use your own **powerlib_models** library, you must first copy the `powerlib_models.mdl` file into your working directory or any other directory. If you are using a directory different from the current directory, you must specify this new directory in the MATLAB search path before the standard blockset directory.

Then you can customize this new **powerlib_models** library, as long as you do not change the names of the blocks, the number of inputs and outputs, and the number of parameters in their dialog boxes. The next time you run the simulation, these modifications take effect in your circuit.

Creating Your Own Library of Models

SimPowerSystems provides a variety of basic building blocks to build more complex electric blocks. Using the masking feature of Simulink, you can assemble several elementary blocks of **powerlib** into a subsystem, build your own parameter dialog box, create the desired block icon, and place this new block in your personal library.

The “Modeling Simple Systems” chapter explained how to build a nonlinear model using a Voltage Measurement block and a Controlled Current Source block. The proposed examples (a nonlinear inductance and a nonlinear resistance) were relatively simple. Using the same principle, you can develop much more complex models using several controlled current sources, or even controlled voltage sources. Refer to the tutorial “Building and Customizing Nonlinear Models” on page 2-40.

Changing Your Circuit Parameters

Each time that you change a parameter of the **powerlib** blocks, you have to restart the simulation in order to evaluate the state-space model and update the parameters of the nonlinear models. However, you can change any source parameter (Magnitude, Frequency, or Phase) during the simulation. The modification takes place as soon as you apply the modification or close the source block menu.

As for the Simulink blocks, all the **powerlib** block parameters that you specify in the dialog box can contain MATLAB expressions using symbolic variable names. Before running the simulation, you must assign a value to each of these variables in your MATLAB workspace. This allows you to perform parametric studies by changing the parameter values in a MATLAB script.

Example of MATLAB Script Performing a Parametric Study

Suppose that you want to perform a parametric study in a circuit named `my_circuit` to find the impact of varying an inductance on switching transients. You want to find the highest overvoltage and the inductance value for which it occurred.

The inductance value of one of the blocks contains variable `L1`, which should be defined in your workspace. `L1` is varied in 10 steps from 10 mH to 100 mH and the values to be tested are saved in a vector, `L1_vec`. The voltage waveform to be analyzed is stored in a ToWorkspace block in matrix format with `V1` variable name.

You can write a MATLAB M-file that loops on the 10 inductance values and displays the worst case.

```
L1_vec= (10:10:100)*1e-3; % 10 inductances values 10/100 mH
V1_max=0;
for i=1:10
   L1=L1_vec(i);
   fprintf('Test No %d L1= %g H\n', i, L1);
   sim('my_circuit'); % performs simulation
   % memorize worst case
   if max(abs(V1))>V1_max,
       imax=i;
       V1_max=max(abs(V1));
```

```
    end
end

fprintf('Maximum overvoltage= %g V occured for L1=%g H\n', V1_max,
L1_vec(imax));
```

5

SimPowerSystems Block Reference

This chapter contains complete information on every block in SimPowerSystems. Refer to this chapter when you need to find detailed information on a particular block.

Blocks – By Category

The SimPowerSystems main library, **powerlib**, organizes its blocks into libraries according to their behavior. The **powerlib** window displays the block library icons and names. This section lists all SimPowerSystems blocks arranged by library.

Use the Simulink Library Browser or the SimPowerSystems library to access the blocks directly, guided by this hierarchical library list.

The main SimPowerSystems **powerlib** block library window contains the Powergui block that opens a graphical user interface for the steady-state analysis of electrical circuits.

Electrical Sources Library

Contains blocks that generate electric signals.

Elements Library

Contains linear and nonlinear circuit elements.

Phasor Elements Library

Contains specialized circuit elements for phasor analysis

Power Electronics Library

Contains power electronics devices.

Machines Library

Contains power machinery models.

Measurements Library

Contains blocks for the current and voltage measurements.

Extras Library

Contains three-phase blocks and specialized measurement and control blocks. You can also open this library by entering `powerlib_extras` at the command line.

Demos Library

Contains useful demos and case studies.

Nonlinear Simulink Blocks for SimPowerSystems Models

The nonlinear Simulink blocks of **powerlib** are stored in a special block library named **powerlib_models**. These masked Simulink models are used by SimPowerSystems to build the equivalent Simulink model of your circuit. See the "Improving Simulation Performance" chapter for a description of the **powerlib_models** library.

Creating Electrical Sources

AC Current Source	Implement a sinusoidal current source
AC Voltage Source	Implement a sinusoidal voltage source
Controlled Current Source	Implement a controlled current source
Controlled Voltage Source	Implement a controlled voltage source
DC Voltage Source	Implement a DC voltage source
Three-Phase Programmable Voltage Source	Implement a three-phase voltage source with programmable time variation of amplitude, phase, frequency, and harmonics
Three-Phase Source	Implement a three-phase source with internal R-L impedance

Creating Circuit Elements

Breaker	Implement a circuit breaker opening at current zero crossing
Connection Port	Create a terminal port for a subsystem
Distributed Parameter Line	Implement an N-phases distributed parameter line model with lumped losses
Ground	Provide a connection to the ground
Linear Transformer	Implement a two- or three-windings linear transformer
Mutual Inductance	Implement a magnetic coupling between two or three windings
Neutral	Implement a local common node in the circuit
Parallel RLC Branch	Implement a parallel RLC branch
Parallel RLC Load	Implement a linear parallel RLC load
PI Section Line	Implement a single-phase transmission line with lumped parameters
Saturable Transformer	Implement a two- or three-windings Saturable Transformer
Series RLC Branch	Implement a series RLC branch
Series RLC Load	Implement a linear series RLC load
Surge Arrester	Implement a metal-oxide surge arrester

Three-Phase Breaker	Implement a three-phase circuit breaker opening at current zero crossing
Three-Phase Dynamic Load	Implements a three-phase dynamic load with active power and reactive power as a function of voltage or controlled from an external input
Three-Phase Fault	Implement a programmable phase-to-phase and phase-to-ground fault breaker system
Three-Phase Mutual Inductance Z1-Z0	Implement a three-phase RL impedance with mutual coupling between phases and allow specification in the form of positive- and zero-sequence parameters
Three-Phase Parallel RLC Branch	Implement a three-phase parallel RLC branch
Three-Phase Parallel RLC Load	Implement a three-phase parallel RLC load with selectable connection
Three-Phase PI Section Line	Implement a three-phase transmission line section with lumped parameters
Three-Phase Series RLC Branch	Implement a three-phase series RLC branch
Three-Phase Series RLC Load	Implement a three-phase series RLC load with selectable connection
Three-Phase Transformer 12 Terminals	Implement three single-phase, two-winding transformers where all terminals are accessible
Three-Phase Transformer (Two Windings)	Implement a three-phase transformer with two windings
Three-Phase Transformer (Three Windings)	Implement a three-phase transformer with three windings
Zigzag Phase-Shifting Transformer	Implement a zigzag phase-shifting transformer with secondary winding connection

Modeling with Phasor Elements

Static Var Compensator	Implement a phasor model of a three-phase, three-wire static var compensator

Modeling Power Electronics Components

Diode	Implement a diode model
GTO	Implement a gate-turn-off (GTO) thyristor model
Ideal Switch	Implement an ideal switch model
IGBT	Implement an insulated-gate-bipolar-transformer (IGBT) model
MOSFET	Implement a metal-oxide-semiconductor-field-effect-transistor (MOSFET) model
Three-Level Bridge	Implement a three-level neutral point clamped (NPC) power converter
Thyristor	Implement a thyristor model
Universal Bridge	Implement a universal three-phase bridge converter

Modeling Electrical Machines

Asynchronous Machine	Model the dynamics of a three-phase asynchronous machine (induction machine)
DC Machine	Model a separately excited DC machine.
Excitation System	Provide an excitation system for the synchronous machine and regulate its terminal voltage in generating mode
Generic Power System Stabilizer	Provide a generic power system stabilizer for the synchronous machine and regulate its electrical power
Hydraulic Turbine and Governor	Model a hydraulic turbine and a proportional-integral-derivative governor system
Machine Measurement Demux	Split machine measurement signal into separate signals
Multiband Power System Stabilizer	Implement a multiband power system stabilizer

Permanent Magnet Synchronous Machine	Model the dynamics of a three-phase permanent magnet synchronous machine with sinusoidal flux distribution
Simplified Synchronous Machine	Model the dynamics of a simplified three-phase synchronous machine
Steam Turbine and Governor	Implement a steam turbine and governor system
Synchronous Machine	Model the dynamics of a three-phase round-rotor or salient-pole synchronous machine

Measuring Electrical Circuits

Current Measurement	Measure a current in a circuit
Impedance Measurement	Measure the impedance in a circuit as a function of the frequency
Multimeter	Measure voltage and current in SimPowerSystems blocks
Three-Phase V-I Measurement	Measure three-phase currents and voltages in a circuit
Voltage Measurement	Measure a voltage in a circuit

Analyzing Electrical Circuits

Powergui	Graphical user interface for the analysis of circuits and systems

Additional Useful Blocks

Signal Measurements

abc_to_dq0 Transformation	Perform a Park transformation from the three-phase (abc) reference frame to the dq0 reference frame
Active & Reactive Power	Measure the active and reactive powers of a voltage-current pair
dq0_to_abc Transformation	Perform a Park transformation from the dq0 reference frame to the three-phase (abc) reference frame
Fourier	Fourier analyze a signal
RMS	Measure the root mean square (RMS) value of a signal

Three-Phase Sequence Analyzer	Measure the positive-, negative-, and zero-sequence components of a three-phase signal
Total Harmonic Distortion	Measure the total harmonic distortion of a voltage or current signal containing harmonics

Signal and Pulse Sources

PWM Generator	Generate pulses for a carried-based Pulse Width Modulator (PWM)
Synchronized 6-Pulse Generator	Implement a synchronized pulse generator to fire the thyristors of a six-pulse converter
Synchronized 12-Pulse Generator	Implement a synchronized pulse generator to fire the thyristors of a twelve-pulse converter
Timer	Generate a signal changing at specified transition times

Blocks – Alphabetical List

abc_to_dq0 Transformation

Purpose

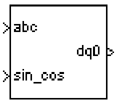
Perform a Park transformation from the three-phase (abc) reference frame to the dq0 reference frame

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description



The abc_to_dq0 Transformation block computes the direct axis, quadratic axis, and zero sequence quantities in a two-axis rotating reference frame for a three-phase sinusoidal signal. The following transformation is used:

$$V_d = \frac{2}{3}(V_a \sin(\omega t) + V_b \sin(\omega t - 2\pi/3) + V_c \sin(\omega t + 2\pi/3))$$

$$V_q = \frac{2}{3}(V_a \cos(\omega t) + V_b \cos(\omega t - 2\pi/3) + V_c \cos(\omega t + 2\pi/3))$$

$$V_0 = \frac{1}{3}(V_a + V_b + V_c)$$

where ω = rotation speed (rad/s) of the rotating frame.

The transformation is the same for the case of a three-phase current; you simply replace the V_a, V_b, V_c, V_d, V_q, and V_0 variables with the I_a, I_b, I_c, I_d, I_q, and I_0 variables.

This transformation is commonly used in three-phase electric machine models, where it is known as a Park transformation. It allows you to eliminate time-varying inductances by referring the stator and rotor quantities to a fixed or rotating reference frame. In the case of a synchronous machine, the stator quantities are referred to the rotor. I_d and I_q represent the two DC currents flowing in the two equivalent rotor windings (d winding directly on the same axis as the field winding, and q winding on the quadratic axis), producing the same flux as the stator I_a, I_b, and I_c currents.

You can use this block in a control system to measure the positive-sequence component V_1 of a set of three-phase voltages or currents. The V_d and V_q (or I_d and I_q) then represent the rectangular coordinates of the positive-sequence component.

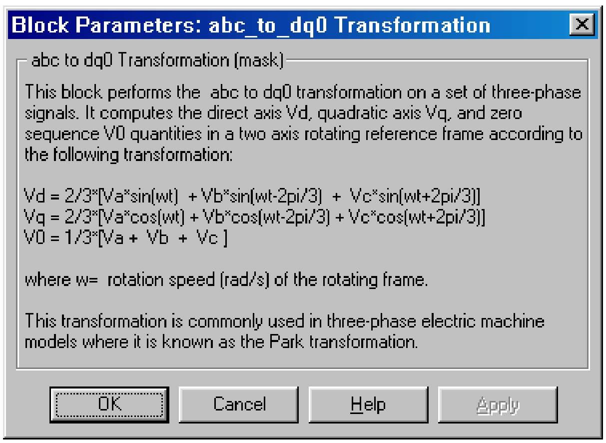
You can use the Math Function block and the Trigonometric Function block to obtain the modulus and angle of V_1:

$$|V_1| = \sqrt{V_q^2 + V_d^2}$$

$$\angle V_1 = \mathrm{atan2}(V_q / V_d)$$

This measurement system does not introduce any delay, but, unlike the Fourier analysis done in the Sequence Analyzer block, it is sensitive to harmonics and imbalances.

Dialog Box and Parameters



Inputs and Outputs

`abc`

Connect to the first input the vectorized sinusoidal phase signal to be converted [phase A phase B phase C].

`sin_cos`

Connect to the second input a vectorized signal containing the [sin(ωt) cos(ωt)] values, where ω is the rotation speed of the reference frame.

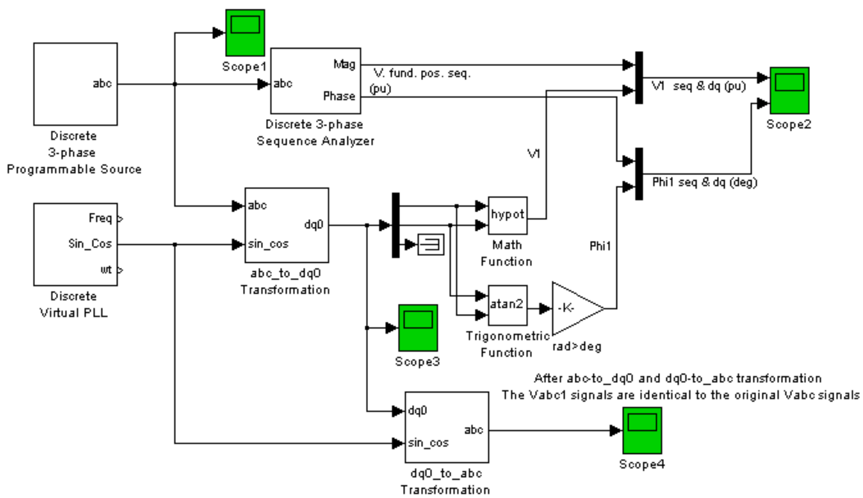
`dq0`

The output is a vectorized signal containing the three sequence components [d q o].

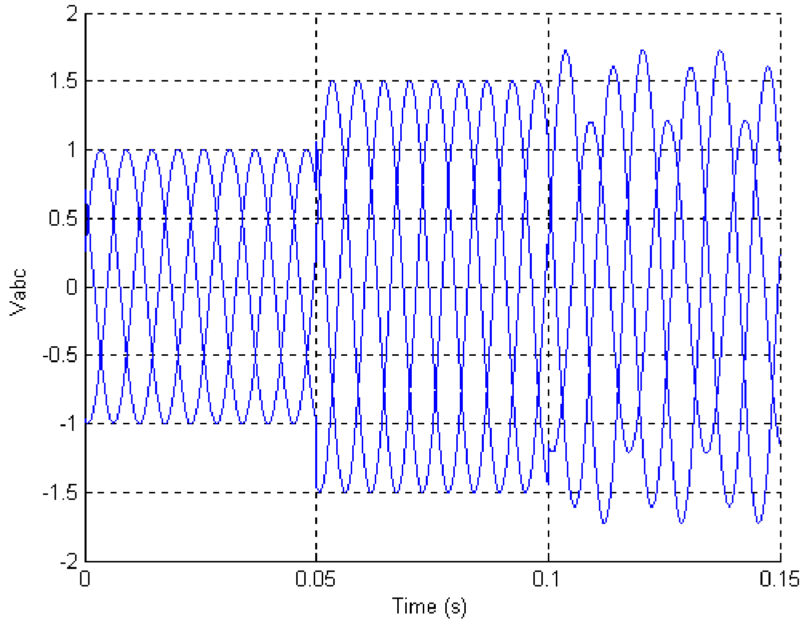
Example

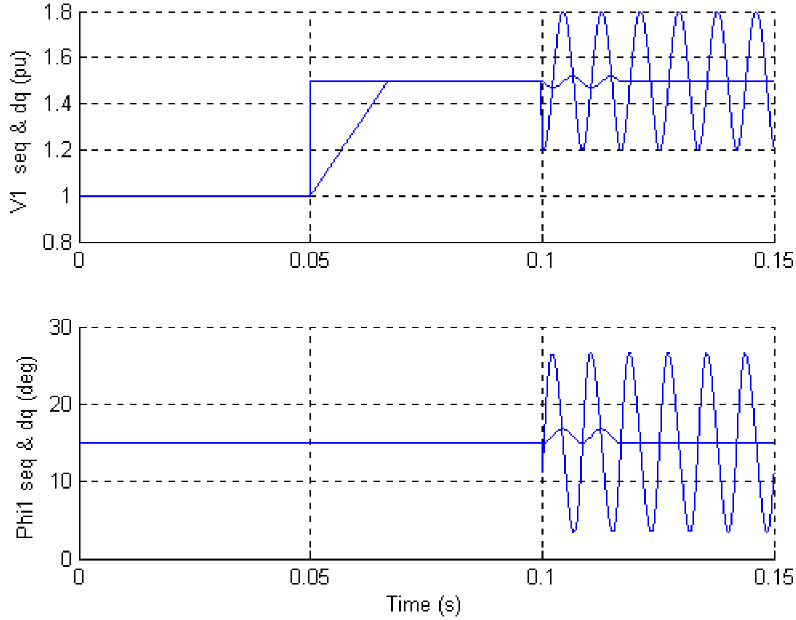
The `power_3phsignaldq` demo uses a Discrete Three-Phase Programmable Source block to generate a 1 p.u., 15 degrees positive sequence voltage. At 0.05 second the positive sequence voltage is increased to 1.5 p.u. and at 0.1 second an imbalance is introduced by the addition of a 0.3 p.u. negative sequence component with a phase of –30 degrees. The magnitude and phase of the positive-sequence component are evaluated in two different ways:

- Sequence calculation of phasors using Fourier analysis
- abc-to-dq0 transformation

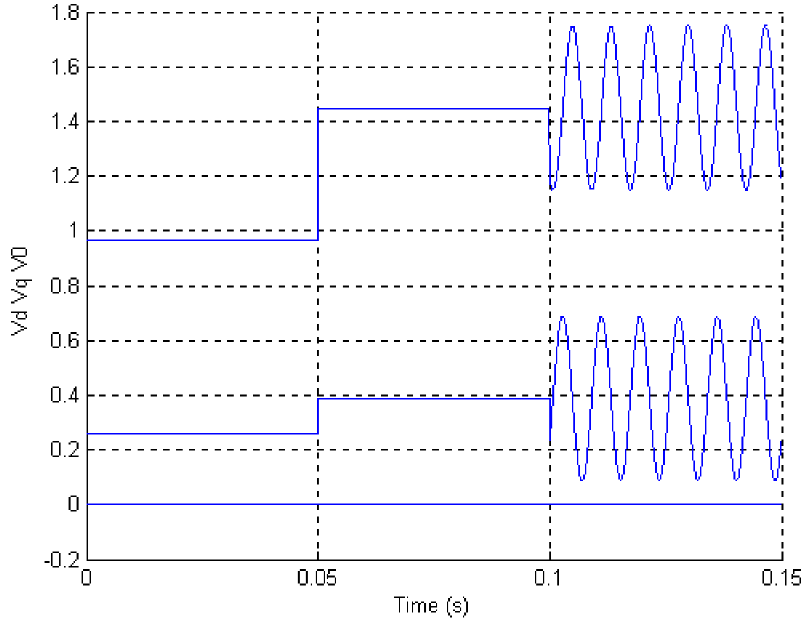


Start the simulation and observe the instantaneous signals Vabc (Scope1), the signals returned by the Sequence Analyzer (Scope2), and the abc-to-dq0 transformation (Scope3).





Note that the Sequence Analyzer, which uses Fourier analysis, is immune to harmonics and imbalance. However, its response to a step is a one-cycle ramp. The abc-to-dqo transformation is instantaneous. However, an imbalance produces a ripple at the V1 and Phi1 outputs.



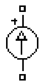
See Also dq0_to_abc Transformation

AC Current Source

Purpose Implement a sinusoidal current source

Library Electrical Sources

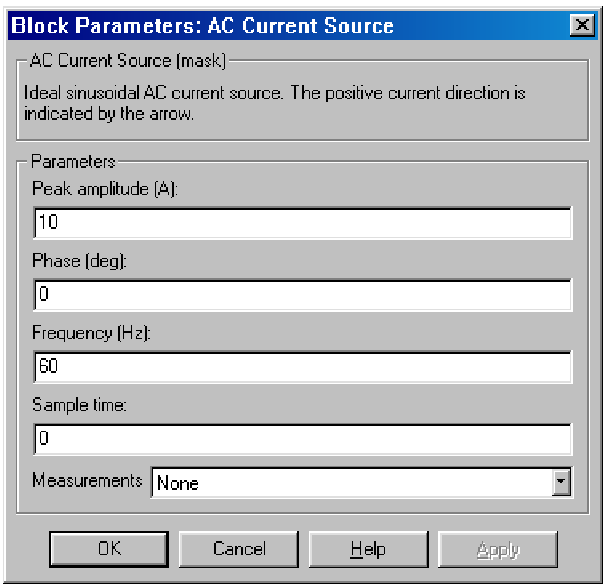
Description The AC Current Source block implements an ideal AC current source. The positive current direction is indicated by the arrow in the block icon. The generated current I is described by the following relationship:



$$I = A\sin(\omega t + \phi) \qquad \omega = 2\pi f \qquad \phi = \text{Phase in radians}$$

Negative values are allowed for amplitude and phase. A zero frequency specifies a DC current source. You cannot enter a negative frequency; Simulink returns an error in that case, and the block displays a question mark in the block icon. You can modify the first three block parameters at any time during the simulation.

Dialog Box and Parameters



Peak amplitude

The peak amplitude of the generated current, in amperes (A).

Phase

The phase in degrees (deg).

Frequency

The source frequency in hertz (Hz).

Sample time

The sample period in seconds (s). The default is `0`, corresponding to a continuous source.

Measurements

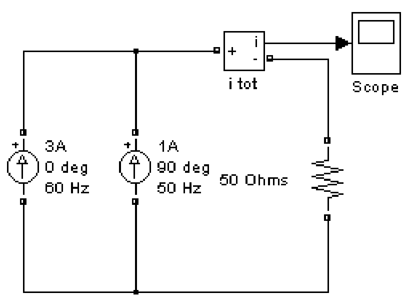
Select `Current` to measure the current flowing through the AC Current Source block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Current	`Isrc:`

Example

The `power_accurrent` demo uses two AC Current Source blocks in parallel to sum two sinusoidal currents in a resistor.



See Also

Controlled Current Source, Multimeter

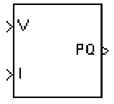
Active & Reactive Power

Purpose Measure the active and reactive powers of a voltage-current pair

Library Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description The Active & Reactive Power block measures the active power P and reactive power Q associated with a periodic voltage-current pair that can contain harmonics. P and Q are calculated by averaging the V I product with a running average window over one cycle of the fundamental frequency, so that the powers are evaluated at fundamental frequency.



$$P = \frac{1}{T}\int_{(t-T)}^{t} (V(\omega t) \times I(\omega t))dt$$

$$Q = \frac{1}{T}\int_{(t-T)}^{t} (V(\omega t) \times I(\omega t - \pi/2))dt$$

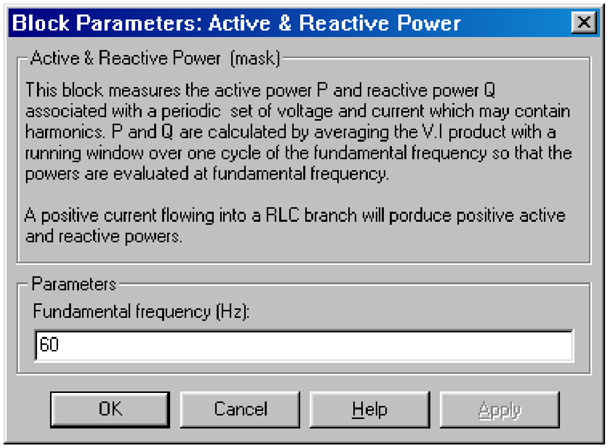
where T = 1/(fundamental frequency).

A current flowing into an RL branch, for example, produces positive active and reactive powers.

As this block uses a running window, one cycle of simulation has to be completed before the output gives the correct active and reactive powers.

The discrete version of this block, available in the Extras/Discrete Measurements library, allows you to specify the initial input voltage and current (magnitude and phase). For the first cycle of simulation the outputs are held constant using the values specified by the initial input parameters.

Dialog Box and Parameters



Fundamental frequency (Hz)

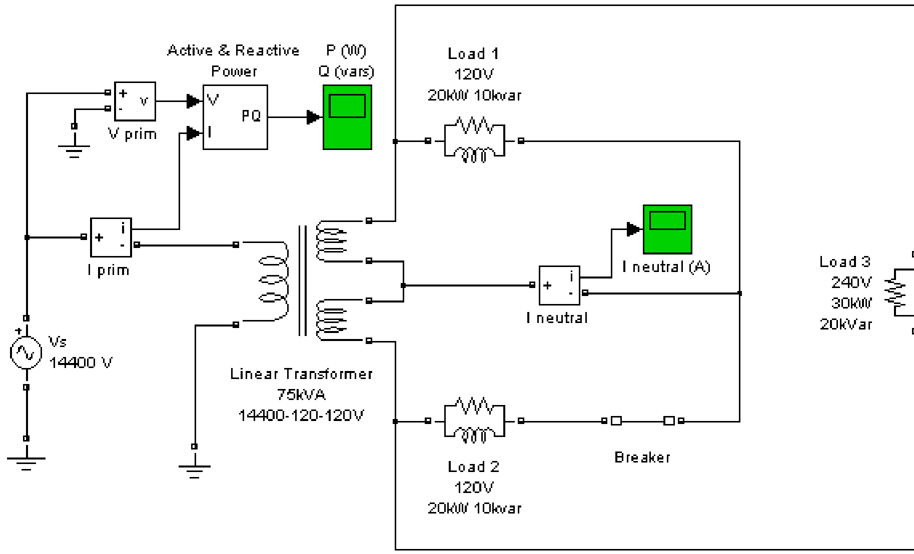
The fundamental frequency, in hertz, of the instantaneous voltage and current.

Inputs and Outputs

The first input is the instantaneous voltage. The second input is the instantaneous current. The output is a vector [P Q] of the active and reactive powers.

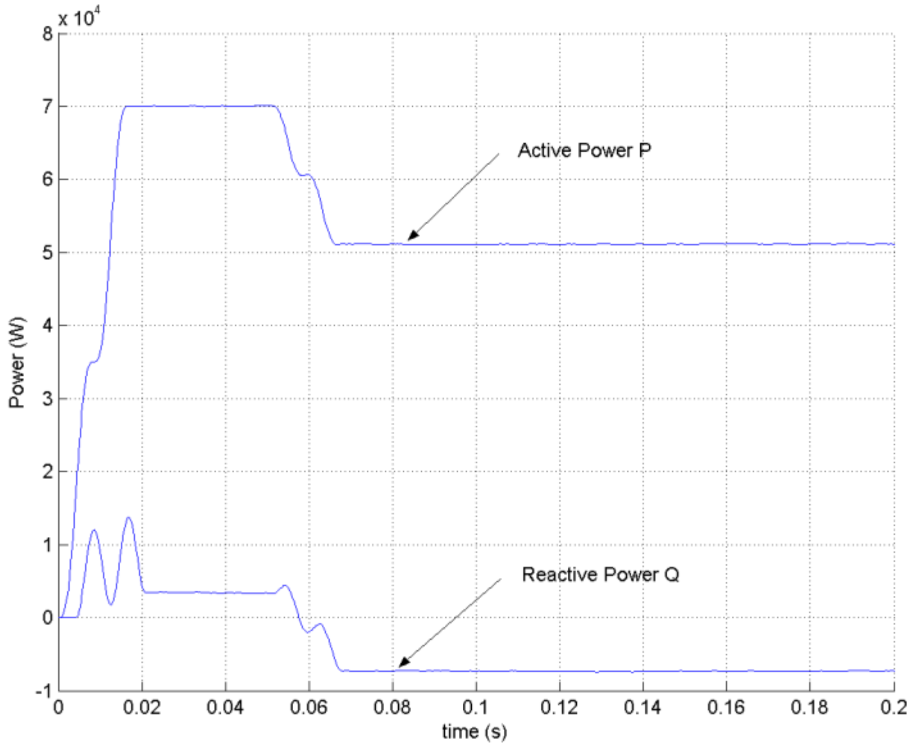
Example

The `power_transfo` demo simulates a three-winding distribution transformer rated at 75 kVA:14400/120/120 V. The transformer primary winding is connected to a high-voltage source of 14400 Vrms. Two identical inductive loads (20 kW-10 kvar) are connected to the two secondary windings. A third capacitive load (30 kW-20 kvar) is fed at 240 V.



Initially, the circuit breaker in series with Load 2 is closed, so that the system is balanced. When the circuit breaker opens, a current starts to flow in the neutral path as a result of the load imbalance.

The active power computed from the primary voltage and current is measured by an Active & Reactive Power block. When the breaker opens, the active power decreases from 70 kW to 50 kW.

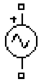

x 10^4
8
7
6
5
4
3
2
1
0
-1
Power (W)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
time (s)
Active Power P
Reactive Power Q

AC Voltage Source

Purpose Implement a sinusoidal voltage source

Library Electrical Sources

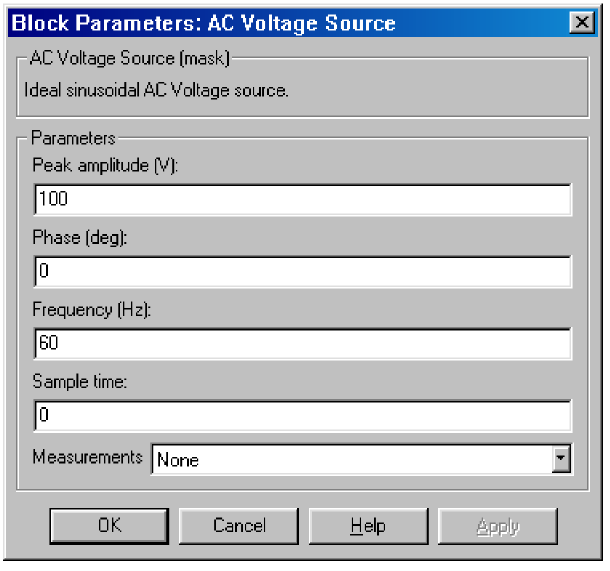
Description The AC Voltage Source block implements an ideal AC voltage source. The generated voltage U is described by the following relationship:



$$U = A\sin(\omega t + \phi) \qquad \omega = 2\pi f \qquad \phi = \text{Phase in radians}$$

Negative values are allowed for amplitude and phase. A 0 frequency specifies a DC voltage source. Negative frequency is not allowed; otherwise Simulink signals an error, and the block displays a question mark in the block icon. You can modify the first three block parameters at any time during the simulation.

Dialog Box and Parameters



Peak amplitude

The peak amplitude of the generated voltage, in volts (V).

Phase

The phase in degrees (deg).

Frequency

The source frequency in hertz (Hz).

Sample time

The sample period in seconds (s). The default is `0`, corresponding to a continuous source.

Measurements

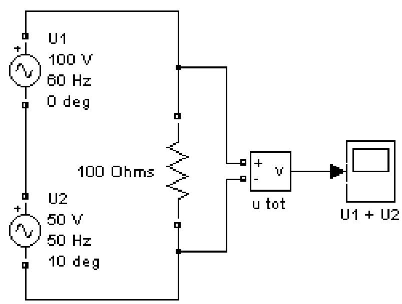
Select `Voltage` to measure the voltage across the terminals of the AC Voltage Source block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Voltage	`Usrc:`

Example

The `power_acvoltage` demo uses two AC Voltage Source blocks at different frequencies connected in series across a resistor. The sum of the two voltages is read by a Voltage Measurement block.



See Also

Controlled Voltage Source, DC Voltage Source, Multimeter

Asynchronous Machine

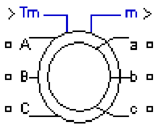
Purpose

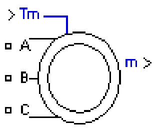
Model the dynamics of a three-phase asynchronous machine, also known as an induction machine

Library

Machines

Description

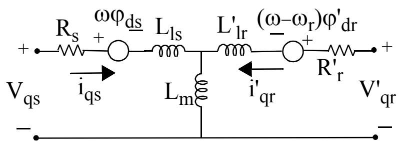




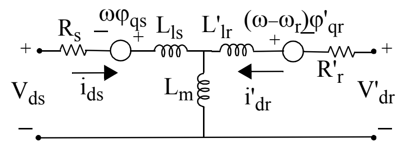
The Asynchronous Machine block operates in either generator or motor mode. The mode of operation is dictated by the sign of the mechanical torque (positive for motoring, negative for generating). The electrical part of the machine is represented by a fourth-order state-space model and the mechanical part by a second-order system. All electrical variables and parameters are referred to the stator. This is indicated by the prime signs in the machine equations given below. All stator and rotor quantities are in the arbitrary two-axis reference frame (dq frame). The subscripts used are defined as follows:

Subscript	Definition
d	d axis quantity
q	q axis quantity
r	Rotor quantity
s	Stator quantity
l	Leakage inductance
m	Magnetizing inductance

Electrical System



q axis



d axis

$$V_{qs} = R_s i_{qs} + \frac{d}{dt}\varphi_{qs} + \omega\varphi_{ds}$$

$$V_{ds} = R_s i_{ds} + \frac{d}{dt}\varphi_{ds} - \omega\varphi_{qs}$$

$$V'_{qr} = R'_r i'_{qr} + \frac{d}{dt}\varphi'_{qr} + (\omega - \omega_r)\varphi'_{dr}$$

$$V'_{dr} = R'_r i'_{dr} + \frac{d}{dt}\varphi'_{dr} - (\omega - \omega_r)\varphi'_{qr}$$

$$T_e = 1.5p(\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds})$$

where

$$\varphi_{qs} = L_s i_{qs} + L_m i'_{qr}$$

$$\varphi_{ds} = L_s i_{ds} + L_m i'_{dr}$$

$$\varphi'_{qr} = L'_r i'_{qr} + L_m i_{qs}$$

$$\varphi'_{dr} = L'_r i'_{dr} + L_m i_{ds}$$

$$L_s = L_{ls} + L_m$$

$$L'_r = L'_{lr} + L_m$$

Mechanical System

$$\frac{d}{dt}\omega_m = \frac{1}{2H}(T_e - F\omega_m - T_m)$$

$$\frac{d}{dt}\theta_m = \omega_m$$

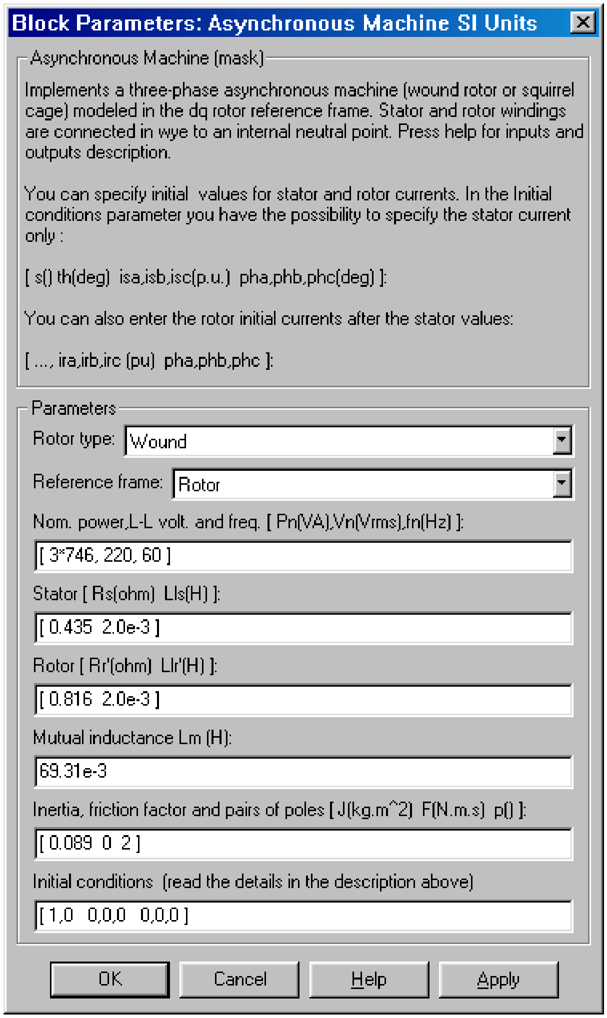
The Asynchronous Machine block parameters are defined as follows (all quantities are referred to the stator):

Parameter	Definition
R_s, L_{ls}	Stator resistance and leakage inductance
R'_r, L'_{lr}	Rotor resistance and leakage inductance
L_m	Magnetizing inductance

Parameter (Continued)	Definition (Continued)
L_s, L'_r	Total stator and rotor inductances
V_{qs}, i_{qs}	q axis stator voltage and current
V'_{qr}, i'_{qr}	q axis rotor voltage and current
V_{ds}, i_{ds}	d axis stator voltage and current
V'_{dr}, i'_{dr}	d axis rotor voltage and current
φ_{qs}, φ_{ds}	Stator q and d axis fluxes
φ'_{qr}, φ'_{dr}	Rotor q and d axis fluxes
ω_m	Angular velocity of the rotor
θ_m	Rotor angular position
p	Number of pole pairs
ω_r	Electrical angular velocity (ω_m x p)
θ_r	Electrical rotor angular position (θ_m x p)
T_e	Electromagnetic torque
T_m	Shaft mechanical torque
J	Combined rotor and load inertia coefficient. Set to infinite to simulate locked rotor.
H	Combined rotor and load inertia constant. Set to infinite to simulate locked rotor.
F	Combined rotor and load viscous friction coefficient

Dialog Boxes and Parameters

You can choose between two Asynchronous Machine blocks to specify the electrical and mechanical parameters of the model.



Rotor type

Specifies the branching for the rotor windings.

Reference frame

Specifies the reference frame that is used to convert input voltages (abc reference frame) to the dq reference frame, and output currents (dq reference frame) to the abc reference frame. You can choose among the following reference frame transformations:

- `Rotor` (Park transformation)
- `Stationary` (Clarke or αβ transformation)
- `Synchronous`

The following relationships describe the abc-to-dq reference frame transformations applied to the Asynchronous Machine phase-to-phase voltages.

$$\begin{bmatrix} V_{qs} \\ V_{ds} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2\cos\theta & \cos\theta + \sqrt{3}\sin\theta \\ 2\sin\theta & \sin\theta - \sqrt{3}\cos\theta \end{bmatrix}\begin{bmatrix} V_{abs} \\ V_{bcs} \end{bmatrix}$$

$$\begin{bmatrix} V'_{qr} \\ V'_{dr} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2\cos\beta & \cos\beta + \sqrt{3}\sin\beta \\ 2\sin\beta & \sin\beta - \sqrt{3}\cos\beta \end{bmatrix}\begin{bmatrix} V'_{abr} \\ V'_{bcr} \end{bmatrix}$$

In the preceding equations, θ is the angular position of the reference frame, while $\beta = \theta - \theta_r$ is the difference between the position of the reference frame and the position (electrical) of the rotor. Because the machine windings are connected in a three-wire Y configuration, there is no homopolar (0) component. This also justifies the fact that two line-to-line input voltages are used inside the model instead of three line-to-neutral voltages. The following relationships describe the dq-to-abc reference frame transformations applied to the Asynchronous Machine phase currents.

$$\begin{bmatrix} i_{as} \\ i_{bs} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \dfrac{-\cos\theta + \sqrt{3}\sin\theta}{2} & \dfrac{-\sqrt{3}\cos\theta - \sin\theta}{2} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix}$$

$$\begin{bmatrix} i'_{ar} \\ i'_{br} \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta \\ \dfrac{-\cos\beta + \sqrt{3}\sin\beta}{2} & \dfrac{-\sqrt{3}\cos\beta - \sin\beta}{2} \end{bmatrix} \begin{bmatrix} i'_{qr} \\ i'_{dr} \end{bmatrix}$$

$$i_{cs} = -i_{as} - i_{bs}$$

$$i'_{cr} = -i'_{ar} - i'_{br}$$

The following table shows the values taken by θ and β in each reference frame (θ_e is the position of the synchronously rotating reference frame).

Reference Frame	θ	β
Rotor	θ_r	0
Stationary	0	$-\theta_r$
Synchronous	θ_e	$\theta_e - \theta_r$

The choice of reference frame affects the waveforms of all dq variables. It also affects the simulation speed and in certain cases the accuracy of the results. The following guidelines are suggested in [1]:

- Use the stationary reference frame if the stator voltages are either unbalanced or discontinuous and the rotor voltages are balanced (or 0).
- Use the rotor reference frame if the rotor voltages are either unbalanced or discontinuous and the stator voltages are balanced.
- Use either the stationary or synchronous reference frames if all voltages are balanced and continuous.

Nominal power, L-L volt, and freq.

The nominal apparent power Pn (VA), RMS line-to-line voltage Vn (V), and frequency fn (Hz).

Stator

The stator resistance Rs (Ω or p.u.) and leakage inductance Lls (H or p.u.).

Rotor

The rotor resistance Rr' (Ω or p.u.) and leakage inductance Llr' (H or p.u.), both referred to the stator.

Mutual inductance

The magnetizing inductance Lm (H or p.u.).

Inertia, friction factor, and pairs of poles

For the **SI units** dialog box: the combined machine and load inertia coefficient J (kg.m^2), combined viscous friction coefficient F (N.m.s), and pole pairs p.

For the **p.u. units** dialog box: the inertia constant H (s), combined viscous friction coefficient F (p.u.), and pole pairs p.

Initial conditions

Specifies the initial slip s, electrical angle θe (degrees), stator current magnitude (A or p.u.), and phase angles (degrees):

[slip, th, i_{as}, i_{bs}, i_{cs}, $phase_{as}$, $phase_{bs}$, $phase_{cs}$]

For the wound-rotor machine, you can also specify optional initial values for the rotor current magnitude (A or p.u.), and phase angles (degrees):

[slip, th, i_{as}, i_{bs}, i_{cs}, $phase_{as}$, $phase_{bs}$, $phase_{cs}$, i_{ar}, i_{br}, i_{cr}, $phase_{ar}$, $phase_{br}$, $phase_{cr}$]

For the squirrel cage machine, the initial conditions can be computed by the load flow utility in the Powergui block.

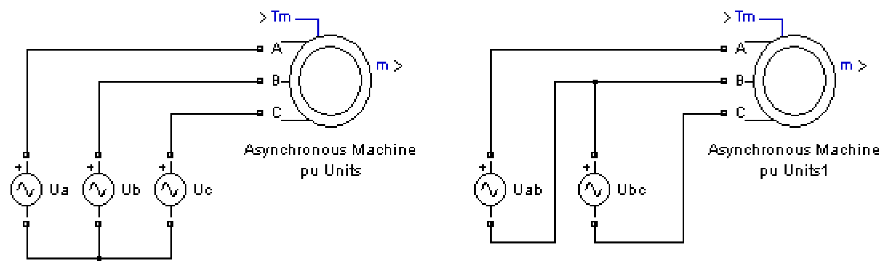
Note Depending on the dialog box you choose to use, SimPowerSystems automatically converts the parameters you enter into per unit parameters. The Simulink model of the Asynchronous Machine block uses p.u. parameters.

Inputs and Outputs

The stator terminals of the Asynchronous Machine block are identified by the A, B, and C letters. The rotor terminals are identified by the a, b, and c letters.

Note that the neutral connections of the stator and rotor windings are not available; three-wire Y connections are assumed.

You must be careful when you connect ideal sources to the machine's stator. If you choose to supply the stator via a three-phase Y-connected infinite voltage source, you must use three sources connected in Y. However, if you choose to simulate a delta source connection, you must only use two sources connected in series.



The Simulink input of the block is the mechanical torque at the machine's shaft. When the input is positive, the asynchronous machine behaves as a motor. When the input is negative, the asynchronous machine behaves as a generator.

The Simulink output of the block is a vector containing 21 signals. They are, in order (refer to the preceding description section, all currents flowing into machine).

Signal	Definition
1 to 3	Rotor currents i'_{ra}, $i'_{rb,}$ and i'_{rc}
4 to 9	i'_{qr}, i'_{dr}, φ'_{qr}, φ'_{dr}, v'_{qr}, and v'_{d}
10 to 12	Stator currents i_{sa}, i_{sb} and i_{sc}
13 to 18	i_{qs}, i_{ds}, φ_{qs}, φ_{ds}, v_{qs}, and v_{ds}
19 to 21	ω_m, T_e, and θ_m

You can demultiplex these signals by using the Machines Measurement Demux block provided in the Machines library.

Limitations

The Asynchronous Machine block does not include a representation of iron losses and saturation.

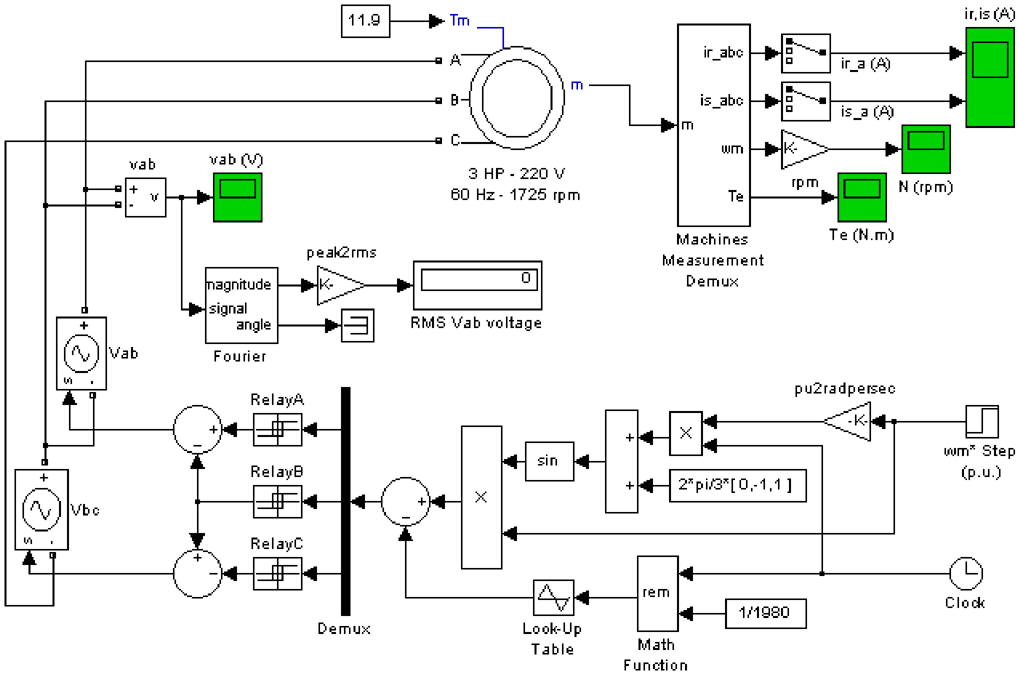
Example

The `power_pwm` demo illustrates the use of the Asynchronous Machine block in motor mode. It consists of an asynchronous machine in an open-loop speed control system.

The machine's rotor is short-circuited, and the stator is fed by a PWM inverter, built with Simulink blocks and interfaced to the Asynchronous Machine block through the Controlled Voltage Source block. The inverter uses sinusoidal pulse-width modulation, which is described in [2]. The base frequency of the sinusoidal reference wave is set at 60 Hz and the triangular carrier wave's frequency is set at 1980 Hz. This corresponds to a frequency modulation factor m_f of 33 (60 Hz x 33 = 1980). It is recommended in [2] that m_f be an odd multiple of three and that the value be as high as possible.

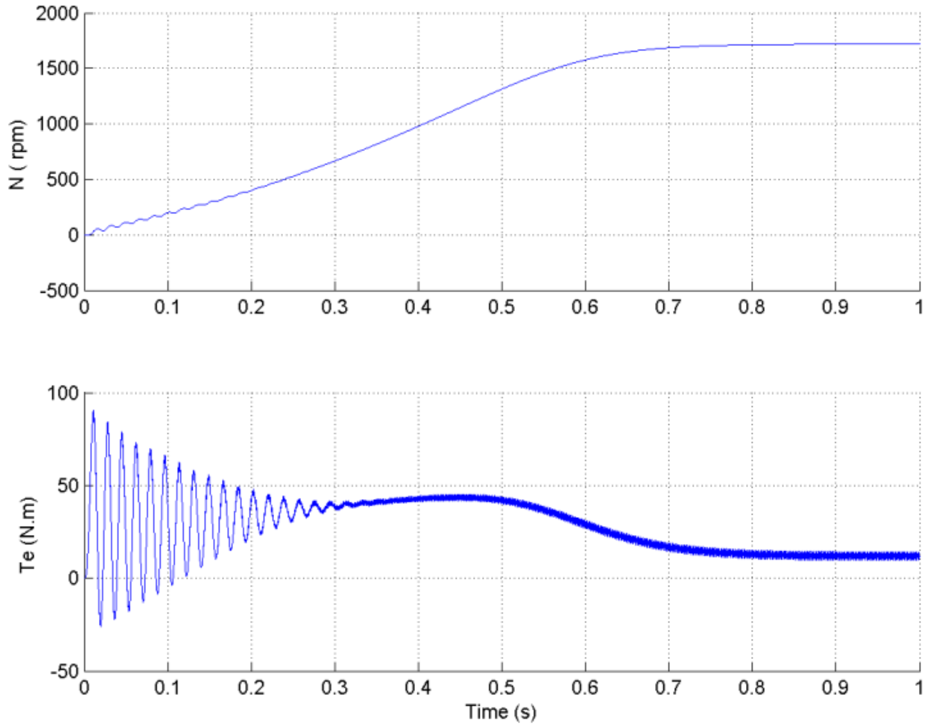
The 3 HP machine is connected to a constant load of nominal value (11.9 N.m). It is started and reaches the set point speed of 1.0 p.u. at t = 0.9 second.

The parameters of the machine are those found in the **SI Units** dialog box above, except for the stator leakage inductance, which is set to twice its normal value. This is done to simulate a smoothing inductor placed between the inverter and the machine. Also, the stationary reference frame was used to obtain the results shown.



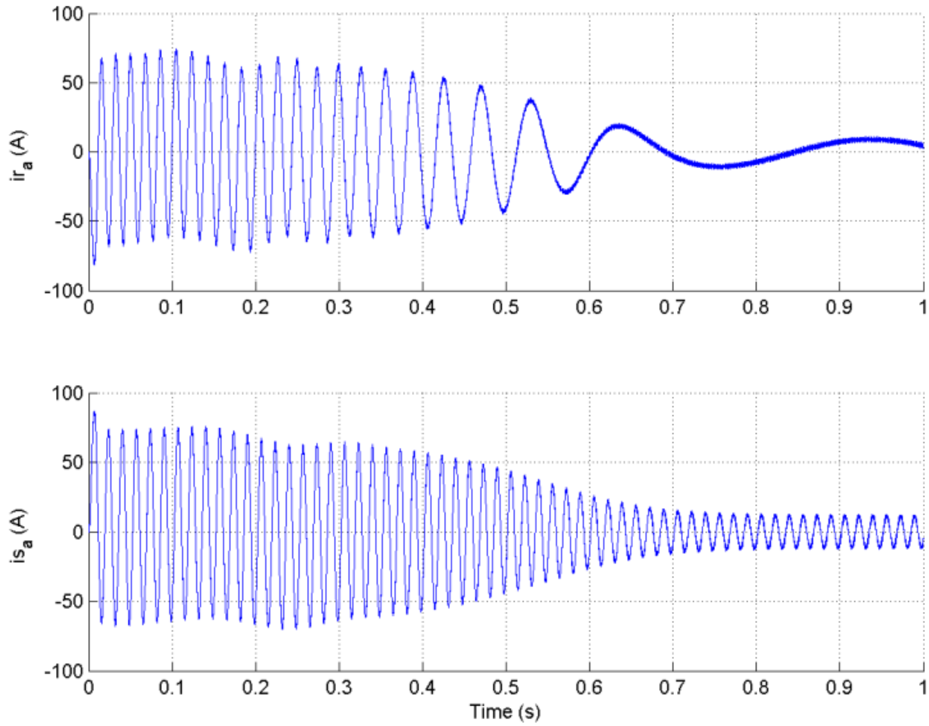
Open the power_pwm demo. Note in the simulation parameters that a small relative tolerance is required because of the high switching rate of the inverter.

Run the simulation and observe the machine's speed and torque.

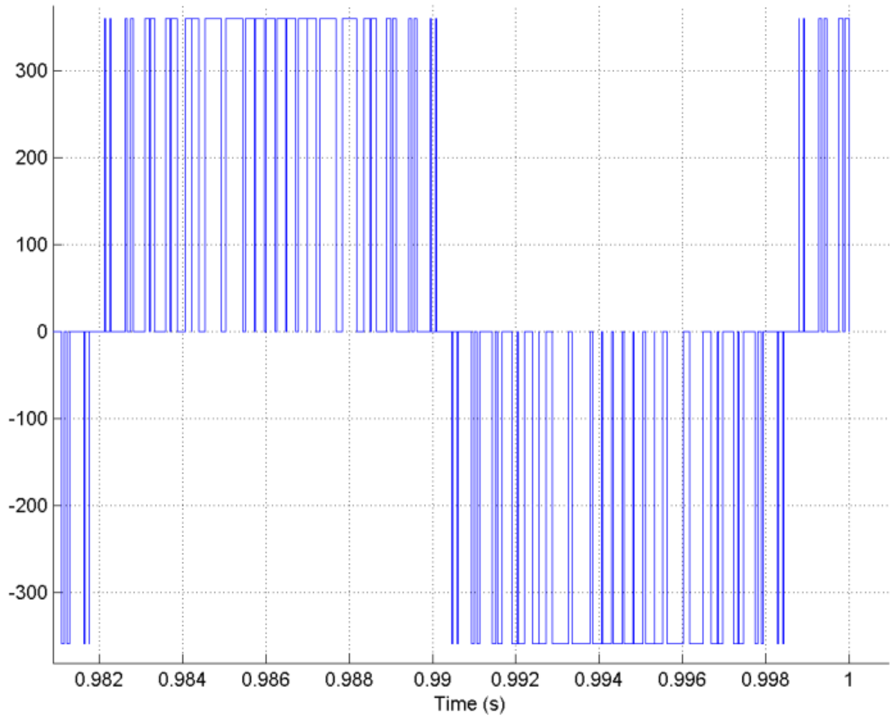


The first graph shows the machine's speed going from 0 to 1725 rpm (1.0 p.u.). The second graph shows the electromagnetic torque developed by the machine. Because the stator is fed by a PWM inverter, a noisy torque is observed.

However, this noise is not visible in the speed because it is filtered out by the machine's inertia, but it can also be seen in the stator and rotor currents, which are observed next.



Finally, look at the output of the PWM inverter. Because nothing of interest can be seen at the simulation time scale, the graph concentrates on the last moments of the simulation.



References

[1] Krause, P.C., O. Wasynczuk, and S.D. Sudhoff, *Analysis of Electric Machinery*, IEEE Press, 1995.

[2] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995, Section 8.4.1.

See Also

Machine Measurement Demux, Powergui

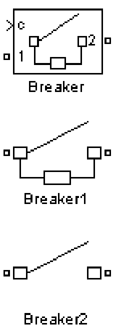
Purpose

Implement a circuit breaker opening at the current zero crossing

Library

Elements

Description



The Breaker block implements a circuit breaker where the opening and closing times can be controlled either from an external Simulink signal (external control mode), or from an internal control timer (internal control mode).

The arc extinction process is simulated by opening the breaker device when the current passes through 0 (first current zero crossing following the transition of the Simulink control input from 1 to 0).

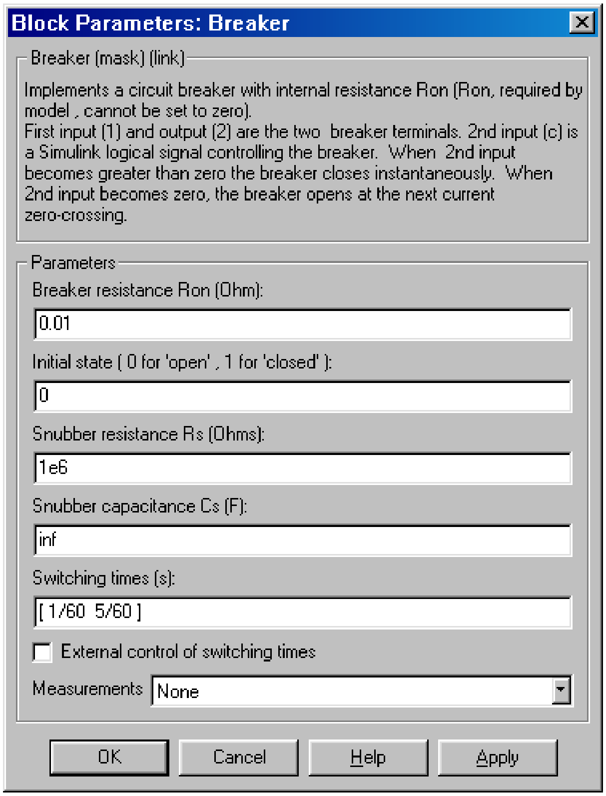
When the breaker is closed it behaves as a resistive circuit. It is represented by a resistance Ron. The Ron value can be set as small as necessary in order to be negligible compared with external components (typical value is 10 mΩ). When the breaker is open it has an infinite resistance.

If the Breaker block is set in external control mode, a Simulink input appears on the block icon. The control signal connected to the Simulink input must be either 0 or 1: 0 to open the breaker, 1 to close it. If the Breaker block is set in internal control mode, the switching times are specified in the dialog box of the block.

If the breaker initial state is set to 1 (closed), SimPowerSystems automatically initializes all the states of the linear circuit and the Breaker block initial current so that the simulation starts in steady state.

A series Rs-Cs snubber circuit is included in the model. It can be connected to the circuit breaker. If the Breaker block happens to be in series with an inductive circuit, an open circuit or a current source, you must use a snubber.

Dialog Box and Parameters



Breaker resistance Ron

The internal breaker resistance, in ohms (Ω). The **Breaker resistance Ron** parameter cannot be set to 0.

Initial state

The initial state of the breaker. A closed contact is displayed in the block icon when the **Initial state** parameter is set to `1`, and an open contact is displayed when it is set to `0`.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance, in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Switching times

Specifies the vector of switching times when using the Breaker block in internal control mode. At each switching time the Breaker block opens or closes depending on its initial state. For example, if the **Initial state** parameter is `0` (open), the breaker closes at the first switching time, opens at the second switching time, and so on. The **Switching times** parameter is not visible in the dialog box if the **External control of switching times** parameter is selected.

External control of switching times

If selected, adds a Simulink input to the Breaker block for external control of the switching times of the breaker. The switching times are defined by a logical signal (`0` or `1`) connected to the Simulink input.

Measurements

Select `Branch voltage` to measure the voltage across the Breaker block terminals.

Select `Branch current` to measure the current flowing through the Breaker block. If the snubber device is connected to the breaker model, the measured current is the one flowing through the breaker contacts only.

Select `Branch voltage and current` to measure the breaker voltage and the breaker current.

Place a Multimeter block in your model to display the selected measurements during the simulation.

In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

Measurement	Label
Branch voltage	`Ub:`
Branch current	`Ib:`

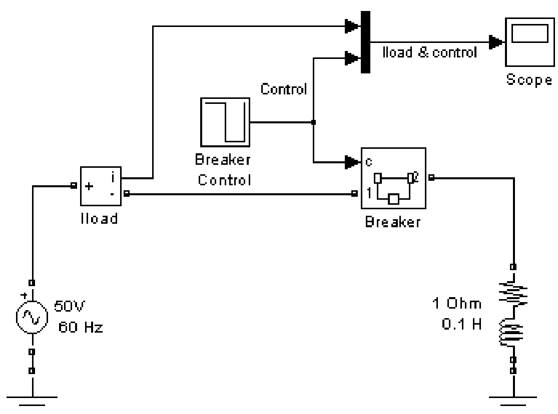
Limitations

When the block is connected in series with an inductor or another current source, you must add the snubber circuit. In most applications you can use a resistive snubber (**Snubber capacitance** parameter set to `inf`) with a large resistor value (**Snubber resistance** parameter set to `1e6` or so). Because of modeling constraints, the internal breaker inductance Ron cannot be set to 0.

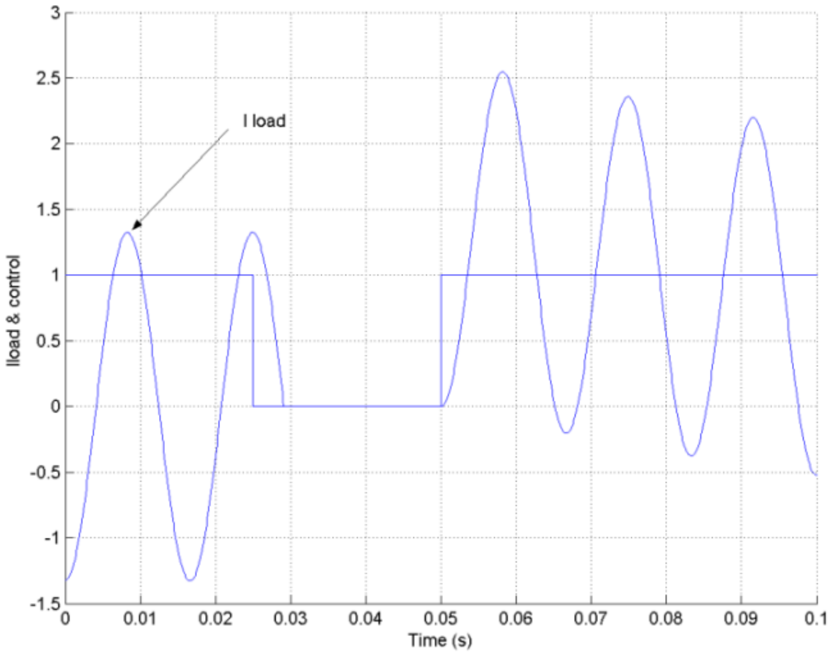
You must use a stiff integration algorithm to simulate circuits with the Breaker block. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

Example

The `power_breaker` demo illustrates a circuit breaker connected in series with a series RL circuit on a 60 Hz voltage source. The switching times of the Breaker block are controlled by a Simulink signal. The breaker device is initially closed and an opening order is given at t = 1.5 cycles, when current reaches a maximum. The current stops at the next zero crossing, then the breaker is reclosed at a zero crossing of voltage at t = 3 cycles.



Simulation produces the following results.



Note that the breaker device opens only when the load current has reached zero, after the opening order.

See Also

Three-Phase Breaker, Three-Phase Fault

Connection Port

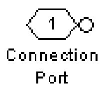
Purpose

Create a Physical Modeling connector port for a subsystem

Library

Elements

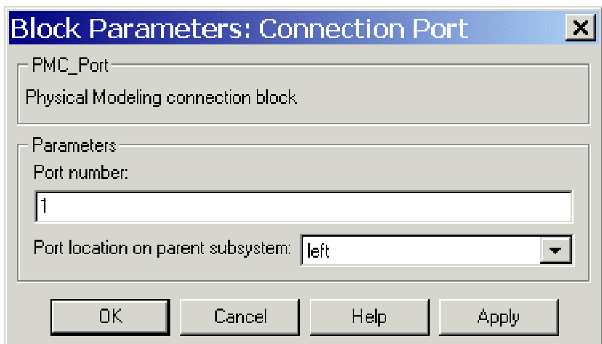
Description



The Connection Port block, placed inside a subsystem composed of SimPowerSystems blocks, creates a Physical Modeling open round connector port ○ on the boundary of the subsystem. Once connected to a connection line, the port becomes solid • . Once you begin the simulation, the solid port • becomes an electrical terminal port, an open square □.

You connect individual SimPowerSystems blocks and subsystems made of SimPowerSystems blocks to one another with SimPowerSystems connection lines, instead of normal Simulink signal lines. These are anchored at the open, round connector ports ○. Subsystems constructed of SimPowerSystems blocks automatically have such open round connector ports. You can add additional connector ports by adding Connection Port blocks to your subsystem.

Dialog Box and Parameters



Port number

This field labels the subsystem connector port created by the block. Multiple connector ports on the boundary of a single subsystem require different numbers as labels. The default value for the first port is `1`.

Port location on parent subsystem

Choose which side of the parent subsystem boundary the port is placed. The choices are `left` or `right`. The default is `left`.

See Also

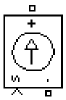
In Simulink, see Creating Subsystems.

Controlled Current Source

Purpose Implement a controlled current source

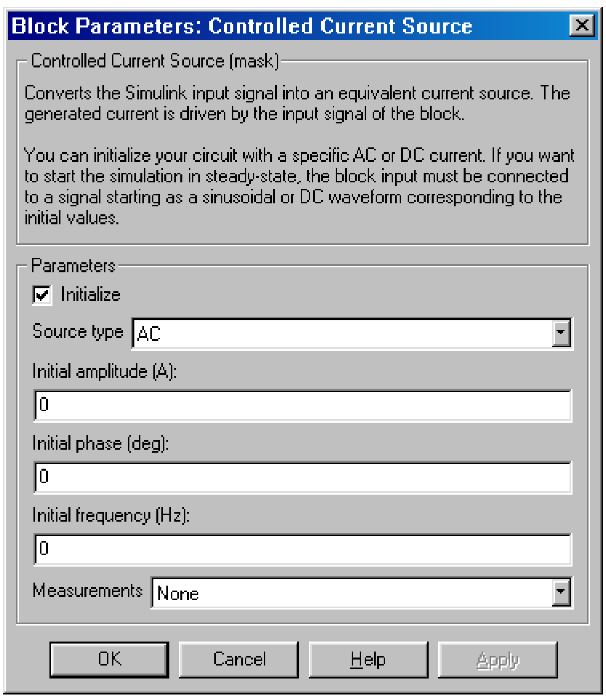
Library Electrical Sources

Description The Controlled Current Source block provides a current source controlled by a Simulink signal. The positive current direction is as shown by the arrow in the block icon.



You can initialize the Controlled Current Source block with a specific AC or DC current. If you want to start the simulation in steady state, the block input must be connected to a signal starting as a sinusoidal or DC waveform corresponding to the initial values.

Dialog Box and Parameters



Initialize

If selected, initializes the Controlled Current Source block with the specified **Initial current**, **Initial phase**, and **Initial frequency** parameters.

Source type

The **Source type** parameter is not visible if the **Initialize** parameter is not selected.

The type of current source. Select `AC` to initialize the Controlled Current Source Block as an AC current source. Select `DC` to initialize the Controlled Current Source block as a DC current.

Initial current

The **Initial current** parameter is not visible in the dialog box if the **Initialize** parameter is not selected. The initial peak current for the initialization of the source, in amperes (A).

Initial phase

The initial phase for the initialization of the source, in degrees. The **Initial phase** parameter is not visible in the dialog box if the **Source type** parameter is set to **DC**.

Initial frequency

The initial frequency for the initialization of the source, in hertz (Hz). The **Initial frequency** parameter is not visible in the dialog box if the **Source type** parameter is set to **DC**.

Measurements

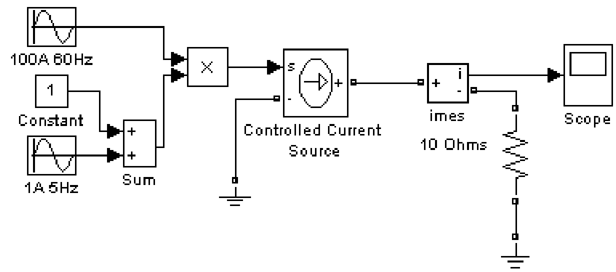
Select `Current` to measure the current flowing through the Controlled Current Source block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

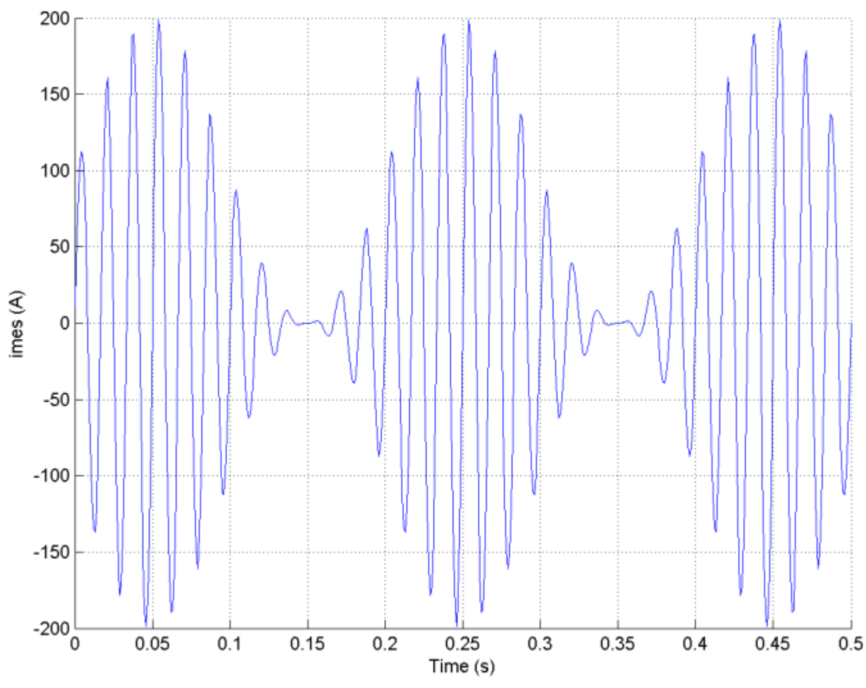
Measurement	Label
Current	`Isrc:`

Example

The power_controlcurr demo uses a Controlled Current Source to generate a 60 Hz current modulated at 5 Hz.



Simulation produces the following waveforms:



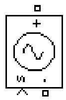
See Also

AC Current Source, Controlled Voltage Source, Multimeter

Purpose Implement a controlled voltage source

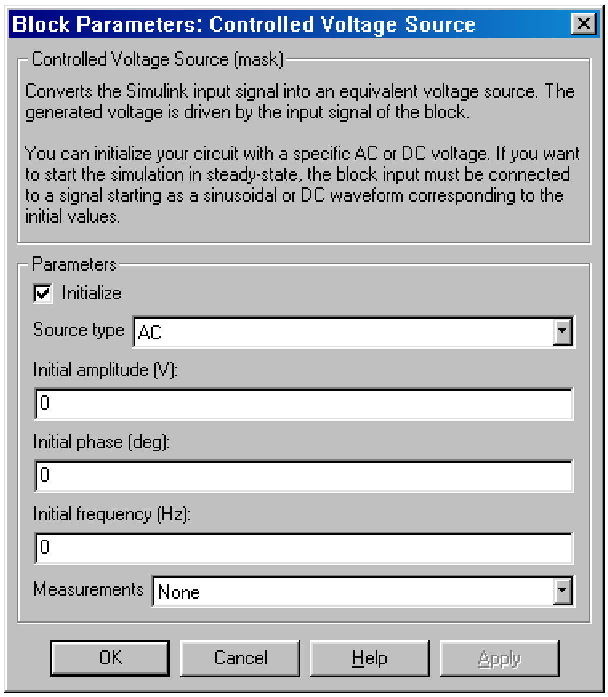
Library Electrical Sources

Description The Controlled Voltage Source block provides a voltage source controlled by a Simulink signal.



You can initialize the Controlled Voltage Source block with a specific AC or DC voltage. If you want to start the simulation in steady state, the Simulink input must be connected to a signal starting as a sinusoidal or DC waveform corresponding to the initial values.

Dialog Box and Parameters



Initialize

If selected, initializes the Controlled Voltage Source block with the specified **Initial voltage**, **Initial phase**, and **Initial frequency** parameters.

Source type

The **Source type** parameter is not available if the **Initialize** parameter is not selected.

The type of voltage source. Select `AC` to initialize the Controlled Voltage Source block with an AC voltage source. Select `DC` to initialize the Controlled Voltage Source Block with a DC voltage.

Initial voltage

The **Initial voltage** parameter is not available if the **Initialize** parameter is not selected. The initial voltage for the initialization of the source, in amperes (A).

Initial phase

The **Initial phase** parameter is not available if the **Source type** parameter is set to **DC**. The initial phase for the initialization of the source, in degrees.

Initial frequency

The initial frequency for the initialization of the source, in hertz (Hz). The **Initial frequency** parameter is not available in the dialog box if the **Source type** parameter is set to **DC**.

Measurements

Select `Voltage` to measure the voltage across the terminals of the Controlled Voltage Source block.

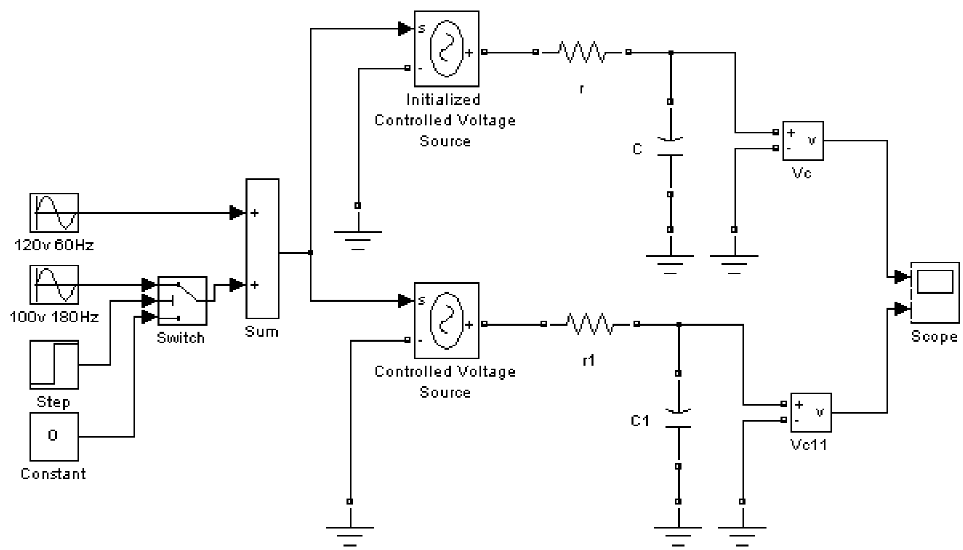
Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

Measurement	Label
Voltage	`Usrc:`

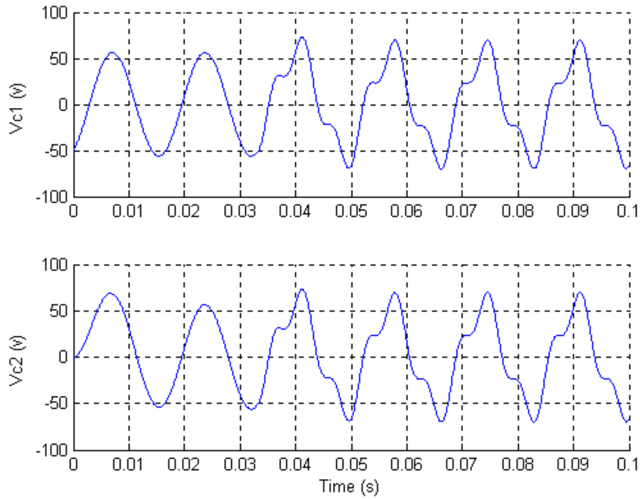
Example

The `power_controlvolt` demo uses Controlled Voltage Source blocks to generate a 60 Hz sinusoidal voltage containing a third harmonic. One Controlled Voltage Source block is initialized as a 120 V AC voltage source with an initial frequency of 60 Hz and initial phase set to 0. The second Controlled Voltage Source block is not initialized.

At t = 0.0333 s a 100 V-180 Hz sinusoidal signal is added to the 120 V Simulink signal. The resulting capacitor voltages are compared on a Scope block.



The Vc voltage starts in steady state, whereas the Vc1 voltage contains a DC offset.



See Also

AC Current Source, Controlled Current Source, Multimeter

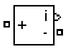
Purpose

Measure a current in a circuit

Library

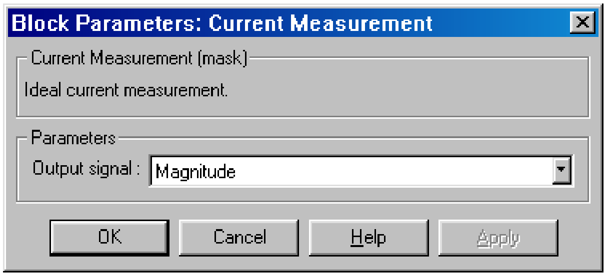
Measurements

Description



The Current Measurement block is used to measure the instantaneous current flowing in any electrical block or connection line. The Simulink output provides a Simulink signal that can be used by other Simulink blocks.

Dialog Box and Parameters



Output signal

Specifies the format of the output signal when the block is used in a phasor simulation. The **Output signal** parameter is disabled when the block is not used in a phasor simulation. The phasor simulation is activated by a Powergui block placed in the model.

Set to `Complex` to output the measured current as a complex value. The output is a complex signal.

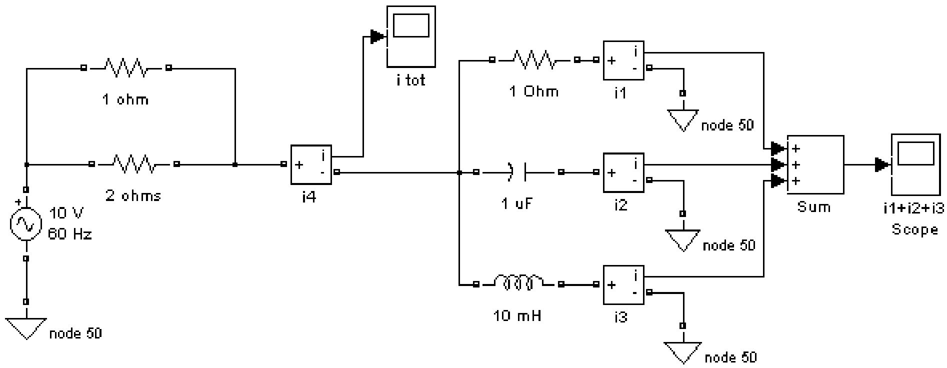
Set to `Real-Imag` to output the real and imaginary parts of the measured current. The output is a vector of two elements.

Set to `Magnitude-Angle` to output the magnitude and angle of the measured current. The output is a vector of two elements.

Set to `Magnitude` to output the magnitude of the measured current. The output is a scalar value.

Example

The `power_currmeasure` demo uses four Current Measurement blocks to read currents in different branches of a circuit. The two scopes display the same current.



See Also

Powergui, Three-Phase V-I Measurement, Voltage Measurement

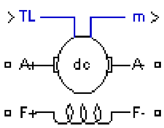
Purpose

Implement a separately excited DC machine

Library

Machines

Description



This block implements a separately excited DC machine. An access is provided to the field terminals (F+, F–) so that the machine model can be used as a shunt-connected or a series-connected DC machine. The torque applied to the shaft is provided at the Simulink input T_L.

The armature circuit (A+, A–) consists of an inductor La and resistor Ra in series with a counter-electromotive force (CEMF) E.

The CEMF is proportional to the machine speed.

$$E = K_E \omega$$

K_E is the voltage constant and ω is the machine speed.

In a separately excited DC machine model, the voltage constant K_E is proportional to the field current I_f:

$$K_E = L_{af} I_f$$

where L_{af} is the field-armature mutual inductance.

The electromechanical torque developed by the DC machine is proportional to the armature current I_a.

$$T_e = K_T I_a$$

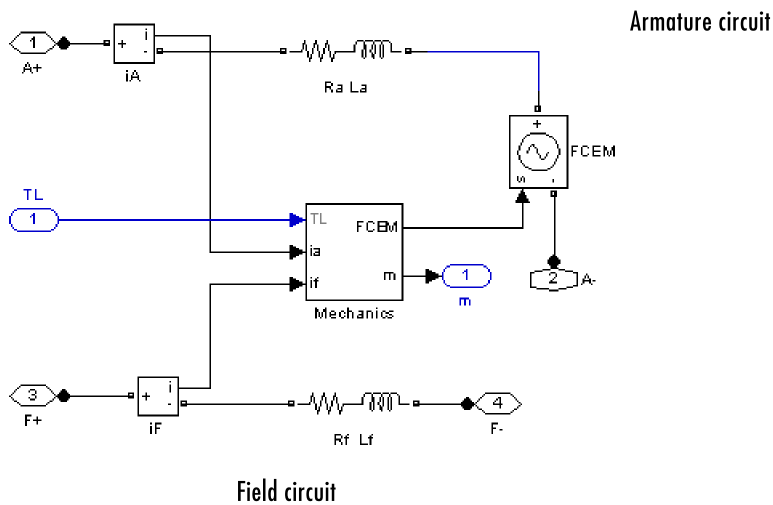
where K_T is the torque constant. The sign convention for T_e and T_L is

$$T_e\ T_L > 0 \quad : \text{Motor mode}$$
$$T_e\ T_L < 0 \quad : \text{Generator mode}$$

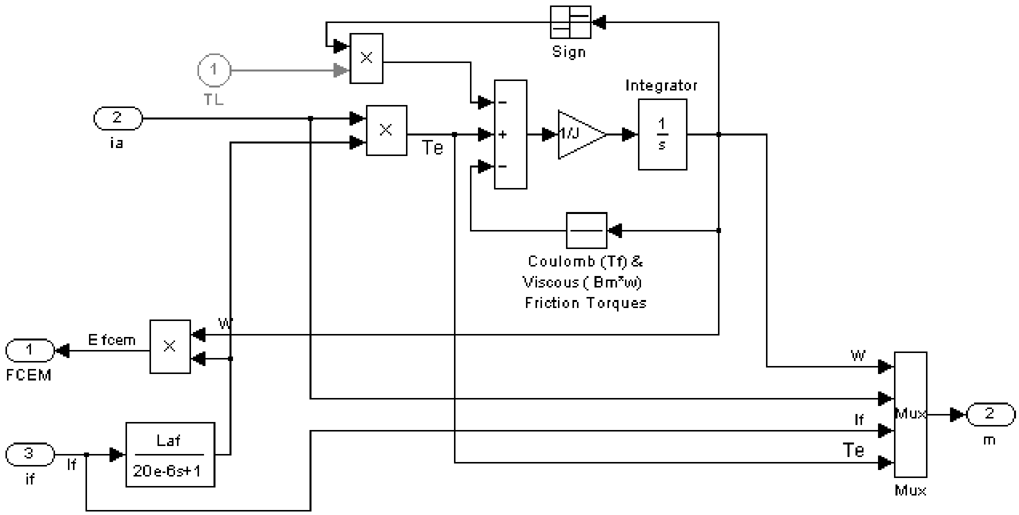
The torque constant is equal to the voltage constant.

$$K_T = K_E$$

The armature circuit is connected between the A+ and A– ports of the DC Machine block. It is represented by a series Ra La branch in series with a Controlled Voltage Source and a Current Measurement block.



Mechanical part:



The field circuit is represented by an RL circuit. It is connected between the F+ and F– ports of the DC Machine block.

The mechanical part computes the speed of the DC machine from the net torque applied to the rotor. The speed is used to implement the CEMF voltage E of the armature circuit.

The mechanical part is represented by Simulink blocks that implement the equation

$$J\frac{d\omega}{dt} = T_e - \mathrm{sgn}(\omega)T_L - B_m\omega - T_f$$

where J = inertia, B_m = viscous friction coefficient, and T_f = Coulomb friction torque.

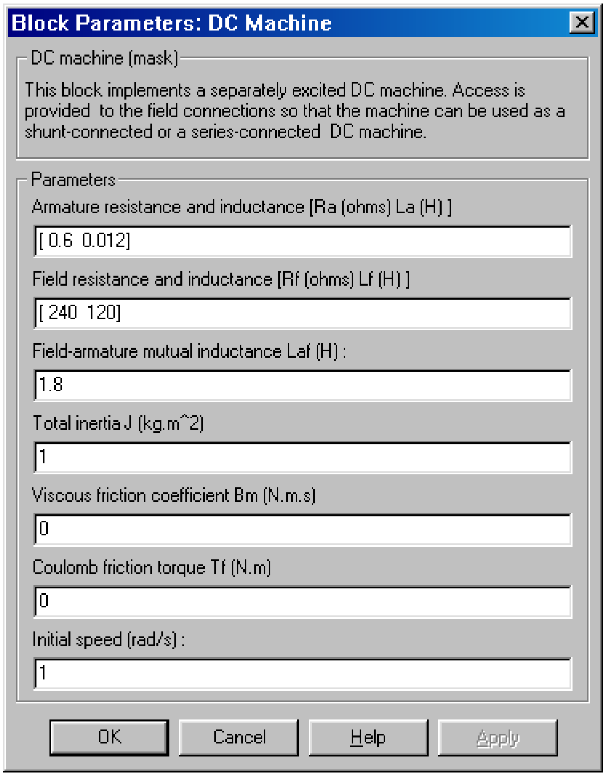
Measurements

Four internal signals are multiplexed on the Simulink measurement output vector returning

- Rotor speed in rad/s

- Armature current in A
- Field current in A
- Electromechanical torque in N.m

Dialog Box and Parameters



Armature resistance and inductance [Ra La]

The armature resistance Ra, in ohms, and the armature inductance La, in henries.

Field resistance and inductance [Rf Lf]

The field resistance Rf, in ohms, and the field inductance Lf, in henries.

Field armature mutual inductance Laf

The field armature mutual inductance, in henries.

Total inertia J

The total inertia of the DC machine, in kg.m^2.

Viscous friction coefficient Bm

The total friction coefficient of the DC machine, in N.m.s.

Coulomb friction torque Tf

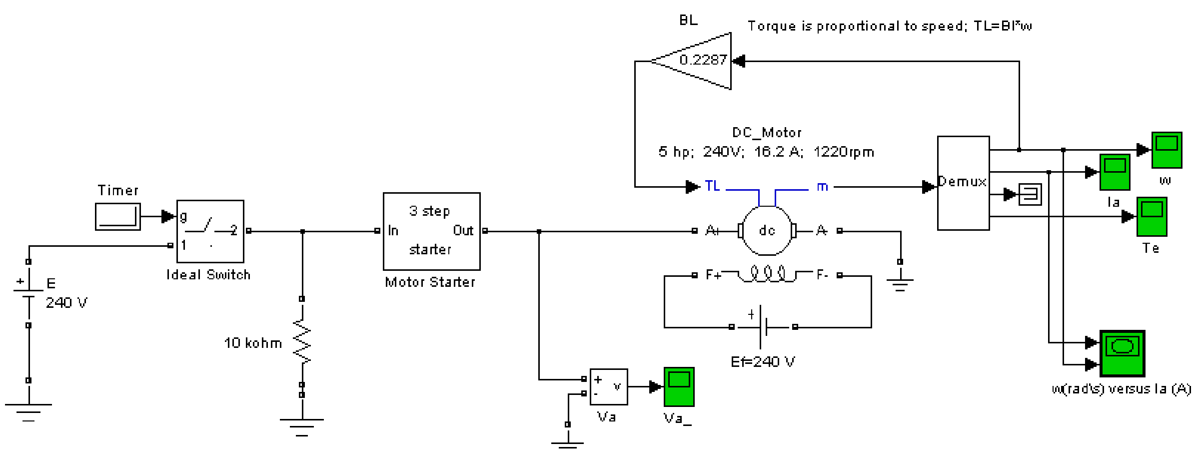
The total Coulomb friction torque constant of the DC machine, in N.m.

Initial speed

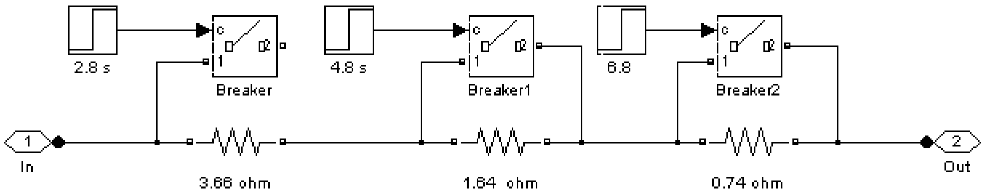
Specifies an initial speed for the DC machine, in rad/s, in order to start the simulation with a specific initial speed. To start the simulation in steady state, the initial value of the input torque signal T_L must be proportional to the initial speed.

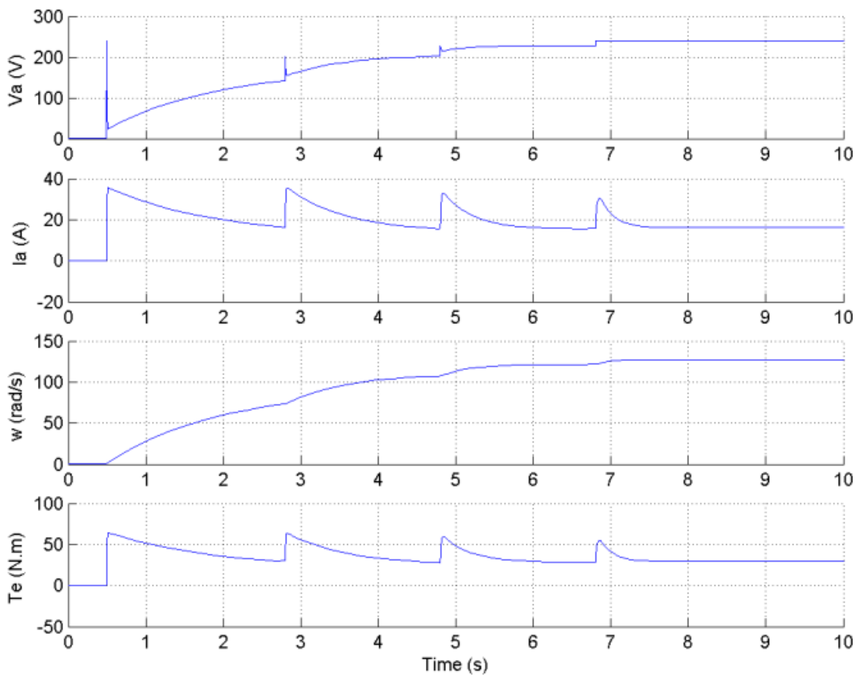
Example

The `power_dcmotor` demo illustrates the starting of a 5 HP 240 V DC machine with a three-step resistance starter.



The Motor Starter subsystem is





References

Analysis of Electric Machinery, Krause et al., pp. 89-92.

See Also

Asynchronous Machine, Synchronous Machine

Purpose

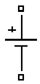
Implement a DC voltage source

Library

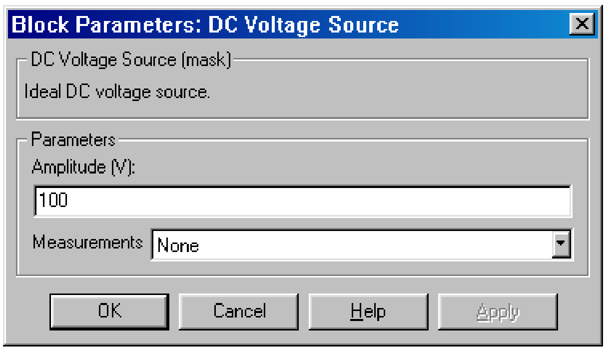
Electrical Sources

Description

The DC Voltage Source block implements an ideal DC voltage source. The positive terminal is represented by a plus sign on one port. You can modify the voltage at any time during the simulation.



Dialog Box and Parameters



Amplitude

The amplitude of the source, in volts (V).

Measurements

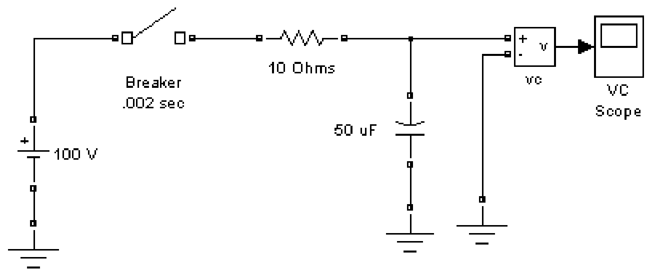
Select `Voltage` to measure the voltage across the terminals of the DC Voltage Source block.

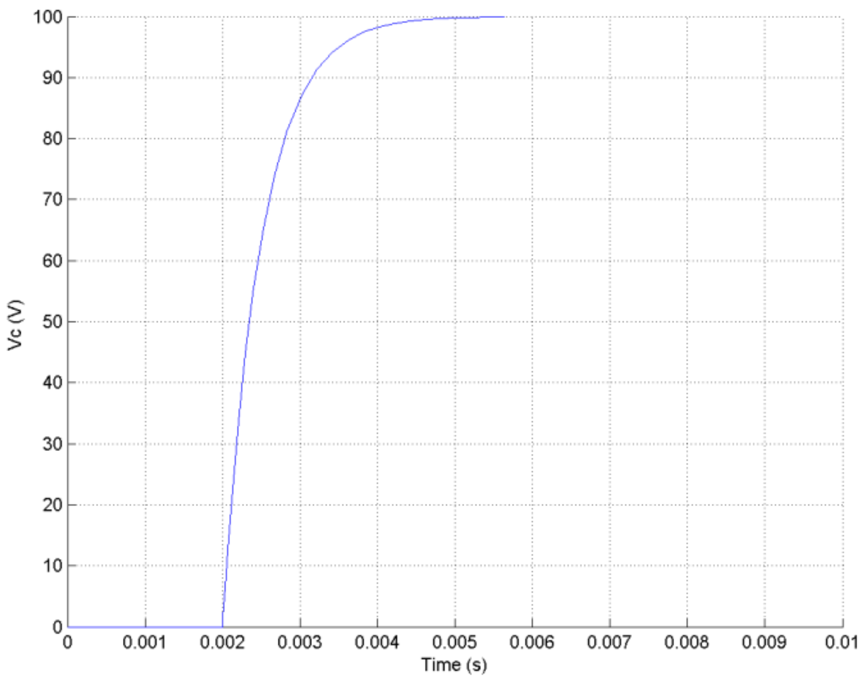
Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

Measurement	Label
Voltage	`Usrc:`

Example

The `power_dcvoltage` demo illustrates the simulation of the transient response of a first-order RC circuit.





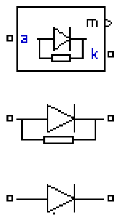
See Also

AC Voltage Source, Controlled Voltage Source

Purpose Implement a diode model

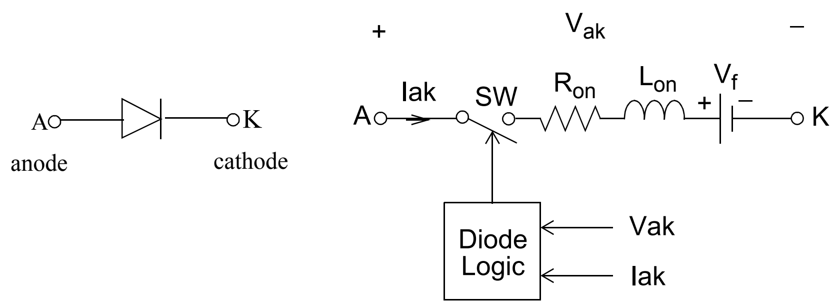
Library Power Electronics

Description



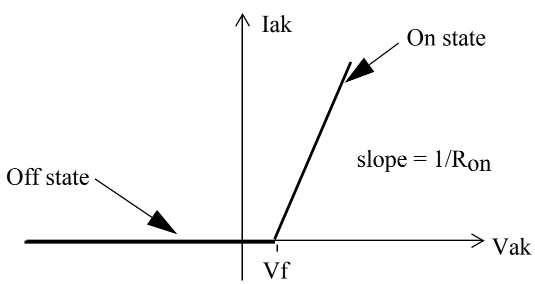
The diode is a semiconductor device that is controlled by its own voltage Vak and current Iak. When a diode is forward biased (Vak>0), it starts to conduct with a small forward voltage Vf across it. It turns off when the current flow into the device becomes 0. When the diode is reverse biased (Vak<0), it stays in the off state.

The Diode block is simulated as a resistor, an inductor, and a DC voltage source connected in series with a switch. The switch is controlled by the voltage Vak and current Iak.

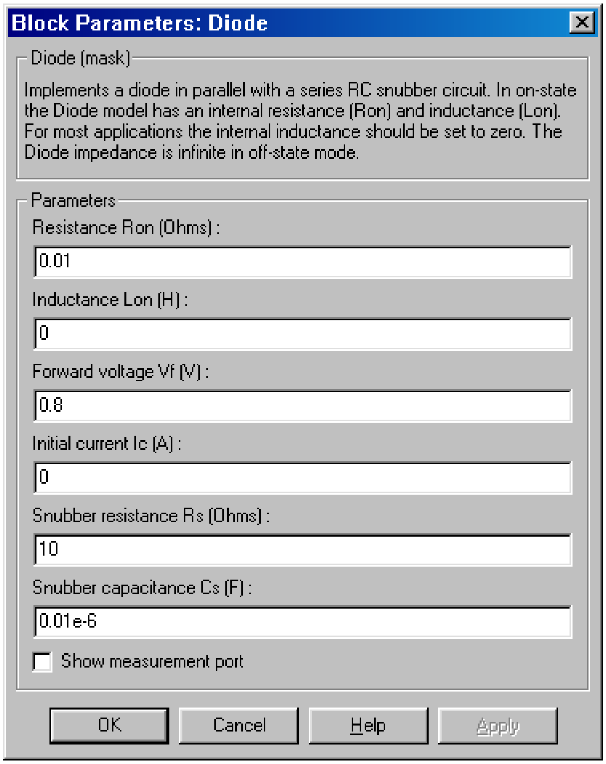


The Diode block also contains a series Rs-Cs snubber circuit that can be connected in parallel with the diode device (between nodes A and K).

The static VI characteristic of this model is shown in the following figure.



Dialog Box and Parameters



Resistance Ron

The diode internal resistance Ron, in ohms (Ω). The **Resistance Ron** parameter cannot be set to `0` when the **Inductance Lon** parameter is set to `0`.

Inductance Lon

The diode internal inductance Lon, in henries (H). The **Inductance Lon** parameter cannot be set to `0` when the **Resistance Ron** parameter is set to `0`.

Forward voltage Vf

The forward voltage of the diode device, in volts (V).

Initial current Ic

Specifies an initial current flowing in the diode device. It is usually set to 0 in order to start the simulation with the diode device blocked. If the **Initial Current IC** parameter is set to a value greater than 0, the steady state calculation of SimPowerSystems considers the initial status of the diode as closed.

Initializing all states of a power electronic converter is a complex task. Therefore, this option is useful only with simple circuits.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, adds a Simulink output to the block returning the diode current and voltage.

Inputs and Outputs

The anode of the diode is identified with the letter a and the cathode is identified by the letter k. The Simulink output is a measurement output vector [Iak Vak] returning the diode current and voltage.

Assumptions and Limitations

The Diode block implements a macromodel of a diode device. It does not take into account either the geometry of the device or the complex physical processes underlying the state change [1]. The leakage current in the blocking state and the reverse-recovery (negative) current are not considered. In most circuits, the reverse current does not affect converter or other device characteristics.

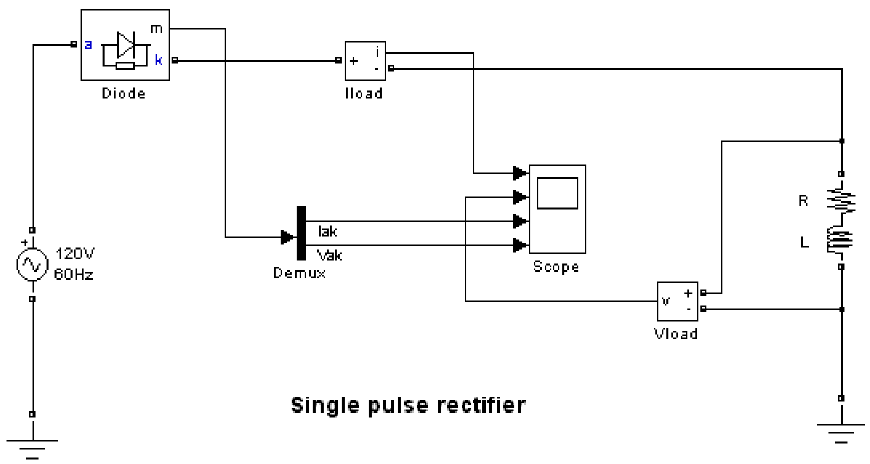
Depending on the value of the inductance Lon, the diode is modeled either as a current source (Lon > 0) or as a variable topology circuit (Lon = 0). The Diode block cannot be connected in series with an inductor, a current source, or an open circuit, unless its snubber circuit is in use. See the “Improving Simulation Performance” chapter for more details on this topic.

You must use a stiff integrator algorithm to simulate circuits containing diodes. ode23tb or ode15s with default parameters usually gives the best simulation speed.

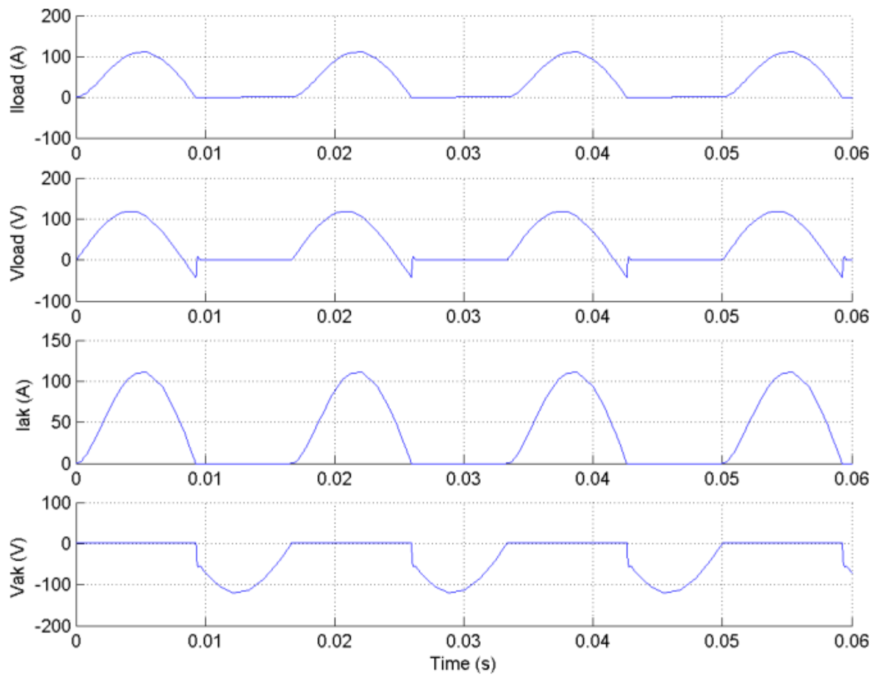
The inductance Lon is forced to 0 if you choose to discretize your circuit.

Example

The power_diode demo illustrates a single pulse rectifier consisting of a Diode block, an RL load, and an AC Voltage source block.



Simulation produces the following results.



References

[1] Rajagopalan, V., *Computer-Aided Analysis of Power Electronic Systems*, Marcel Dekker, Inc., New York, 1987.

[2] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

Thyristor, Universal Bridge

Discrete System

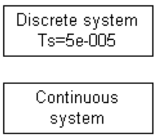
Purpose

Discretize the state-space model of a circuit

Library

powerlib

Description



The Discrete System block, in previous versions of SimPowerSystems, served to discretize the state-space model of an electrical model. Discrete time models are used for the linear elements as well as for the nonlinear blocks of the Elements, Machines, and Power Electronics libraries of **powerlib**.

Note This block is now obsolete. Use the Powergui block to replace this block.

See Also

Powergui

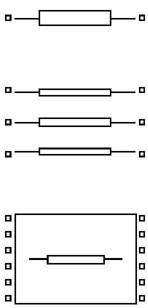
Purpose

Implement an N-phase distributed parameter transmission line model with lumped losses

Library

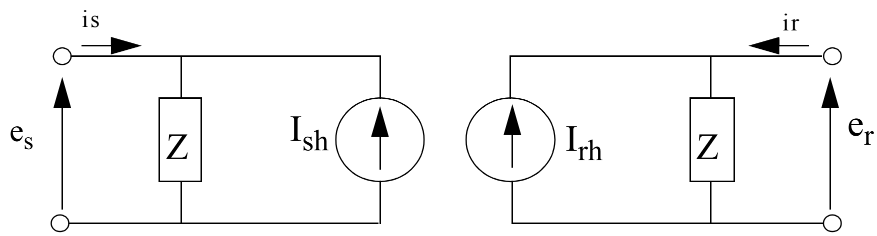
Elements

Description



The Distributed Parameter Line block implements an N-phase distributed parameter line model with lumped losses. The model is based on the Bergeron's traveling wave method used by the Electromagnetic Transient Program (EMTP) [1]. In this model, the lossless distributed LC line is characterized by two values (for a single-phase line): the surge impedance $Zc = \sqrt{L/C}$ and the phase velocity $v = 1/\sqrt{LC}$.

The model uses the fact that the quantity $e+Zi$ (where e is line voltage and i is line current) entering one end of the line must arrive unchanged at the other end after a transport delay of $\tau = d/v$, where d is the line length. By lumping $R/4$ at both ends of the line and $R/2$ in the middle and using the current injection method of SimPowerSystems, the following two-port model is derived.



$$I_{sh}(t) = \left(\frac{1+h}{2}\right)\left[\frac{1}{Z}e_r(t-\tau) + hi_r(t-\tau)\right] + \left(\frac{1-h}{2}\right)\left[\frac{1}{Z}e_s(t-\tau) + hi_s(t-\tau)\right]$$

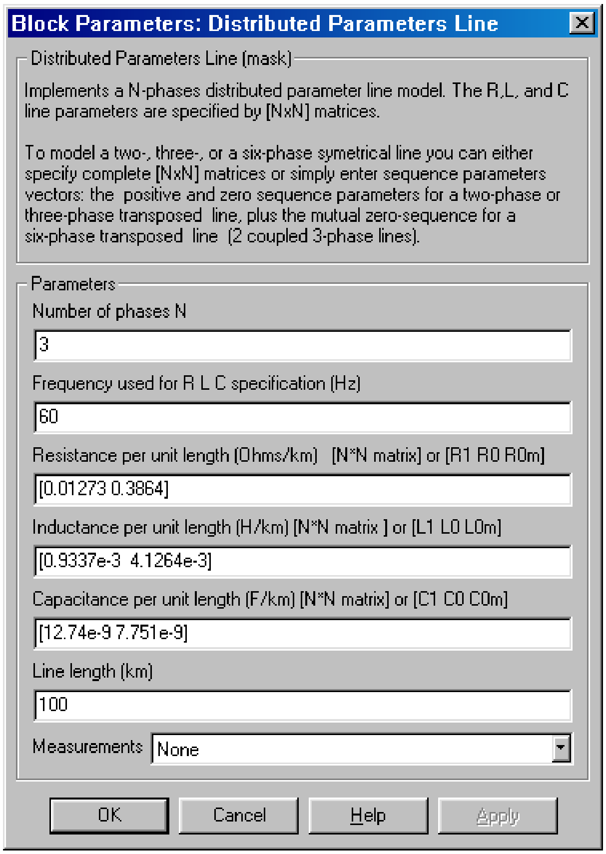
$$I_{rh}(t) = \left(\frac{1+h}{2}\right)\left[\frac{1}{Z}e_s(t-\tau) + hi_s(t-\tau)\right] + \left(\frac{1-h}{2}\right)\left[\frac{1}{Z}e_r(t-\tau) + hi_r(t-\tau)\right]$$

where $Z = Z_C + \frac{R}{4}$, $h = \frac{Z_C - \frac{R}{4}}{Z_C + \frac{R}{4}}$, $Z_C = \sqrt{\frac{L}{C}}$, and $\tau = d\sqrt{LC}$.

For multiphase line models, modal transformation is used to convert line quantities from phase values (line currents and voltages) into modal values independent of each other. The previous calculations are made in the modal domain before being converted back to phase values.

In comparison to the PI section line model, the distributed line represents wave propagation phenomena and line end reflections with much better accuracy. See the comparison between the two models in the Example section.

Dialog Box and Parameters



Number of phases N

Specifies the number of phases, N, of the model. The block icon dynamically changes according to the number of phases that you specify. When you apply the parameters or close the dialog box, the number of inputs and outputs is updated.

Frequency used for RLC specifications

Specifies the frequency used to compute the resistance *R*, inductance *L*, and capacitance *C* matrices of the line model.

Resistance per unit length

The resistance R per unit length, as an N-by-N matrix in ohms/km (Ω/km).

For a symmetrical line, you can either specify the N-by-N matrix or the sequence parameters. For a two-phase or three-phase continuously transposed line, you can enter the positive and zero-sequence resistances [*R1 R0*]. For a symmetrical six-phase line you can enter the sequence parameters plus the zero-sequence mutual resistance [*R1 R0 R0m*].

For asymmetrical lines, you must specify the complete N-by-N resistance matrix.

Inductance per unit length

The inductance *L* per unit length, as an N-by-N matrix in henries/km (H/km).

For a symmetrical line, you can either specify the N-by-N matrix or the sequence parameters. For a two-phase or three-phase continuously transposed line, you can enter the positive and zero-sequence inductances [*L1 L0*]. For a symmetrical six-phase line, you can enter the sequence parameters plus the zero-sequence mutual inductance [*L1 L0 L0m*].

For asymmetrical lines, you must specify the complete N-by-N inductance matrix.

Capacitance per unit length

The capacitance C per unit length, as an N-by-N matrix in farads/km (F/km).

For a symmetrical line, you can either specify the N-by-N matrix or the sequence parameters. For a two-phase or three-phase continuously transposed line, you can enter the positive and zero-sequence capacitances [*C1 C0*]. For a symmetrical six-phase line you can enter the sequence parameters plus the zero-sequence mutual capacitance [*C1 C0 C0m*].

For asymmetrical lines, you must specify the complete N-by-N capacitance matrix.

Line length

The line length, in km.

Measurements

Select `Phase-to-ground voltages` to measure the sending end and receiving end voltages for each phase of the line model.

Place a Multimeter block in your model to display the selected measurements during the simulation.

In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

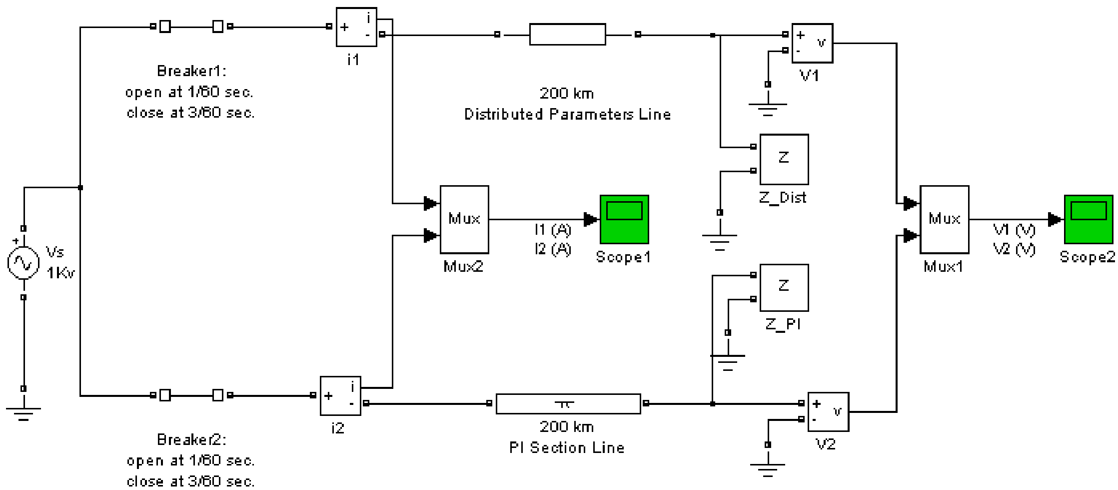
Measurement	Label
Phase-to-ground voltages, sending end	`Us_ph1_gnd:`, `Us_ph2_gnd:`, `Us_ph3_gnd:`, etc.
Phase-to-ground voltages, receiving end	`Ur_ph1_gnd:`, `Ur_ph2_gnd:`, `Ur_ph3_gnd:`, etc.

Limitations

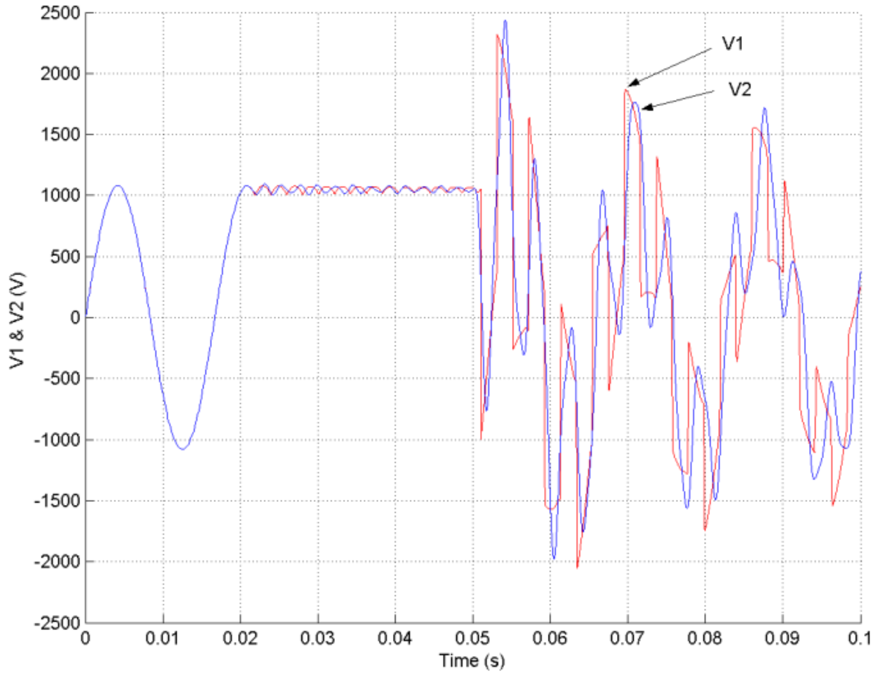
This model does not represent accurately the frequency dependence of RLC parameters of real power lines. Indeed, because of the skin effects in the conductors and ground, the R and L matrices exhibit strong frequency dependence, causing an attenuation of the high frequencies.

Example

The `power_monophaseline` demo illustrates a 200 km line connected on a 1 kV, 60 Hz infinite source. The line is deenergized and then reenergized after 2 cycles. The simulation is performed simultaneously with the Distributed Parameter Line block and with the PI Section Line block.

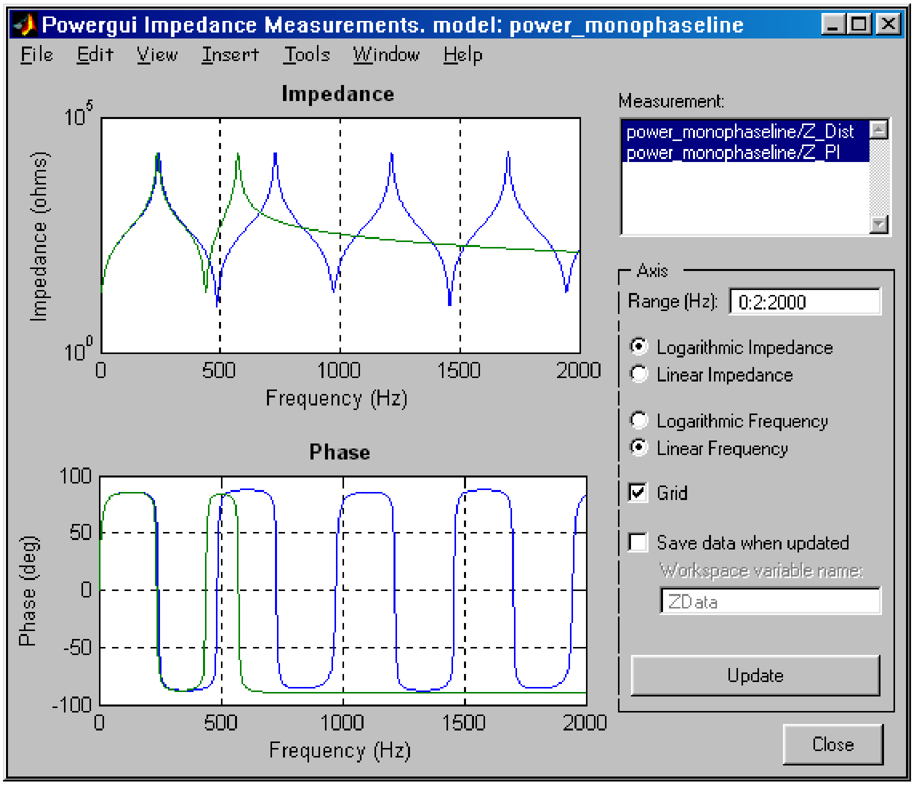


The receiving end voltage obtained with the Distributed Parameter Line block is compared with the one obtained with the PI Section Line block (two sections).



Open the Powergui. Click the **Impedance vs Frequency Measurement** button. A new window appears, listing the two Impedance Measurement blocks connected to your circuit. Set the parameters of **Impedance vs Frequency**

Measurement to compute impedance in the [0,2000] Hz frequency range, select the two measurements in the list, then click the **Update** button.



The distributed parameter line shows a succession of poles and zeros equally spaced, every 486 Hz. The first pole occurs at 243 Hz, corresponding to frequency $f = 1/(4 * T)$ where

$$T = \text{traveling time} = l\sqrt{LC} = 1.028 \text{ ms}$$

The PI section line only shows two poles because it consists of two PI sections. Impedance comparison shows that a two-section PI line gives a good approximation of the distributed line for the 0 to 350 Hz frequency range.

References

[1] Dommel, H., "Digital Computer Solution of Electromagnetic Transients in Single and Multiple Networks," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-88, No. 4, April, 1969.

See Also

PI Section Line

dq0_to_abc Transformation

Purpose

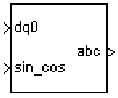
Perform a Park transformation from the dq0 reference frame to the abc reference frame

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description



The dq0_to_abc Transformation block performs the reverse of the so-called Park transformation, which is commonly used in three-phase electric machine models. It transforms three quantities (direct axis, quadratic axis, and zero-sequence components) expressed in a two-axis reference frame back to phase quantities. The following transformation is used:

$$V_a = V_d \sin(\omega t) + V_q \cos(\omega t) + V_0$$
$$V_b = V_d \sin(\omega t - 2\pi/3) + V_q \cos(\omega t - 2\pi/3) + V_0$$
$$V_c = V_d \sin(\omega t + 2\pi/3) + V_q \cos(\omega t + 2\pi/3) + V_0$$

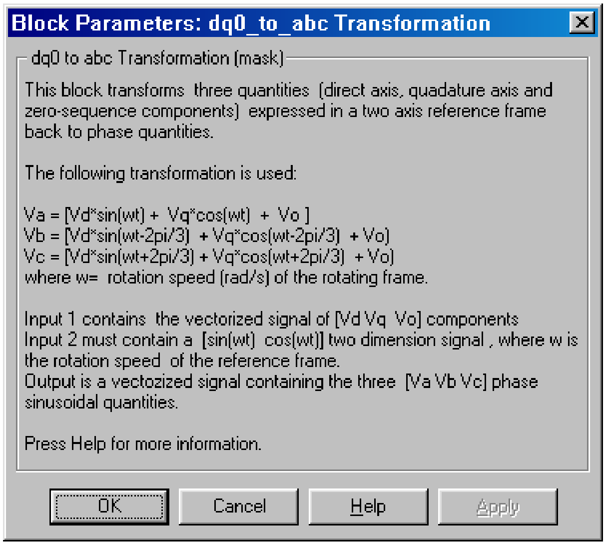
where

ω = rotation speed (rad/s) of the rotating frame

The transformation is the same for the case of a three-phase current; you simply replace the V_a, V_b, V_c, V_d, V_q, and V_0 variables with the I_a, I_b, I_c, I_d, I_q, and I_0 variables.

The dq0_to_abc Transformation block is used in the model of the Synchronous Machine block where the stator quantities are referred to the rotor. The Park transformation then eliminates time-varying inductances by referring the stator and rotor quantities to a fixed or rotating reference frame. The I_d and I_q currents represent the two DC currents flowing in the two equivalent rotor windings (d winding on the same axis as the field winding, and q winding in quadratic) producing the same flux as the stator I_a, I_b, and I_c currents.

Dialog Box and Parameters



Inputs and Outputs

`dq0`

Connect to the first input a vectorized signal containing the sequence components [d q 0] to be converted.

`sin_cos`

Connect to the second input a vectorized signal containing the [sin(ωt) cos(ωt)] values, where ω is the rotation speed of the reference frame.

`abc`

The output is a vectorized signal containing the three-phase sinusoidal quantities [phase A phase B phase C].

Example

See the demo of the abc_to_dq0 Transformation block for an example using the dq0_to_abc Transformation block.

See Also

abc_to_dq0 Transformation

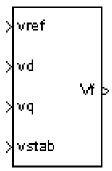
Purpose

Provide an excitation system for the synchronous machine and regulate its terminal voltage in generating mode

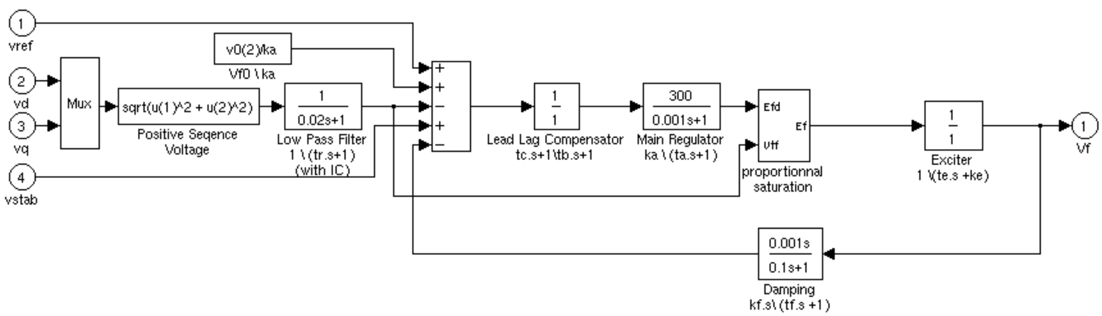
Library

Machines

Description



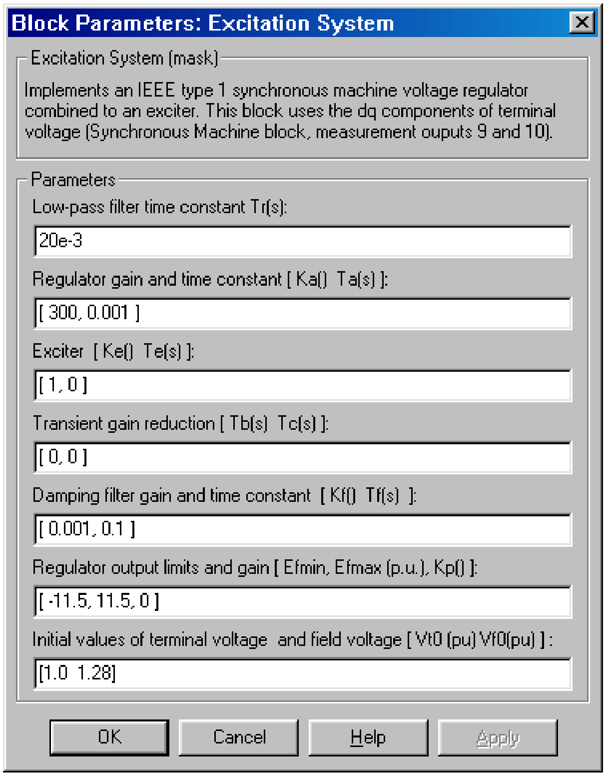
The Excitation System block is a Simulink system implementing a DC exciter described in [1], without the exciter's saturation function. The basic elements that form the Excitation System block are the voltage regulator and the exciter.



The exciter is represented by the following transfer function between the exciter voltage Vfd and the regulator's output ef:

$$\frac{V_{fd}}{ef} = \frac{1}{Ke + sTe}$$

Dialog Box and Parameters



Low-pass filter time constant

The time constant Tr, in seconds (s), of the first-order system that represents the stator terminal voltage transducer.

Regulator gain and time constant

The gain Ka and time constant Ta, in seconds (s), of the first-order system representing the main regulator.

Exciter

The gain Ke and time constant Te, in seconds (s), of the first-order system representing the exciter.

Transient gain reduction

The time constants Tb, in seconds (s), and Tc, in seconds (s), of the first-order system representing a lead-lag compensator.

Damping filter gain and time constant

The gain Kf and time constant Tf, in seconds (s), of the first-order system representing a derivative feedback.

Regulator output limits and gain

Limits Efmin and Efmax are imposed on the output of the voltage regulator. The upper limit can be constant and equal to Efmax, or variable and equal to the rectified stator terminal voltage Vtf times a proportional gain Kp. If Kp is set to 0, the former applies. If Kp is set to a positive value, the latter applies.

Initial values of terminal voltage and field voltage

The initial values of terminal voltage Vt0 (p.u.) and field voltage Vf0 (p.u.). When set correctly, they allow you to start the simulation in steady state. Initial terminal voltage should normally be set to 1 p.u. Both Vt0 and Vf0 values are automatically updated by the load flow utility of the Powergui block.

Example

See the Hydraulic Turbine and Governor block.

Inputs and Outputs

The first input of the block is the desired value of the stator terminal voltage. The following two inputs are the v_q and v_d components of the terminal voltage. The fourth input can be connected to a power system stabilizer to provide additional stabilization of power system oscillations. All inputs are in p.u. The output of the block is the field voltage Vf for the Synchronous Machine block (p.u.).

References

[1] "Recommended Practice for Excitation System Models for Power System Stability Studies," *IEEE Standard 421.5-1992*, August, 1992.

See Also

Generic Power System Stabilizer, Hydraulic Turbine and Governor, Multiband Power System Stabilizer, Steam Turbine and Governor, Synchronous Machine

Fourier

Purpose

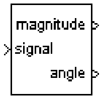
Perform a Fourier analysis of a signal

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description



The Fourier block performs a Fourier analysis of the input signal over a running window of one cycle of the fundamental frequency of the signal. The Fourier block can be programmed to calculate the magnitude and phase of the DC component, the fundamental, or any harmonic component of the input signal.

Recall that a signal *f(t)* can be expressed by a Fourier series of the form

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t)$$

where n represents the rank of the harmonics ($n = 1$ corresponds to the fundamental component). The magnitude and phase of the selected harmonic component are calculated by the following equations:

$$|H_n| = \sqrt{a_n^{\ 2} + b_n^{\ 2}} \qquad \angle H_n = \operatorname{atan}\left(\frac{b_n}{a_n}\right)$$

where

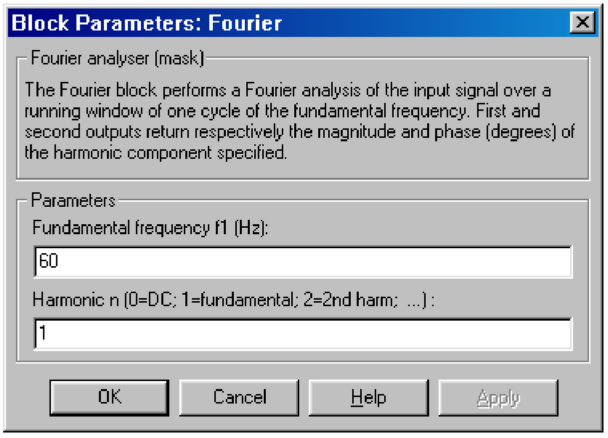
$$a_n = \frac{2}{T} \int_{(t-T)}^{t} f(t) \cos(n\omega t) dt$$

$$b_n = \frac{2}{T} \int_{(t-T)}^{t} f(t) \sin(n\omega t) dt$$

$$T = \frac{1}{f_1} \qquad f_1\text{: Fundamental frequency}$$

As this block uses a running average window, one cycle of simulation has to be completed before the outputs give the correct magnitude and angle. The discrete version of this block allows you to specify the initial magnitude and phase of the output signal. For the first cycle of simulation the outputs are held to the values specified by the initial input parameter.

Dialog Box and Parameters



Fundamental frequency f1

The fundamental frequency, in hertz, of the input signal.

Harmonic n (0 = DC; 1 = fundamental; 2 = 2nd harm; ...)

Specify the harmonic component you want to perform the Fourier analysis. Enter 0 if you want to analyze the DC component. Enter 1 if you want to analyze the fundamental frequency, or enter a number corresponding to the desired harmonic.

Inputs and Outputs

`signal`

Connect to the signal to be analyzed. Typical input signals are voltages or currents measured by Current Measurement blocks or Voltage Measurement blocks.

`magnitude`

The first output returns the magnitude of the harmonic component specified, in the same units as the input signal.

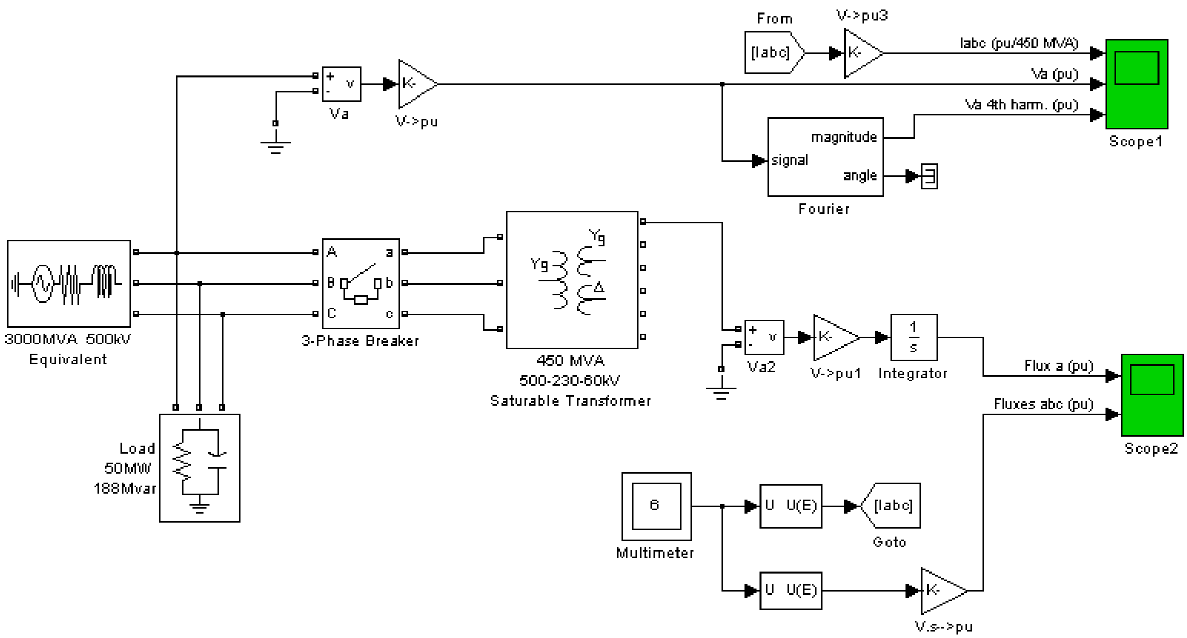
`phase`

The second output returns the phase, in degrees, of the harmonic component specified.

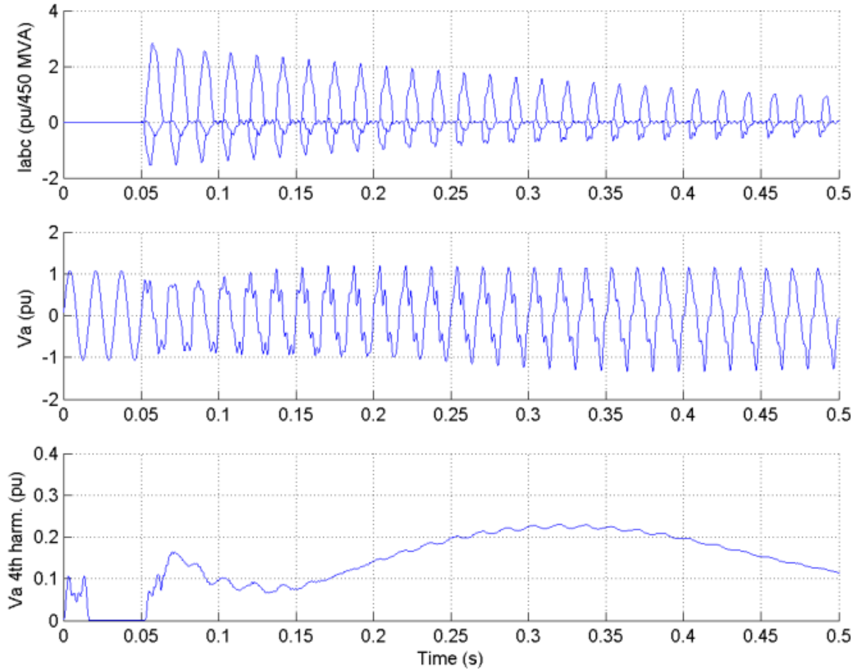
Example

The `power_transfosat` demo shows the energization of a 450 MVA three-phase transformer on a 500 kV network. The power system is simulated by an equivalent circuit consisting of an inductive source having a short-circuit power of 3000 MVA and a parallel RC load.

The load capacitance is set to produce a resonance at 240 Hz (fourth harmonic). A Fourier block is used to measure the fourth harmonic content of phase A of the primary voltage.



The Fourier block measures a high level fourth harmonic in the voltage (on the second trace of Scope1) because of the fourth harmonic content of the current injected into the network resonating at that particular frequency (240 Hz).


4
2
0
-2
Iabc (pu/450 MVA)
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5
2
1
0
-1
-2
Va (pu)
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5
0.4
0.3
0.2
0.1
0
Va 4th harm. (pu)
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5
Time (s)

Generic Power System Stabilizer

Purpose

Implement a generic power system stabilizer for the synchronous machine

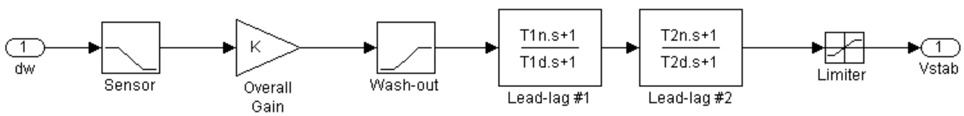
Library

Machines

Description

The Generic Power System Stabilizer (PSS) block can be used to add damping to the rotor oscillations of the synchronous machine by controlling its excitation. The disturbances occurring in a power system induce electromechanical oscillations of the electrical generators. These oscillations, also called power swings, must be effectively damped to maintain the system stability. The output signal of the PSS is used as an additional input (`vstab`) to the Excitation System block. The PSS input signal can be either the machine speed deviation, dw, or its acceleration power, Pa = Pm – Peo (difference between the mechanical power and the electrical power).

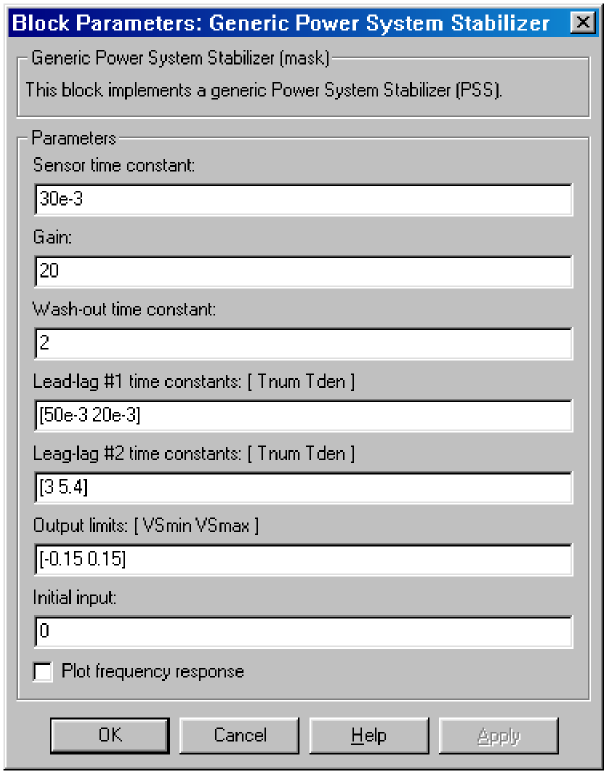
The Generic Power System Stabilizer is modeled by the following nonlinear system:



To ensure a robust damping, the PSS should provide a moderate phase advance at frequencies of interest in order to compensate for the inherent lag between the field excitation and the electrical torque induced by the PSS action.

The model consists of a low-pass filter, a general gain, a washout high-pass filter, a phase-compensation system, and an output limiter. The general gain K determines the amount of damping produced by the stabilizer. The washout high-pass filter eliminates low frequencies that are present in the dw signal and allows the PSS to respond only to speed changes. The phase-compensation system is represented by a cascade of two first-order lead-lag transfer functions used to compensate the phase lag between the excitation voltage and the electrical torque of the synchronous machine.

Dialog Box and Parameters



Sensor time constant

The time constant, in seconds (s), of the first-order low-pass filter used to filter the block's input signal.

Gain

The overall gain K of the generic power system stabilizer.

Wash-out time constant

The time constant, in seconds (s), of the first-order high-pass filter used by the washout system of the model.

Lead-lag #1 time constants: [Tnum Tden]

The numerator time constant T1n and denominator time constant T1d, in seconds (s), of the first lead-lag transfer function.

Lead-lag #2 time constants: [Tnum Tden]

The numerator time constant T2n and denominator time constant T2d, in seconds (s), of the second lead-lag transfer function.

Output limits: [Vsmin Vsmax]

The limits VSmin and VSmax, in p.u., imposed on the output of the stabilizer.

Initial input

The initial DC voltage, in p.u., of the block's input signal. Specification of this parameter is required to initialize all states and start the simulation in steady state with `vstab` set to zero.

Plot frequency response

If selected, a plot of the frequency response of the stabilizer is displayed when you click the **Apply** button.

Magnitude in dB

The **Magnitude in dB** parameter is not visible if the **Plot frequency response** is not selected. If selected, the magnitude of the frequency response is plotted in dB.

Frequency range

The **Frequency range** parameter is not visible in the dialog box if the **Plot frequency response** is not selected. Specify the frequency range used to plot the frequency response of the stabilizer.

Inputs and Outputs

`In`

Two types of signals can be used at the input `In`:

- The synchronous machine speed deviation dw signal (in p.u.)
- The synchronous machine acceleration power Pa = Pm – Peo (difference between the machine mechanical power and output electrical power (in p.u.))

`Vstab`

The output is the stabilization voltage (in p.u.) to connect to the Vstab input of the Excitation System block used to control the terminal voltage of the synchronous machine.

Example

See the help text of the `power_PSS` demo model.

References

Kundur, P., *Power System Stability and Control*, McGraw-Hill, 1994, Section 12.5.

See Also

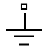
Multiband Power System Stabilizer

Ground

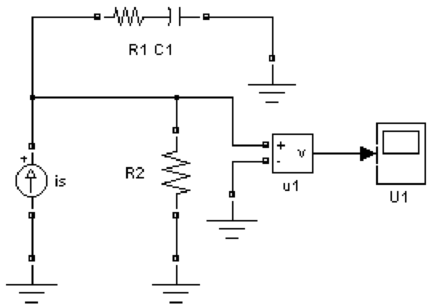
Purpose Provide a connection to the ground

Library Elements

Description The Ground block implements a connection to the ground.



Example The `power_ground` demo shows an application of the Ground block.



See Also Neutral

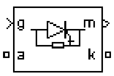
Purpose

Implement a gate turn off (GTO) thyristor model

Library

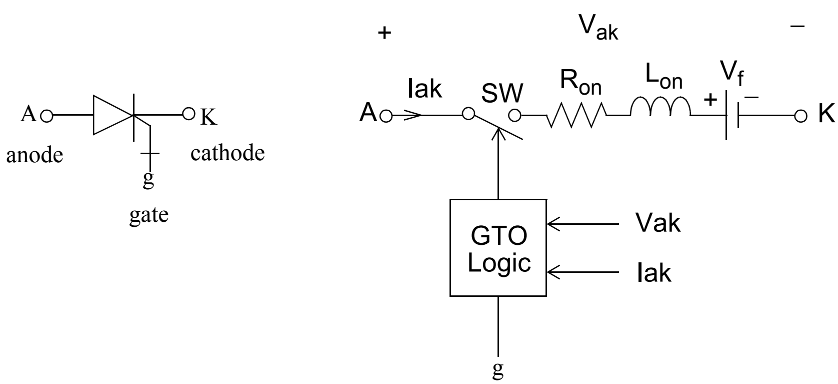
Power Electronics

Description



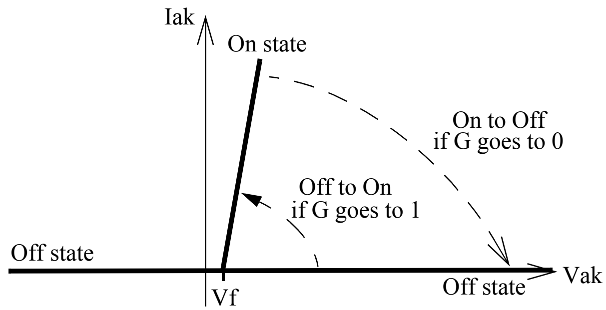
The gate turn-off (GTO) thyristor is a semiconductor device that can be turned on and off via a gate signal. Like a conventional thyristor, the GTO thyristor can be turned on by a positive gate signal (g > 0). However, unlike the thyristor, which can be turned off only at a zero crossing of current, the GTO can be turned off at any time by the application of a gate signal equal to 0.

The GTO thyristor is simulated as a resistor Ron, an inductor Lon, and a DC voltage source Vf connected in series with a switch. The switch is controlled by a logical signal depending on the voltage Vak, the current Iak, and the gate signal g.

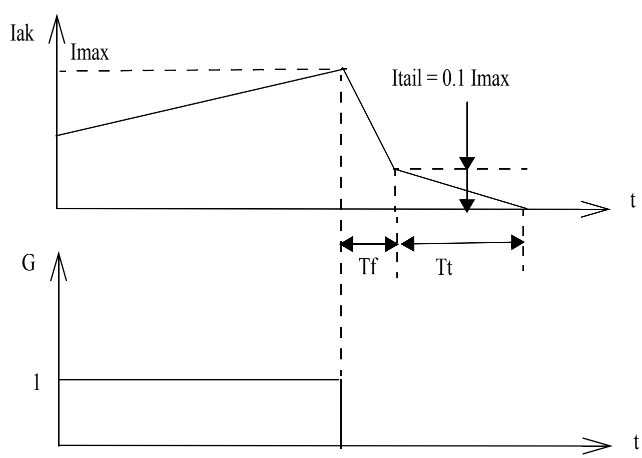


The Vf, Ron, and Lon parameters are the forward voltage drop while in conduction, the forward conducting resistance, and the inductance of the device. The GTO block also contains a series Rs-Cs snubber circuit that can be connected in parallel with the GTO device (between terminal ports A and K).

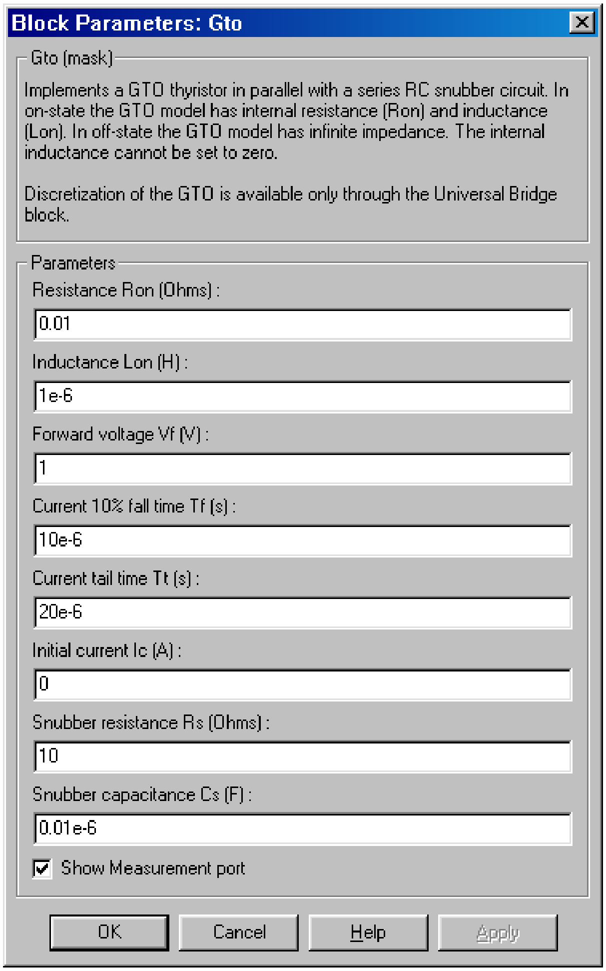
The GTO thyristor turns on when the anode-cathode voltage is greater than Vf and a positive pulse signal is present at the gate input (g > 0). When the gate signal is set to 0, the GTO thyristor starts to block but its current does not stop instantaneously.



Because the current extinction process of a GTO thyristor contributes significantly to the turnoff losses, the turnoff characteristic is built into the model. The current decrease is approximated by two segments. When the gate signal becomes 0, the current Iak first decreases from the value Imax (value of Iak when the GTO thyristor starts to open) to Imax/10, during the fall time (Tf), and then from Imax/10 to 0 during the tail time (Tt). The GTO thyristor turns off when the current Iak becomes 0. The latching and holding currents are not considered.



Dialog Box and Parameters



Resistance Ron

The internal resistance Ron, in ohms (Ω).

Inductance Lon

The internal inductance Lon, in henries (H). The **Inductance Lon** parameter cannot be set to `0`.

Forward voltage Vf

The forward voltage of the GTO thyristor device, in volts (V).

Current 10% fall time

The current fall time Tf, in seconds (s).

Current tail time

The current tail time Tt, in seconds (s).

Initial current Ic

You can specify an initial current flowing in the GTO thyristor. It is usually set to 0 in order to start the simulation with the device blocked.

If the **Initial Current IC** parameter is set to a value greater than 0, the steady state calculation of SimPowerSystems considers the initial status of the GTO as closed. Initializing all states of a power electronic converter is a complex task. Therefore, this option is useful only with simple circuits.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance, in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, add a Simulink output to the block returning the GTO current and voltage.

Inputs and Outputs

The input port (`g`) is a Simulink signal applied to the gate of the GTO thyristor. The output port (`m`) is a Simulink measurement vector [Iak Vak] returning the GTO thyristor current and voltage.

Assumptions and Limitations

The GTO block implements a macromodel of a real GTO thyristor. It does not take into account either the geometry of the device or the underlying physical processes of the device [1].

The GTO block requires a continuous application of the gate signal (`g` > 0) in order to be in the on state (with Iak > 0). The latching current and the holding current are not considered. The critical value of the derivative of the reapplied anode-cathode voltage is not considered.

The GTO block is modeled as a current source. It cannot be connected in series with an inductor, a current source, or an open circuit, unless its snubber circuit is in use. In order to avoid an algebraic loop, you cannot set the inductance Lon to 0.

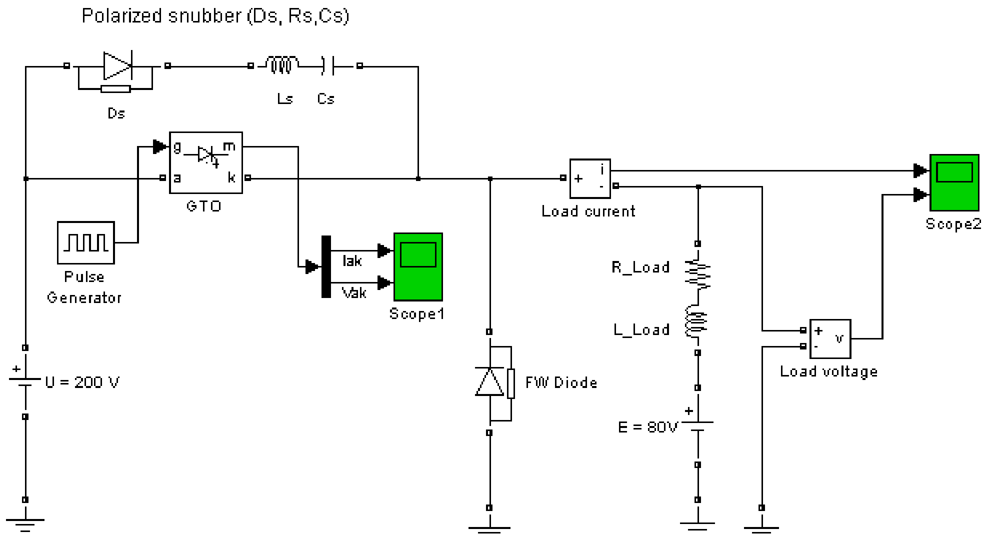
Each GTO block adds an extra state to the electrical circuit model. Circuits containing GTO blocks cannot be discretized. In order to discretize circuits using GTO converters, use the Universal Bridge block or the Three-Level Bridge block. See the "Improving Simulation Performance" chapter for more details on this topic.

You must use a stiff integrator algorithm to simulate circuits containing GTO blocks. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

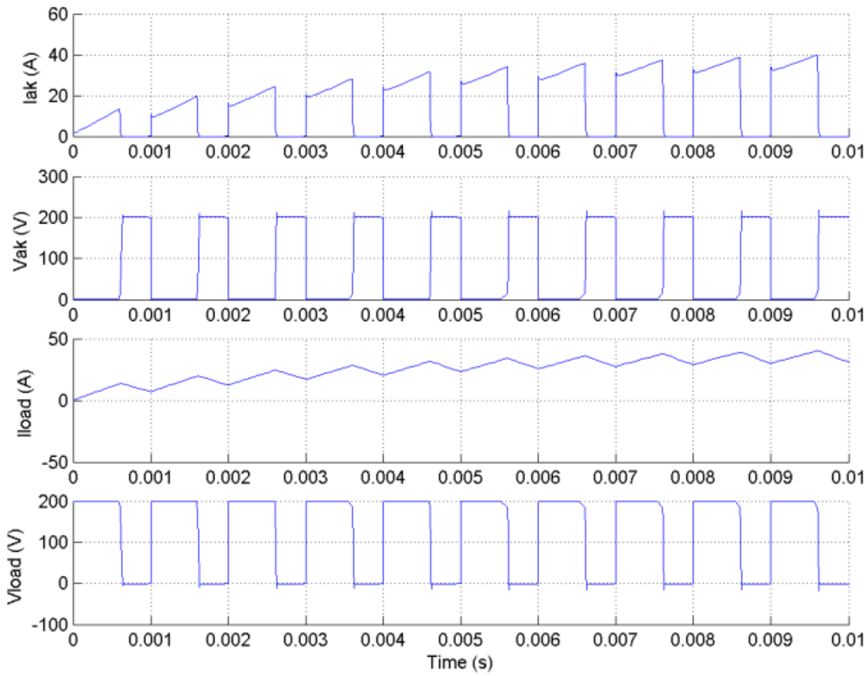
Example

The `power_buckconv` demo illustrates the use of the GTO block in a buck converter topology. The basic polarized snubber circuit is connected across the GTO block. The snubber circuit consists of a capacitor Cs, a resistor Rs, and a diode Ds. The parasitic inductance Ls of the snubber circuit is also taken into consideration.

The parameters of the GTO block are those found in the dialog box section, except for the internal snubber, which is not used (**Rs** = `inf`; **Cs** = `0`). The switching frequency is 1000 Hz and the pulse width is 216 degrees (duty cycle: 60%).



Run the simulation. Observe the GTO voltage and current as well as the load voltage and current.



References

[1] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

IGBT, MOSFET, Three-Level Bridge, Thyristor, Universal Bridge

Hydraulic Turbine and Governor

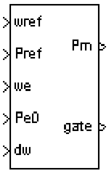
Purpose

Model a hydraulic turbine and a proportional-integral-derivative (PID) governor system

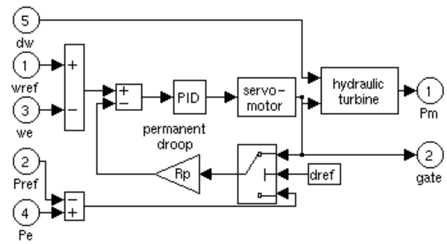
Library

Machines

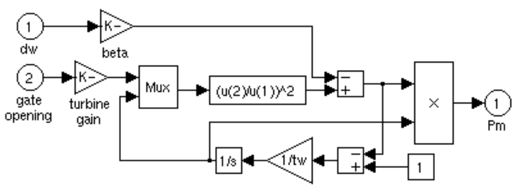
Description



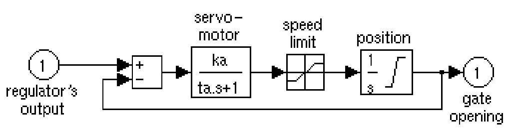
The Hydraulic Turbine and Governor block implements a nonlinear hydraulic turbine model, a PID governor system, and a servomotor [1].



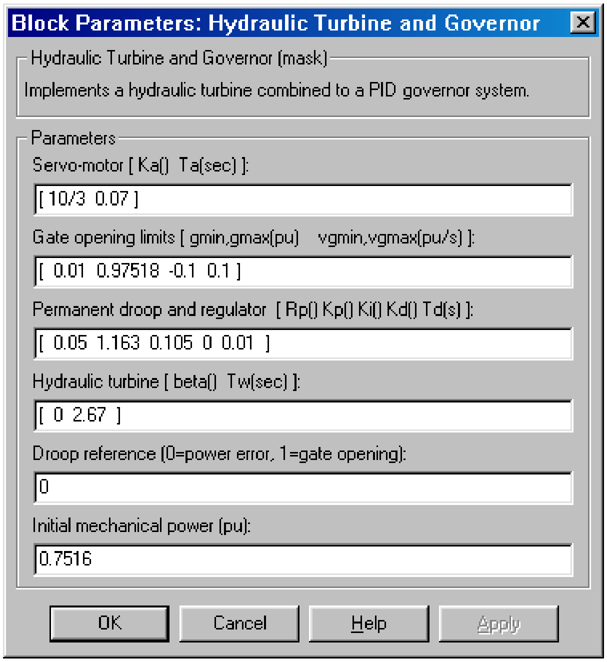
The hydraulic turbine is modeled by the following nonlinear system.



The gate servomotor is modeled by a second-order system.



Dialog Box and Parameters



Servo-motor

The gain Ka and time constant Ta, in seconds (s), of the first-order system representing the servomotor.

Gate opening limits

The limits gmin and gmax (p.u.) imposed on the gate opening, and vgmin and vgmax (p.u./s) imposed on gate speed.

Permanent droop and regulator

The static gain of the governor is equal to the inverse of the permanent droop Rp in the feedback loop. The PID regulator has a proportional gain Kp, an integral gain Ki, and a derivative gain Kd. The high-frequency gain of the PID is limited by a first-order low-pass filter with time constant Td (s).

Hydraulic turbine

The speed deviation damping coefficient β and water starting time Tw (s).

Droop reference

Specifies the input of the feedback loop: gate position (set to 1) or electrical power deviation (set to 0).

Initial mechanical power

The initial mechanical power Pm0 (p.u.) at the machine's shaft. This value is automatically updated by the load flow utility of the Powergui block.

Inputs and Outputs

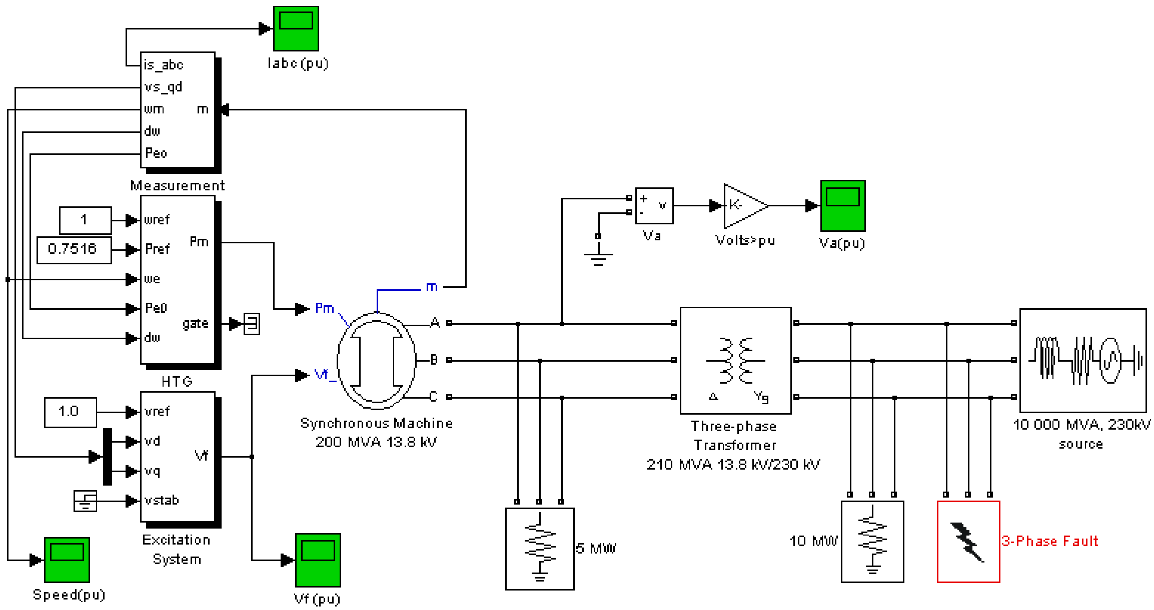
The first two inputs are the desired speed and mechanical power. The third and fourth inputs are the machine's actual speed and electrical power. The fifth input is the speed deviation. Inputs 2 and 4 can be left unconnected if you want to use the gate position as input to the feedback loop instead of the power deviation. All inputs are in p.u. The outputs of the block are mechanical power Pm for the Synchronous Machine block and gate opening (both in p.u.).

Example

This `power_turbine` demo illustrates the use of the Synchronous Machine associated with the Hydraulic Turbine and Governor (HTG) and Excitation System blocks. It also demonstrates the use of the load flow tool of the Powergui block to initialize machine currents and initial mechanical power of the HTG block. A three-phase generator rated 200 MVA, 13.8 kV, 112.5 rpm is connected to a 230 kV network through a Delta-Y 210 MVA transformer. The system starts in steady state with the generator supplying 150 MW of active power. At t = 0.1 s, a three-phase to ground fault occurs on the 230 kV bus of the transformer. The fault is cleared after six cycles (t = 0.2 s).

In order to start the simulation in steady state, you must initialize the Synchronous Machine block for the desired load flow. Open the Powergui and select **Load flow and machine initialization**. The machine **Bus type** should be already initialized as `PV generator`, indicating that the load flow is performed with the machine controlling the active power and its terminal voltage. Specify the desired values by entering the following parameters:

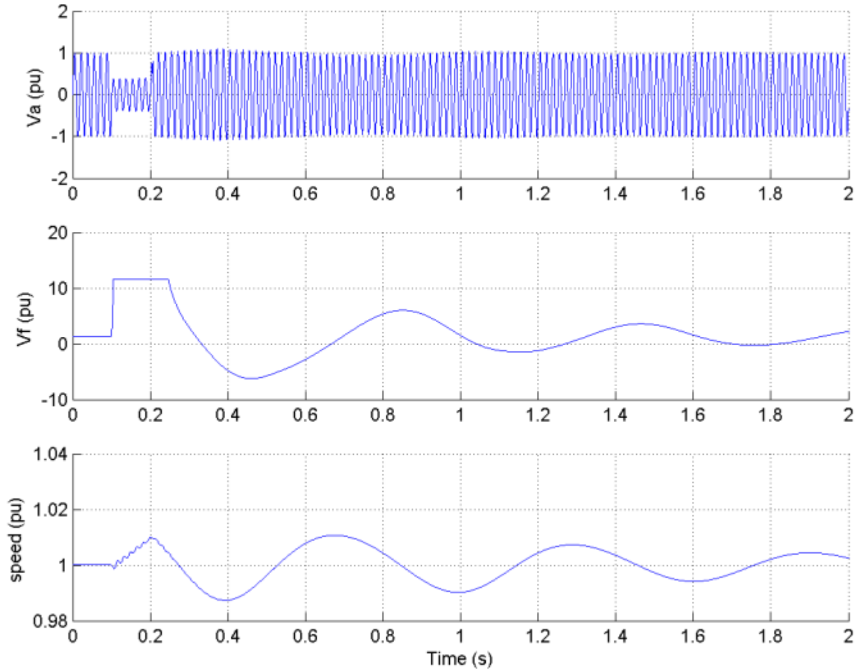
- Terminal voltage U AB (Vrms) = 13800
- Active power (watts) = 150e6



Then click the **Update Load Flow** button. Once the load flow has been solved, the line-to-line machine voltages as well as the phase currents flowing out of the machine. The machine reactive power, mechanical power, and field voltage requested to supply the electrical power should also be displayed:

- Q = 3.4 Mvar
- Pmec = 150.32 MW (0.7516 p.u.)
- Field voltage Vf = 1.291 p.u.

The load flow also initializes the HTG and Excitation System blocks. Open the HTG block menu and notice that the initial mechanical power is set to 0.5007 p.u. (100.14 MW). Then open the Excitation System block menu and note that the initial terminal voltage and field voltage are set respectively to 1.0 and 1.291 p.u. Open the four scopes and start the simulation. The simulation starts in steady state.



Observe that the terminal voltage Va is 1.0 p.u. at the beginning of the simulation. It falls to about 0.4 p.u. during the fault and returns to nominal quickly after the fault is cleared. This quick response in terminal voltage is due to the fact that the Excitation System output Vf can go as high as 11.5 p.u., which it does during the fault. The speed of the machine increases to 1.01 p.u. during the fault, then it oscillates around 1 p.u. as the governor system regulates it. The speed takes much longer than the terminal voltage to stabilize, mainly because the rate of valve opening/closing in the governor system is limited to 0.1 p.u./s.

References

[1] IEEE Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies, "Hydraulic Turbine and Turbine Control Models for Dynamic Studies," *IEEE Transactions on Power Systems*, Vol. 7, No. 1, February, 1992, pp. 167-179.

See Also

Excitation System, Steam Turbine and Governor, Synchronous Machine

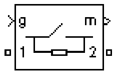
Purpose

Implement an ideal switch device

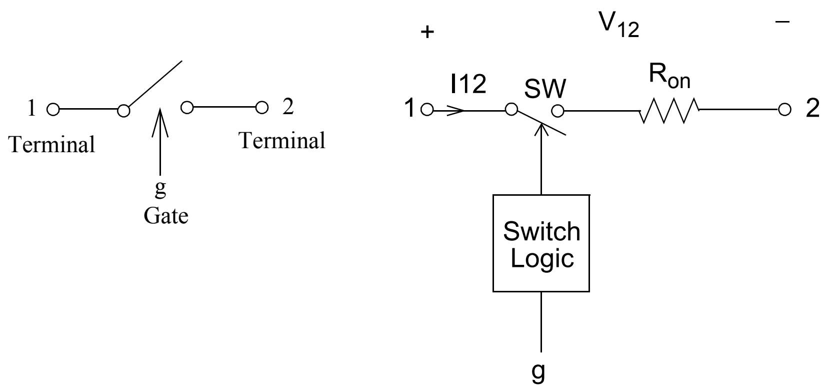
Library

Power Electronics

Description



The Ideal Switch block does not correspond to a particular physical device. When used with appropriate switching logic, it can be used to model simplified semiconductor devices such as a GTO or a MOSFET, or even a power circuit breaker with current chopping. The switch is simulated as a resistor Ron in series with a switch controlled by a logical gate signal g.

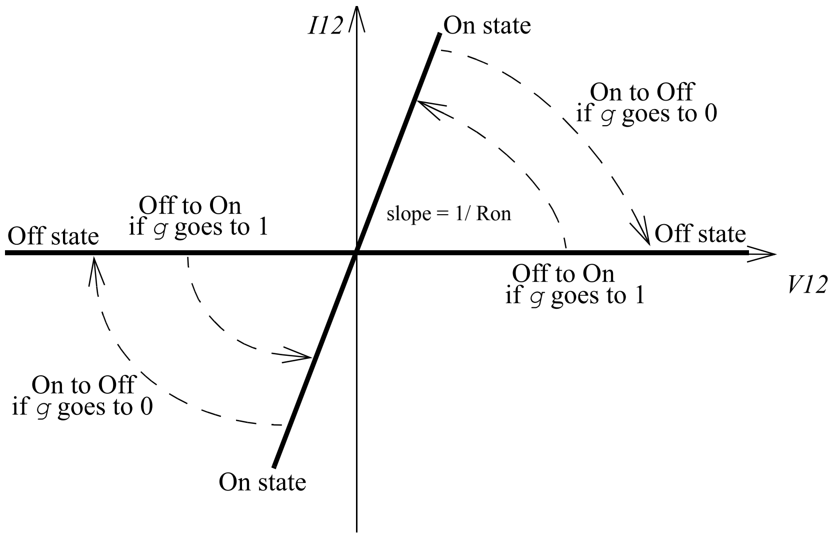


The Ideal Switch block is fully controlled by the gate signal (g > 0 or g = 0). It has the following characteristics:

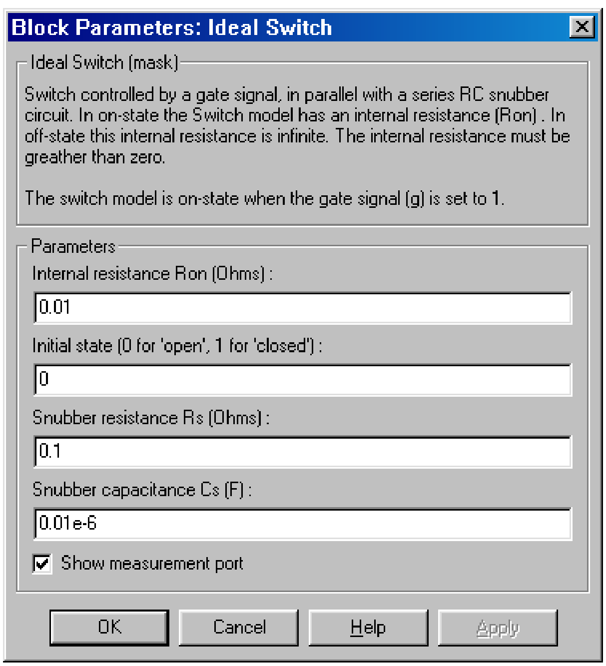
- Blocks any forward or reverse applied voltage with 0 current flow when `g` = 0
- Conducts any bidirectional current with quasi-zero voltage drop when `g` > 0
- Switches instantaneously between on and off states when triggered

The Ideal Switch block turns on when a positive signal is present at the gate input (g > 0). It turns off when the gate signal equals 0 (g = 0).

The Ideal Switch block also contains a series Rs-Cs snubber circuit that can be connected in parallel with the ideal switch (between nodes 1 and 2).



Dialog Box and Parameters



Internal Resistance Ron

The internal resistance of the switch device, in ohms (Ω). The **Internal resistance Ron** parameter cannot be set to 0.

Initial state

The initial state of the Ideal Switch block. The initial status of the Ideal Switch block is taken into account in the steady-state calculation of SimPowerSystems.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, add a Simulink output to the block returning the ideal switch current and voltage.

Inputs and Outputs

The input port (g) controls the opening and closing of the switch. The output port (m) is a measurement output vector [Iak Vak] returning the Ideal Switch block current and voltage.

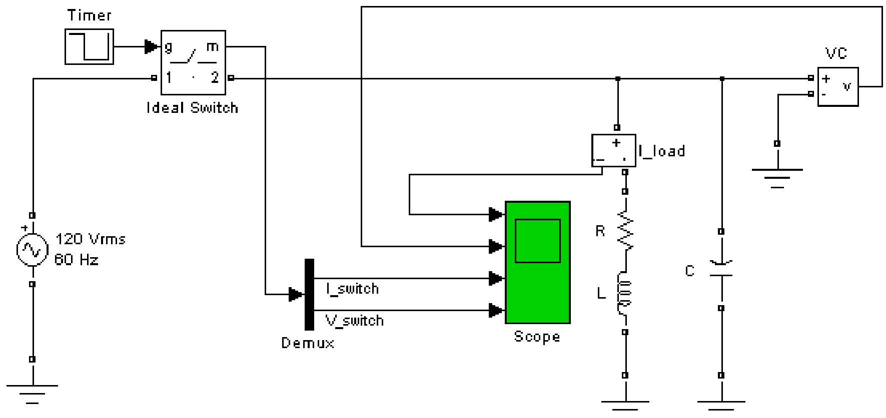
Assumptions and Limitations

The Ideal Switch block is modeled as a current source. It cannot be connected in series with an inductor, a current source, or an open circuit, unless its snubber circuit is in use. See the "Improving Simulation Performance" chapter for more details on this topic.

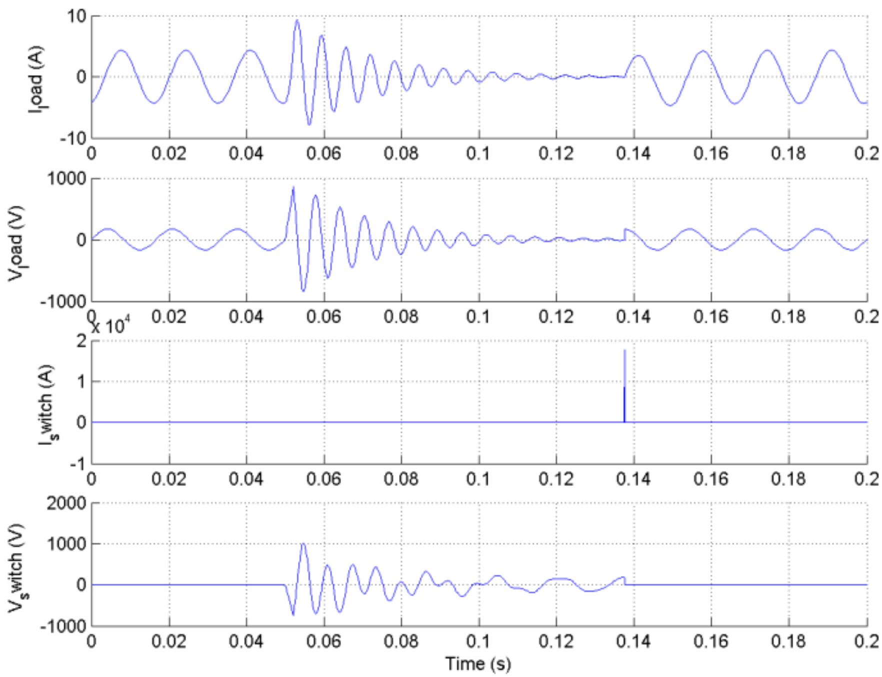
You must use a stiff integrator algorithm to simulate circuits containing Ideal Switch blocks. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

Example

The `power_switch` demo uses the Ideal Switch block to switch an RLC circuit on an AC source (60 Hz). The switch, which is initially closed, is first opened at t = 50 ms (3 cycles) and then reclosed at t = 138 ms (8.25 cycles). The Ideal Switch block has 0.01 ohms resistance and no snubber is used.



Run the simulation and observe the inductor current, the switch current, and the capacitor voltage. Notice the high-frequency overvoltage produced by inductive current chopping. Note also the high switch current spike when the switch is reclosed on the capacitor at maximum source voltage.



References

Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

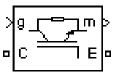
Breaker

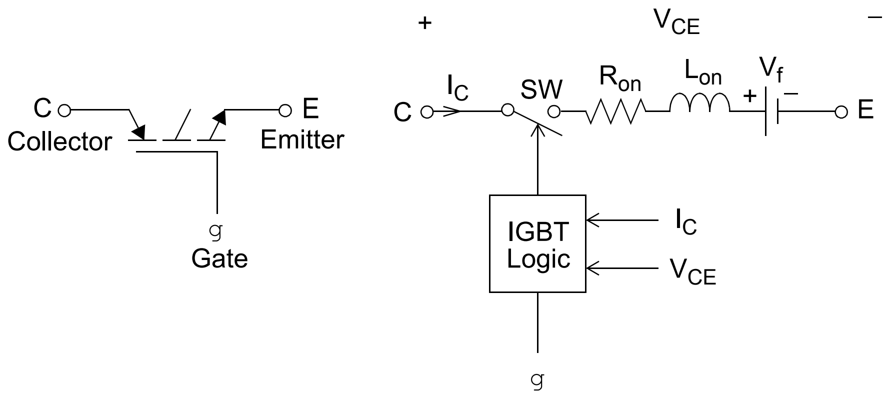
IGBT

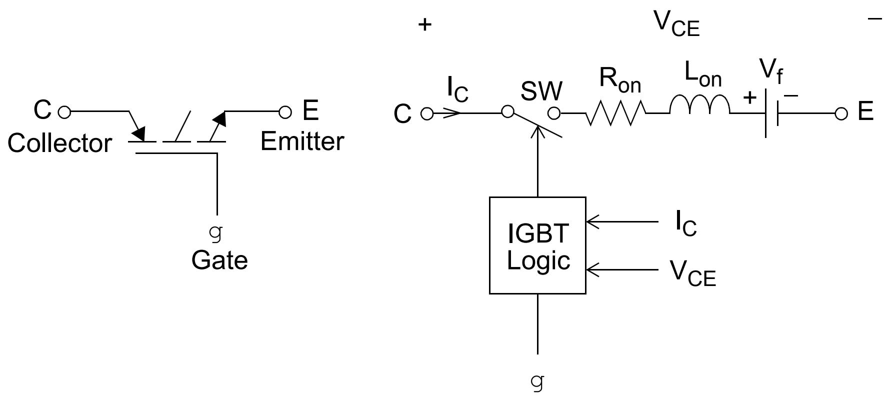
Purpose Implement an insulated gate bipolar transistor (IGBT)

Library Power Electronics

Description The IGBT block implements a semiconductor device controllable by the gate signal. The IGBT is simulated as a series combination of a resistor Ron, inductor Lon, and a DC voltage source Vf in series with a switch controlled by a logical signal (g > 0 or g = 0).



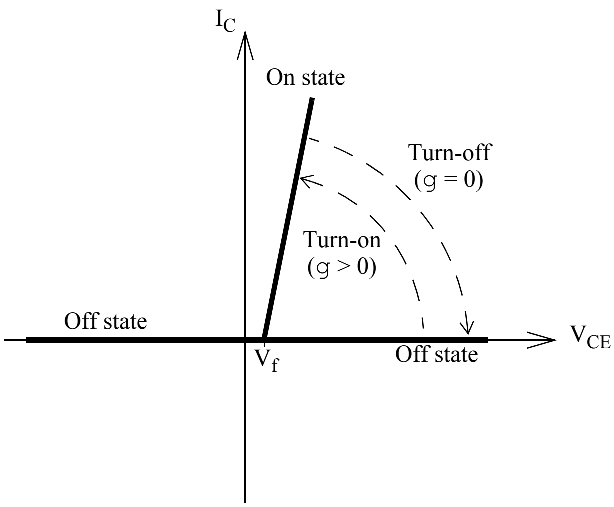




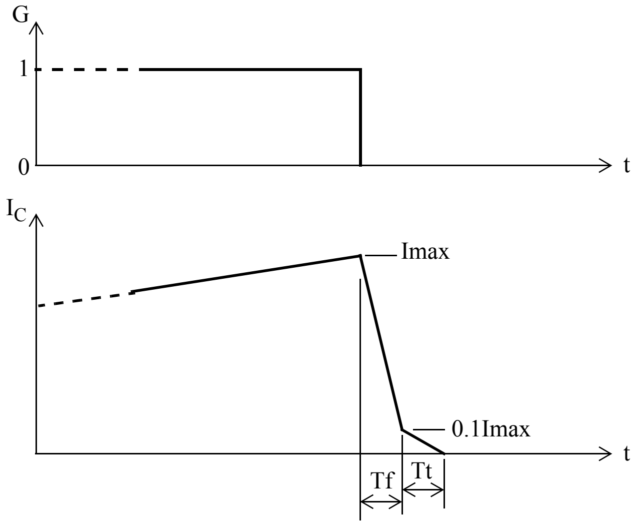
The IGBT turns on when the collector-emitter voltage is positive and greater than Vf and a positive signal is applied at the gate input (g > 0). It turns off when the collector-emitter voltage is positive and a 0 signal is applied at the gate input (g = 0).

The IGBT device is in the off state when the collector-emitter voltage is negative. Note that many commercial IGBTs do not have the reverse blocking capability. Therefore, they are usually used with an antiparallel diode.

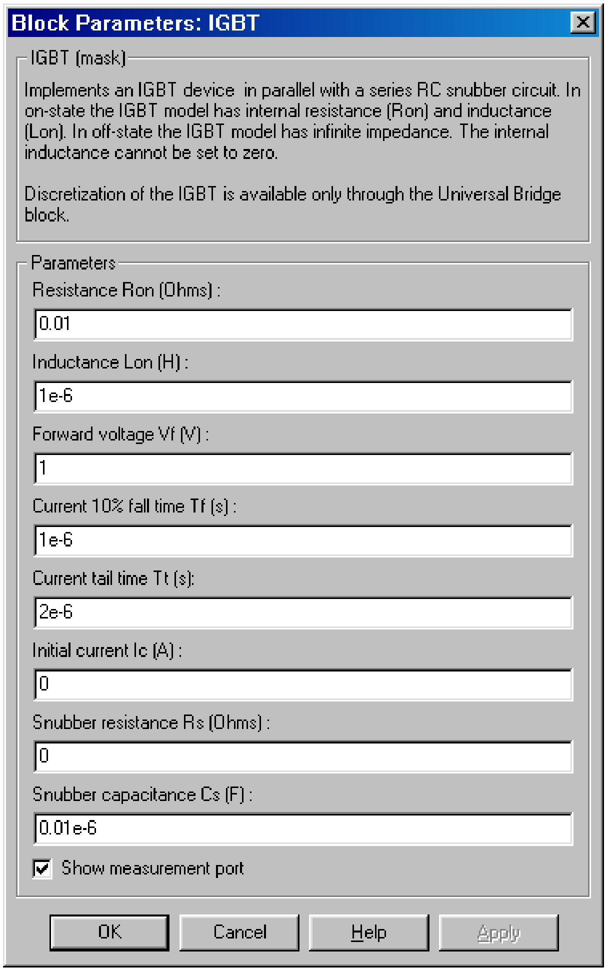
The IGBT block contains a series Rs-Cs snubber circuit, which is connected in parallel with the IGBT device (between terminals C and E).



The turn-off characteristic of the IGBT model is approximated by two segments. When the gate signal falls to 0, the collector current decreases from Imax to 0.1 Imax during the fall time (Tf), and then from 0.1 Imax to 0 during the tail time (Tt).


G
1
0
t
I_C
Imax
0.1Imax
t
Tf
Tt

Dialog Box and Parameters



Resistance Ron

The internal resistance Ron, in ohms (Ω).

Inductance Lon

The internal inductance Lon, in henries (H). The **Inductance Lon** parameter cannot be set to 0.

Forward voltage Vf

The forward voltage of the IGBT device, in volts (V).

Current 10% fall time

The current fall time Tf, in seconds (s).

Current tail time

The current tail time Tt, in seconds (s).

Initial current Ic

You can specify an initial current flowing in the IGBT. It is usually set to 0 in order to start the simulation with the device blocked.

If the **Initial Current IC** parameter is set to a value greater than 0, the steady state calculation of SimPowerSystems considers the initial status of the IGBT as closed. Initializing all states of a power electronic converter is a complex task. Therefore, this option is useful only with simple circuits.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, add a Simulink output to the block returning the diode IGBT current and voltage.

Inputs and Outputs

The input port (g) is a logical Simulink signal applied to the gate. The output port is a measurement vector [Ic Vce] returning the IGBT current and voltage.

Assumptions and Limitations

The IGBT block implements a macromodel of the real IGBT device. It does not take into account either the geometry of the device or the complex physical processes [1].

The IGBT block is modeled as a current source. It cannot be connected in series with an inductor, a current source, or an open circuit, unless its snubber circuit is in use. In order to avoid an algebraic loop, you cannot set the IGBT block inductance Lon to 0. Each IGBT block adds an extra state to the electrical circuit model. See the “Improving Simulation Performance” chapter for more details on this topic.

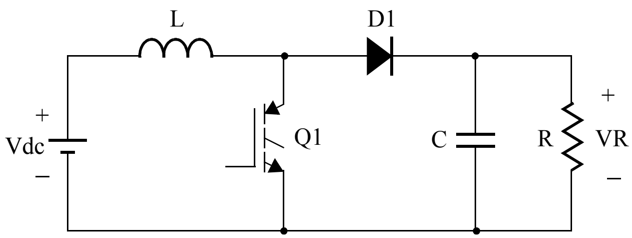
Circuits containing individual IGBT blocks cannot be discretized. However, discretization is permitted for IGBT/Diode bridges simulated with the Universal Bridge block or the Three-Level Bridge block.

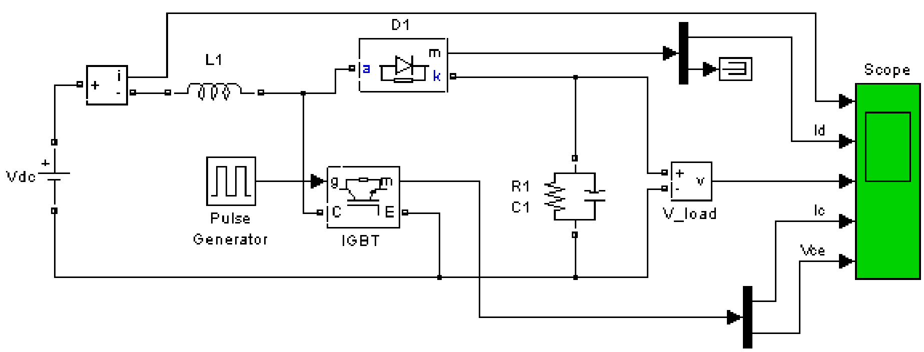
You must use a stiff integrator algorithm to simulate circuits containing IGBTs. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

Example

The `power_igbtconv` demo illustrates the use of the IGBT block in a boost DC-DC converter. The IGBT is switched on and off at a frequency of 10 kHz to transfer energy from the DC source to the load (RC). The average output voltage (V_R) is a function of the duty cycle (α) of the IGBT switch:

$$V_R = \frac{1}{1-\alpha} V_{dc}$$

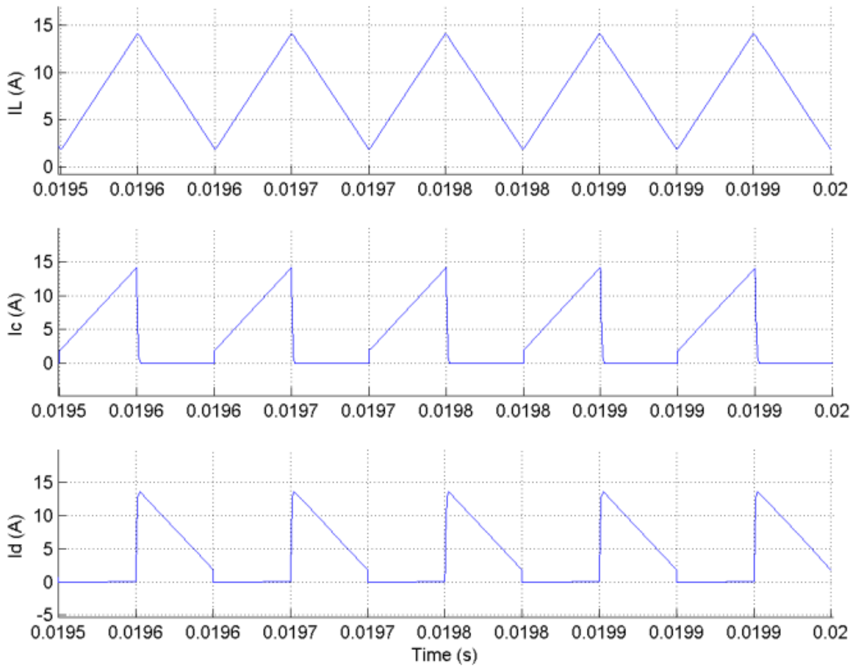


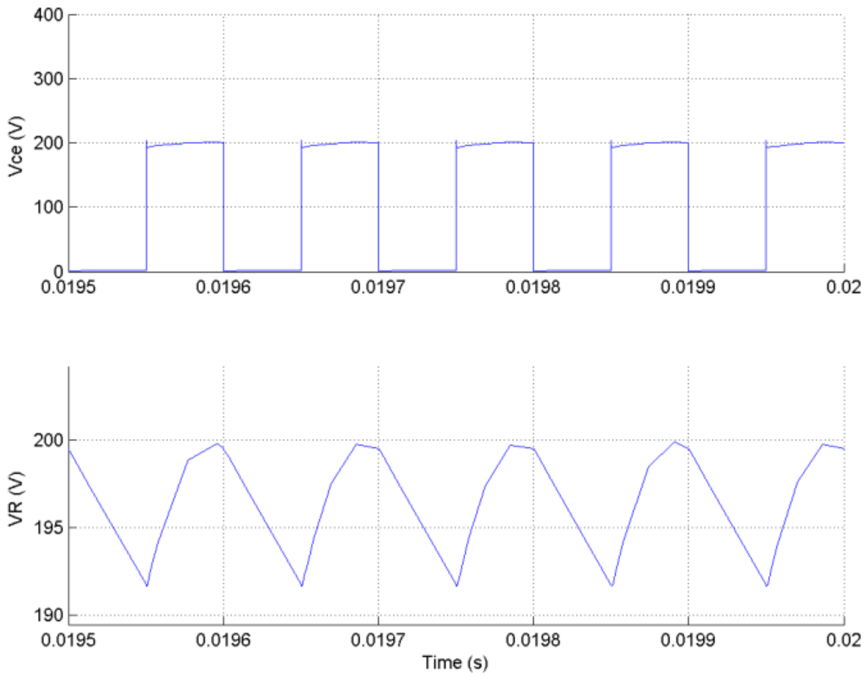


In our example, $\alpha = 0.5$ so that the theoretical value of V_R is 200 V, assuming no voltage drop across the diode and the IGBT.

Run the simulation and observe the inductor current (I_L), the IGBT collector current (I_C), the diode current (I_D), the IGBT device collector-emitter voltage (V_{CE}), and the load voltage (V_R).

The load voltage (197 V) is slightly lower than the theoretical value (200 V) mainly because of the forward voltage (V_f) of the diode (0.8 V) and of the IGBT ($V_f = 1$ V).





References

[1] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

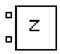
GTO, MOSFET, Three-Level Bridge, Thyristor

Impedance Measurement

Purpose Measure the impedance of a circuit as a function of frequency

Library Measurements

Description



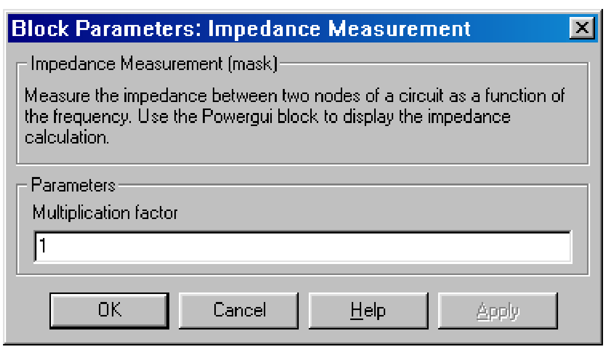
The Impedance Measurement block measures the impedance between two nodes of a linear circuit as a function of the frequency. It consists of a current source Iz, connected between inputs one and two of the Impedance Measurement block, and a voltage measurement Vz, connected across the terminals of the current source. The network impedance is calculated as the transfer function H(s) from the current input to the voltage output of the state-space model.

$$H(s) = \frac{V_z(s)}{I_z(s)}$$

The impedance (magnitude and phase) as function of frequency is displayed by using the Impedance vs Frequency Measurement tool of the Powergui block.

The measurement takes into account the initial states of the Breaker and Ideal Switch blocks. It also allows impedance measurements with Distributed Parameter Line blocks in your circuit.

Dialog Box and Parameter



Multiplication factor

If you plan to use the Impedance Measurement block in a three-phase circuit, you can use the **Multiplication factor** parameter to rescale the measured impedance. For example, measuring the impedance between two phases of a three-phase circuit gives two times the positive-sequence impedance. Therefore you must apply a multiplication factor of 1/2 to the impedance in order to obtain the correct positive-sequence impedance value.

Similarly, to measure the zero-sequence impedance of a balanced three-phase circuit, you can connect the Impedance Measurement block between ground or neutral and the three phases connected together.

In that case, you are measuring one third of the zero-sequence impedance and you must apply a multiplication factor of 3 to obtain the correct zero-sequence value.

Limitations

The only nonlinear blocks that are taken into account during the impedance measurement are the Breaker, Three-Phase Breaker, Three-Phase Fault, Ideal Switch, and Distributed Parameter Line blocks. All other nonlinear blocks, such as machines and power electronic devices, are not considered, and they are disconnected during the measurement.

If you plan to connect the Impedance Measurement block in series with an inductance, a current source, or any nonlinear element, you must add a large resistor across the terminals of the block, because the Impedance Measurement block is simulated as a current source block.

Example

See the Powergui block reference page for an example using the Impedance Measurement block.

See Also

Powergui

Linear Transformer

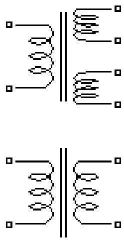
Purpose

Implement a two-winding or three-winding linear transformer

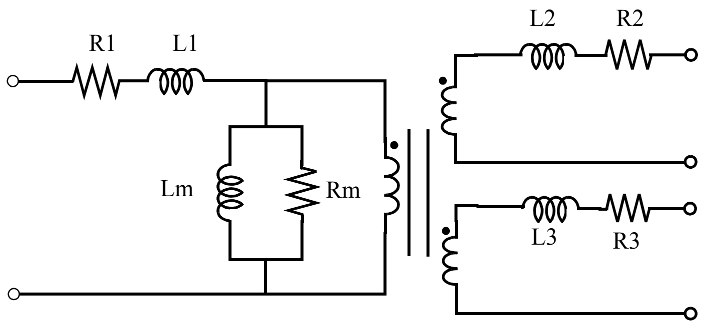
Library

Elements

Description



The Linear Transformer block model shown consists of three coupled windings wound on the same core.



The model takes into account the winding resistances (R1 R2 R3) and the leakage inductances (L1 L2 L3), as well as the magnetizing characteristics of the core, which is modeled by a linear (Rm Lm) branch.

The Per Unit Conversion

In order to comply with industry, you must specify the resistance and inductance of the windings in per unit (p.u.). The values are based on the transformer rated power Pn, in VA, nominal frequency fn, in Hz, and nominal voltage Vn, in Vrms, of the corresponding winding. For each winding, the per unit resistance and inductance are defined as

$$R(\text{p.u.}) = \frac{R(\Omega)}{R_{base}}$$

$$L(\text{p.u.}) = \frac{L(H)}{L_{base}}$$

The base resistance and base inductance used for each winding are

$$R_{base} = \frac{(Vn)^2}{Pn}$$

$$L_{base} = \frac{R_{base}}{2\pi fn}$$

For the magnetization resistance Rm and inductance Lm, the p.u. values are based on the transformer rated power and on the nominal voltage of winding 1.

For example, the default parameters of winding 1 specified in the dialog box section give the following bases:

$$R_{base} = \frac{(735e3/(\sqrt{3}))^2}{250e6} = 720.3\Omega \qquad L_{base} = \frac{720.3}{2\pi 60} = 1.91H$$

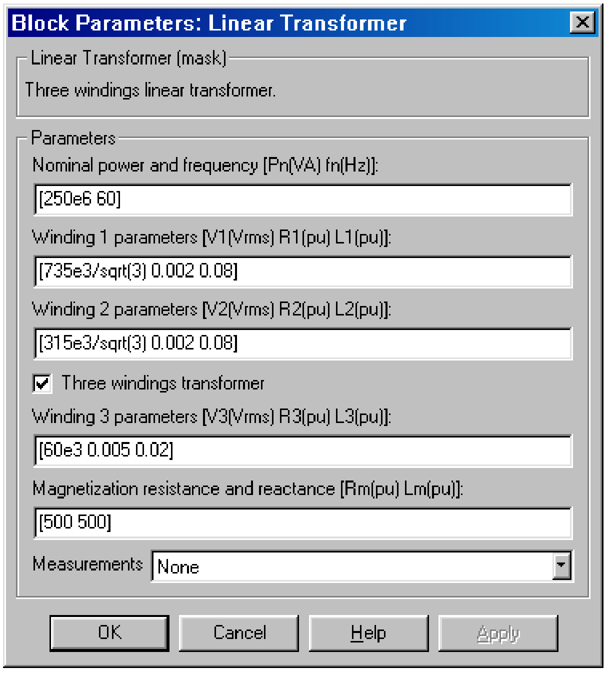
Suppose that the winding 1 parameters are R1 = 1.44 Ω and L1 = 0.1528 H; the corresponding values to be entered in the dialog box are

$$R_1 = \frac{1.44\Omega}{720.3\Omega} = 0.002 \text{ p.u.}$$

$$L_1 = \frac{0.1528H}{1.91H} = 0.08 \text{ p.u.}$$

To specify a magnetizing current of 0.2% (resistive and inductive) based on nominal current, you must enter per unit values of 1/0.002 = 500 p.u. for the resistance and the inductance of the magnetizing branch. Using the base values calculated previously, these per unit values correspond to Rm = 8.6e5 ohms and Lm = 995 henries.

Dialog Box and Parameters



Nominal power and frequency

The nominal power rating Pn in volt-amperes (VA) and frequency fn, in hertz (Hz), of the transformer.

Winding 1 parameters

The nominal voltage V, in volts RMS, resistance, and leakage inductance in p.u. The p.u. values are based on the nominal power Pn and on V1.

Winding 2 parameters

The nominal voltage V2 in volts RMS, resistance, and leakage inductance in p.u. The p.u. values are based on the nominal power Pn and on V2.

Three windings transformer

If selected, implements a linear transformer with three windings; otherwise, it implements a two-windings transformer.

Winding 3 parameters

The **Winding 3 parameters** parameter is not available if the **Three windings transformer** parameter is not selected.

The nominal voltage in volts RMS (Vrms), resistance, and leakage inductance in p.u. The p.u. values are based on the nominal power Pn and on V3.

Magnetization resistance and reactance

The resistance and inductance simulating the core active and reactive losses, both in p.u. The p.u. values are based on the nominal power Pn and on V1. For example, to specify 0.2% of active and reactive core losses, at nominal voltage, use Rm = 500 p.u. and Lm = 500 p.u.

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals of the Linear Transformer block.

Select `Winding currents` to measure the current flowing through the windings of the Linear Transformer block.

Select `Magnetization current` to measure the magnetization current of the Linear Transformer block.

Select `All voltages and currents` to measure the winding voltages and currents plus the magnetization current.

Place a Multimeter block in your model to display the selected measurements during the simulation.

In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement	Label
Winding voltages	`Uw1:, Uw2:, Uw3:`
Winding currents	`Iw1:, Iw2:, Iw3:`
Magnetization current	`Imag:`

Linear Transformer

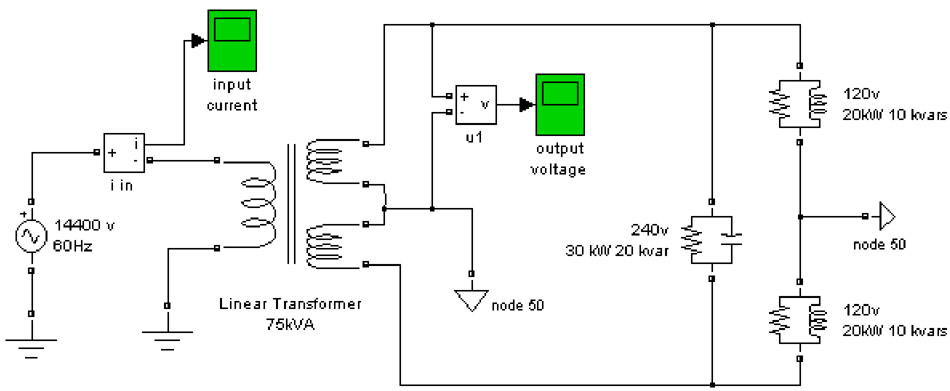
Note To implement a quasi-ideal transformer model, set the winding resistances and inductances to 0, and the magnetization inductance Lm to `inf`. The Rm value must have a finite value. Use a large value such as 1e4 (0.01% losses).

Limitations

Windings can be left floating (that is, not connected to the rest of the circuit). However, an internal resistor is automatically added between the floating winding and the main circuit. This internal connection does not affect voltage and current measurements.

Example

The `power_transformer` demo shows a typical residential distribution transformer network feeding line-to-neutral and line-to-line loads.



See Also

Mutual Inductance, Saturable Transformer, Three-Phase Transformer (Two Windings), Three-Phase Transformer (Three Windings)

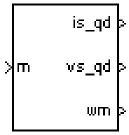
Purpose

Split measurement signal of machine models into separate signals

Library

Machines

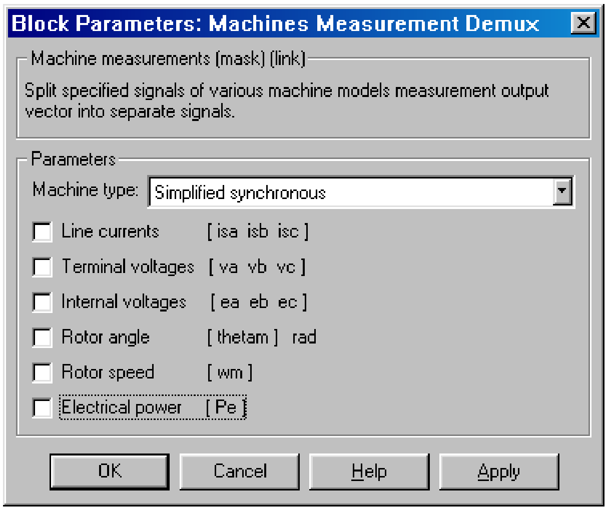
Description



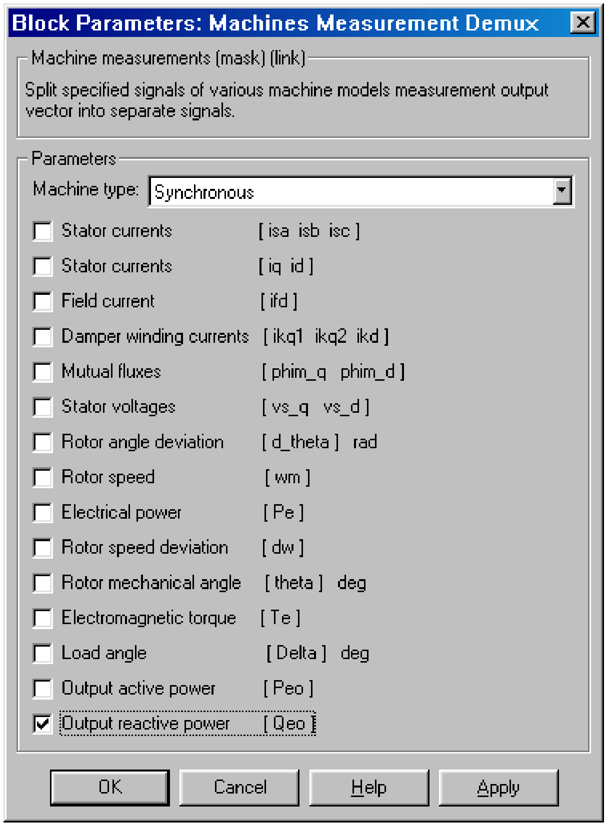
The Machine Measurement Demux block is used to demux the measurement signals of the Simplified Synchronous Machine, the Synchronous Machine, the Asynchronous Machine, and the Permanent Magnet Synchronous Machine blocks.

The Machine Measurement Demux block is connected directly to the measurement output of the machine blocks. You select the type of machine connected to the block and you select the measurements you want to observe. An output is added to the block for each measurement in the list.

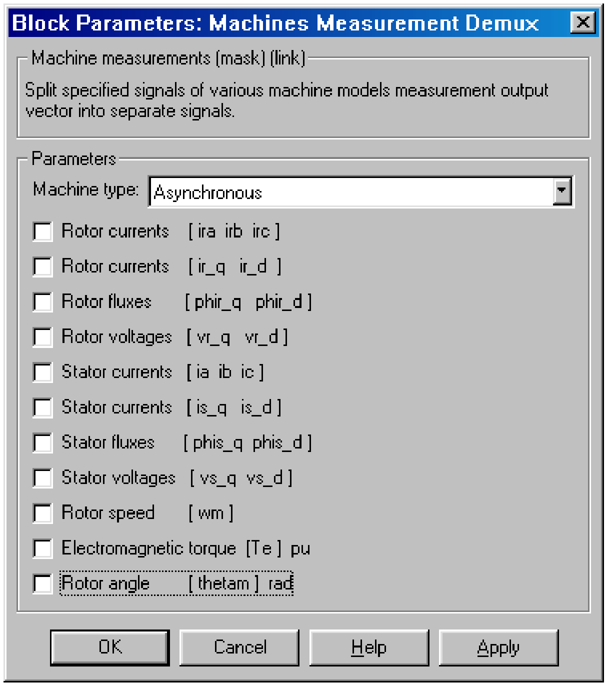
Dialog Box and Parameters



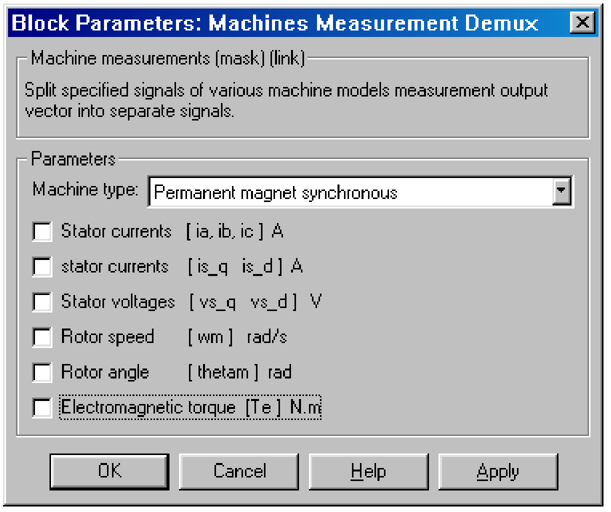
Machines Measurement Demux dialog: Simplified synchronous type



Machines Measurement Demux dialog: Synchronous type



Machines Measurement Demux dialog: Asynchronous type



Machines Measurement Demux dialog: Permanent magnet synchronous type

Machine Type

Set to `Simplified synchronous` to display the measurement list for the Simplified Synchronous Machine block.

Set to `Synchronous` to display the measurement list for the Synchronous Machine block.

Set to `Asynchronous` to display the measurement list for the Asynchronous Machine block.

Set to `Permanent magnet synchronous` to display the measurement list for the Permanent Magnet Synchronous Machine block.

Measurement list

Select the block parameters you want to output.

See Also

Asynchronous Machine, Permanent Magnet Synchronous Machine, Simplified Synchronous Machine, Synchronous Machine

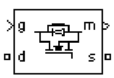
Purpose

Implement a MOSFET model

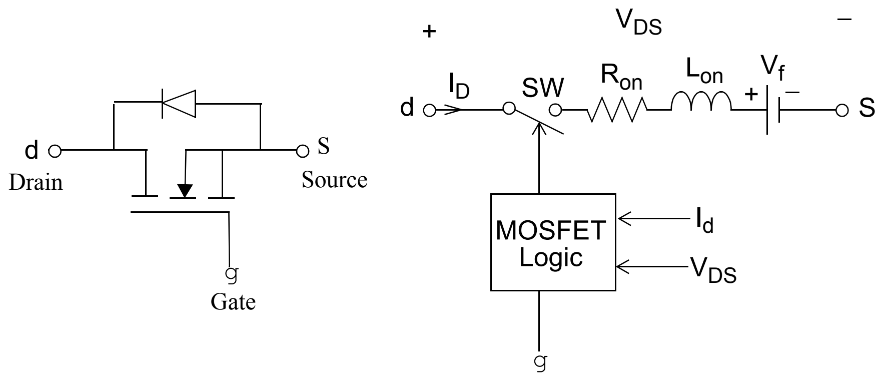
Library

Power Electronics

Description



The metal-oxide semiconductor field-effect transistor (MOSFET) is a semiconductor device controllable by the gate signal (g > 0) if its current Id is positive (Id > 0). The MOSFET device is connected in parallel with an internal diode that turns on when the MOSFET device is reverse biased (Vds < 0). The model is simulated as a series combination of a variable resistor (R_t) and inductor (L_{on}) in series with a switch controlled by a logical signal (g > 0 or g = 0).



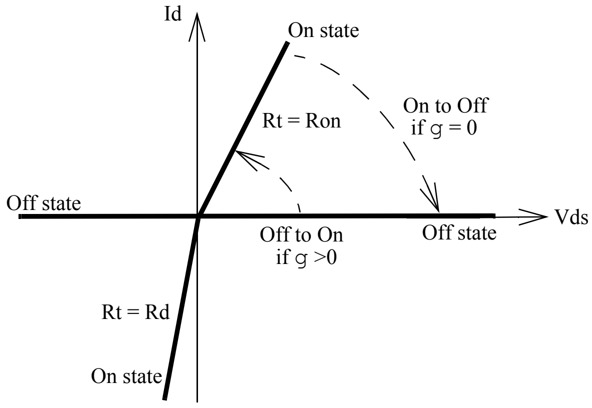
The MOSFET device turns on when the drain-source voltage is positive and a positive signal is applied at the gate input (g > 0).

With a positive current flowing through the device, the MOSFET turns off when the gate input becomes 0. If the current Id is negative (Id flowing in the internal diode) and without a gate signal (g = 0), the MOSFET turns off when the current Id becomes 0 (Id = 0).

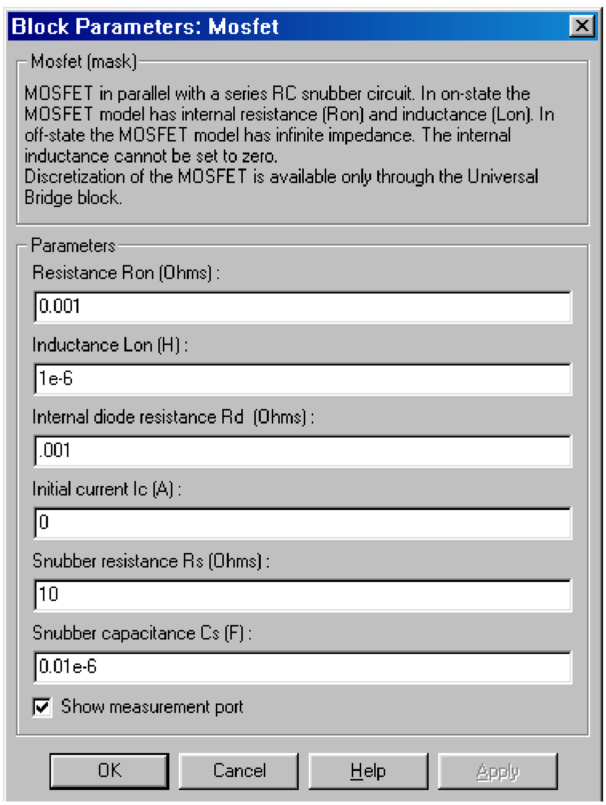
Note that the on state resistance Rt depends on the drain current direction:

- Rt = Ron if Id > 0, where Ron represents the typical value of the forward conducting resistance of the MOSFET device.
- Rt = Rd if Id < 0, where Rd represents the internal diode resistance.

The MOSFET block also contains a series Rs-Cs snubber circuit that can be connected in parallel with the MOSFET (between nodes d and s).



Dialog Box and Parameters



Resistance Ron

The internal resistance Ron, in ohms (Ω).

Inductance Lon

The internal inductance Lon, in henries (H). The **Inductance Lon** parameter cannot be set to 0.

Internal diode resistance Rd

The internal resistance of the internal diode, in ohms (Ω).

Initial current Ic

You can specify an initial current flowing in the MOSFET device. It is usually set to 0 in order to start the simulation with the device blocked.

If the **Initial Current IC** parameter is set to a value greater than 0, the steady state calculation of SimPowerSystems considers the initial status of the MOSFET as closed. Initializing all states of a power electronic converter is a complex task. Therefore, this option is useful only with simple circuits.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance, in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, add a Simulink output to the block returning the MOSFET current and voltage.

Inputs and Outputs

The input port is a logical signal applied to the gate. The output port is a measurement vector [Id Vds] returning the MOSFET device current and voltage.

Assumptions and Limitations

The MOSFET block implements a macromodel of the real MOSFET device. It does not take into account either the geometry of the device or the complex physical processes [1].

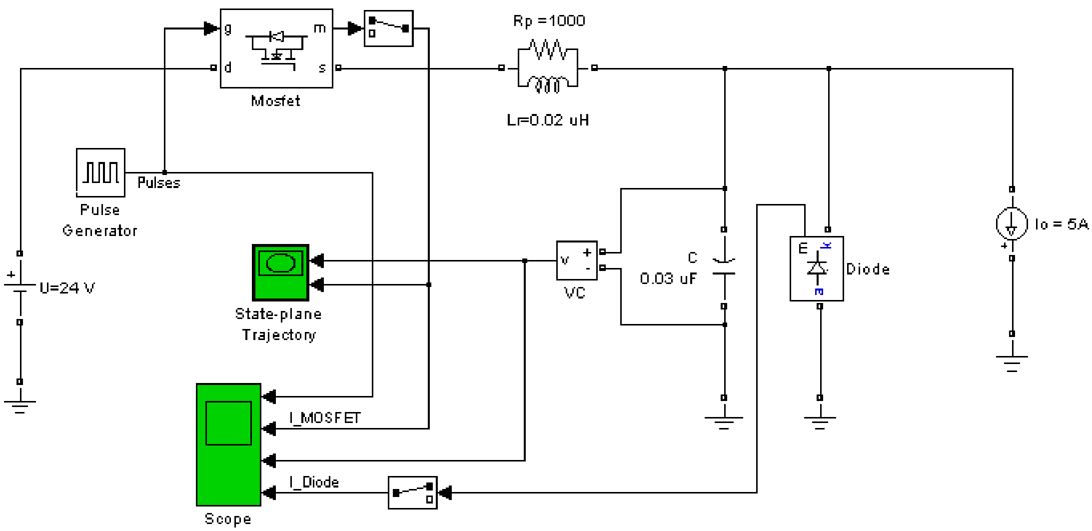
The MOSFET block is modeled as a current source. It cannot be connected in series with an inductor, a current source, or an open circuit, unless its snubber circuit is in use. In order to avoid an algebraic loop, you cannot set the MOSFET block inductance Lon to 0. Each MOSFET block adds an extra state to the electrical circuit model. See the “Improving Simulation Performance” chapter for more details on this topic.

Circuits containing individual MOSFET blocks cannot be discretized. However discretization is permitted for MOSFET/Diode bridges simulated with the Universal Bridge block or the Three-Level Bridge block.

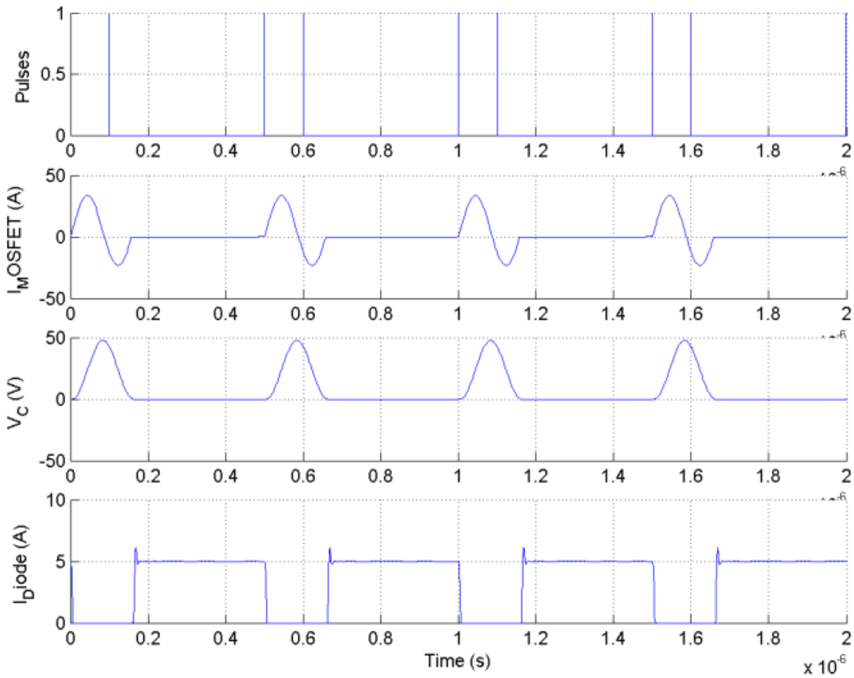
You must use a stiff integrator algorithm to simulate circuits containing MOSFETs. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

Example

The `power_mosconv` demo illustrates the use of the MOSFET block in a zero-current quasi-resonant switch converter. In such a converter, the current produced by the Lr-Cr resonant circuit flows through the MOSFET and internal diode. The negative current flows through the internal diode that turns off at 0 current [1]. The switching frequency is 2 MHz and the pulse width is 72 degrees (duty cycle: 20%).



Run the simulation and observe the gate pulse signal, the MOSFET current, the capacitor voltage, and the diode current on the four-trace Scope block.



References

[1] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

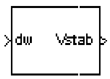
Diode, GTO, Ideal Switch, Three-Level Bridge, Thyristor, Universal Bridge

Multiband Power System Stabilizer

Purpose Implement a multiband power system stabilizer

Library Machines

Description



The disturbances occurring in a power system induce electromechanical oscillations of the electrical generators. These oscillations, also called power swings, must be effectively damped to maintain the system's stability. Electromechanical oscillations can be classified in four main categories:

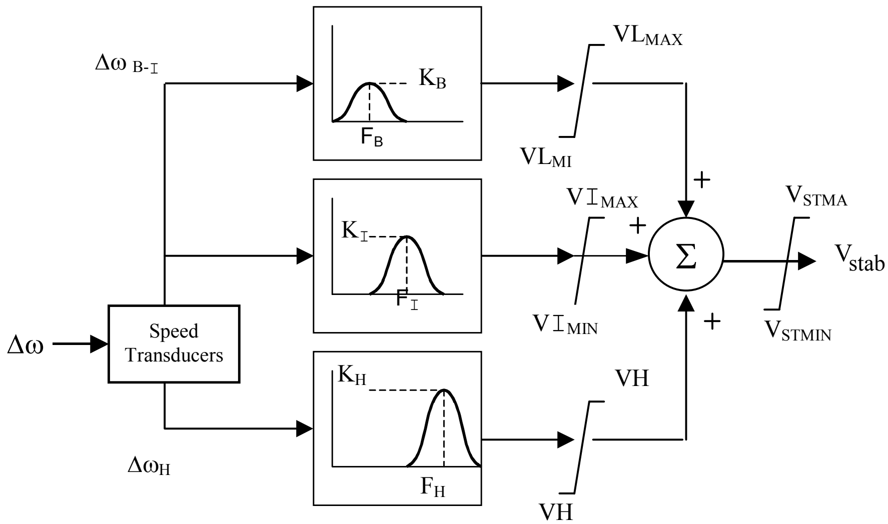
- Local oscillations: between a unit and the rest of the generating station and between the latter and the rest of the power system. Their frequencies typically range from 0.8 to 4.0 Hz.
- Interplant oscillations: between two electrically close generation plants. Frequencies can vary from 1 to 2 Hz.
- Interarea oscillations: between two major groups of generation plants. Frequencies are typically in a range of 0.2 to 0.8 Hz.
- Global oscillation: characterized by a common in-phase oscillation of all generators as found on an isolated system. The frequency of such a global mode is typically under 0.2 Hz.

The need for effective damping of such a wide range, almost two decades, of electromechanical oscillations motivated the concept of the multiband power system stabilizer (MB-PSS).

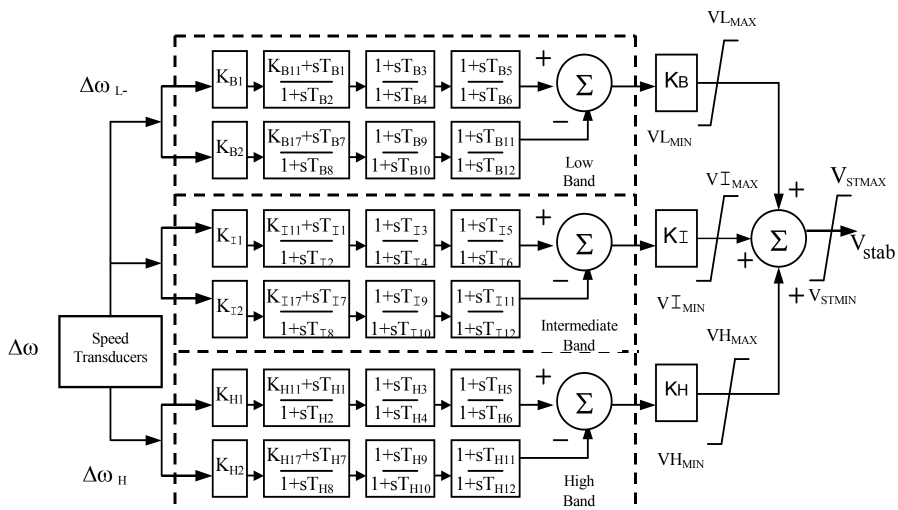
As its name reveals, the MB-PSS structure is based on multiple working bands. Three separate bands are used, respectively dedicated to the low-, intermediate-, and high-frequency modes of oscillations: the low band is typically associated with the power system global mode, the intermediate with the interarea modes, and the high with the local modes.

Each of the three bands is made of a differential bandpass filter, a gain, and a limiter (see Figure). The outputs of the three bands are summed and passed through a final limiter producing the stabilizer output V_{stab}. This signal then modulates the set point of the generator voltage regulator so as to improve the damping of the electromechanical oscillations.

To ensure robust damping, the MB-PSS should include a moderate phase advance at all frequencies of interest to compensate for the inherent lag between the field excitation and the electrical torque induced by the MB-PSS action.



Conceptual Representation



Internal Specifications

The MB-PSS is represented by the IEEE St. 421.5 PSS 4B type model [2], illustrated in Figure , with built-in speed transducers whose parameters are fixed according to manufacturer's specifications.

Generally, only a few of the lead-lag blocks in Figure should be used in a given PSS application. Two different approaches are available to configure the settings in order to facilitate the tuning process:

1. Simplified settings:

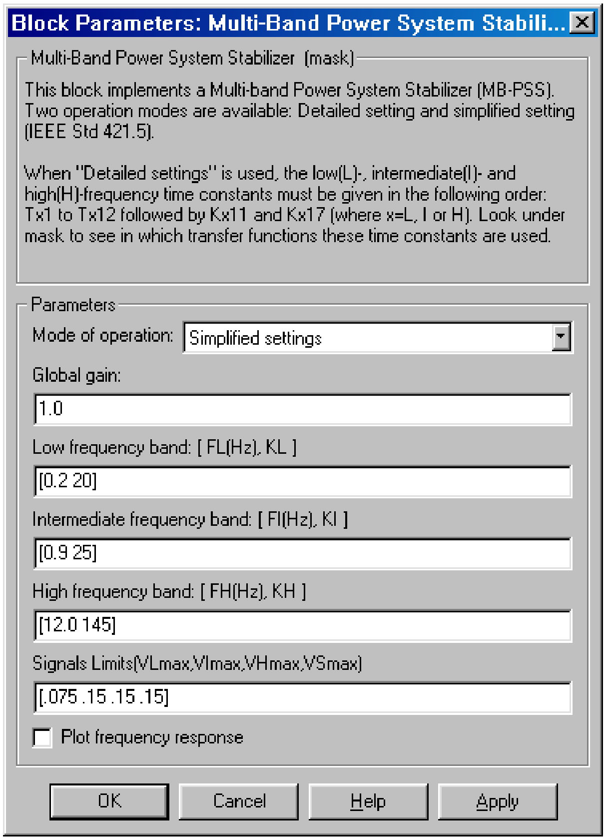
   Only the first lead-lag block of each frequency band is used to tune the Multiband Power System Stabilizer block. The differential filters are assumed to be symmetrical bandpass filters respectively tuned at the center frequency F_L, F_I, and F_H. The peak magnitude of the frequency responses (Figure) can be adjusted independently through the three gains K_L, K_I, and K_H. Only six parameters are therefore required for a simplified tuning of the MB-PSS.

2. Detailed settings:

   The designer is free to use all the flexibility built into the MB-PSS structure to achieve nontrivial controller schemes and to tackle even the most constrained problem (for example, multi unit plant including an intermachine mode, in addition to a local mode and multiple interarea modes). In this case, all the time constants and gains appearing in Figure have to be specified in the dialog box.

Dialog Box and Parameters

Simplified Settings Mode



Global gain

The overall gain K of the multiband power system stabilizer.

Low frequency band: [FL KL]

The center frequency, in hertz, and peak gain of the low-frequency bandpass filter.

Intermediate frequency band: [FI KI]

The center frequency, in hertz, and peak gain of the intermediate-frequency bandpass filter.

High frequency band: [FH KH]

The center frequency, in hertz, and peak gain of the high-frequency bandpass filter.

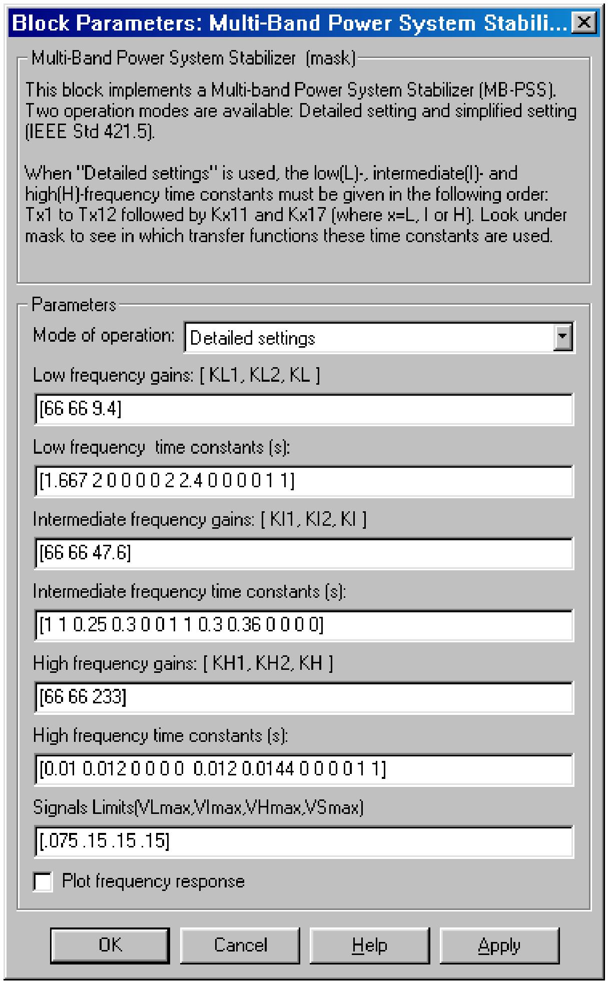
Signal limits [VLmax VImax VHmax VSmax]

The limits imposed on the output of the low-, intermediate-, and high-frequency bands and the limit VSmax imposed on the output of the stabilizer, all in p.u.

Plot frequency response

If selected, a plot of the frequency response of the stabilizer is displayed when you click the **Apply** button.

Detailed Settings Mode



Low frequency gains: [KL1 KL2 KL]

The gains of the positive and negative branches of the differential filter in the low-frequency band and the overall gain K_L of the low-frequency band, in p.u.

Low frequency time constants

The time constants, in seconds, of the lead-lag blocks in the positive and negative branches of the low-frequency filter. You need to specify the following twelve time constants and two gains:

$[T_{B1}\ T_{B2}\ T_{B3}\ T_{B4}\ T_{B5}\ T_{B6}\ T_{B7}\ T_{B8}\ T_{B9}\ T_{B10}\ T_{B11}\ T_{B12}\ K_{B11}\ K_{B17}]$

Set K_{B11} to 0 in order to make the first block of the positive filter branch a washout block. Set K_{B11} to 1 in order to make the block a lead-lag block.

Set K_{B17} to 0 in order to make the first block of the negative filter branch a washout block. Set K_{B17} to 1 in order to make the block a lead-lag block.

Intermediate frequency gains: [KI1 KI2 KI]

The gains of the positive and negative branches of the differential filter in the intermediate-frequency band and the overall gain K_I of the intermediate-frequency band, in p.u.

Intermediate frequency time constants

The time constants, in seconds, of the lead-lag blocks in the positive and negative branches of the intermediate-frequency filter. You need to specify the following twelve time constants and two gains:

$[T_{I1}\ T_{I2}\ T_{I3}\ T_{I4}\ T_{I5}\ T_{I6}\ T_{I7}\ T_{I8}\ T_{I9}\ T_{I10}\ T_{I11}\ T_{I12}\ K_{I11}\ K_{I17}]$

Set K_{I11} to 0 in order to make the first block of the positive filter branch a washout block. Set K_{I11} to 1 in order to make the block a lead-lag block.

Set K_{I17} to 0 in order to make the first block of the negative filter branch a washout block. Set K_{I17} to 1 in order to make the block a lead-lag block.

High frequency gains: [KH1 KH2 KH]

The gains of the positive and negative branches of the differential filter in the high-frequency band and the overall gain K_I of the high-frequency band, in p.u.

High frequency time constants

The time constants, in seconds, of the lead-lag blocks in the positive and negative branches of the high-frequency filter. You need to specify the following twelve time constants and two gains:

$[T_{H1}\ T_{H2}\ T_{H3}\ T_{H4}\ T_{H5}\ T_{H6}\ T_{H7}\ T_{H8}\ T_{H9}\ T_{H10}\ T_{H11}\ T_{H12}\ K_{H11}\ K_{H17}]$

Set K_{H11} to 0 in order to make the first block of the positive filter branch a washout block. Set K_{H11} to 1 in order to make the block a lead-lag block.

Set K_{H17} to 0 in order to make the first block of the negative filter branch a washout block. Set K_{H17} to 1 in order to make the block a lead-lag block.

Signal limits [VLmax VImax VHmax VSmax]

The limits imposed on the output of the low-, intermediate-, and high-frequency bands and the limit VSmax imposed on the output of the stabilizer, all in p.u.

Plot frequency response

If selected, a plot of the frequency response of the stabilizer is displayed when you click the **Apply** button.

Input and Output

`dw`

Connect to the first input the synchronous machine speed deviation *dw* signal (in p.u.).

`Vstab`

The output is the stabilization voltage, in p.u., to connect to the vstab input of the Excitation System block used to control the terminal voltage of the Synchronous Machine block.

Example

See the help text of the `power_PSS` demo model.

References

[1] Grondin, R., I. Kamwa, L. Soulieres, J. Potvin, and R. Champagne, "An approach to PSS design for transient stability improvement through supplementary damping of the common low frequency," *IEEE Transactions on Power Systems*, **8**(3), August 1993, pp. 954-963.

[2] IEEE recommended practice for excitation system models for power system stability studies: *IEEE St. 421.5-2002* (Section 9).

See Also

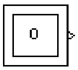
Generic Power System Stabilizer

Multimeter

Purpose Measure the voltages and currents specified in dialog boxes of SimPowerSystems blocks

Library Measurements

Description The Multimeter block is used to measure voltages and currents of the measurements described by the dialog boxes of SimPowerSystems blocks.



The **powerlib** blocks listed in the following table have a special parameter (**Measurements**) that allows you to measure voltages or currents related to the block. Choosing voltages or currents through this measurement parameter is equivalent to connecting an internal voltage or current measurement block inside your blocks. The measured signals can be observed through a Multimeter block placed in your circuit.

Drag the Multimeter block into the top-level system of your circuit and double-click the icon to open the dialog box.

Block Name	Block Name
AC Current Source	PI Section Line
AC Voltage Source	Saturable Transformer
Breaker	Series RLC Branch
Controlled Current Source	Series RLC Load
Controlled Voltage Source	Surge Arrester
DC Voltage Source	Three-Level Bridge
Distributed Parameter Line	Three-Phase Load (Series and Parallel)
Linear Transformer	Three-Phase Branch (Series and Parallel)
Mutual Inductance	Three-Phase Transformer (Two and Three Windings)

Block Name (Continued)	Block Name (Continued)
Parallel RLC Branch	Universal Bridge
Parallel RLC Load	Zigzag Phase-Shifting Transformer

Sign Conventions for Voltages and Currents

When you measure a current using a Current Measurement block, the positive direction of current is indicated on the block icon (positive current flowing from + terminal to – terminal). Similarly, when you measure a voltage using a Voltage Measurement block, the measured voltage is the voltage of the + terminal with respect to the – terminal. However, when voltages and currents of blocks from the Elements library are measured using the Multimeter block, the voltage and current polarities are not immediately obvious because blocks might have been rotated and there are no signs indicating polarities on the block icons.

Unlike Simulink signal lines and input and output ports, the Physical Modeling connection lines and terminal ports □ of SimPowerSystems lack intrinsic directionality. The voltage and current polarities are determined, not by line direction, but instead by block orientation. To find out a block orientation, first click the block to select it. Then enter the following command:

```
get_param(gcb,'Orientation')
```

The following table indicates the polarities of the currents and voltages measured with the Multimeter block for single-phase and three-phase RLC elements (branches or loads), surge arresters, and single-phase and three-phase breakers. The table also indicates the polarities of their state variables (inductor currents and capacitor voltages).

Block Orientation	Positive Current Direction	Measured Voltage
`right`	left —> right	Vleft – Vright
`left`	right —> left	Vright – Vleft

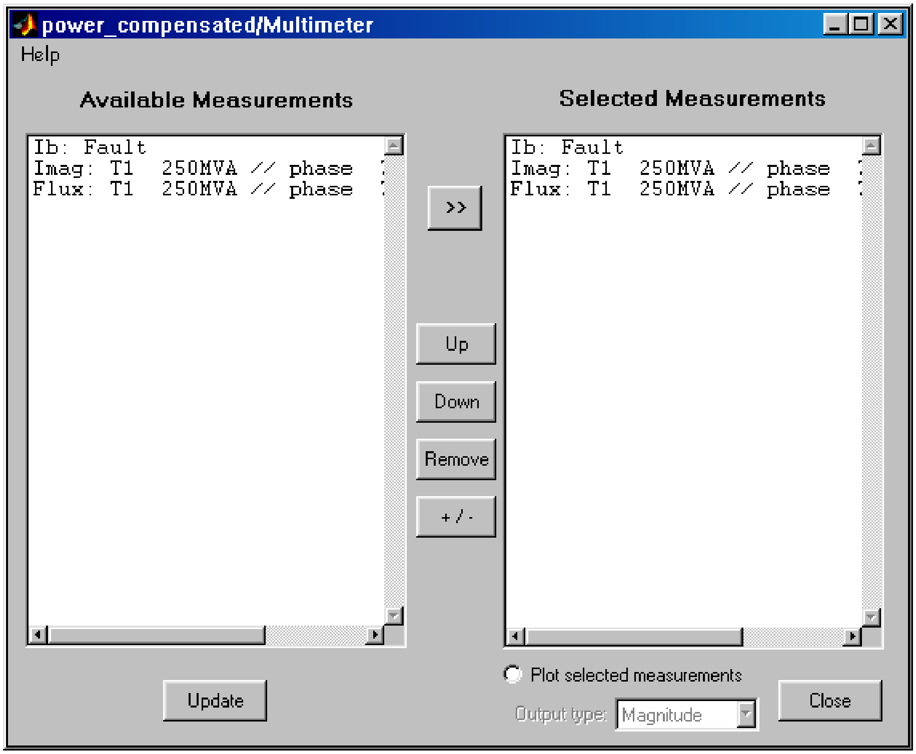
Block Orientation (Continued)	Positive Current Direction (Continued)	Measured Voltage (Continued)
`down`	top —> bottom	Vtop – Vbottom
`up`	bottom —> top	Vbottom – Vtop

The natural orientation of the blocks (that is, their orientation in the Element library) is *right* for horizontal blocks and *down* for vertical blocks.

For single-phase transformers (linear or saturable), with the winding connectors appearing on the left and right sides, the winding voltages are the voltages of the top connector with respect to the bottom connector whatever the block orientation (*right* or *left*). The winding currents are the currents entering the top connector.

For three-phase transformers, the voltage polarities and positive current directions are indicated by the signal labels used in the Multimeter block. For example, Uan_w2 = phase A-to-neutral voltage of the Y connected winding #2, Iab_w1 = winding current flowing from A to B in the delta-connected winding #1.

Dialog Box and Parameters



Available Measurements

The **Available Measurements** list box shows the measurements in the Multimeter block. Use the **>>** button to select measurements from the **Available Measurements** list box. Click the **Update** button to refresh the list of available measurements in the Multimeter block.

The measurements in the list box are identified by the name of the block where the measurement is done. The type of measurement (voltage measurement, current measurement, or flux) is defined by a label preceding the block name. See the reference sections of blocks listed in the previous table for a description of these measurements.

Selected Measurements

The **Selected Measurements** list box shows the measurements sent to the output of the block. You can reorder the measurements by using the **Up**, **Down**, and **Remove** buttons. The **+/–** button allows you to reverse the polarity of any selected measurement.

Plot selected measurements

If selected, displays a plot of selected measurements using a MATLAB figure window. The plot is generated when the simulation stops.

Output type

Specifies the format of the output signals when the block is used in a phasor simulation. The **Output signal** parameter is disabled when the block is not used in a phasor simulation. The phasor simulation is activated by a Powergui block placed in the model.

Set to `Complex` to output the selected measurements as complex values. The outputs are complex signals.

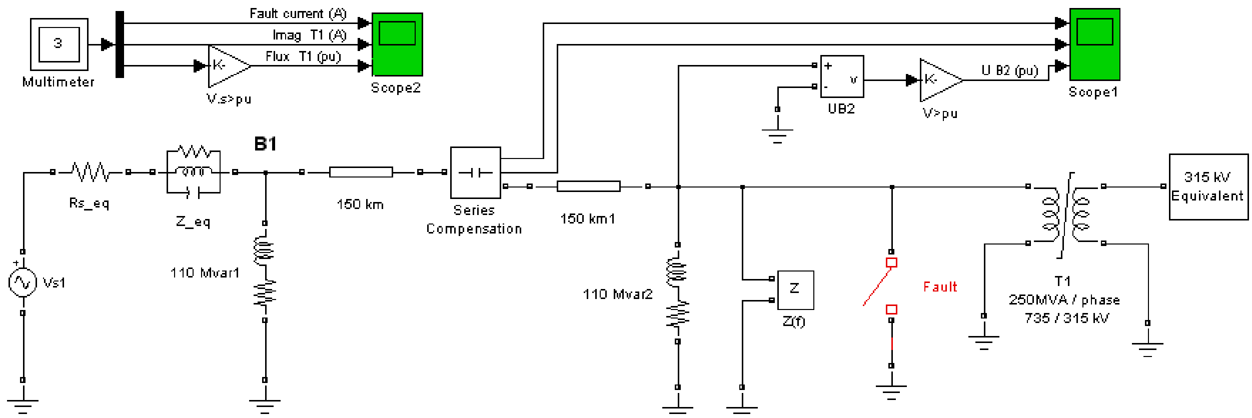
Set to `Real-Imag` to output the real and imaginary parts of the measurements. For each selected measurement, the multimeter outputs the real and imaginary parts.

Set to `Magnitude-Angle` to output the magnitude and angle of the selected measurements. For each selected measurement, the multimeter outputs the magnitude and angle values.

Set to `Magnitude` to output the magnitude of the selected measurements.

Example

The `power_compensated` demo uses a Multimeter block to measure the voltage across the secondary winding of a Saturable Transformer block and the currents flowing through two Series RLC Load blocks.



A Multimeter block is dragged into the model. In the dialog box of the 250 MVA block, set the **Measurements** parameter to `All measurements (V,I,flux)`. In the 110 Mvar block, set it to `Branch voltage` and in the 110 Mvar1 block, set it to `Branch voltage and current`.

The output of the Multimeter block is connected to a Scope block in order to display the measurements during the simulation. In addition, you can select the **Plot selected measurements** parameter to display a plot of selected measurements when simulation stops.

Open the Multimeter block dialog box and select the signals you want to observe, as shown on the Dialog Box and Parameters section. Notice the labels used to define the available measurements in the Multimeter block. See the reference section of the Saturable Transformer block and Series RLC Load block for a description of these labels.

Start the simulation. After 0.4 seconds, the simulation stops and a MATLAB figure window opens to display the selected measurements in the Multimeter block.

See Also

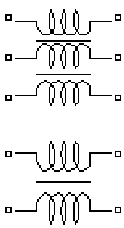
Current Measurement, Voltage Measurement

Mutual Inductance

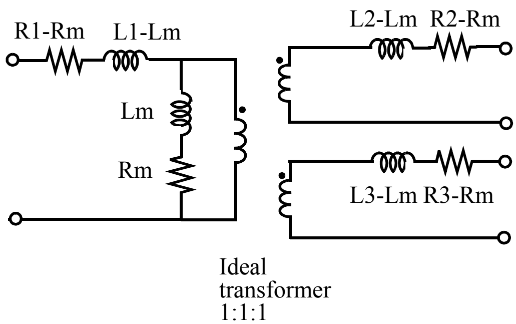
Purpose Implement a magnetic coupling between two or three windings

Library Elements

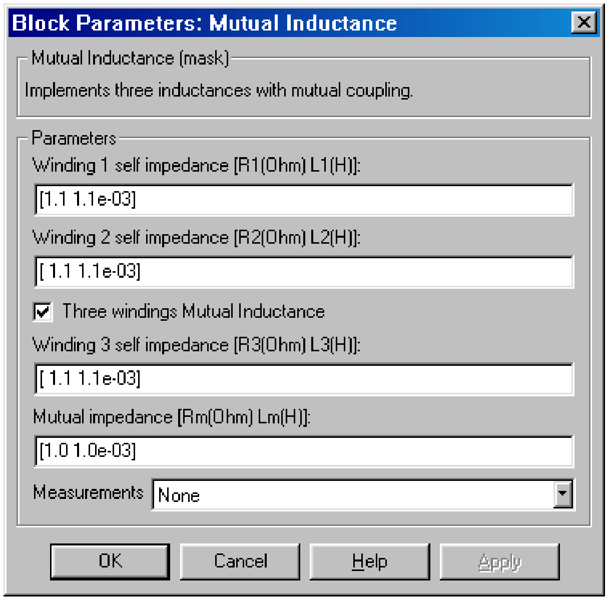
Description The Mutual Inductance block implements a magnetic coupling among three separate windings. Specify the self-resistance and inductance of each winding on the first three entries of the dialog box and the mutual resistance and inductance in the last entry.



The electrical model for this block is given below:



Dialog Box and Parameters



Winding 1 self impedance

The self-resistance and inductance for winding 1, in ohms (Ω) and henries (H).

Winding 2 self impedance

The self-resistance and inductance for winding 2, in ohms (Ω) and henries (H).

Three windings Mutual Inductance

If selected, implements three coupled windings; otherwise, it implements two coupled windings.

Winding 3 self impedance

The **Winding 3 self impedance** parameter is not available if the **Three windings Mutual Inductance** parameter is not selected. The self-resistance and inductance in ohms (Ω) and henries (H) for winding 3.

Mutual impedance

The mutual resistance and inductance between windings, in ohms (Ω) and henries (H). If the mutual resistance and reactance are set to `[0 0]`, the block implements three separate inductances with no mutual coupling.

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals.

Select `Winding currents` to measure the current flowing through the windings.

Select `Winding voltages and currents` to measure the winding voltages and currents.

Place a Multimeter block in your model to display the selected measurements during the simulation.

In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement	Label
Winding voltages	`Uw1:, Uw2:, Uw3:`
Winding currents	`Iw1:, Iw2:, Iw3:`

Inputs and Outputs

Winding 1 is connected between input 1 and output 1 of the Mutual Inductance block. Winding 2 is connected between input 2 and output 2 and winding 3, if defined, is connected between input 3 and output 3.

The input ports of the Mutual Inductance block are at the same instantaneous polarity.

Limitations

Because of modeling constraints, the following restrictions apply:

$$R_1, R_2, R_3 \neq R_m$$

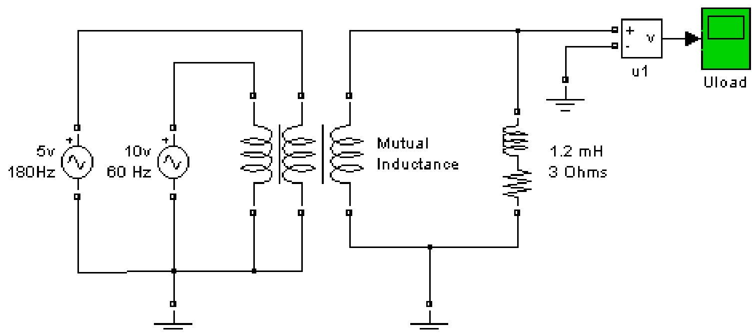
$$L_1, L_2, L_3 \neq L_m$$

Negative values are allowed for the self- and mutual inductances as long as the self-inductances are different from the mutual inductance.

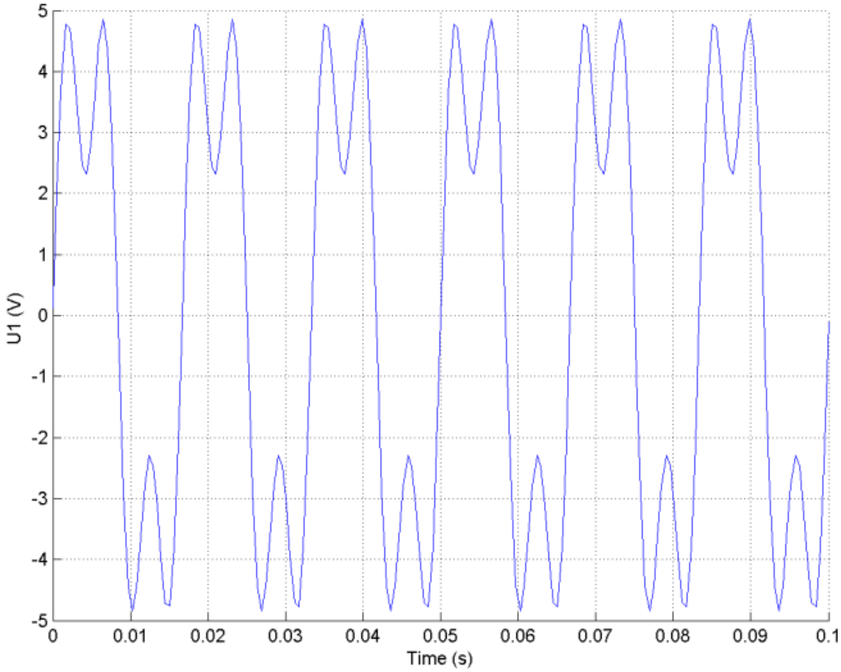
Windings can be left floating (not connected by an impedance to the rest of the circuit). However an internal resistor between the floating winding and the main circuit is automatically added. This internal connection does not affect voltage and current measurements.

Example

The power_mutual demo uses three coupled windings to inject a third harmonic voltage into a circuit fed at 60 Hz.



Simulation produces the following load voltage waveform:



See Also

Linear Transformer, Saturable Transformer, Three-Phase Mutual Inductance Z1-Z0

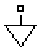
Purpose

Implement a common node in the circuit

Library

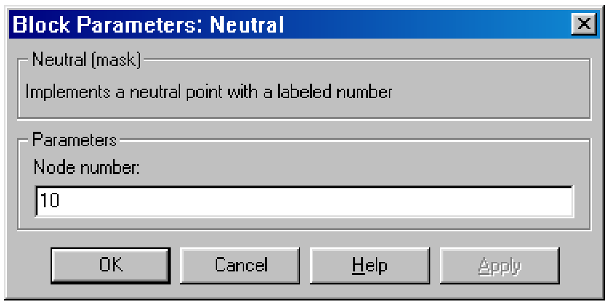
Elements

Description



The Neutral block implements a common node with a specific node number. You can use this block to create a floating neutral or to interconnect two points without drawing a connection line.

Dialog Box and Parameters

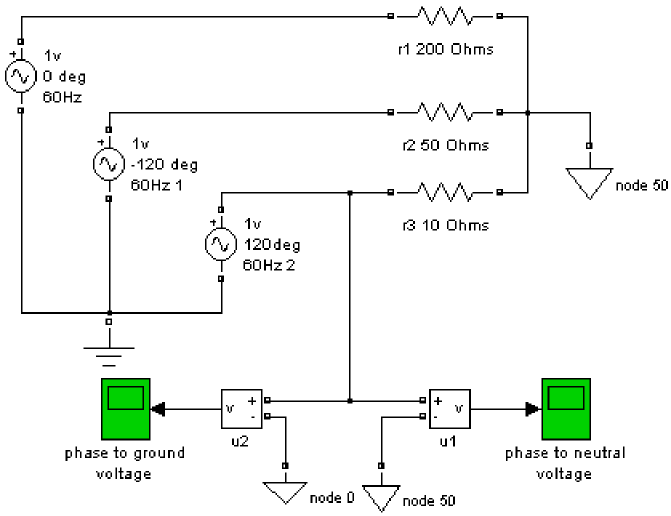


Node number

Specify a number of the neutral node. If the **Node number** parameter is set to `0`, the Neutral block makes a connection to ground. The node number is displayed next to the block icon.

Example

The `power_neutral` demo uses three Neutral blocks. One Neutral block is used to refer a Voltage Measurement block to the ground (node 0).



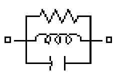
See Also

Ground

Purpose Implement a parallel RLC branch

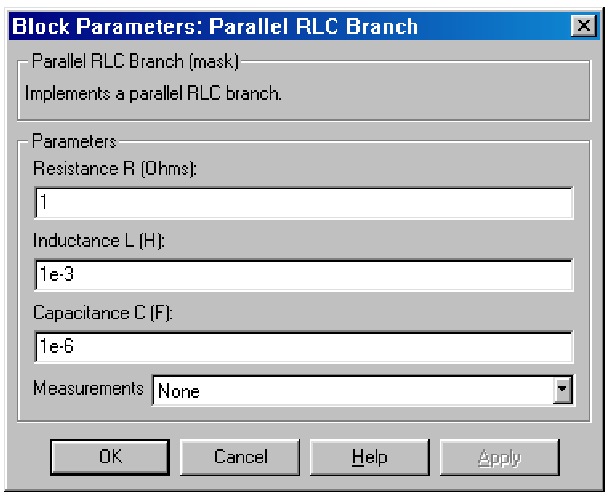
Library Elements

Description The Parallel RLC Branch block implements a single resistor, inductor, and capacitor or a parallel combination of these. To eliminate either the resistance, inductance, or capacitance of the branch, the R, L, and C values must be set respectively to infinity (inf), infinity (inf), and 0. Only existing elements are displayed in the block icon.



Negative values are allowed for resistance, inductance, and capacitance.

Dialog Box and Parameters



Resistance R

The branch resistance, in ohms (Ω).

Inductance L

The branch inductance, in henries (H).

Capacitance C

The branch capacitance, in farads (F).

Measurements

Select Branch voltage to measure the voltage across the Parallel RLC Branch block terminals.

Select Branch current to measure the total current (sum of R, L, C currents) flowing through the Parallel RLC Branch block.

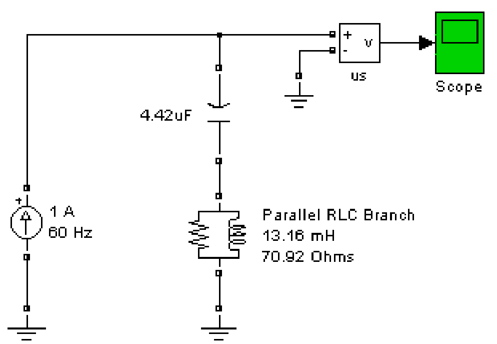
Select Branch voltage and current to measure the voltage and the current of the Parallel RLC Branch block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Branch voltage	Ub:
Branch current	Ib:

Example

The power_paralbranch demo is used to obtain the frequency response of an eleventh-harmonic filter (tuned frequency at 660 Hz) connected on a 60 Hz power system:



The network impedance in the Laplace domain is

$$Z(s) = \frac{V(s)}{I(s)} = \frac{RLCs^2 + Ls + R}{LCs^2 + RCs}$$

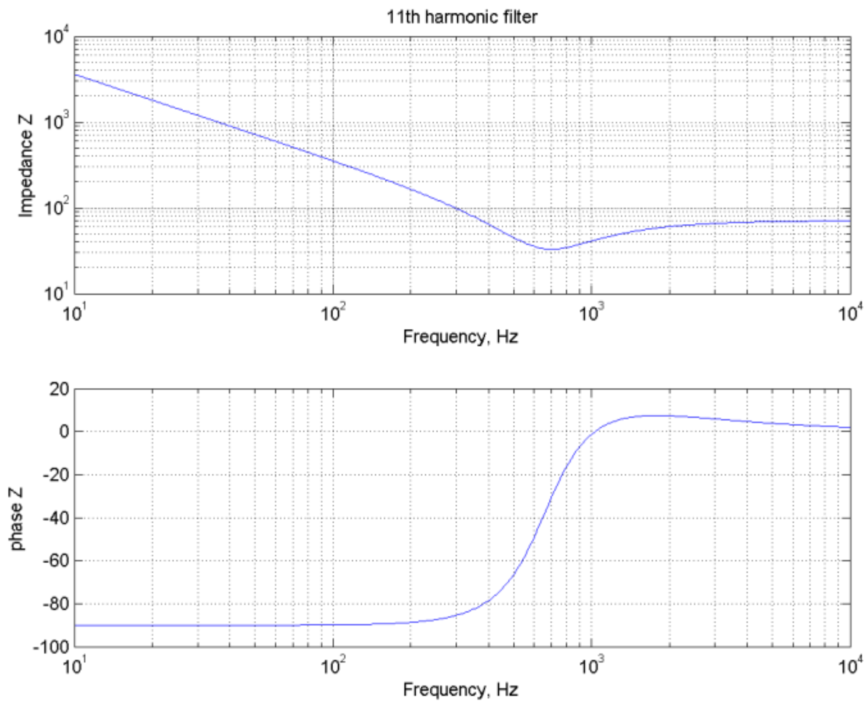
To obtain the frequency response of the impedance you have to get the state-space model (A B C D matrices) of the system.

This system is a one input (Is) and one output (Vs) system.

Note If you have the Control System Toolbox installed, you can get the transfer function *Z(s)* from the state-space matrices and the bode function.

```
[A,B,C,D] = power_analyze('power_paralbranch');
freq = logspace(1,4,500);
w = 2*pi*freq;
[Zmag,Zphase] = bode(A,B,C,D,1,w);
subplot(2,1,1)
loglog(freq,Zmag)
grid
title('11th harmonic filter')
xlabel('Frequency, Hz')
ylabel('Impedance Z')
subplot(2,1,2)
semilogx(freq,Zphase)
xlabel('Frequency, Hz')
ylabel('phase Z')
grid
```

You can also use the Impedance Measurement block and the Powergui block to plot the impedance as a function of frequency.



See Also

Multimeter, Parallel RLC Load, Powergui, Series RLC Branch, Series RLC Load

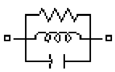
Purpose

Implement a linear parallel RLC load

Library

Elements

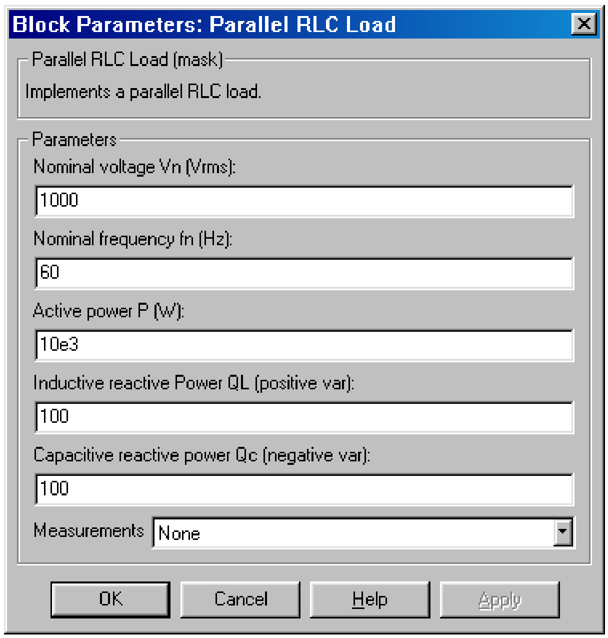
Description



The Parallel RLC Load block implements a linear load as a parallel combination of RLC elements. At the specified frequency, the load exhibits a constant impedance. The active and reactive powers absorbed by the load are proportional to the square of the applied voltage.

Only elements associated with nonzero powers are displayed in the block icon.

Dialog Box and Parameters



Nominal voltage Vn

The nominal voltage of the load, in volts RMS (Vrms).

Nominal frequency fn

The nominal frequency, in hertz (Hz).

Active power P

The active power of the load, in watts.

Inductive reactive power QL

The inductive reactive power QL, in vars. Specify a positive value, or 0.

Capacitive reactive power QC

The capacitive reactive power QC, in vars. Specify a positive value, or 0.

Measurements

Select **Branch voltage** to measure the voltage across the Parallel RLC Load block terminals.

Select **Branch current** to measure the current flowing through the Parallel RLC Load block.

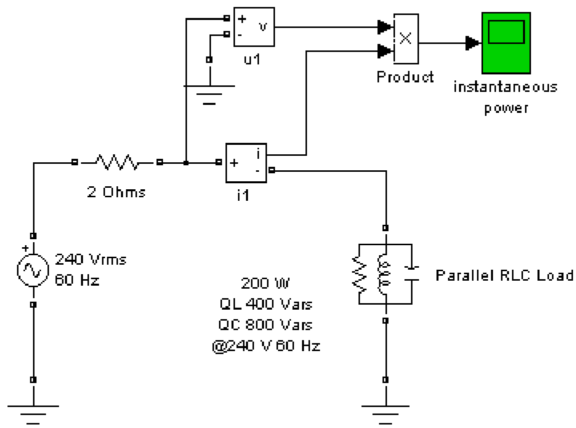
Select **Branch voltage and current** to measure the voltage and the current of the Parallel RLC Load block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Branch voltage	`Ub:`
Branch current	`Ib:`

Example

The `power_paralload` demo uses a parallel RLC load block to implement a load.



See Also

Multimeter, Parallel RLC Branch, Series RLC Branch, Series RLC Load

Permanent Magnet Synchronous Machine

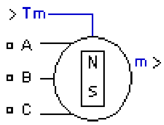
Purpose

Model the dynamics of a three-phase permanent magnet synchronous machine with sinusoidal flux distribution

Library

Machines

Description



The Permanent Magnet Synchronous Machine block operates in either generator or motor mode. The mode of operation is dictated by the sign of the mechanical torque (positive for motor mode, negative for generator mode). The electrical and mechanical parts of the machine are each represented by a second-order state-space model. The model assumes that the flux established by the permanent magnets in the stator is sinusoidal, which implies that the electromotive forces are sinusoidal.

The block implements the following equations expressed in the rotor reference frame (qd frame).

Electrical System

$$\frac{d}{dt}i_d = \frac{1}{L_d}v_d - \frac{R}{L_d}i_d + \frac{L_q}{L_d}p\omega_r i_q$$

$$\frac{d}{dt}i_q = \frac{1}{L_q}v_q - \frac{R}{L_q}i_q - \frac{L_d}{L_q}p\omega_r i_d - \frac{\lambda p\omega_r}{L_q}$$

$$T_e = 1.5p[\lambda i_q + (L_d - L_q)i_d i_q]$$

where (all quantities in the rotor reference frame are referred to the stator)

L_q, L_d	q and d axis inductances
R	Resistance of the stator windings
i_q, i_d	q and d axis currents
v_q, v_d	q and d axis voltages
ω_r	Angular velocity of the rotor
λ	Amplitude of the flux induced by the permanent magnets of the rotor in the stator phases

p	Number of pole pairs
T_e	Electromagnetic torque

Mechanical System

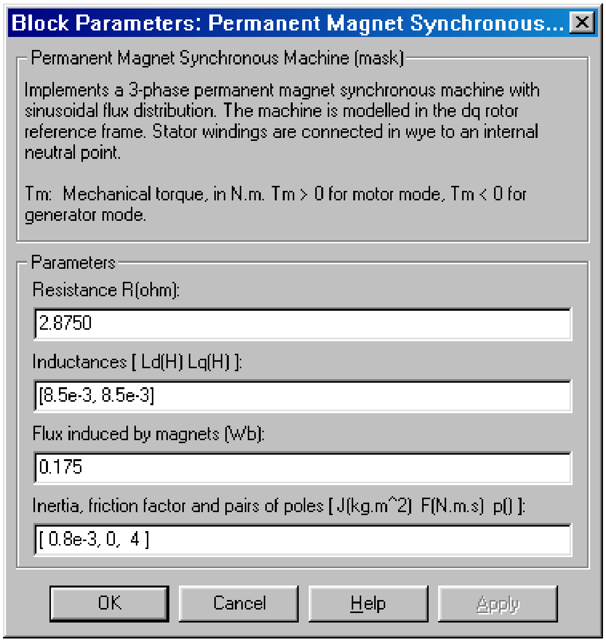
$$\frac{d}{dt}\omega_r = \frac{1}{J}(T_e - F\omega_r - T_m)$$

$$\frac{d\theta}{dt} = \omega_r$$

where

J	Combined inertia of rotor and load
F	Combined viscous friction of rotor and load
θ	Rotor angular position
T_m	Shaft mechanical torque

Dialog Box and Parameters



Resistance

The stator resistance R (Ω).

Inductances

The d-axis and q-axis stator inductances Ld (H) and Lq (H).

Flux induced by magnets

The constant flux λ (Wb) induced in the stator windings by the magnets.

Inertia, friction factor and pairs of poles

The combined machine and load inertia coefficient J (kg.m^2), combined viscous friction coefficient F (N.m.s), and pole pairs p.

Inputs and Outputs

The first three inputs are the electrical connections of the machine's stator. The fourth input is the mechanical torque at the machine's shaft (Simulink signal). This input should normally be positive because the Permanent Magnet Synchronous Machine block is usually used as a motor. Nevertheless, you can apply a negative torque input if you choose to use the block in generator mode.

The block outputs a vector containing the following 10 signals (all currents flowing into machine):

Signal	Definition
1 to 3	Line currents i_a, i_b, and i_c, in A
4 and 5	q and d axis currents i_q and i_d, in A
6 and 7	q and d axis voltages v_q and v_d, in V
8	Rotor mechanical speed ω_r, in rad/s
9	Rotor mechanical angle θ, in rad
10	Electromagnetic torque T_e, in N.m

You can demultiplex these signals by using the Machines Measurement Demux block provided in the Machines library.

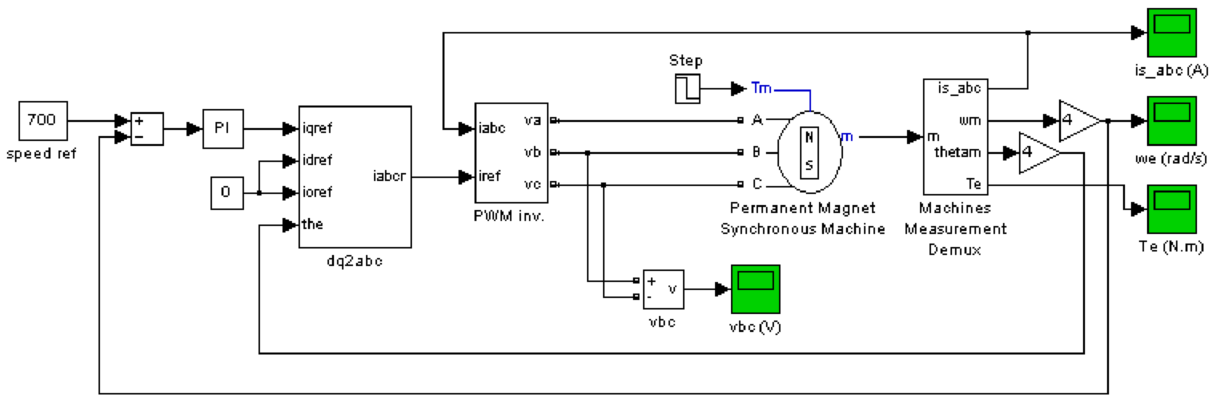
Assumption

The Permanent Magnet Synchronous Machine block assumes a linear magnetic circuit with no saturation of the stator and rotor iron. This assumption can be made because of the large air gap usually found in permanent magnet synchronous machines.

Example

This `power_pmmotor` demo illustrates the use of the Permanent Magnet Synchronous Machine block in motoring mode with a closed-loop control system built entirely in Simulink. The interfacing is done using Controlled Voltage Source blocks from the Electrical Sources library. The complete system consists of a PWM inverter built with ideal switches (Simulink Relay blocks). Two control loops are used; the inner loop is used to regulate the motor line currents and the outer loop regulates the motor's speed. More elaborate and efficient control schemes for the Permanent Magnet Synchronous Machine block can be found, for instance, in [1]. The mechanical torque applied at the motor's shaft is originally 3 N.m (nominal) and steps to 1 N.m at t = 0.04

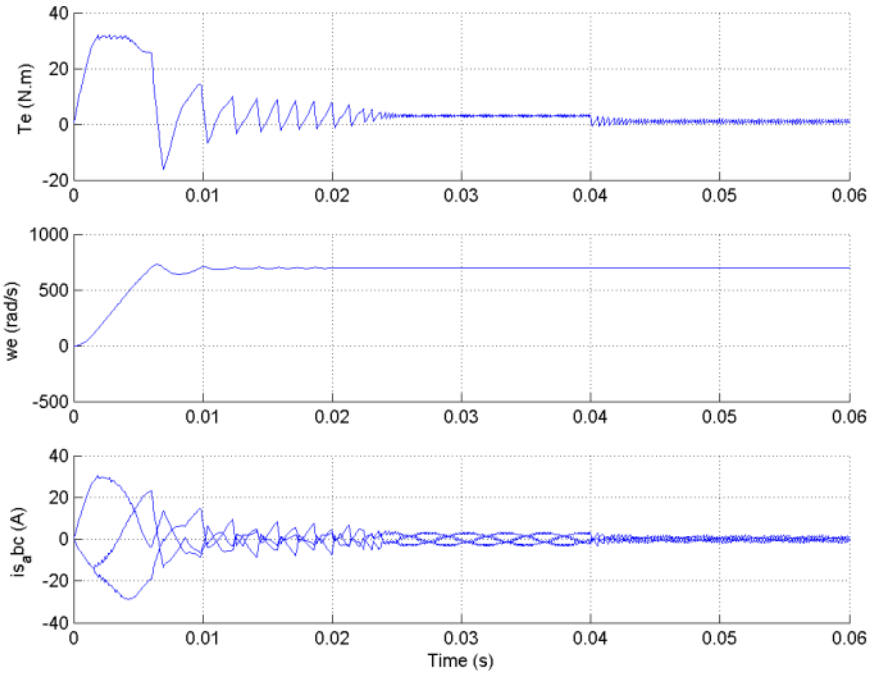
seconds. The parameters of the machine are those found in the dialog box section.



Set the simulation parameters as follows:

- Integrator type: `stiff`, `ode15s`
- Stop time: `0.06`
- Integration options: Use default options, except for absolute tolerance that you can set to `1e-3`

Run the simulation and observe the motor's torque, speed, and currents.



The torque climbs to nearly 32 N.m when the motor starts but stabilizes rapidly to its nominal value (3 N.m), until the step is applied, at which point the torque oscillates slightly before stabilizing to its new value (1 N.m). As for the speed, you can see that it stabilizes quite fast at start-up and is not affected by the load step.

The currents are initially high when the machine starts, like the torque, but stabilize quickly to their nominal values until the step is applied, at which point they oscillate before stabilizing to a lower value, corresponding to the load torque decrease.

References

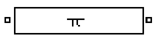
[1] Grenier, D., L.-A. Dessaint, O. Akhrif, Y. Bonnassieux, and B. LePioufle, "Experimental Nonlinear Torque Control of a Permanent Magnet Synchronous Motor Using Saliency," *IEEE Transactions on Industrial Electronics*, Vol. 44, No. 5, October 1997, pp. 680-687.

PI Section Line

Purpose Implement a single-phase transmission line with lumped parameters

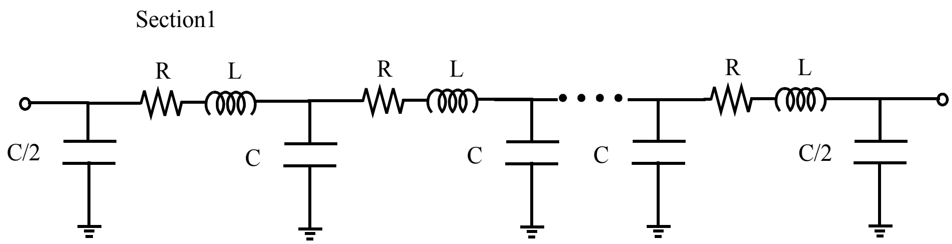
Library Elements

Description



The PI Section Line block implements a single-phase transmission line with parameters lumped in PI sections.

For a transmission line, the resistance, inductance, and capacitance are uniformly distributed along the line. An approximate model of the distributed parameter line is obtained by cascading several identical PI sections, as shown in the following figure.



Unlike the Distributed Parameter Line block, which has an infinite number of states, the PI section linear model has a finite number of states that permit you to compute a linear state-space model. The number of sections to be used depends on the frequency range to be represented.

A good approximation of the maximum frequency range represented by the PI line model is given by the following equation:

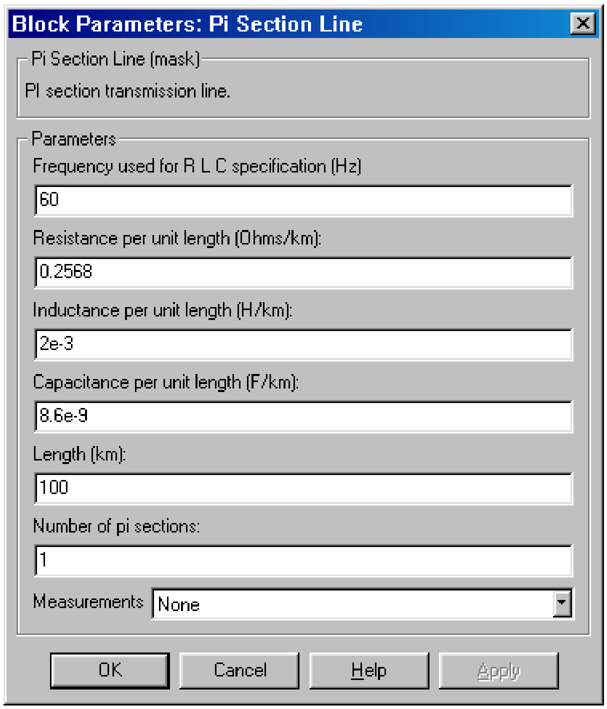
$$f_{max} = \frac{Nv}{8l}$$

where

N	Number of PI sections
v	Propagation speed in km/s = $1/\sqrt{LC}$ L in H/km, C in F/km
l	Line length in km

For example, for a 100 km aerial line having a propagation speed of 300,000 km/s, the maximum frequency range represented with a single PI section is approximately 375 Hz. For studying interactions between a power system and a control system, this simple model could be sufficient. However for switching surge studies involving high-frequency transients in the kHz range, much shorter PI sections should be used. In fact, you can obtain the most accurate results by using a distributed parameters line model.

Dialog Box and Parameters



Frequency used for RLC specifications

Frequency used to compute the line parameters, in hertz (Hz).

Resistance per unit length

The resistance per unit length of the line, in ohms/km (Ω).

Inductance per unit length

The inductance per unit length of the line, in henries/km (H/km).

Capacitance per unit length

The capacitance per unit length of the line, in farads/km (F/km).

Length

The line length in km.

Number of pi sections

The number of PI sections. The minimum value is 1.

Measurements

Select `Input and output voltages` to measure the sending end (input port) and receiving end (output port) voltages of the line model.

Select `Input and output voltages` to measure the sending end and receiving end currents of the line model.

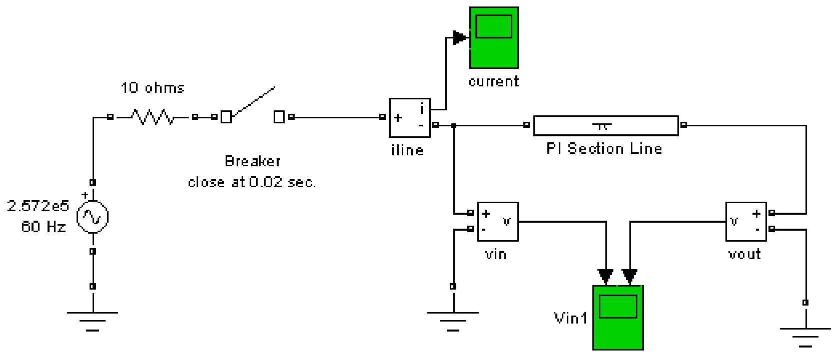
Select `All voltages and currents` to measure the sending end and receiving end voltages and currents of the line model.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

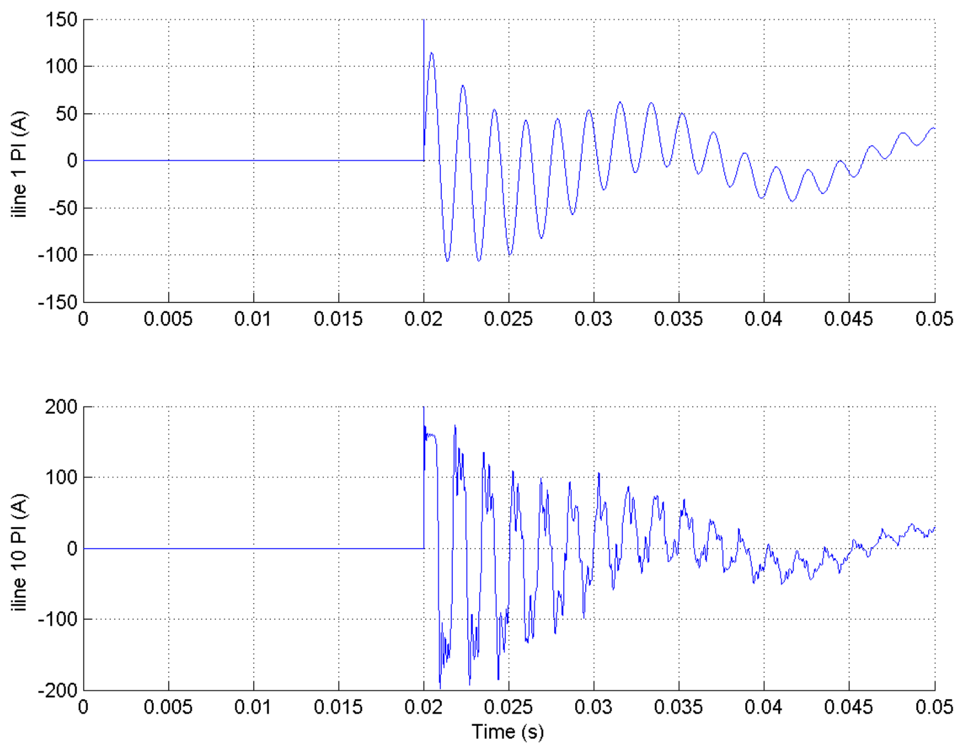
Measurement	Label
Sending end voltage (block input)	`Us:`
Receiving end voltage (block output)	`Ur:`
Sending end current (input current)	`Is:`
Receiving end current (output current)	`Ir:`

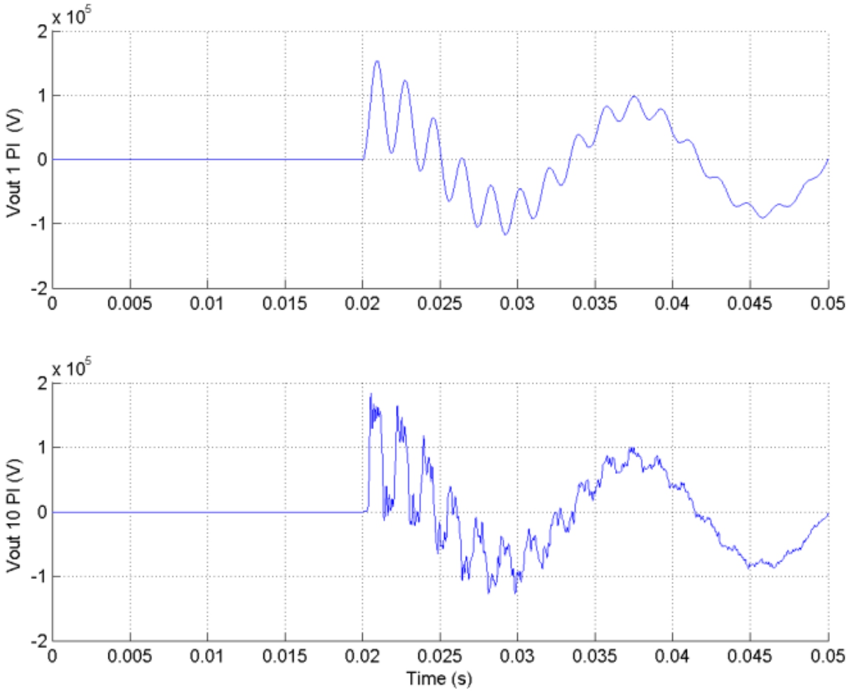
Example

The `power_piline` demo shows the line energization voltages and currents of a PI section line.



The results obtained with the line modeled by one PI section of 100 km and 10 PI sections of 10 km are shown.





See Also Distributed Parameter Line

Purpose

Graphical user interface for the analysis of circuits and systems

Library

powerlib

Description

The Powergui block provides useful graphical user interface (GUI) tools for the analysis of SimPowerSystems models. Copy the Powergui block into the top level of your model and double-click the block to open the interface.

What the Powergui Block Does

The Powergui block allows you to choose one of three methods to solve your circuit:

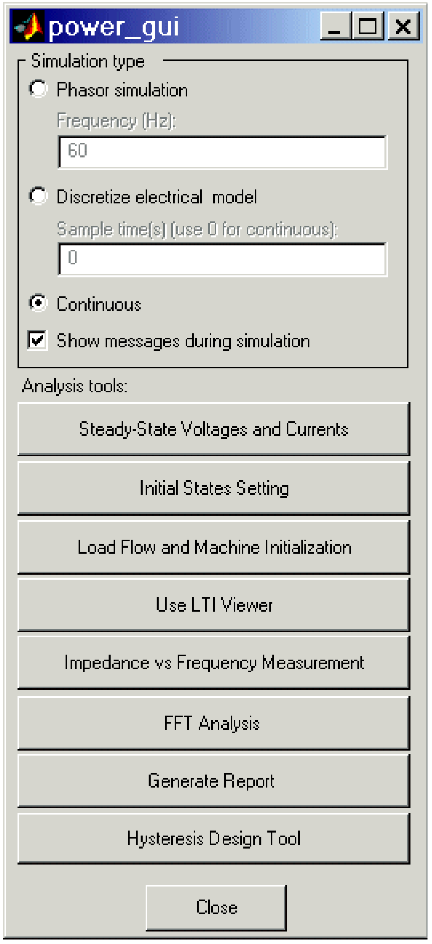
1. Continuous method, which uses a variable step solver from Simulink
2. Discretization of the electrical system for a solution at fixed time steps
3. Phasor solution method

The Powergui block also allows you to

- Display steady-state values of measured current and voltages as well as all state variables (inductor currents and capacitor voltages) in a circuit.
- Modify the initial states in order to start the simulation from any initial conditions. The names of the state variables are the name of the block where the capacitor or the inductor is found, preceded by the `Uc_` label for capacitor voltages and by the `IL_` label for the inductor currents.
- Perform load flows and initialize three-phase networks containing three-phase machines so that the simulation starts in steady state. This option is available with circuits containing the following types of machines: Simplified Synchronous Machine, Synchronous Machine, or Asynchronous Machine (squirrel cage) blocks.
- Display impedance versus frequency plots when Impedance Measurement blocks are present in your circuit.
- Perform FFT analysis of the simulation results.
- Generate the state-space model (SS) of your system (if you have the Control System Toolbox installed) and automatically open the LTI Viewer interface for time and frequency domain responses

- Generate a report containing steady-state values of the measurement blocks, the sources, the nonlinear models, and the states of your circuit. The report is saved in a file with the `.rep` extension.
- Model the hysteresis characteristic of the Saturable Transformer blocks.

Dialog Boxes and Parameters



Simulation Type

Phasor simulation

If selected, SimPowerSystems performs a phasor simulation of the model, at the frequency specified by the **Frequency** parameter.

Frequency (Hz)

Specify the frequency used by SimPowerSystems to perform the phasor simulation of the model. The **Frequency** field is gray if **Phasor simulation** is not selected.

Discretize electrical model

If selected, SimPowerSystems performs a discretization of the model. The sample time is specified by the **Sample time** parameter.

Sample time(s)

Specify the sample time used to discretize the electrical circuit. Set the **Sample time** parameter to a value greater than 0. The icon displays the value of the sample time. If sample time is specified as `0`, discretization is not performed, and the continuous solution method is used. The **Sample time** field is gray if the **Discretize electrical model** parameter is not selected.

Continuous

If selected, SimPowerSystems performs a continuous solution of the model.

Show messages during simulation

If selected, the command line echo messages of SimPowerSystems are enabled during the analysis and simulation of the model.

Analysis Tools

Steady-State Voltages and Currents

Open a window that displays the steady-steady-state voltages and currents of the model.

Initial States Setting

Open a window that allows you to display and modify initial voltages and currents of the model.

Load Flow and Machine Initialization

Open a window to perform load flow and machine initialization.

Use LTI Viewer

Open a window to use the LTI Viewer of the Control System Toolbox.

Impedance vs Frequency Measurement

Open a window that allows you to display the impedance versus frequency measurements performed by the Impedance Measurement blocks of the model.

FFT Analysis

Open a window to use the FFT analysis tool.

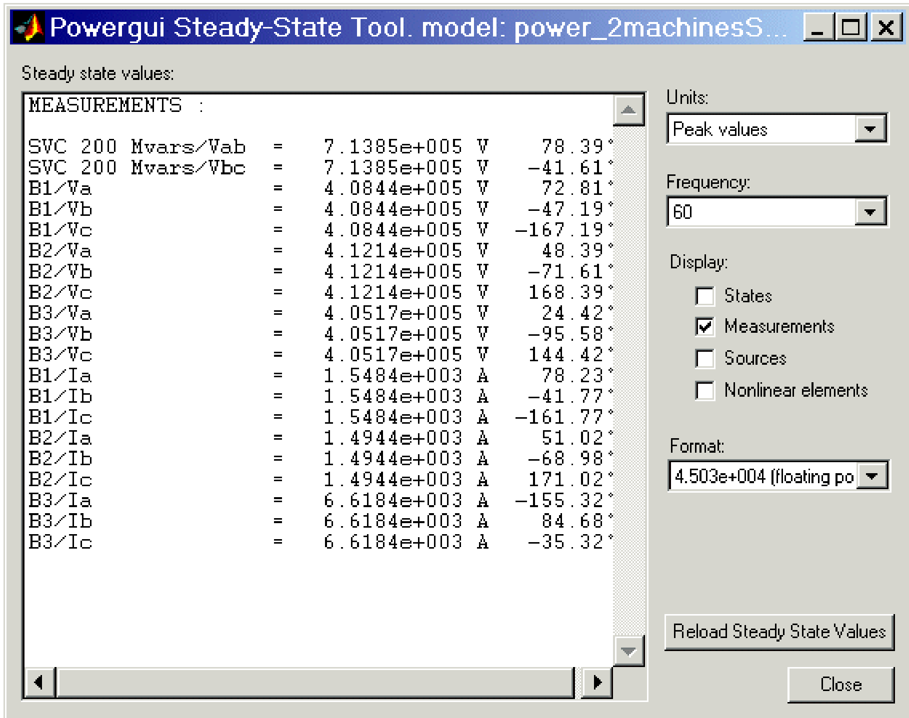
Generate Report

Open a window and generate a report of the steady-state calculations.

Hysteresis Design Tool

Open a window to design a hysteresis characteristic for the saturable core of the Saturable Transformer block and the Three-Phase Transformer blocks (two- and three-windings).

Steady-State Voltages and Currents GUI



Steady state values

Display measurements of steady-state voltages and currents in the model.

Units

Set the **Units** parameter to `Peak values` to display the peak values of the selected values. Set the **Units** parameter to `RMS` to display the root-mean-square (RMS) values of the selected values.

Frequency

Allows you to choose the frequency, in hertz (Hz), that you want for display of the voltage and current phasors. The **Frequency** parameter lists all the different frequencies of the electrical sources of the model.

States

If selected, the window displays the steady-state phasors of the capacitor voltages and inductor currents of the circuit. The default is unselected.

Measurements

If selected, the window displays the steady-state voltage and current phasors of the measurement blocks of the circuit. The default is selected.

Sources

If selected, the window displays the steady-state voltage and current phasors of the electrical sources of the circuit. The default is unselected.

Nonlinear elements

If selected, the window displays the steady-state voltages and currents of the nonlinear blocks of the circuit. The default is unselected.

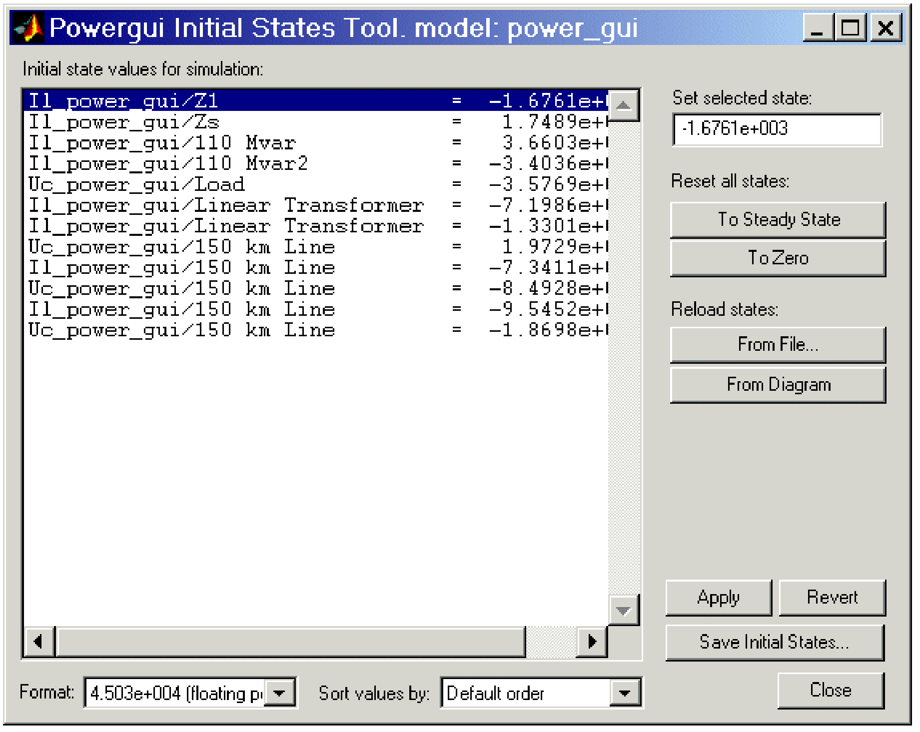
Format

In the pull-down menu, choose the format in which you want your measurements displayed. The `floating point` option is displayed in mantissa-exponent form with five significant figures. The `best of` option displays with four significant figures and uses mantissa-exponent form only for numbers larger than 9999. The final option is displayed in plain numbers with two figures to the right of the decimal point. The default is `floating point`.

Reload Steady State Values

Recompute and redisplay the steady-state measurements.

Initial States Setting GUI



Initial state values for simulation

Display names of model state variables and their initial values.

Set selected state

Enter a value here to set the initial value of the variable selected in the **Initial state values for simulation** list.

Reset all states

If **To Steady State** is selected, sets all initial state values to steady-state values. If **To Zero** is selected, sets all variables to zero.

Reload states

If **From File** is selected, allows you to choose a previously saved file storing the model's states. If **From Diagram** is selected, sets all initial state values to their current values (either steady state values or last modified values).

Apply

Apply the chosen settings to the simulation.

Revert

Reapply the model's original settings from when this GUI was opened.

Save Initial States

Save the model's initial state settings in a file.

Format

In the pull-down menu, choose the format in which you want your measurements displayed. The `floating point` option is displayed in mantissa-exponent form with five significant figures. The `best of` option displays with four significant figures and uses mantissa-exponent form only for numbers larger than 9999. The final option is displayed in plain numbers with two figures to the right of the decimal point. The default is `floating point`.

Sort values by

Select order of displayed initial state values. Selecting `Default order` displays the value by block order in the diagram. Selecting `State number` displays the values according to the states' ordering in the state-space model. Selecting `Type` displays the values grouped by capacitors and inductors. The default is `Default order`.

Sign Conventions for Voltages and Currents

Unlike Simulink signal lines and input and output ports, the Physical Modeling connection lines and terminal ports □ of SimPowerSystems lack intrinsic directionality. The voltage and current polarities are determined, not by line direction, but instead by block orientation. To find out a block orientation, first click on the block to select it. Then enter the following command:

```
get_param(gcb,'Orientation')
```

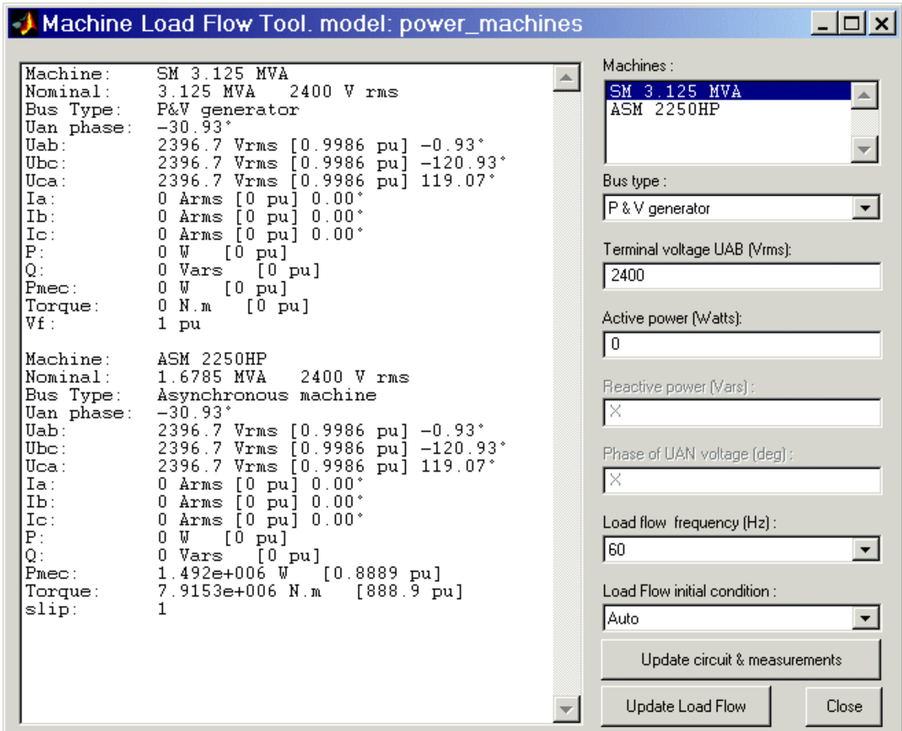
The following table indicates the polarities of the currents and voltages for single-phase and three-phase RLC elements (branches or loads), surge arresters, and single-phase and three-phase breakers. The table also indicates the polarities of their state variables (inductor currents and capacitor voltages).

Block Orientation	Positive Current Direction	Measured Voltage
`right`	left —> right	Vleft – Vright
`left`	right —> left	Vright – Vleft
`down`	top —> bottom	Vtop – Vbottom
`up`	bottom —> top	Vbottom – Vtop

The natural orientation of the blocks (that is, their orientation in the Element library) is *right* for horizontal blocks and *down* for vertical blocks.

For single-phase transformers (linear or saturable), with the winding connectors appearing on the left and right sides, the winding voltages are the voltages of the top connector with respect to the bottom connector whatever the block orientation (*right* or *left*). The winding currents are the currents entering the top connector. For three-phase transformers, the voltage polarities and positive current directions are indicated by the signal labels used in the Multimeter block.

Load Flow and Machine Initialization GUI



Machine load flow

Displays load flow characteristics of the machine selected in the **Machines** field.

Machines

Display the names of the Simplified Synchronous Machines, the Synchronous Machines, the Asynchronous Machine, and the Three-Phase Dynamic Load blocks of your model. Select a machine or a load in the list box in order to set its parameters for the load flow.

Bus type

If **Bus type** is set to `P&V Generator`, you can set the desired terminal voltage and active power of the machine. If **Bus type** is set to `PQ` generator, you can set the desired active and reactive powers. If **Bus type** is set to Swing Bus, you can set the desired terminal voltage, enter an active power guess, and specify the phase of the UAN terminal voltage of the machine.

If you select an Asynchronous Machine block machine, you only have to enter the desired mechanical power delivered by the machine. If you select a Three-Phase Dynamic Load block, you have to specify the active and reactive powers consumed by the load.

Terminal voltage UAB

Specify the terminal line-to-line voltage of the selected machine.

Active power

Specify the active power of the selected machine or load.

Active power guess

Specify active power guess to start iterations when the specified machine bus type is Swing Bus.

Reactive power

Specify the reactive power of the selected machine or load.

Phase of UAN voltage

This parameter is activated only when the bus type is Swing Bus.

Specify the phase of the phase-to-neutral voltage of phase A of the selected machine.

Mechanical power

In motor mode, specify the mechanical power developed by the squirrel cage induction machine. In generator mode, specify the mechanical power absorbed by the machine as a negative number.

Load flow frequency

Specify the frequency to be used in the load flow calculations (normally 60 Hz or 50 Hz).

Load flow initial condition

Normally, you should keep the default setting **Auto** to let the load flow automatically adjust the initial conditions before starting iterations. If you select **Start from previous solution**, the load flow starts with initial conditions corresponding to the previous solution. Try this option if the load flow fails to converge after a change has been made to the power and voltage settings of the machines or to the circuit parameters.

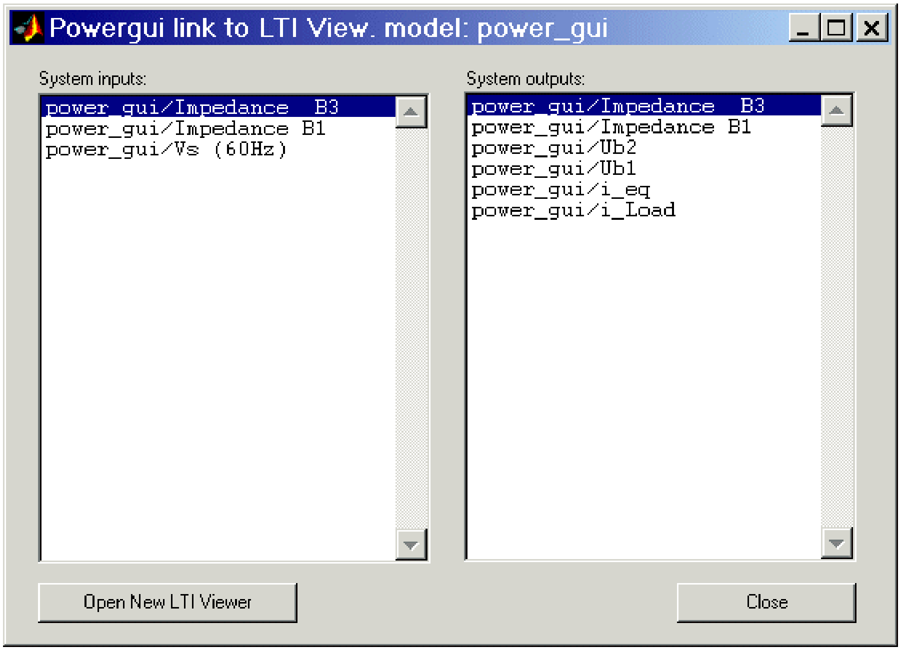
Update Circuit & Measurements

Update the list of machines, voltage and current phasors, as well as the powers in the load flow window if you have made a change in your model while the load flow window is open. The new voltages and powers displayed in the load flow window are computed by using the machine currents obtained from the last load flow (the three currents stored in the **Initial conditions** parameter of the machine blocks).

Execute Load Flow

Executes the load flow calculations for the given load flow parameters.

Use LTI Viewer GUI



System inputs

Lists the inputs of the state-space model of your circuit. Select the inputs to be used by the LTI Viewer.

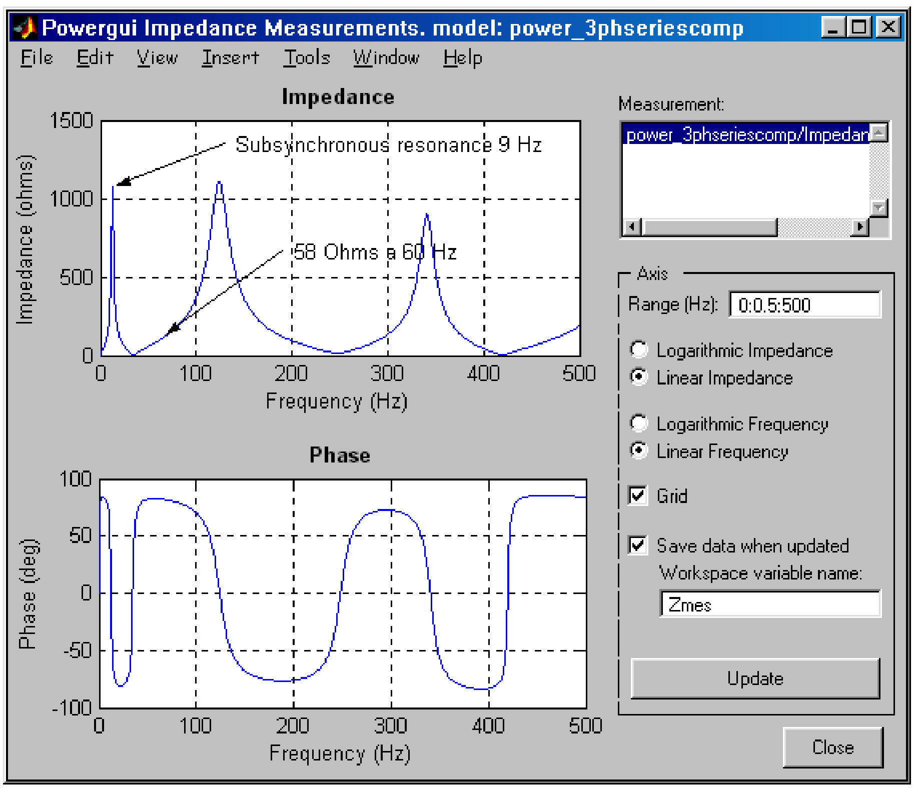
System outputs

Lists the outputs of the state-space model of your circuit. Select the outputs to be used by the LTI Viewer.

Open New LTI Viewer

Generate the state-space model of the circuit and opens the LTI viewer for the selected system inputs and outputs.

Impedance vs Frequency Measurement GUI



Measurement
Lists the Impedance Measurement blocks of the model. Select the blocks for which you want to obtain the frequency response. Use the CTRL key to select several impedances to be displayed on the same plot.

Axis

Range (Hz)
Specify the frequency vector, in hertz (Hz). You can specify in that field any valid MATLAB expression defining a vector of frequencies; for example, `0:2:1000` or `linspace(0,1000,500)`. The default is `logspace(0,3,50)`.

Logarithmic Impedance/Linear Impedance
Choose logarithmic or linear scale for the vertical impedance scale.

Logarithmic Frequency/Linear Frequency
Choose logarithmic or linear scale for the horizontal frequency scales.

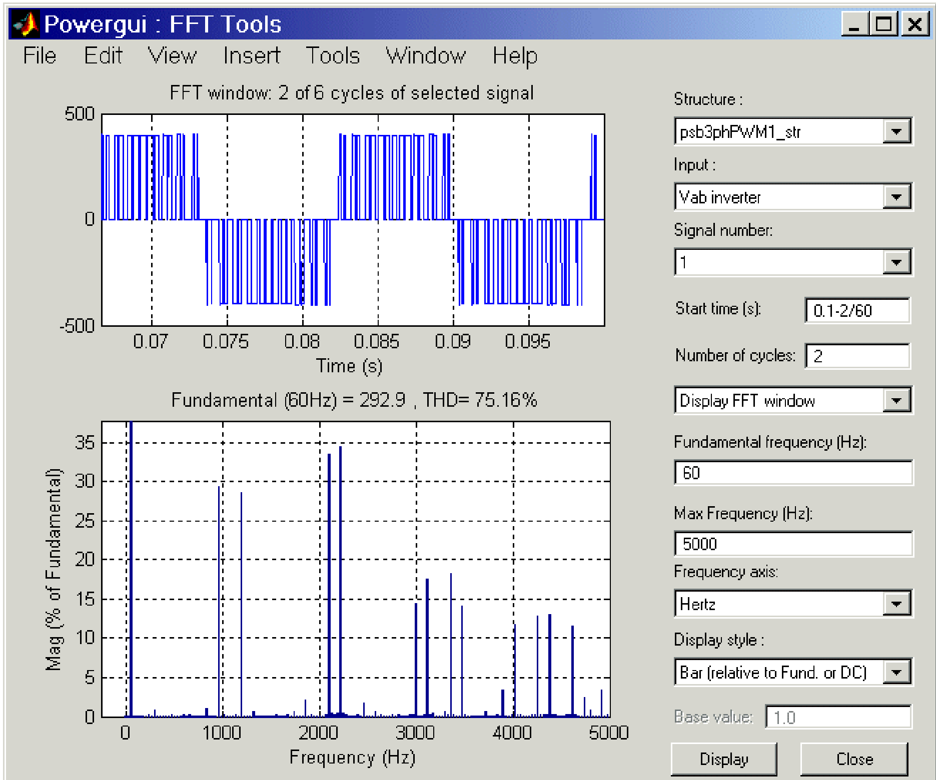
Grid
If selected, a grid is displayed for the two plots. Default is unselected.

Save data when updated
If selected, data are saved in a variable in the workspace. The name of the variable is defined by the **Workspace variable name** parameter. The complex impedances are saved in an array together with the corresponding frequencies. Frequency is saved in column 1 and impedances are saved in the next columns. Default is unselected.

Display/Save or **Update**
Click to initially display the impedance versus frequency measurement and, if the **Save data when updated** check box is selected, save the data to your workspace.

Click to start the impedance versus frequency measurement again and display results after multiple runs of your model.

FFT Analysis GUI



Structure

Lists the structures with time variables that are present in your workspace. These structures are generated by the Scope or To Workspace blocks in your model. Use the pull-down menu to select the variable you want to analyze.

Input

Select the input signal of the selected structure with time variables specified in the **Structure** field. Structures with time variables with multiple inputs can be generated by a Scope block having multiple input ports.

Signal Number

Specify the index of the selected input signal specified by the **Input** parameter. For example, the **Signal Number** parameter allows you to select the phase A signal of a three-phase signal connected to input 2 of a Scope block.

Start time(s)

Specify the start times for the FFT analysis.

Number of cycles

Specify the number of cycles for the FFT analysis.

Display FFT window/Display entire signal

In the pulldown menu, select **Display entire signal** to display the entire selected signal in the upper plot. Select **Display FFT window** to display only the portion of the signal where the FFT analysis is performed.

Fundamental frequency

Specify the fundamental frequency, in hertz (Hz), for the FFT analysis.

Max Frequency

Specify the maximum frequency, in hertz (Hz), for the FFT analysis.

Frequency axis

In the pull-down menu, select **Hertz** to display the spectrum frequency axis in hertz. Select **Harmonic order** to display the spectrum frequency axis in harmonic order relative to the fundamental frequency.

Display style

In the pull-down menu, select **Bar (relative to Fund. or DC)** to display the spectrum as a bar graph relative to the fundamental frequency. Select **Bar (relative to specified base)** to display the spectrum as a bar graph relative to the base defined by the **Base value** parameter.

Select **List (relative to Fund. or DC)** to display the spectrum as a list in % relative to the fundamental or DC component. Select **List (relative to specified base)** to display the spectrum as a list in % relative to the base value defined by the **Base value** parameter.

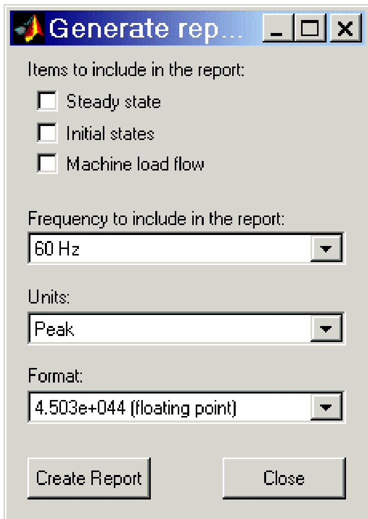
Base value

Enter a base value for the display of harmonics.

Display

Display the FFT analysis results for the selected measurement.

Generate Report GUI



Items to include in the report

In the check boxes, select any combination of measurements to include in the generated report, **Steady state**, **Initial states**, and **Machine load flow**. The default is unselected for all three.

Frequency to include in the report

Select the frequency or frequencies to include in the generated report, `60 Hz` or `All`. The default is `60 Hz`.

Units

Set the **Units** parameter to `Peak values` to display the peak values of the selected values. Set the **Units** parameter to `RMS` to display the root-mean-square (RMS) values of the selected values.

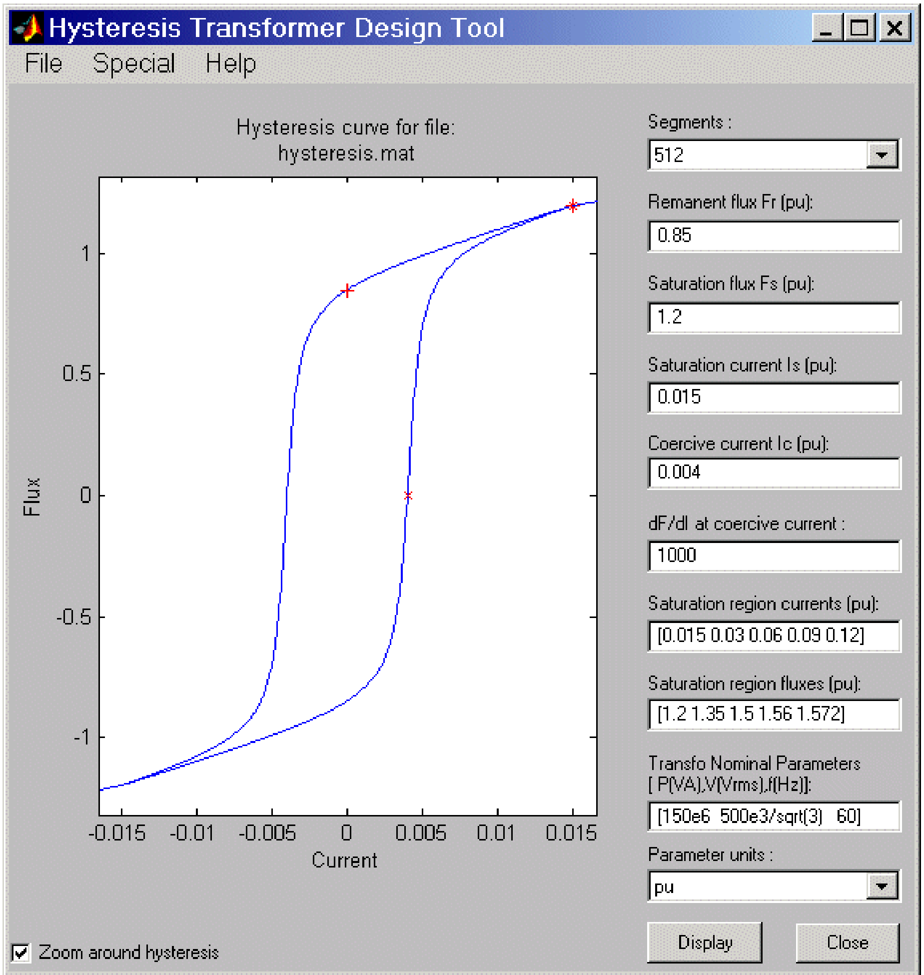
Format

In the pull-down menu, choose the format in which you want your measurements displayed. The `floating point` option is displayed in mantissa-exponent form with five significant figures. The `best of` option is displayed with four significant figures and uses mantissa-exponent form only for numbers larger than 9999. The final option is displayed in plain numbers with two figures to the right of the decimal point. The default is `floating point`.

Create Report

Generate a report and save it to a file.

Hysteresis Design Tool GUI



To learn more about hysteresis modeling, see the Saturable Transformer block reference pages.

Segments

In the pull-down menu, specify the number of linear segments used to define the right side of the hysteresis loop. The left side of the loop is the symmetric image of the right side.

Remanent flux Fr

Specify the remanent flux point of the hysteresis characteristic (flux at zero current).

Saturation Flux Fs

Specify the saturation flux point where the hysteresis loop becomes a single-valued saturation curve.

Saturation current Is

Specify the saturation current point where the hysteresis loop becomes a single-valued saturation curve. The saturation region is defined by the **Saturation region currents** parameter.

Coercive current Ic

Specify the coercive current point of the hysteresis characteristic.

dF/dI at coercive current

Set the slope of the flux at the coercive current point (current at zero flux).

Saturation region currents

Specify the vector of current values that define the saturation characteristic. The number of specified points must be the same as for the **Saturation region fluxes** parameter. You only need to specify the positive part of the characteristic.

Saturation region fluxes

Specify the vector of flux values that define the saturation characteristic. The number of specified points must be the same as for the **Saturation region currents** parameter. You only need to specify the positive part of the characteristic.

Transfo Nominal Parameters

Specify the nominal parameters (nominal power in VA, nominal voltage of winding 1 in volts RMS, and nominal frequency in Hz) used in the conversion of the hysteresis parameters.

Parameter units

Convert the fluxes and currents that define the hysteresis characteristic from SI to p.u. or from p.u. to SI.

Zoom around the hysteresis

If selected, zoom the plot around the hysteresis curve. The default is selected.

Example

Open the demos of SimPowerSystems and double-click the Powergui block contained in the model. For each demo, you can use the tools of the Powergui to look at the initial values and steady-state values of the inductor currents and capacitor voltages. For demos containing machines, you can edit and perform a machine load flow analysis.

See Also

Multimeter, Saturable Transformer

PWM Generator

Purpose

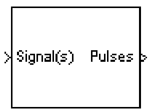
Generate pulses for a carrier-based two-level pulse width modulator (PWM) in converter bridge

Library

Extras/Control Blocks

A discrete version of this block is available in the Extras/Discrete Control Blocks library.

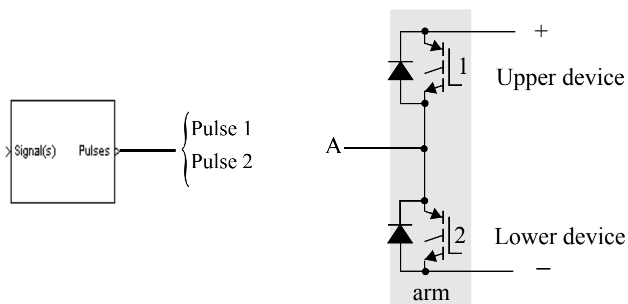
Description



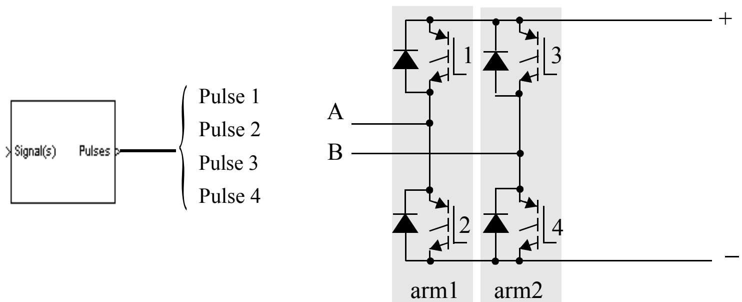
The PWM Generator block generates pulses for carrier-based pulse width modulation (PWM) converters using two-level topology. The block can be used to fire the forced-commutated devices (FETs, GTOs, or IGBTs) of single-phase, two-phase, three-phase, two-level bridges or a combination of two three-phase bridges.

The number of pulses generated by the PWM Generator block is determined by the number of bridge arms you have to control:

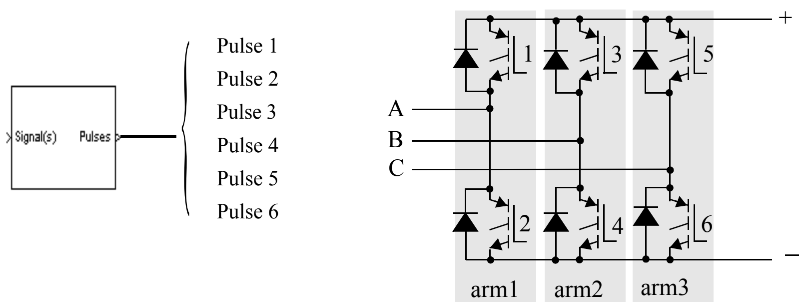
- Two pulses are generated for a one-arm bridge. Pulse 1 fires the upper device and pulse 2 fires the lower device (shown for the IGBT device).



- Four pulses are generated for a two-arm bridge. Pulses 1 and 3 fire the upper devices of the first and second arm. Pulses 2 and 4 fire the lower devices.



- Six pulses are generated for a three-arm bridge. Pulses 1, 3, and 5 fire the upper devices of the first, second, and third arms. Pulses 2, 4, and 6 fire the lower devices.



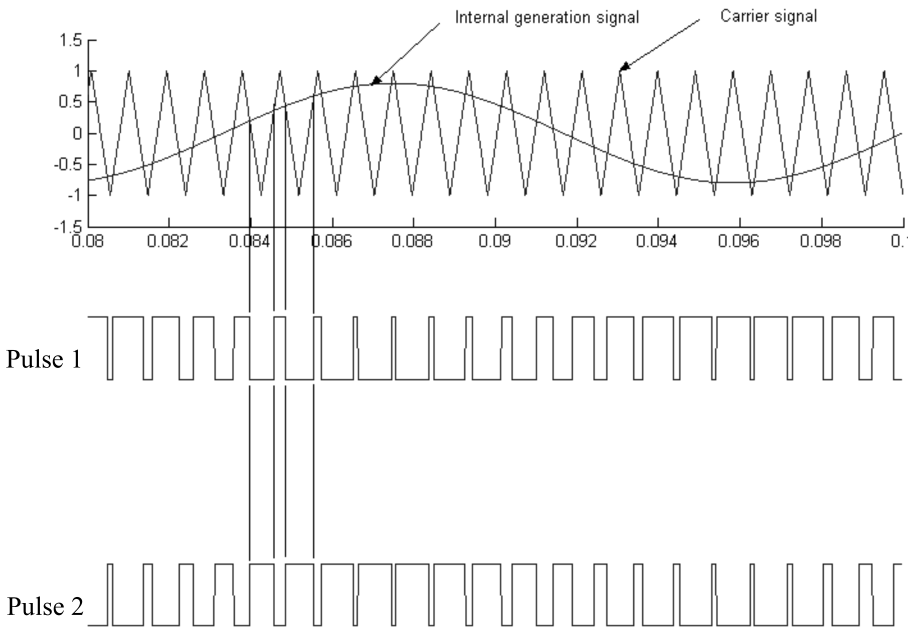
- Twelve pulses are generated for a double three-arm bridge. The first six pulses (1 to 6) fire the six devices of the first three-arm bridge and the last six pulses (7 to 12) fire the six devices of the second three-arm bridge.

For each arm the pulses are generated by comparing a triangular carrier waveform to a reference modulating signal. The modulating signals can be generated by the PWM generator itself, or they can be a vector of external signals connected at the input of the block. One reference signal is needed to generate the pulses for a single- or a two-arm bridge, and three reference signals are needed to generate the pulses for a three-phase, single or double bridge.

The amplitude (modulation), phase, and frequency of the reference signals are set to control the output voltage (on the AC terminals) of the bridge connected to the PWM Generator block.

The two pulses firing the two devices of an arm bridge are complementary. For example, pulse 4 is low (0) when pulse 3 is high (1). This is illustrated in the next two figures.

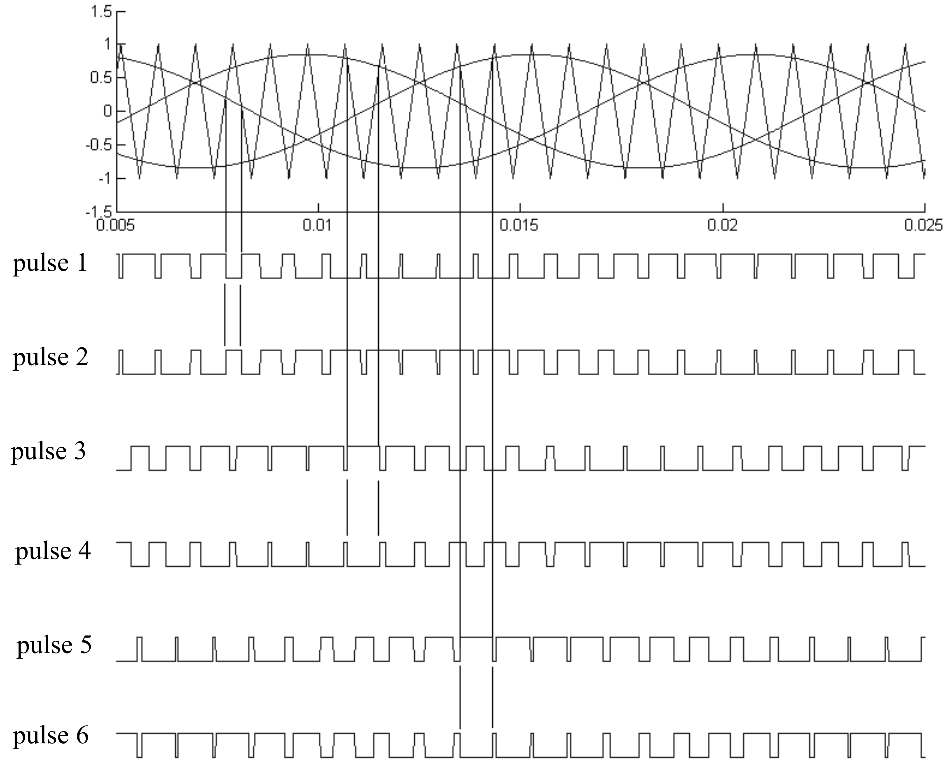
The following figure displays the two pulses generated by the PWM Generator block when it is programmed to control a one-arm bridge.



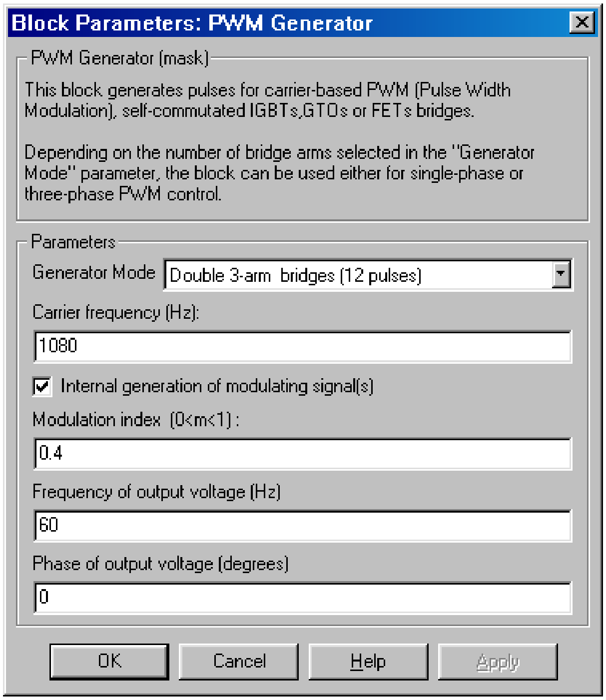
The triangular carrier signal is compared with the sinusoidal modulating signal. When the carrier is greater than the modulating signal, pulse 1 is high (1) and pulse 2 is low (0).

For a single-phase two-arm bridge the modulating signal used for arm 2 is the negative of modulating signal used for arm 1 (180 degrees phase shift). For a three-phase six-arm bridge the three modulating signals used for bridge 2 are the negative of the modulating signals applied to bridge 1.

The following figure displays the six pulses generated by the PWM Generator block when it is programmed to control a three-arm bridge.



Dialog Box and Parameters



Generator Mode

Specify the number of pulses to generate. The number of pulses is proportional to the number of bridge arms to fire. Select for example `Double 3-arm bridges (12 pulses)` to fire the self-commutated devices of two six-pulse bridges connected in a twelve-pulse bridge configuration.

Carrier frequency

The frequency, in hertz, of the carrier triangular signal.

Internal generation of modulating signal

If selected, the modulating signal is generated by the block. Otherwise, external modulating signals are used for pulse generation.

Modulation index (0 < m < 1)

The **Modulation index** parameter is visible only if the **Internal generation of modulating signal (s)** parameter is selected.

The amplitude of the internal sinusoidal modulating signal. The **Modulation index** must be greater than 0, and lower than or equal to 1. This parameter is used to control the amplitude of the fundamental component of the output voltage of the controlled bridge.

Frequency of output voltage

The **Frequency of output voltage (Hz)** parameter is visible only if the **Internal generation of modulating signal (s)** parameter is selected.

The frequency, in hertz, of the internal modulating signals. This parameter is used to control the fundamental frequency of the output voltage of the controlled bridge.

Phase of output voltage

The **Phase of output voltage** parameter is visible only if the **Internal generation of modulating signal (s)** parameter is selected.

The phase, in degrees, of the internal modulating signal. This parameter is used to control the phase of the fundamental component of the output voltage of the controlled bridge.

Inputs and Outputs

`Signal(s)`

The input is not visible when **Internal generation of modulating signal (s)** is selected.

The input is the vector of modulating signals when **Internal generation of modulating signal** is not selected. Connect this input to a single-phase sinusoidal signal when the block is used to control a single- or a two-arm bridge, or to a three-phase sinusoidal signal when the PWM Generator block is controlling one or two three-phase bridges.

`Pulses`

The output contains the two, four, six, or twelve pulse signals used to fire the self-commutated devices (MOSFETs, GTOs, or IGBTs) of single-phase, two-phase, or three-phase bridges or a combination of two three-phase bridges.

Example

See the `power_1phPWM` and `power_3phPWM` demos for examples of single-phase and three-phase two-level inverters.

See Also

Universal Bridge

Purpose

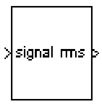
Measure the root mean square (RMS) value of a signal

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description



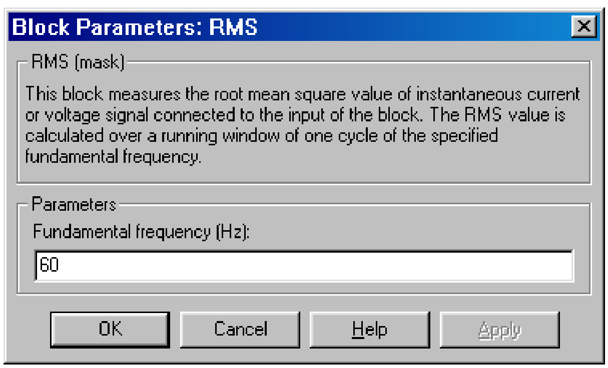
This block measures the root mean square value of an instantaneous current or voltage signal connected to the input of the block. The RMS value of the input signal is calculated over a running average window of one cycle of the specified fundamental frequency.

$$RMS(f(t)) = \sqrt{\frac{1}{T}\int_{(t-T)}^{t} f(t)^2}$$

$f(t)$: input signal, T = 1/fundamental frequency

as this block uses a running average window, one cycle of simulation has to be completed before the output gives the correct value. The discrete version of this block allows you to specify the initial magnitude of inputs. For the first cycle of simulation the output is held to the RMS value of the specified initial input.

Dialog Box and Parameters

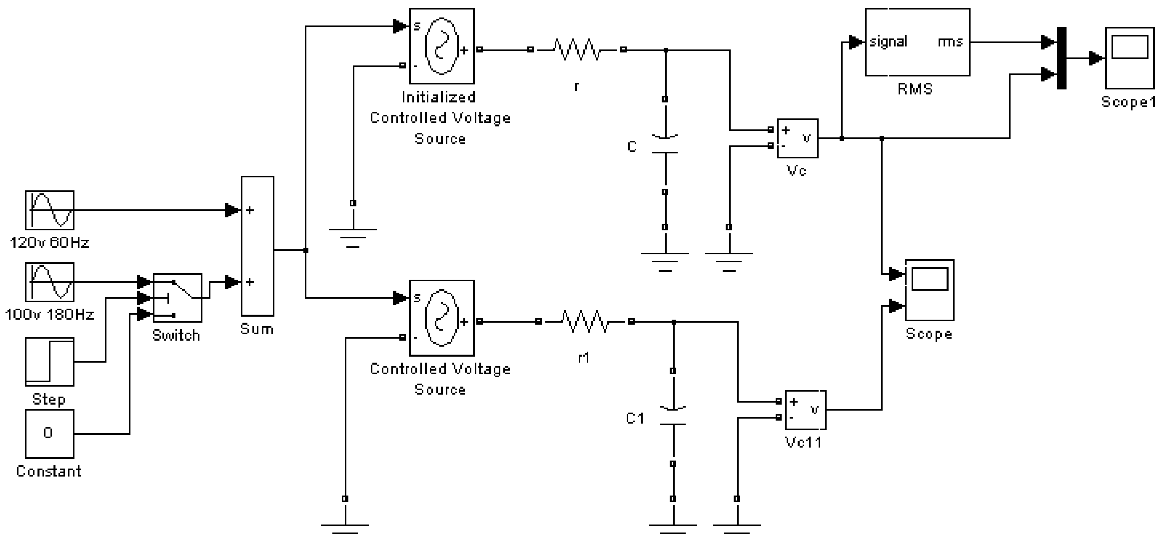


Fundamental frequency

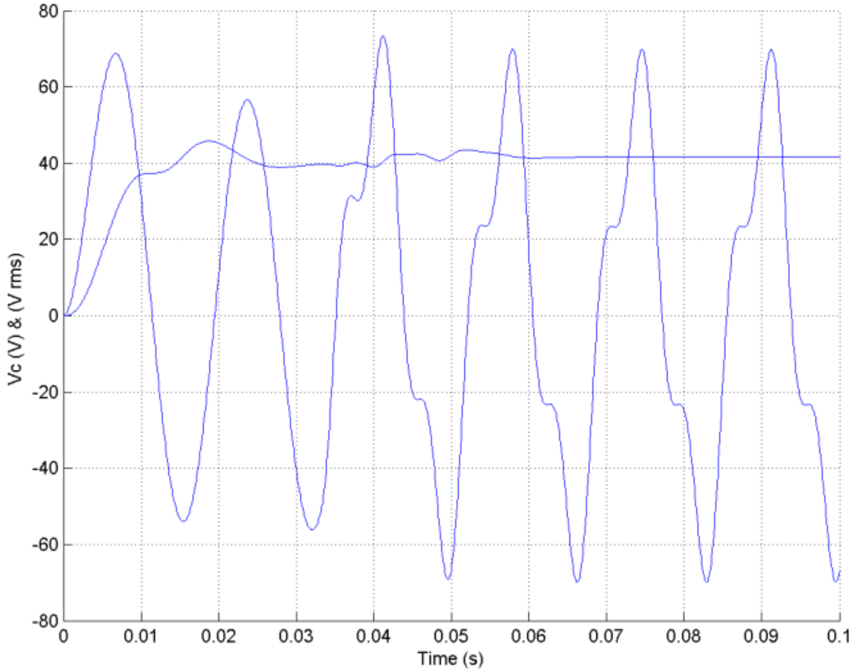
The fundamental frequency, in hertz, of the input signal.

Example

In the `power_controlvolt` demo, you can add an RMS block as shown below to measure the RMS value of the capacitor voltage. The Controlled Voltage Source block introduces a third harmonic (180 Hz) in the voltage at t = 0.4 seconds.



At the beginning of the simulation, the RMS block needs one cycle of the fundamental frequency (60Hz) to calculate the RMS value of the voltage. At t = 0.4 seconds the RMS value slightly increases because of the addition of the third harmonic in the signal. Again, the RMS block needs one cycle of the fundamental signal to stabilize and give the correct result.

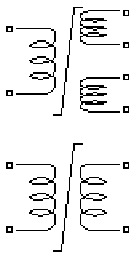

Vc (V) & (V rms)
Time (s)
80
60
40
20
0
-20
-40
-60
-80
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

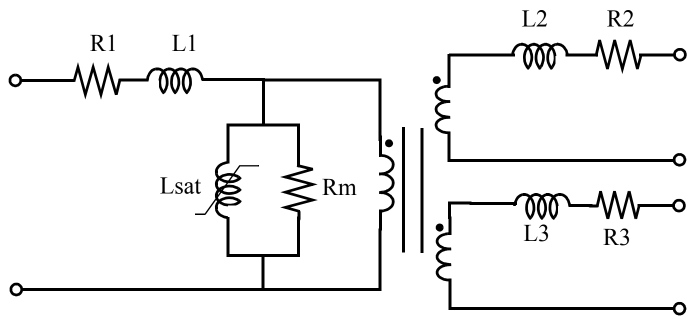
Saturable Transformer

Purpose Implement a two- or three-winding saturable transformer

Library Elements

Description The Saturable Transformer block model shown consists of three coupled windings wound on the same core.





The model takes into account the winding resistances (R1 R2 R3) and the leakage inductances (L1 L2 L3) as well as the magnetizing characteristics of the core, which is modeled by a resistance Rm simulating the core active losses and a saturable inductance Lsat.

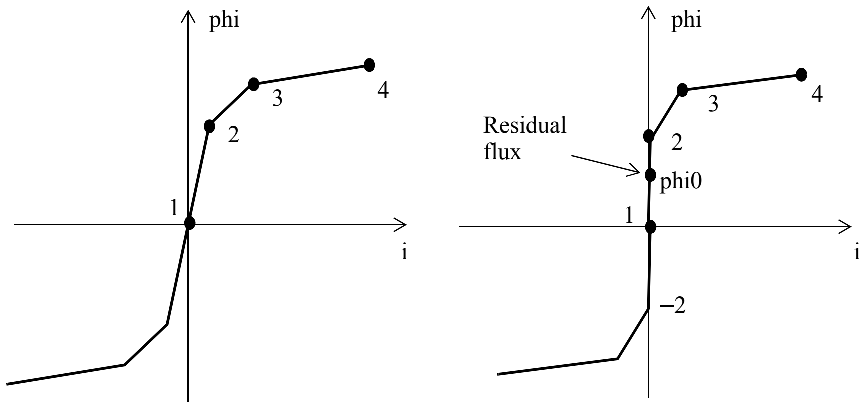
You can choose one of the following two options for the modeling of the nonlinear flux-current characteristic

1. Model saturation without hysteresis. The total iron losses (eddy current + hysteresis) are modeled by a linear resistance, Rm.
2. Model hysteresis and saturation. Specification of the hysteresis is done by means of the **Hysteresis Design Tool** of the Powergui block. The eddy current losses in the core are modeled by a linear resistance, Rm.

Note Modeling the hysteresis requires additional computation load and therefore slows down the simulation. The hysteresis model should be reserved for specific applications where this phenomenon is important.

Saturation Characteristic Without Hysteresis

When the hysteresis is not modeled, the saturation characteristic of the Saturable Transformer block is defined by a piecewise linear relationship between the flux and the magnetization current.



(a) No residual flux can be specified.

(b) A residual flux can be specified between points 2 and −2.

Therefore, if you want to specify a residual flux, phi0, the second point of the saturation characteristic should correspond to a null current, as shown in the figure (b).

The saturation characteristic is entered as (i, phi) pair values in per units, starting with pair (0, 0). SimPowerSystems converts the vector of fluxes Φ_{pu} and the vector of currents I_{pu} into standard units to be used in the saturation model of the Saturable Transformer block:

$$\Phi = \Phi_{pu}\Phi_{base}$$

$$I = I_{pu}I_{base}$$

where the base flux linkage (Φbase) and base current (Ibase) are the peak values obtained at nominal voltage power and frequency:

$$I_{base} = \frac{Pn}{V_1}\sqrt{2} \qquad \Phi_{base} = \frac{V_1}{2\pi f_n}\sqrt{2} \text{ (Flux linkage in volts-seconds)}$$

The base flux is defined as the peak value of the sinusoidal flux (in webers) when winding 1 is connected to a 1 p.u. sinusoidal voltage source (nominal voltage). The Φ_{base} value defined above represents the base flux linkage (in volt-seconds). It is related to the base flux by the following equation

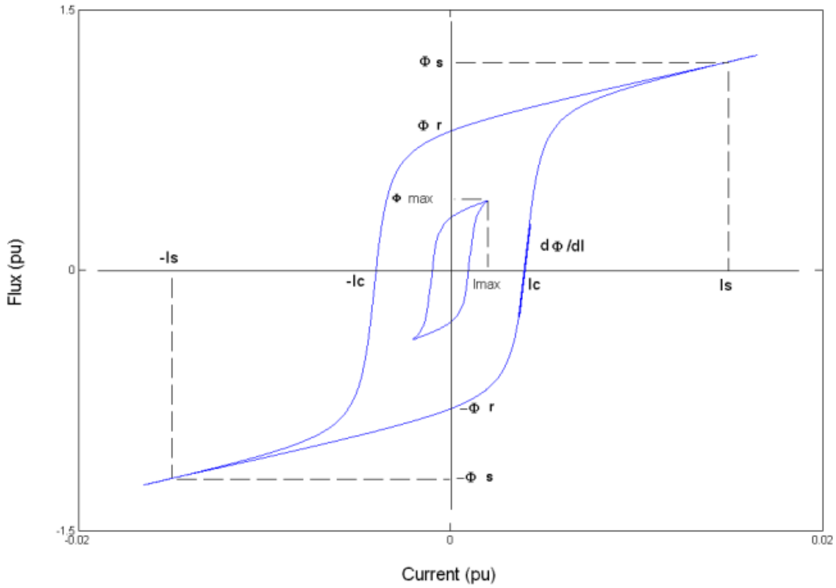
$$\Phi_{base} = \text{Base flux} \times \text{number of turns of winding 1}$$

When they are expressed in p.u., the flux and the flux linkage have the same value.

Saturation Characteristic with Hysteresis

The magnetizing current I is computed from the flux Φ obtained by integrating voltage across the magnetizing branch. The static model of hysteresis defines the relation between flux and the magnetization current evaluated in DC, when the eddy current losses are not present.

The hysteresis model is based on a semiempirical characteristic, using an arctangent analytical expression Φ(I) and its inverse I(Φ) to represent the operating point trajectories. The analytical expression parameters are obtained by curve fitting empirical data defining the major loop and the single-valued saturation characteristic. The **Hysteresis design tool** of the Powergui block is used to fit the hysteresis major loop of a particular core type to basic parameters. These parameters are defined by the remanent flux (Φr), the coercive current (Ic), and the slope (dΦ/dI) at (0, Ic) point as shown in the next figure.



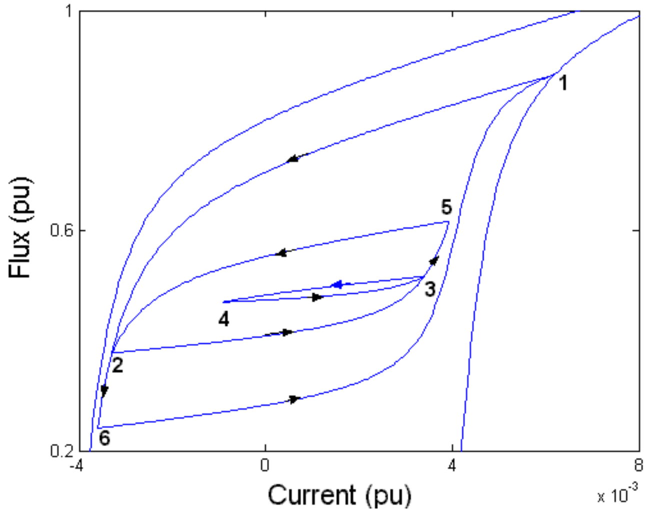
The major loop half cycle is defined by a series of N equidistant points connected by line segments. The value of N is defined in the **Hysteresis design tool** of the Powergui block. Using N = 256 yields a smooth curve and usually gives satisfactory results.

The single-valued saturation characteristic is defined by a set of current-flux pairs defining a saturation curve which should be asymptotic to the air core inductance Ls.

The main characteristics of the hysteresis model are summarized below:

1. A symmetrical variation of the flux produces a symmetrical current variation between –Imax and +Imax, resulting in a symmetrical hysteresis loop whose shape and area depend on the value of Φmax. The major loop is produced when Φmax is equal to the saturation current (Φs). Beyond that point the characteristic reduces to a single-valued saturation characteristic.
2. In transient conditions, an oscillating magnetizing current produces minor asymmetrical loops, as shown in the next figure, and all points of operation

are assumed to be within the major loop. Loops once closed have no more influence on the subsequent evolution.



The trajectory starts from the initial (or residual) flux point, which must lie on the vertical axis inside the major loop. You can specify this initial flux value phi0, or it is automatically adjusted so that the simulation starts in steady state.

The Per Unit Conversion

In order to comply with industry practice, you must specify the resistance and inductance of the windings in per unit (p.u.). The values are based on the transformer rated power Pn in VA, nominal frequency fn in Hz, and nominal voltage Vn, in Vrms, of the corresponding winding. For each winding the per unit resistance and inductance are defined as

$$R(\text{p.u.}) = \frac{R(\Omega)}{R_{base}}$$

$$L(\text{p.u.}) = \frac{L(H)}{L_{base}}$$

The base resistance and base inductance used for each winding are

$$R_{base} = \frac{(Vn)^2}{Pn}$$

$$L_{base} = \frac{R_{base}}{2\pi fn}$$

For the magnetization resistance Rm, the p.u. values are based on the transformer rated power and on the nominal voltage of winding 1.

The default parameters of winding 1 specified in the dialog box section give the following base values:

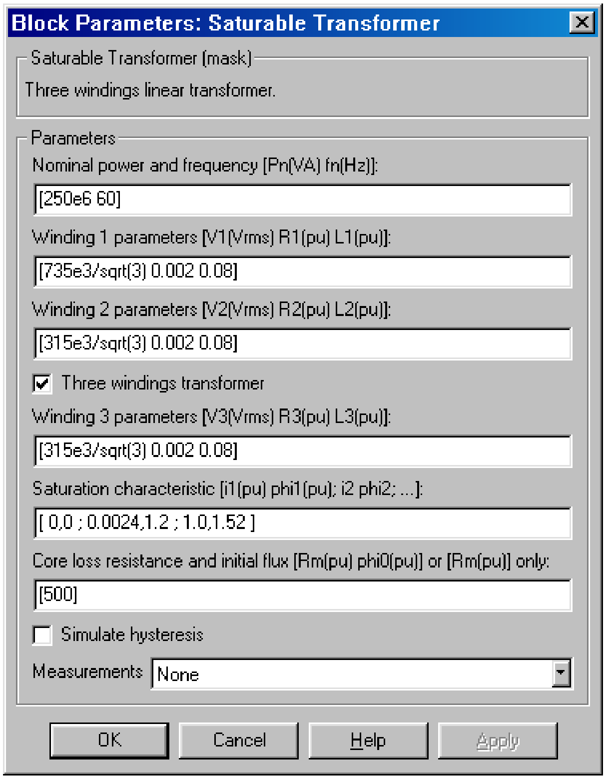
$$R_{base} = \frac{(735e3/\sqrt{3})^2}{250e6} = 720.3\Omega \qquad L_{base} = \frac{720.3}{2\pi 60} = 1.91H$$

For example, if winding 1 parameters are R1 = 1.44 Ω and L1 = 0.1528 H, the corresponding values to enter in the dialog box are

$$R_1 = \frac{1.44\Omega}{720.3\Omega} = 0.002 \text{ p.u.}$$

$$L_1 = \frac{0.1528H}{1.91H} = 0.08 \text{ p.u.}$$

Dialog Box and Parameters



Nominal power and frequency

The nominal power rating, Pn, in volt-amperes (VA), and frequency, in hertz (Hz), of the transformer.

Winding 1 parameters

The nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 1.

Winding 2 parameters

The nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 2.

Three windings transformer

If selected, specify a saturable transformer with three windings; otherwise it implements a two windings transformer.

Winding 3 parameters

The **Winding 3 parameters** are not available if the **Three windings transformer** parameter is not selected. The nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 3.

Saturation characteristic

Specify a series of magnetizing current (p.u.) - flux (p.u.) pairs starting with (0,0).

Core loss resistance and initial flux

Specify the active power dissipated in the core by entering the equivalent resistance Rm in p.u. For example, to specify a 0.2% of active power core loss at nominal voltage, use Rm = 500 p.u. You can also specify the initial flux phi0 (p.u). This initial flux becomes particularly important when the transformer is energized. If phi0 is not specified, the initial flux is automatically adjusted so that the simulation starts in steady state. When simulating hysteresis, Rm models the eddy current losses only.

Simulate hysteresis

Select to model hysteresis saturation characteristic instead of a single-valued saturation curve.

Hysteresis data MAT file

The **Hysteresis data MAT file** parameter is visible only if the **Simulate hysteresis** parameter is selected.

Specify a `.mat` file containing the data to be used for the hysteresis model. When you open the **Hysteresis Design tool** of the Powergui, the default hysteresis loop and parameters saved in the hysteresis.mat file are displayed. Use the **File —> Load a model** menu of the **Hysteresis Design tool** to load another `.mat` file. Use the **File —> Save this model** menu of the **Hysteresis Design tool** to save your model in a new `.mat` file.

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals of the Saturable Transformer block.

Select `Winding currents` to measure the current flowing through the windings of the Saturable Transformer block.

Select `Flux and excitation current (Im + IRm)` to measure the flux linkage, in volt seconds (V.s), and the total excitation current including iron losses modeled by Rm.

Select `Flux and magnetization current (Im)` to measure the flux linkage, in volt seconds (V.s), and the magnetization current, in amperes (A), not including iron losses modeled by Rm.

Select `All measurement (V, I, Flux)` to measure the winding voltages, currents, magnetization currents, and the flux linkage.

Place a Multimeter block in your model to display the selected measurements during the simulation.

In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement	Label
Winding voltages	`Uw1:, Uw2:, Uw3:`
Winding currents	`Iw1:, Iw2:, Iw3:`
Excitation current	Iexc:
Magnetization current	`Imag:`
Flux linkage	`Flux:`

Inputs and Outputs

The winding terminals of Input 1, output 1, and output 3 (if it exists) are at the same instantaneous polarity.

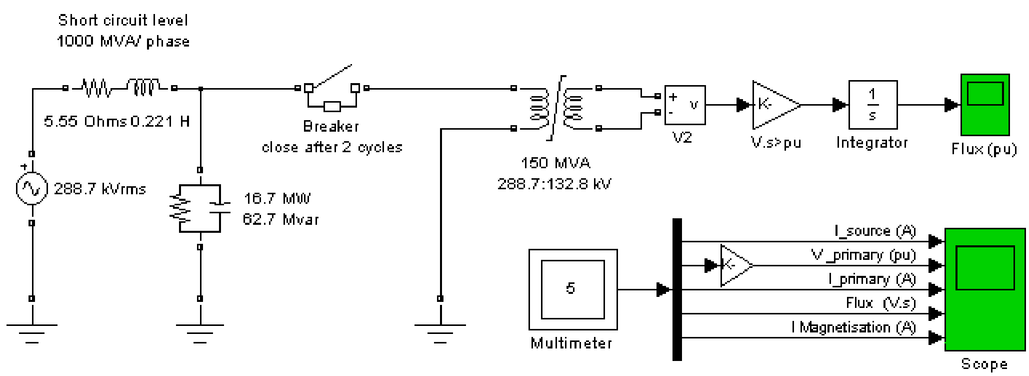
Limitations

Windings can be left floating (that is, not connected by an impedance to the rest of the circuit). However, the floating winding is connected internally to the main circuit through a resistor. This invisible connection does not affect voltage and current measurements.

Example

The `power_xfosaturable` demo illustrates the energization of one phase of a three-phase 450 MVA, 500/230 kV transformer on a 3000 MVA source. The transformer parameters are

Nominal power and frequency	Pn = 150e6 VA	fn = 60 Hz	
Winding 1 parameters (primary)	V1 = 500e3 Vrms/sqrt(3)	R1 = 0.002 p.u.	L1 = 0.08 p.u
Winding 2 parameters (secondary)	V2 = 230e3 Vrms/sqrt(3)	R2 = 0.002 p.u.	L2 = 0.08 p.u.
Saturation characteristic	[0 0; 0.0 1.2; 1.0 1.52]		
Core loss resistance and initial flux	Rm = 500 p.u.	phi0 = 0.8 p.u.	

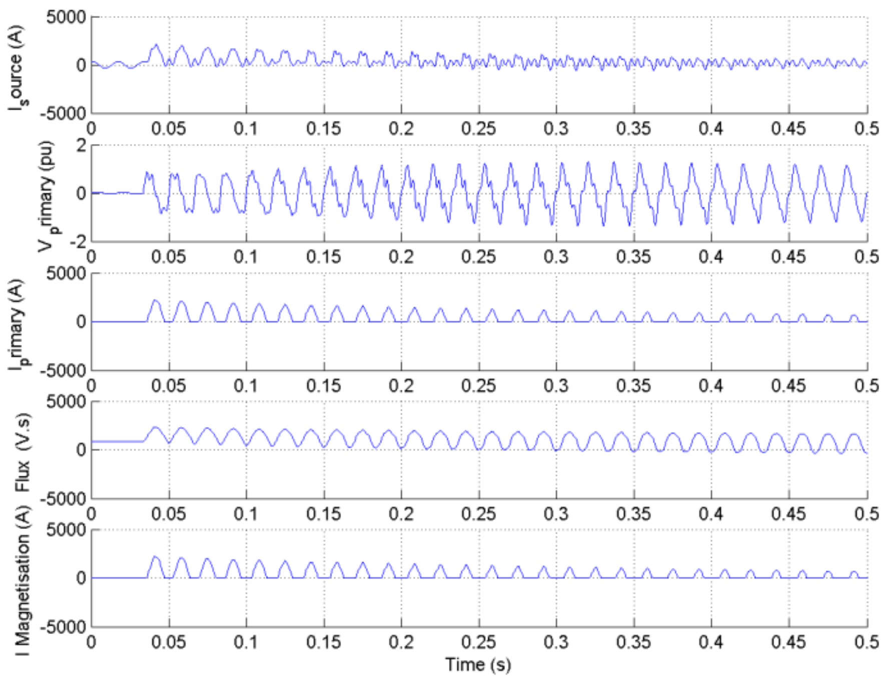


Simulation of this circuit illustrates the saturation effect on the transformer current and voltage.

As the source is resonant at the fourth harmonic, you can observe a high fourth-harmonic content in the secondary voltage. In this circuit, the flux is calculated in two ways:

- By integrating the secondary voltage
- By using the Multimeter block

The simulation results demonstrate these points:



References

Casoria, S., P. Brunelle, and G. Sybille, “Hysteresis Modeling in the MATLAB/Power System Blockset,” Electrimacs 2002, École de technologie supérieure, Montreal, 2002.

Frame, J.G., N. Mohan, and Tsu-huei Liu, “Hysteresis modeling in an Electro-Magnetic Transients Program,” presented at the IEEE PES winter meeting, New York, January 31 to February 5, 1982.

See Also

Linear Transformer, Multimeter, Mutual Inductance, Powergui, Three-Phase Transformer (Two Windings), Three-Phase Transformer (Three Windings)

Series RLC Branch

Purpose

Implement a series RLC branch

Library

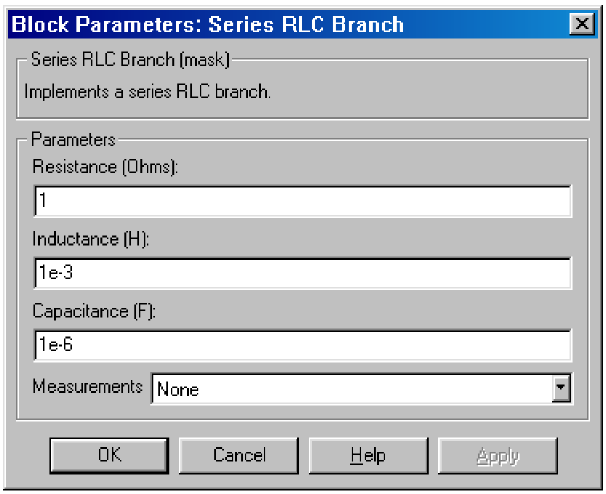
Elements

Description

The Series RLC Branch block implements a single resistor, inductor, or capacitor, or a series combination of these. To eliminate either the resistance, inductance, or capacitance of the branch, the R, L, and C values must be set respectively to 0, 0, and infinity (`inf`). Only existing elements are displayed in the block icon.

Negative values are allowed for resistance, inductance, and capacitance.

Dialog Box and Parameters



Resistance

The branch resistance, in ohms (Ω).

Inductance

The branch inductance, in henries (H).

Capacitance

The branch capacitance, in farads (F).

Measurements

Select Branch voltage to measure the voltage across the Series RLC Branch block terminals.

Select Branch current to measure the current flowing through the Series RLC Branch block.

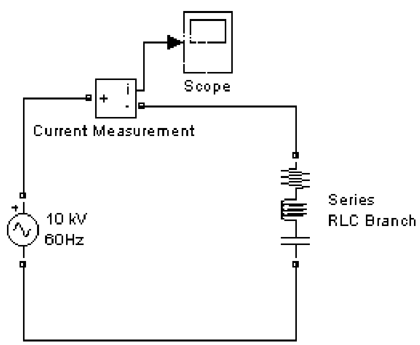
Select Branch voltage and current to measure the voltage and the current of the Series RLC Branch block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Branch voltage	Ub:
Branch current	Ib:

Example

Obtain the frequency response of a fifth-harmonic filter (tuned frequency = 300 Hz) connected on a 60 Hz power system. This example is available in the power_seriesbranch model.



The network impedance in the Laplace domain is

$$Z(s) = \frac{V(s)}{I(s)} = \frac{LCs^2 + RCs + 1}{Cs}$$

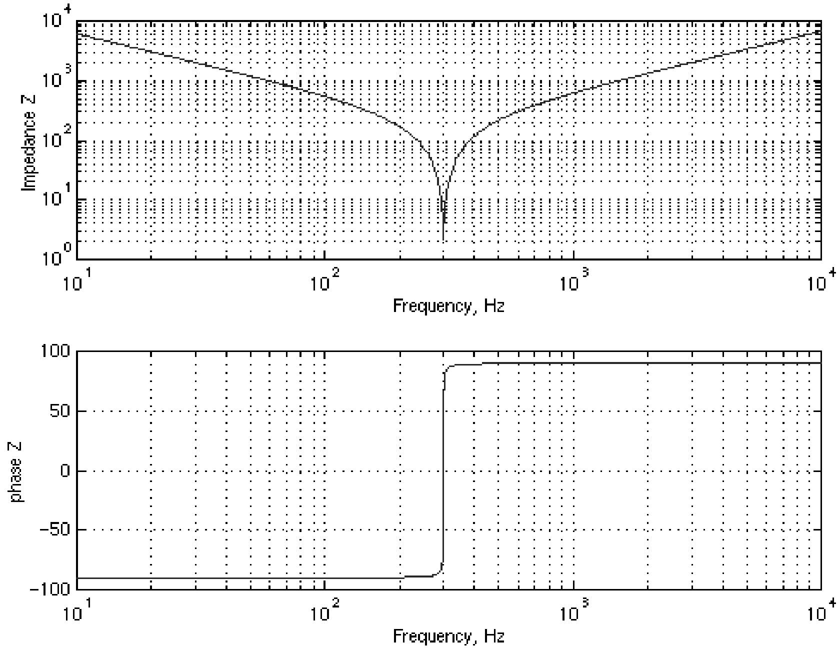
To obtain the frequency response of the impedance you have to get the state-space model (A B C D matrices) of the system.

This system is a one-input (Vsource) and one-output (Current Measurement block) system.

Note If you have the Control System Toolbox installed, you can use the `bode` function to get the transfer function *Z(s)* from the state-space matrices as follows:

```
[A,B,C,D] = power_analyze('power_seriesbranch');
freq = logspace(1,4,500);
w = 2*pi*freq;
[Ymag,Yphase] = bode(A,B,C,D,1,w);
% invert Y(s) to get Z(s)
Zmag = 1./Ymag;
Zphase = -Yphase;
subplot(2,1,1)
loglog(freq,Zphase)
grid
title('5th harmonic filter')
xlabel('Frequency, Hz')
ylabel('Impedance Zmag')
subplot(2,1,2)
semilogx(freq,Zphase)
xlabel('Frequency, Hz')
ylabel('phase Z')
grid
```

You can also use the Impedance Measurement block and the Powergui block to plot the impedance as a function of frequency. In order to measure the impedance you must disconnect the voltage source.



See Also Multimeter, Parallel RLC Branch, Parallel RLC Load, Series RLC Load

Series RLC Load

Purpose

Implement a linear series RLC load

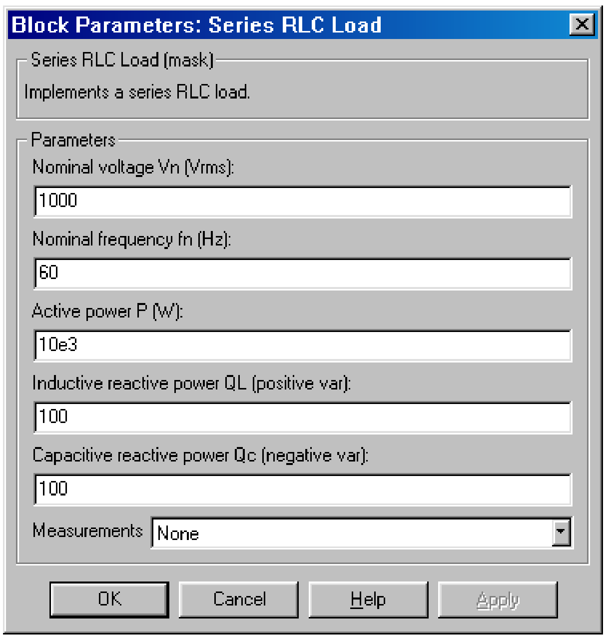
Library

Elements

Description

The Series RLC Load block implements a linear load as a series combination of R L C elements. At the specified frequency, the load exhibits a constant impedance. The active and reactive powers absorbed by the load are proportional to the square of the applied voltage. Only elements associated with nonzero powers are displayed in the block icon.

Dialog Box and Parameters



Nominal voltage Vn

The nominal voltage of the load, in volts RMS.

Nominal frequency fn

The nominal frequency, in hertz.

Active power P

The active power of the load, in watts.

Inductive reactive power QL

The inductive reactive power QL, in vars. Specify a positive value, or 0.

Capacitive reactive power QC

The capacitive reactive power QC, in vars. Specify a positive value, or 0.

Measurements

Select `Branch voltage` to measure the voltage across the Series RLC Load block terminals.

Select `Branch current` to measure the current flowing through the Series RLC Load block.

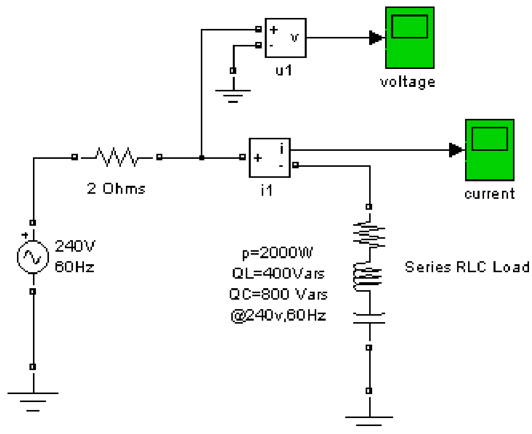
Select `Branch voltage and current` to measure the voltage and the current of the Series RLC Load block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name:

Measurement	Label
Branch voltage	`Ub:`
Branch current	`Ib:`

Example

The `power_seriesload` demo uses a Series RLC Load block to implement a simple load.

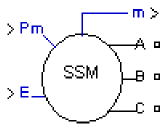


See Also Multimeter, Parallel RLC Branch, Parallel RLC Load, Series RLC Branch

Purpose Model the dynamics of a simplified three-phase synchronous machine

Library Machines

Description The Simplified Synchronous Machine block models both the electrical and mechanical characteristics of a simple synchronous machine.



The electrical system for each phase consists of a voltage source in series with an RL impedance, which implements the internal impedance of the machine. The value of R can be zero but the value of L must be positive.

The Simplified Synchronous Machine block implements the mechanical system described by

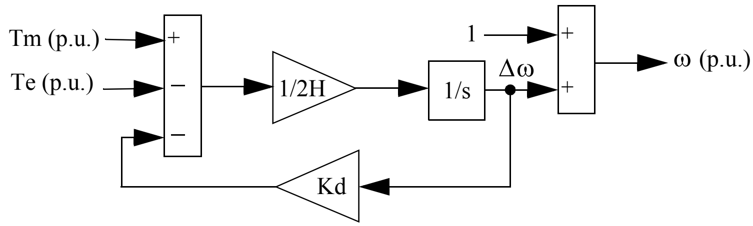
$$\Delta\omega(t) = \frac{1}{2H}\int_0^t (Tm - Te)dt - Kd\Delta\omega(t)$$

$$\omega(t) = \Delta\omega(t) + \omega_0$$

where

$\Delta\omega$ = Speed variation with respect to speed of operation
H = Constant of inertia
Tm = Mechanical torque
Te = Electromagnetic torque
Kd = Damping factor representing the effect of damper windings
$\omega(t)$ = Mechanical speed of the rotor
ω_0 = Speed of operation (1 p.u.)

Although the parameters can be entered in either SI units or per unit in the dialog box, the internal calculations are done in per unit. The following block diagram illustrates how the mechanical part of the model is implemented. Notice that the model computes a deviation with respect to the speed of operation, and not the absolute speed itself.



The Kd damping coefficient simulates the effect of damper windings normally used in synchronous machines. When the machine is connected to an infinite network (zero impedance), the variation of machine power angle delta (δ) resulting from a change of mechanical power (P_m) can be approximated by the following second-order transfer function:

$$\delta / P_m = (\omega_s / 2H)/(s^2 + 2\zeta\omega_n s + \omega_n^2)$$

where

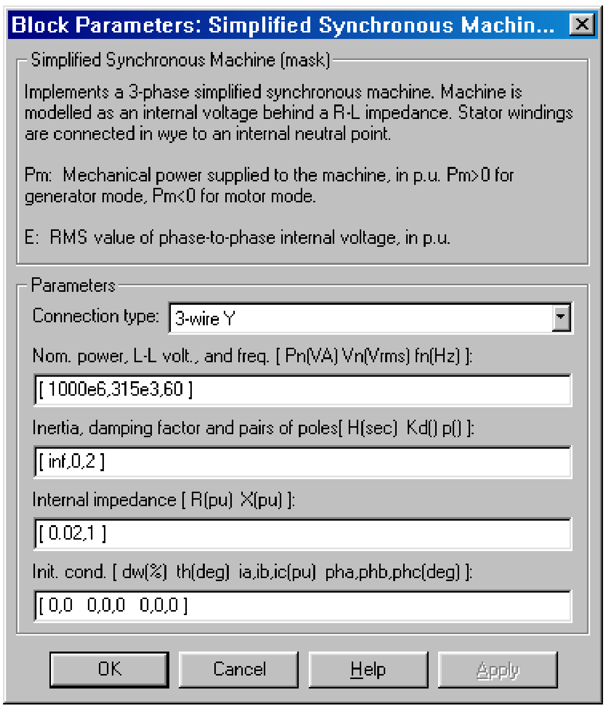
δ	Power angle delta: angle of internal voltage E with respect to terminal voltage, in radians
P_m	Mechanical power in p.u.
ω_n	Frequency of electromechanical oscillations = $\sqrt{\omega_s \cdot P_{max}/(2H)}$ in rad/s
ζ	Damping ratio $= (K_d/4)\sqrt{2/(\omega_s \cdot H \cdot P_{max})}$
ω_s	Electrical frequency in rad/s
$P_{\max}$	Maximum power in p.u. transmitted through reactance X at terminal voltage V_t and internal voltage E. $P_{\max} = V_t \cdot E / X$ (p.u.) where V_t, E, and X are in p.u.
H	Inertia constant(s)
K_d	Damping factor (p.u._of_torque / p.u._of_speed)

This approximate transfer function, which has been derived by assuming sin(δ) = δ, is valid for small power angles (δ < 30 degrees). It follows from the above ζ expression that the Kd value required to obtain a given ζ damping ratio is

$$K_d = 4\zeta\sqrt{\omega_s \cdot H \cdot P_{max}/2}$$

Dialog Box and Parameters

In the **powerlib** library you can choose between the SI units or the p.u. units Simplified Synchronous Machine blocks to specify the electrical and mechanical parameters of the model.



Connection type

Specify the number of wires used in three-phase Y connection: either three-wire (neutral not accessible) or four-wire (neutral is accessible).

Nominal power, L-L voltage, and frequency

The nominal apparent power Pn (VA), frequency fn (Hz), and RMS line-to-line voltage Vn (V). Used to compute nominal torque and convert SI units to p.u.

Inertia, friction factor and pairs of poles

The inertia (J in kg.m^2 or H in seconds) damping factor (Kd) and number of pairs of poles (p). The damping factor should be specified in (p.u. of torque)/(p.u. of speed) in both machine dialog boxes (in p.u. and in SI).

Internal impedance

The resistance R (Ω or p.u.) and reactance L (H or p.u.) for each phase.

Initial conditions

The initial speed deviation (% of nominal), rotor angle (degrees), line current magnitudes (A or p.u.), and phase angles (degrees). These values can be computed by the load flow utility of the Powergui block.

Note These two blocks simulate exactly the same simplified synchronous machine model; the only difference is the way of entering the parameter units.

Inputs and Outputs

The first input of the Simplified Synchronous Machine block is the mechanical power supplied to the machine. This input can be a constant or the output of the Hydraulic Turbine and Governor block. The frequency of the internal voltage sources depends on the mechanical speed of the machine. The amplitude of these voltages is given by the second input of the block, which can be a constant or the output of a voltage regulator. If you use SI units these two inputs should be in watts and volts phase-to-phase RMS. If you use p.u. both inputs should be in p.u.

The first three outputs are the electrical terminals of the stator. The last output of the block is a vector containing the following 12 signals:

Signal	Definition
1 to 3	Line currents (flowing out of the machine) ia, ib, ic
4 - 6	Terminal voltages va, vb, vc
7 - 9	Internal voltages ea, eb, ec
10	Mechanical angle θ
11	Rotor speed ω
12	Electrical power Pe

You can demultiplex these signals by using the Machines Measurement Demux block in the Machines library.

Assumptions

The electrical system of the Simplified Synchronous Machine block consists solely of a voltage source behind a synchronous reactance and resistance. All the other self- and magnetizing inductances of the armature, field, and damping windings are neglected. The effect of damper windings is approximated by the damping factor Kd. The three voltage sources and RL impedance branches are Y-connected (three wires or four wires). The load might or might not be balanced.

Example

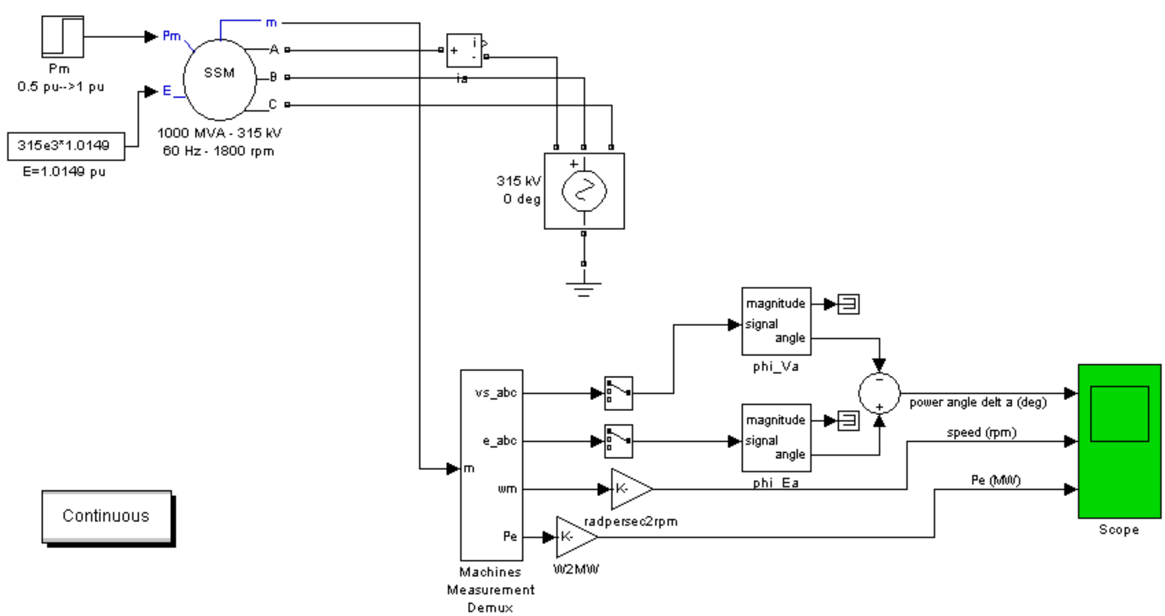
The `power_simplealt` demo uses the Simplified Synchronous Machine block to represent a 1000 MVA, 315 kV, 60 Hz equivalent source connected to an infinite bus (Three-Phase Programmable Voltage Source block). The Simplified Synchronous Machine (SI Units) block is used as a synchronous generator. The internal resistance and reactance are set respectively to 0.02 p.u. (1.9845 Ω) and 0.2 p.u. (X = 19.845 Ω; L = 0.0526 H). The inertia of the machine is J = 168,870 kg.m^2, corresponding to an inertia constant H = 3 s. The electrical frequency is $\omega_s = 2*\pi*60/2 = 377$ rad/s. The machine has two pairs of poles such that its synchronous speed is 2*π*60/2 = 188.5 rad/s or 1800 rpm.

The **Load Flow** option of the Powergui has been used to initialize the machine in order to start simulation in steady state with the machine generating 500

MW. The required internal voltage computed by the load flow is 1.0149 p.u. Therefore an internal voltage E = 315e3*1.0149 = 319,690 Vrms phase-to-phase is specified in the Constant block connected to the E input. The maximum power that can be delivered by the machine with a terminal voltage V_t = 1.0 p.u. and an internal voltage E = 1.0149 p.u. is $P_{max} = V_t*E/X$ = 1.0149/0.2 = 5.0745 p.u.

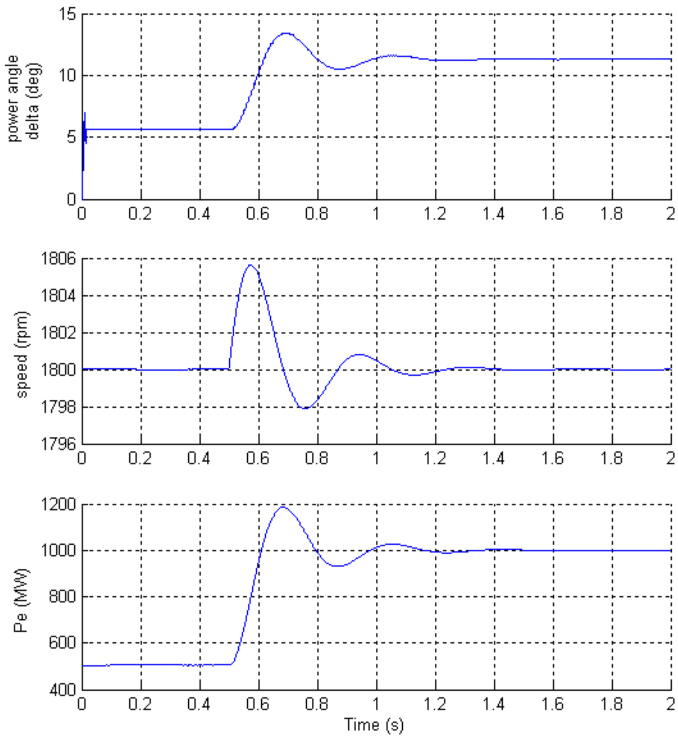
The damping factor Kd is adjusted in order to obtain a damping ratio ζ = 0.3. According to the formula given in the Description section, the required Kd value is $K_d = 4\zeta\sqrt{(\omega_s \cdot H \cdot P_{max})/2} = 64.3$.

Two Fourier blocks are used to measure the power angle δ. This angle is computed as the difference between the phase angle of phase A internal voltage and the phase angle of phase A terminal voltage.



In this demo, a step is performed on the mechanical power applied to the shaft. The machine is initially running in steady state with a mechanical power of 505 MW (mechanical power required for an output electrical power of 500 MW, considering the resistive losses). At t = 0.5 s the mechanical power is suddenly increased to 1000 MW.

Run the demo and observe the electromechanical transient on the Scope block displaying the power angle δ in degrees, the machine speed in rpm, and the electrical power in MW. Simulation results are shown in the following figure.



For an initial electrical power Pe = 500 MW (0.5 p.u.), the load angle δ is 5.65 degrees, which corresponds to the expected value:

$$Pe = \frac{V_t \cdot E \cdot \sin\delta}{X} = \frac{1.0 \cdot 1.0149 \cdot \sin(5.65°)}{0.2} = 0.5 \text{ p.u.}$$

As the mechanical power is stepped from 0.5 p.u. to 1.0 p.u., the load angle increases and goes through a series of underdamped oscillations (damping ratio $\zeta = 0.3$) before stabilizing to its new value of 11.3 degrees. The frequency of the oscillations is given by

$$f_n = (1/2\pi) \cdot \sqrt{\omega_s \cdot P_{max}/(2H)} = 2.84 \text{ Hz}$$

See Also

Excitation System, Hydraulic Turbine and Governor, Machine Measurement Demux, Powergui, Steam Turbine and Governor, Synchronous Machine

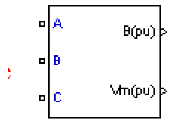
Purpose

Implement a phasor model of a three-phase, three-wire static var compensator

Library

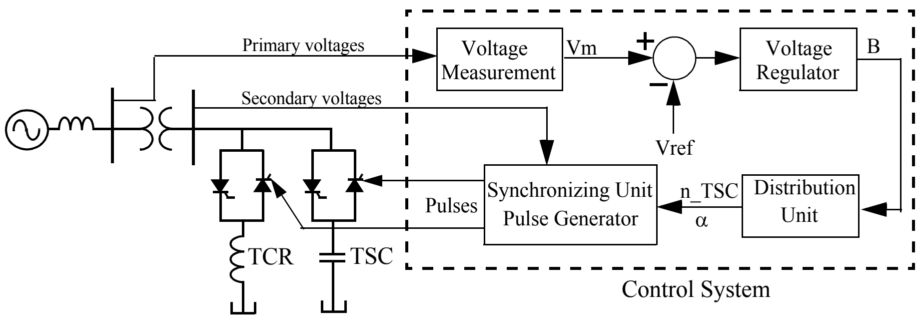
Phasor Elements

Description



The Static Var Compensator (SVC) is a device of the Flexible AC Transmission Systems (FACTS) family using power electronics to control power flow on power grids. The SVC regulates voltage at its terminal by controlling the amount of reactive power injected into or absorbed from the power system. When system voltage is low, the SVC generates reactive power (SVC capacitive). When system voltage is high, it absorbs reactive power (SVC inductive). The variation of reactive power is performed by switching three-phase capacitor banks and inductor banks connected on the secondary side of a coupling transformer. Each capacitor bank is switched on and off by three thyristor switches (Thyristor Switched Capacitor or TSC). Reactors are either switched on-off (Thyristor Switched Reactor or TSR) or phase-controlled (Thyristor Controlled Reactor or TCR).

The figure below shows a single-line diagram of a static var compensator and a simplified block diagram of its control system.



Single-line Diagram of an SVC and Its Control System Block Diagram

The control system consists of

- A measurement system measuring the positive-sequence voltage to be controlled

- A voltage regulator that uses the voltage error (difference between the measured voltage Vm and the reference voltage Vref) to determine the SVC susceptance B needed to keep the system voltage constant
- A distribution unit that determines the TSCs (and eventually TSRs) that must be switched in and out, and computes the firing angle α of TCRs
- A synchronizing system and a pulse generator that send appropriate pulses to the thyristors

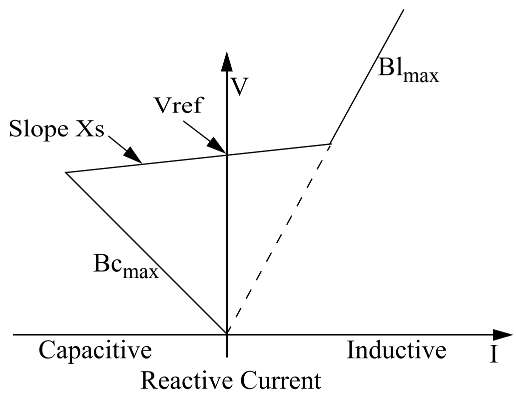
The Static Var Compensator block is a phasor model, and you must use it with the phasor simulation method, activated with the Powergui block. It can be used in three-phase power systems together with synchronous generators, motors, and dynamic loads to perform transient stability studies and observe impact of the SVC on electromechanical oscillations and transmission capacity. This model does not include detailed representations of the power electronics, the measurement system, or the synchronization system. These systems are approximated rather by simple transfer functions and delays that yield a correct representation at the system's fundamental frequency.

SVC V-I Characteristic

The SVC can be operated in two different modes:

- In voltage regulation mode (the voltage is regulated within limits as explained below)
- In var control mode (the SVC susceptance is kept constant)

When the SVC is operated in voltage regulation mode, it implements the following V-I characteristic.



SVC V-I characteristic

As long as the SVC susceptance B stays within the maximum and minimum susceptance values imposed by the total reactive power of capacitor banks (Bc_{max}) and reactor banks (Bl_{max}), the voltage is regulated at the reference voltage Vref. However, a voltage droop is normally used (usually between 1% and 4% at maximum reactive power output), and the V-I characteristic has the slope indicated in the figure. The V-I characteristic is described by the following three equations:

$V = Vref + Xs \cdot I$	SVC is in regulation range ($-Bc_{max} < B < Bl_{max}$)
$V = -\frac{I}{Bc_{max}}$	SVC is fully capacitive ($B = Bc_{max}$)
$V = \frac{I}{Bl_{max}}$	SVC is fully inductive ($B = Bl_{max}$)

where

V	Positive sequence voltage (p.u.)
I	Reactive current (p.u./Pbase) (I > 0 indicates an inductive current)
Xs	Slope or droop reactance (p.u./Pbase)

Bcmax	Maximum capacitive susceptance (p.u./Pbase) with all TSCs in service, no TSR or TCR
Blmax	Maximum inductive susceptance (p.u./Pbase) with all TSRs in service or TCRs at full conduction, no TSC
Pbase	Three-phase base power specified in the block dialog box

SVC Dynamic Response

When the SVC is operating in voltage regulation mode, its response speed to a change of system voltage depends on the voltage regulator gains (proportional gain Kp and integral gain Ki), the droop reactance Xs, and the system strength (short-circuit level).

For an integral-type voltage regulator (Kp = 0), if the voltage measurement time constant Tm and the average time delay Td due to valve firing are neglected, the closed-loop system consisting of the SVC and the power system can be approximated by a first-order system having the following closed-loop time constant:

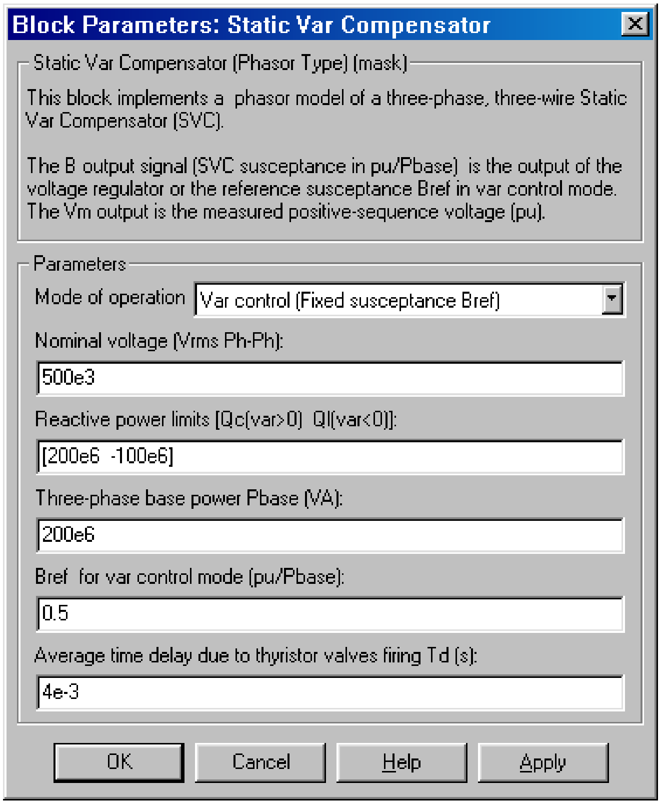
$$T_c = \frac{1}{Ki \cdot (Xs + Xn)}$$

where

Tc	Closed loop time constant
Ki	Proportional gain of the voltage regulator (p.u._B/p.u._V/s)
Xs	Slope reactance p.u./Pbase
Xn	Equivalent power system reactance (p.u./Pbase)

This equation demonstrates that you obtain a faster response speed when the regulator gain is increased or when the system short-circuit level decreases (higher Xn values). If you take into account the time delays due to voltage measurement system and valve firing, you obtain an oscillatory response and, eventually, an instability with too weak a system or too large a regulator gain.

Dialog Box and Parameters



Mode of operation

Specifies the SVC mode of operation. Select either `Voltage regulation` or `Var control (fixed susceptance Bref)`.

Nominal voltage

The nominal phase-to-phase voltage in Vrms.

Reactive power limits [Qc Ql]

The maximum SVC reactive powers at 1 p.u. voltage, in vars. Enter a positive value for the capacitive reactive power Qc (vars generated by the SVC) and a negative value for the inductive reactive power Ql (vars absorbed by the SVC).

Three-phase base power Pbase

Three-phase base power, in VA, used to specify the following parameters in p.u.: droop reactance Xs, gains Kp and Ki of the voltage PI regulator, and reference susceptance Bref. This base power is also used to normalize the output B susceptance signal.

Reference voltage Vref

This parameter is not visible when the **Mode of operation** parameter is set to `Var Control`.

Reference voltage, in p.u., used by the voltage regulator.

Droop Xs

This parameter is not visible when the **Mode of operation** parameter is set to `Var Control`.

Droop reactance, in p.u./Pbase, defining the slope of the V-I characteristic.

Voltage regulator [Kp Ki]

This parameter is not visible when the **Mode of operation** parameter is set to `Var Control`.

Proportional gain, in (p.u. of B)/(p.u. of V), and integral gain, in p.u._B/p.u._V/s, of the voltage regulator.

Bref for var control mode

This parameter is not visible when the **Mode of operation** parameter is set to `Voltage regulation`.

Reference susceptance, in p.u./Pbase, when the SVC is operating in var control mode.

Time constant of voltage measurement system Tm

This parameter is not visible when the **Mode of operation** parameter is set to `Var Control`.

Time constant, in seconds, of the first-order low-pass filter simulating the measurement system response time to a change of the fundamental voltage. A 1/2-cycle time constant can be used to approximate the transfer function of a Fourier-based measurement system using a one-cycle running average.

Average time delay due to thyristor valves firing Td

Average time delay simulating the noninstantaneous variation of thyristor fundamental current when the distribution unit sends a switching order to the pulse generator. Because pulses have to be synchronized with thyristor commutation voltages, this delay normally varies between 0 and 1/2 cycle. The suggested average value is 1/4 cycle.

Inputs and Outputs

`A B C`

The three terminals of the static var compensator.

`B (p.u.)`

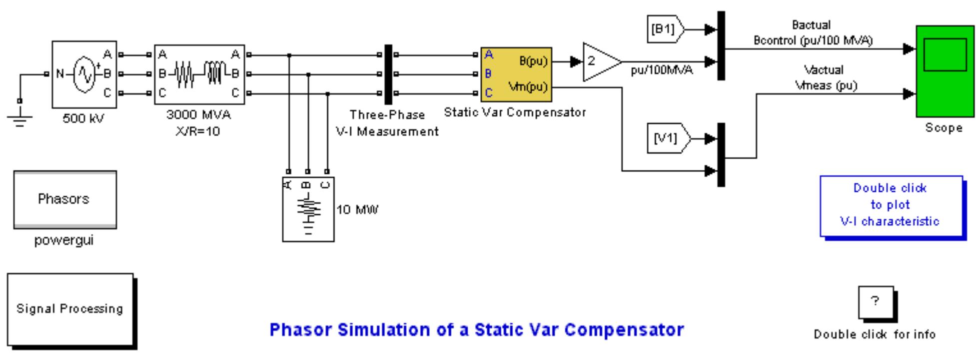
Simulink signal of the SVC susceptance, in p.u./Pbase. In voltage control mode, this control signal is the output of the voltage regulator. In var control mode, this value is the Bref value specified by the user. A positive value indicates that the SVC is capacitive. A negative value indicates that the SVC is inductive.

`Vm (p.u.)`

Simulink signal of the positive-sequence measured voltage, in p.u. This control signal is the output of the voltage measurement system.

Example

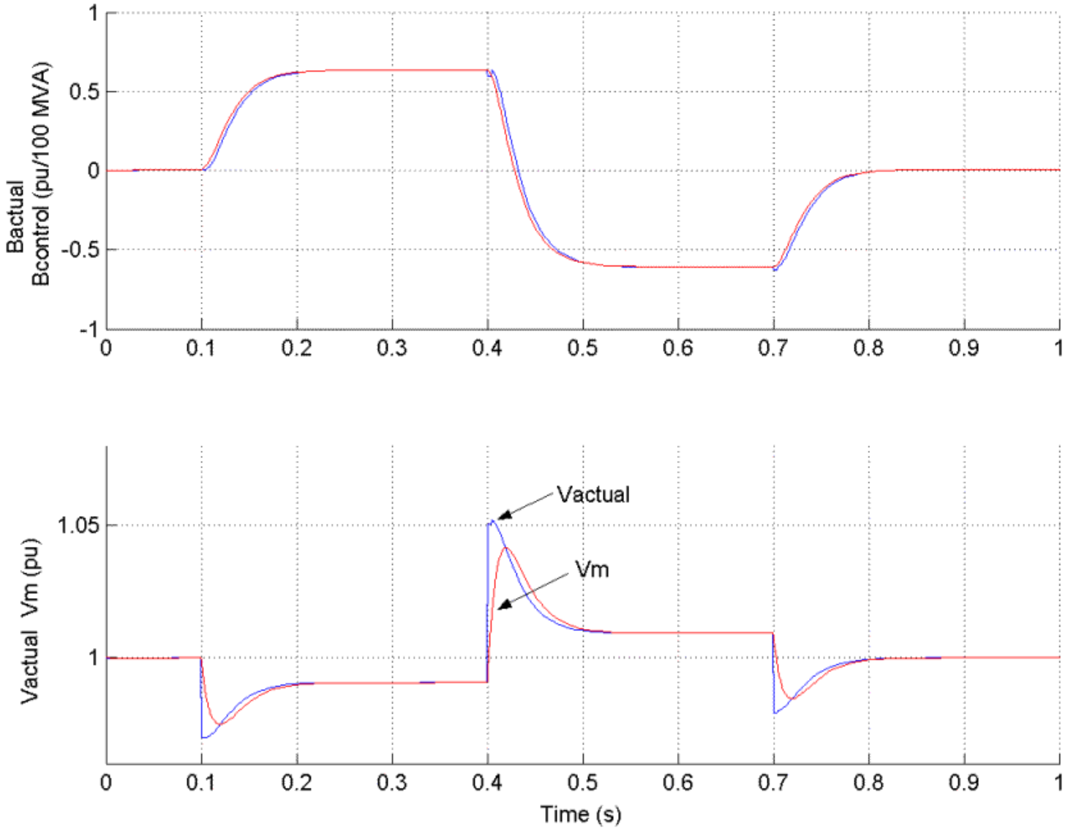
The `power_SVC` demo illustrates the steady-state and dynamic performance of an SVC regulating voltage on a 500 kV, 60 Hz, 3000 MVA system. The Static Var Compensator block models a +200 Mvar/–100 Mvar SVC.



Open the SVC block menu and look at its parameters. The SVC is set to `Voltage regulation` mode with a reference voltage Vref = 1.0 pu. The voltage droop reactance is 0.03 p.u./200 MVA, so that the voltage varies from 0.97 p.u. to 1.015 p.u. when the SVC current goes from fully capacitive to fully inductive. Double-click the blue block located below the Scope block to display the SVC V-I characteristic.

The Three-Phase Programmable Voltage Source is used to vary the system voltage and observe the SVC performance. Initially the source is generating its nominal voltage (500 kV). Then, voltage is successively decreased (0.97 p.u. at t = 0.1 s), increased (1.03 p.u. at t = 0.4 s) and finally returned to nominal voltage (1 p.u. at t = 0.7 s).

Start the simulation and observe the SVC dynamic response to voltage steps on the Scope. Waveforms are reproduced on the figure below. Trace 1 shows the actual positive-sequence susceptance B1 and control signal output B of the voltage regulator. Trace 2 shows the actual system positive-sequence voltage V1 and output Vm of the SVC measurement system.



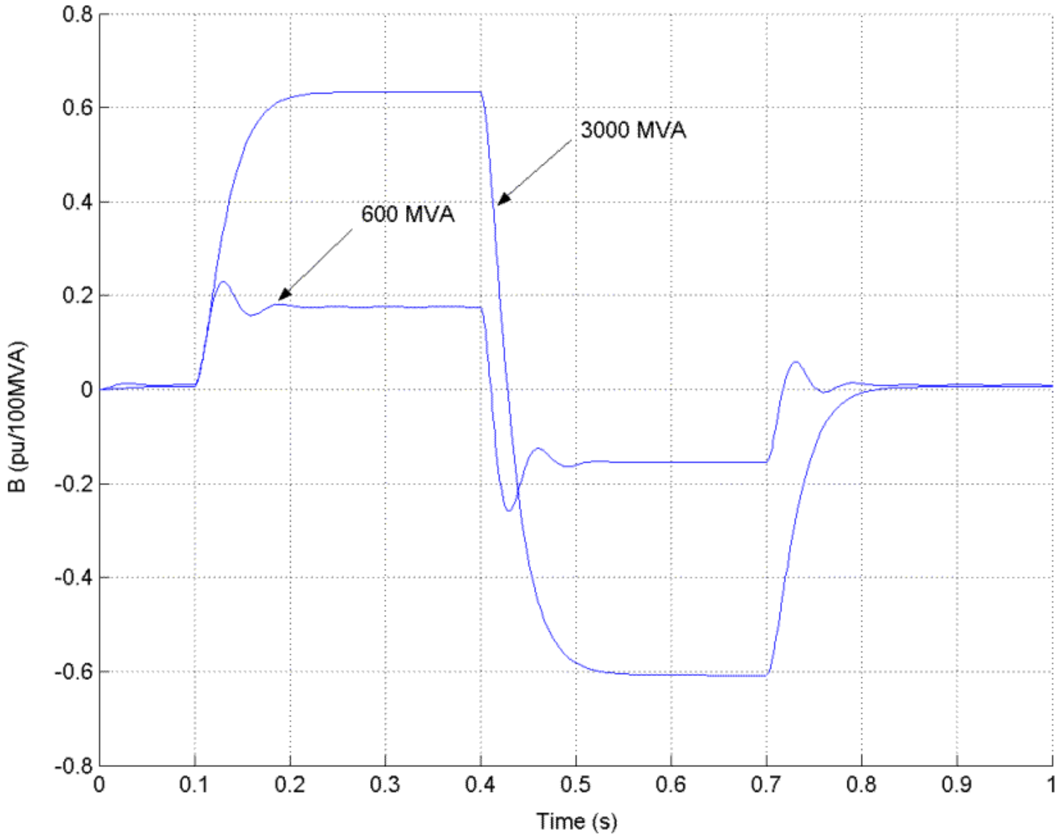
The SVC response speed depends on the voltage regulator integral gain Ki (proportional gain Kp is set to zero), system strength (reactance Xn), and droop (reactance Xs).

As mentioned above, neglecting the voltage measurement time constant Tm and the average time delay Td due to valve firing, the system can be approximated by a first-order system having a closed-loop time constant:

$$T_c = \frac{1}{Ki \cdot (Xs + Xn)}$$

With given system parameters (Ki = 300; Xn = 0.0667 p.u./200 MVA; Xs = 0.03 p.u./200 MVA), the closed-loop time constant is Tc = 0.0345 s.

If you increase the regulator gain or decrease the system strength, Tm and Td are no longer negligible, and you instead observe an oscillatory response and eventually instability. The figure below compares the SVC susceptance (B output of the voltage regulator) for two different short-circuit levels: 3000 VA and 600 MVA.



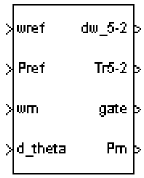
See Also

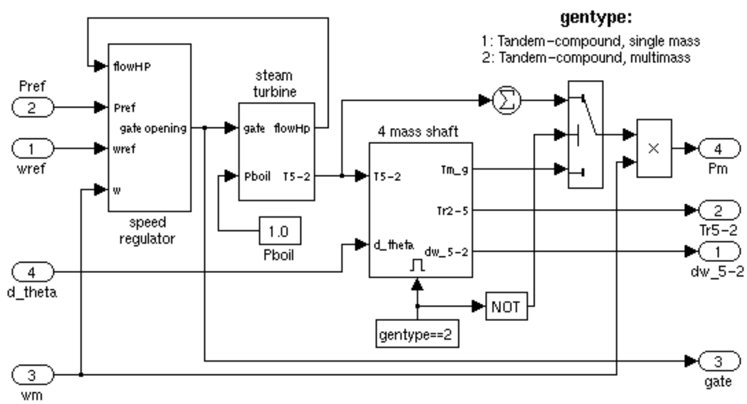
Powergui, Thyristor

Purpose Model the dynamics of a speed governing system, steam turbine, and multimass shaft

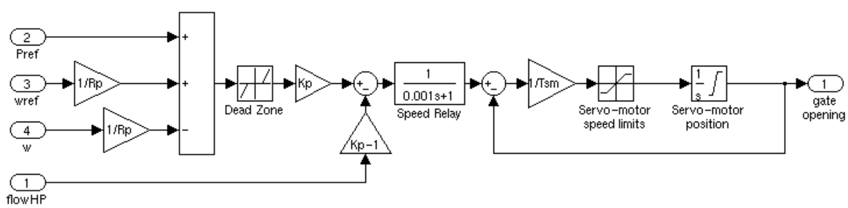
Library Machines

Description The Steam Turbine and Governor block implements a complete tandem-compound steam prime mover, including a speed governing system, a four-stage steam turbine, and a shaft with up to four masses.

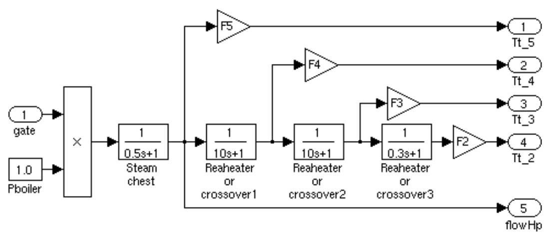




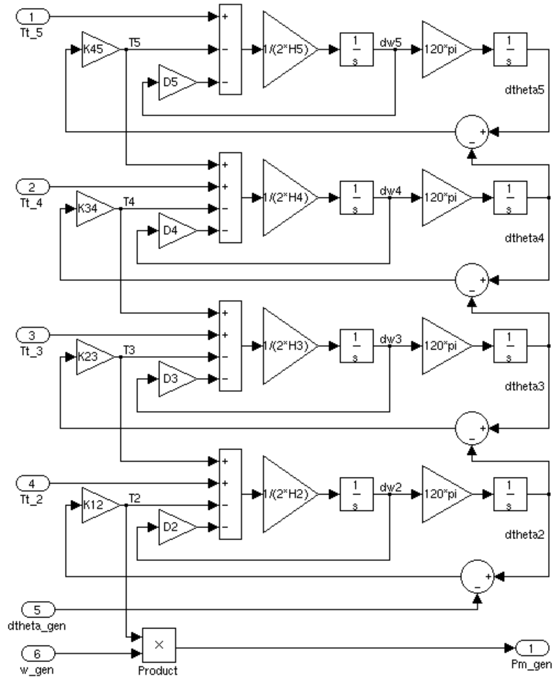
The speed governing system consists of a proportional regulator, a speed relay, and a servomotor controlling the gate opening. It is similar to one of the models proposed in [1].



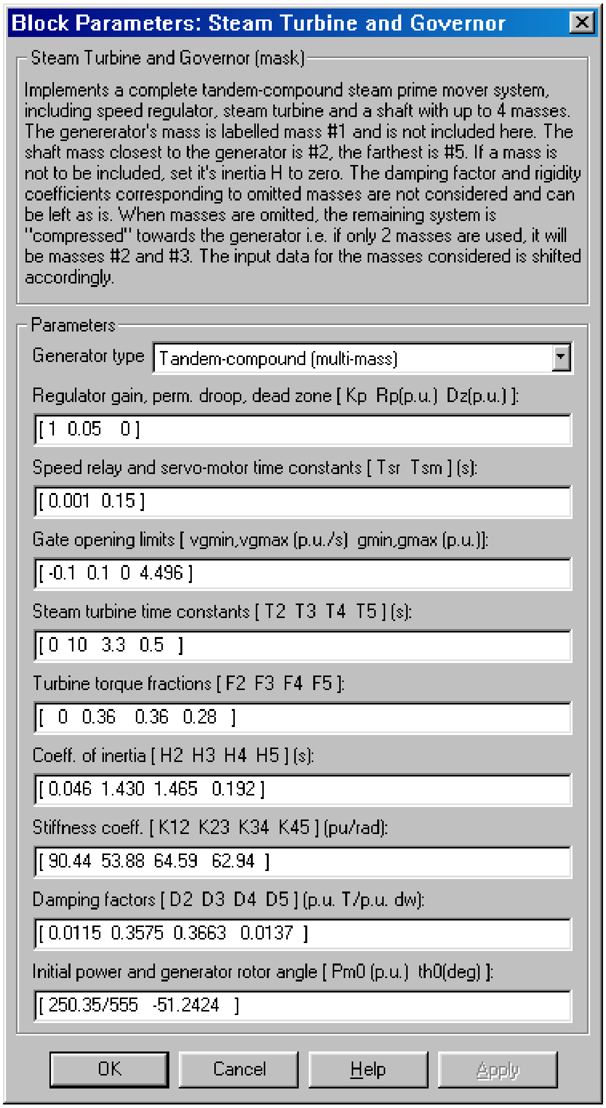
The steam turbine has four stages, each modeled by a first-order transfer function. The first stage represents the steam chest while the three other stages represent either reheaters or crossover piping. The boiler is not modeled and boiler pressure is constant at 1.0 p.u. Fractions F2 to F5 are used to distribute the turbine power to the various shaft stages:



The shaft models a four-mass system, which is coupled to the mass in the Synchronous Machine model for a total of five masses. The machine's mass is labeled mass #1. The mass in the Steam Turbine and Governor block, which is closest to the machine's mass, is mass #2, while the mass farthest from the machine is mass #5. The shaft is characterized by mass inertias H, damping factors D, and rigidity coefficients K. If you choose to simulate a single-mass shaft, the entire four-mass shaft subsystem in the Steam Turbine and Governor block is disabled and all the torque from the turbine is added together and applied to the machine's mass:


1
Tt_5
K45
T5
D5
1/(2*H5)
dw5
120*pi
dtheta5
2
Tt_4
K34
T4
D4
1/(2*H4)
dw4
dtheta4
3
Tt_3
K23
T3
D3
1/(2*H3)
dw3
dtheta3
4
Tt_2
K12
T2
D2
1/(2*H2)
dw2
dtheta2
5
dtheta_gen
6
w_gen
Product
1
Pm_gen

Dialog Box and Parameters



Generator type

Specifies rotor type: single mass or multimass tandem-compound. If you choose a single-mass system, the multimass shaft subsystem in the Steam

Turbine and Governor block is disabled and the turbine's output torques are summed together and applied to the single mass in the Synchronous Machine block.

Regulator gain, permanent droop, dead zone

The gain Kp, permanent droop Rp (p.u.), and dead-zone width Dz (p.u.). Set gain to 3 if you want to use the steam flow feedback loop. Otherwise, set gain to 1.

Speed relay and servo-motor time constants

The speed relay and gate servomotor time constants Tsr (s) and Tsm (s).

Gate opening limits

The minimum and maximum gate opening speed vgmin and vgmax (both in p.u./s), and minimum and maximum gate opening gmin and gmax (both in p.u.).

Steam turbine time constants

The turbine time constants T2 to T5 (s). Numbered consistently with turbine torque fractions and mass numbers; i.e., T5 is the time constant of the first turbine stage, which models the steam chest.

Turbine torque fractions

The turbine torque fractions F2 to F5. Must total 1, otherwise an error message appears. Fraction numbers correspond to mass numbers; i.e., F2 is the fraction of torque to be applied to mass #2 of the multimass shaft.

Coefficient of inertia; Stiffness coefficient; Damping factors

Only visible if generator type is `multimass`. Coefficients of inertia H2 to H5 (s), stiffness coefficients K12 to K45 (p.u./rad), and damping factors D2 to D5 (p.u. torque / p.u. speed deviation) are associated with the masses of the multimass shaft. K12 corresponds to the rigidity coefficient between masses #1 and #2, and so on.

Note If you do not want to simulate all four masses in the multimass shaft, simply set the inertia of unwanted masses to 0. The rigidity coefficient and damping factor corresponding to omitted masses are not considered. When masses are not simulated, the remaining system is "compressed" toward the generator; i.e., if only two masses are used (excluding the generator), they are masses #2 and #3. The input data for the masses considered are shifted accordingly. In any case, inertias must be consistent with torque fractions. You cannot set an inertia to 0 and set the corresponding torque fraction to a nonzero value. However, you can set a torque fraction to 0 and set the corresponding mass inertia to a nonzero value.

Initial power and generator rotor angle

If the shaft is multimass, enter the initial mechanical power Pm0 (p.u.) and initial generator angle θe0 (degrees). If the shaft is single mass, enter only initial mechanical power.

Initial mechanical power is automatically updated by the load flow utility of the Powergui block. Initial angle is also computed by the load flow utility and is written in the associated Synchronous Machine block dialog box.

Inputs and Outputs

The first input is the speed reference, in p.u. It is normally connected to a Constant block with the value set to 1.0 p.u.

The second input is the electrical power reference, in p.u. It is set to a constant value corresponding to the initial active power drawn from the Synchronous Machine block connected to the Steam Turbine and Governor block.

The third input is the generator's speed, in p.u. This is one of the signals in the last output of the Synchronous Machine model (internal variables).

The fourth input is the generator's power angle deviation. It is also one of the signals in the last output of the Synchronous Machine model (internal variables).

The first output is a vector containing the speed deviations, in p.u., of masses #5 to #2, in that order.

The second output is also a vector containing the torques, in p.u., transmitted by masses #5 to #2.

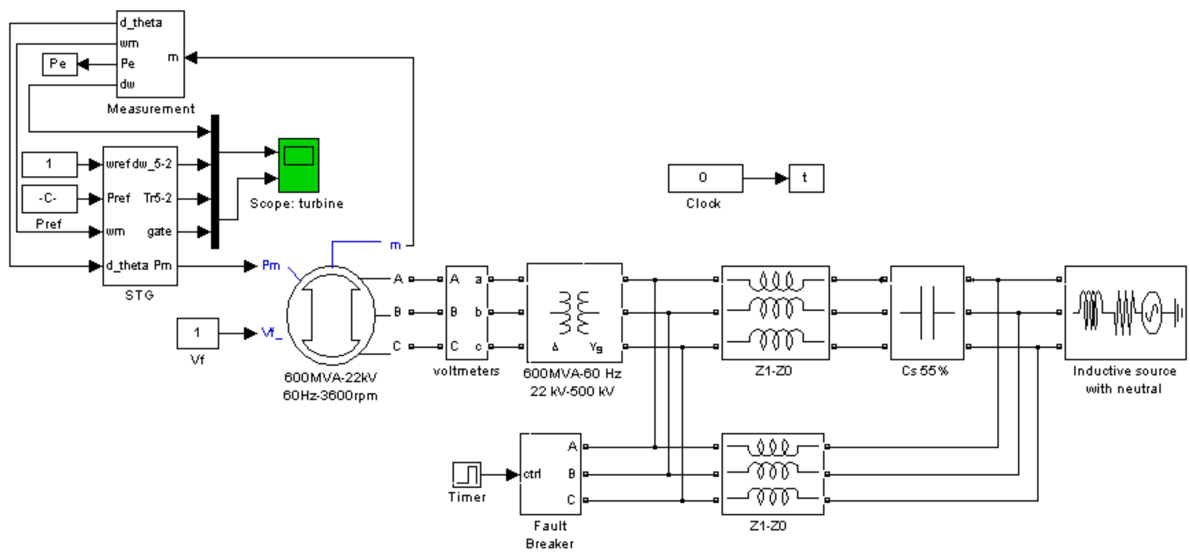
The third output is the gate opening in p.u.

The fourth output is the mechanical power, in p.u., that you must connect to the first input of a Synchronous Machine block.

Example

The power_thermal demo illustrates the use of the Steam Turbine and Governor block. This system is an IEEE benchmark used to study subsynchronous resonance and particularly torque amplification after a fault on a series-compensated power system [2]. It consists in a single generator connected to an infinite bus via two transmission lines, one of which is series compensated. The subsynchronous mode introduced by the compensation capacitor after a fault has been applied and cleared excites the oscillatory torsional modes of the multimass shaft and the torque amplification phenomenon can be observed. Open the Simulink diagram by typing power_thermal.

This system is slightly different from the one presented in [2]. Since we are using the Synchronous Machine mass as the first mass, we cannot model the exciter's mass as is done in [2]. Therefore, our system has only three masses, representing the generator's rotor (mass #1) and the turbine's low and high pressure stages (masses #2 and #3, respectively).

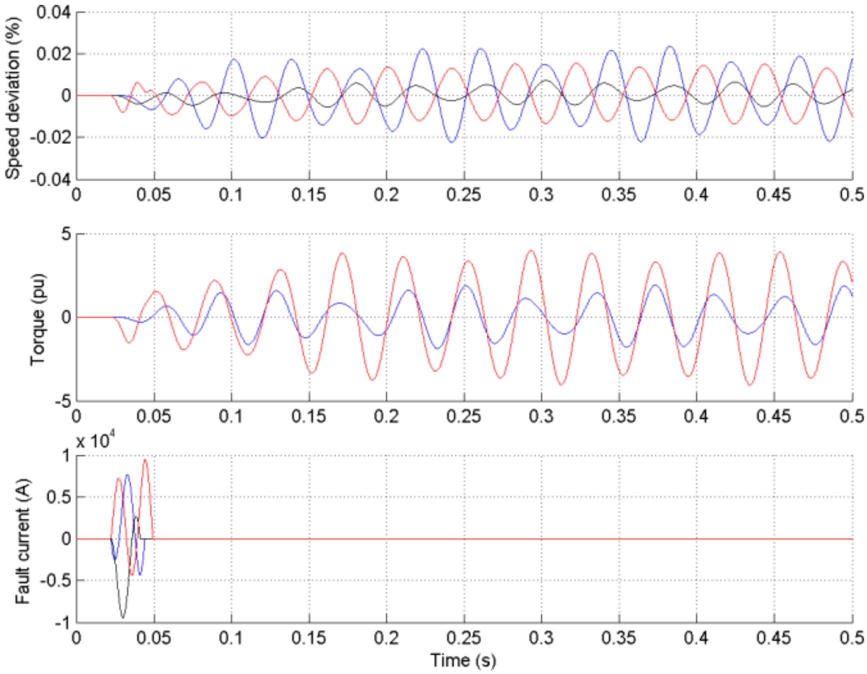


In order to start the simulation in steady state, you must first initialize the synchronous machine and steam turbine by using the **Load Flow and Machine Initialization** utility of the Powergui. Set the generator as a PV generator with initial power of 100 kW (1e5 W) or 0.0167% of nominal power.

This is done to simulate an initially unloaded generator. The load flow returns initial mechanical power of 100,020 W. This value was converted into p.u. by dividing it by the generator's nominal VA rating (600e6 VA) and the result was entered as the first initial condition in the Steam Turbine and Governor block. The second initial condition is the generator's initial angle. This value is computed by the load flow and is written in the initial conditions vector of the generator. The Steam Turbine and Governor block is now correctly initialized. The electrical power (load) reference, the second input of the Steam Turbine and Governor block, is set to the desired electrical power supplied by the generator, in p.u. (1e5/600e6, or 0.1/600).

This test is performed without regulators. The speed governing system is forced to output a constant value by setting the gate opening limits very close to each other, around the initial gate opening, which is also the initial mechanical power in p.u. (100 010/600e6, or 0.00016668 p.u.). The machine's excitation voltage is also set to a constant value (1.00358 p.u.), which is computed by the load flow.

Run the simulation. Once the simulation is completed, observe the mass speed deviations and torques and the fault current.



The peak values of all these signals correspond within 3% to those given in Table 5, case 1A, of [2]. The torque amplification is clearly observed on all masses of the shaft system. The high-pressure mass (#3) transmits a peak torque of 1.91 p.u. to the low-pressure mass (#2), while the low-pressure mass transmits a peak torque of 4.05 p.u. to the generator's rotor (mass #1).

References

[1] IEEE committee report, "Dynamic models for steam and hydro turbines in power system studies," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-92, No. 6, 1973, pp. 1904-1915.

[2] IEEE Subsynchronous resonance working group, "Second benchmark model for computer simulation of subsynchronous resonance," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 5, 1985, pp. 1057-1066.

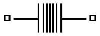
See Also

Excitation System, Hydraulic Turbine and Governor, Powergui, Synchronous Machine

Purpose Implement a metal-oxide surge arrester

Library Elements

Description

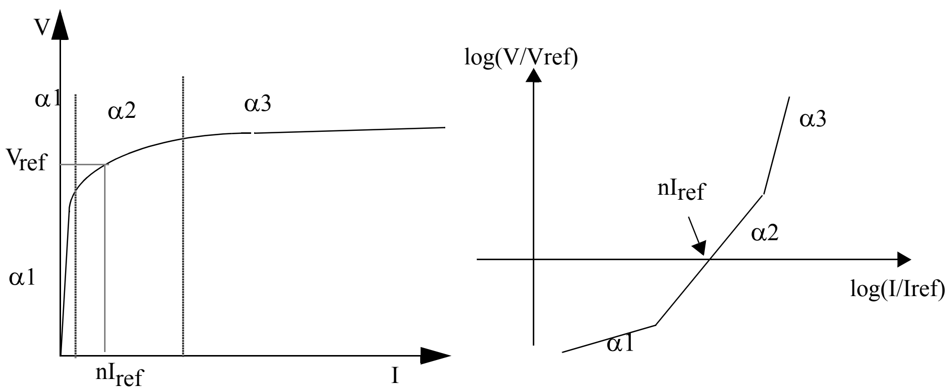


The Surge Arrester block implements a highly nonlinear resistor used to protect power equipment against overvoltages. For applications requiring high power dissipation, several columns of metal-oxide discs are connected in parallel inside the same porcelain housing. The nonlinear V-I characteristic of each column of the surge arrester is modeled by a combination of three exponential functions of the form

$$\frac{V}{V_{ref}} = k_i\left(\frac{I}{I_{ref}}\right)^{1/\alpha_i}$$

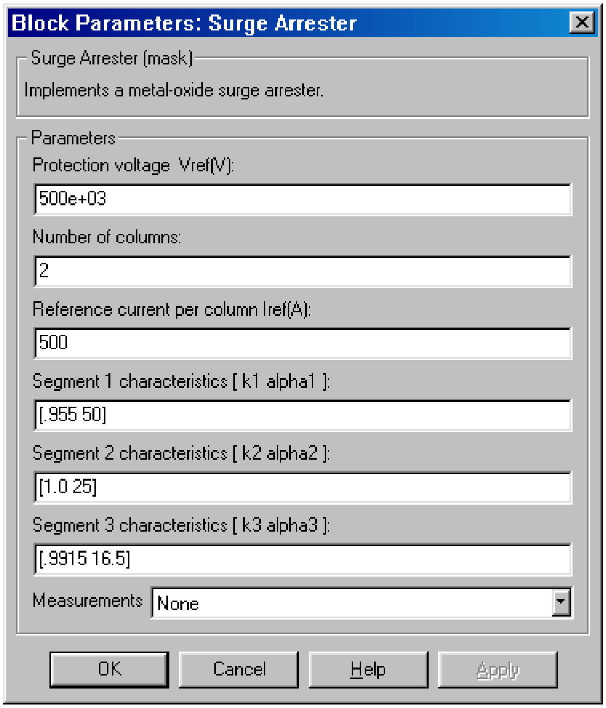
The protection voltage obtained with a single column is specified at a reference current (usually 500 A or 1 kA). Default parameters k and α given in the dialog box fit the average V-I characteristic provided by the main metal-oxide arrester manufacturers and they do not change with the protection voltage. The required protection voltage is obtained by adding discs of zinc oxide in series in each column.

This V-I characteristic is graphically represented as follows (on a linear scale and on a logarithmic scale).



n = Number of columns in parallel

Dialog Box and Parameters



Protection voltage Vref

The protection voltage of the Surge Arrester block, in volts (V).

Number of columns

The number of metal-oxide disc columns. The minimum is one.

Reference current per column Iref

The reference current of one column used to specify the protection voltage, in amperes (A).

Segment 1 characteristics

The k and α parameters of segment 1.

Segment 2 characteristics

The k and α parameters of segment 2.

Segment 3 characteristics

The k and α characteristics of segment 3.

Measurements

Select `Branch voltage` to measure the voltage across the Surge Arrester block terminals.

Select `Branch current` to measure the current flowing through the Surge Arrester block.

Select `Branch voltage and current` to measure the surge arrester voltage and current.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Branch voltage	`Ub:`
Branch current	`Ib:`

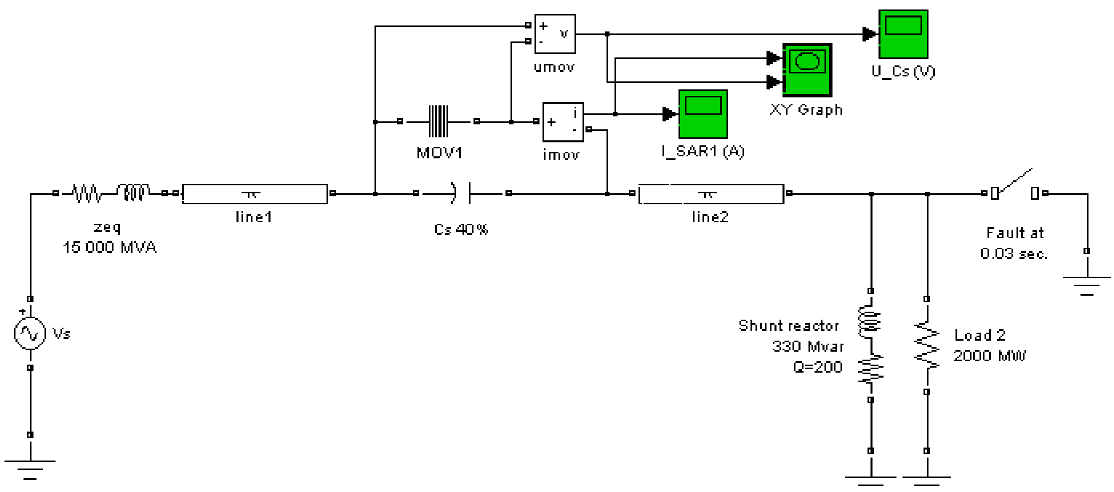
Limitations

The Surge Arrester block is modeled as a current source driven by the voltage appearing across its terminals. Therefore, it cannot be connected in series with an inductor or another current source. As the Surge Arrester block is highly nonlinear, a stiff integrator algorithm must be used to simulate the circuit. `ode15s` or `ode23tb` with default parameters usually gives the best simulation speed. For continuous simulation, in order to avoid an algebraic loop, the voltage applied to the nonlinear resistance is filtered by a first-order filter with a time constant of 0.01 microseconds. This very fast time constant does not significantly affect the result accuracy. When the Surge Arrester block is used in a discrete system, a time delay of one simulation step is used. This delay can cause numerical oscillations if the sample time is too large.

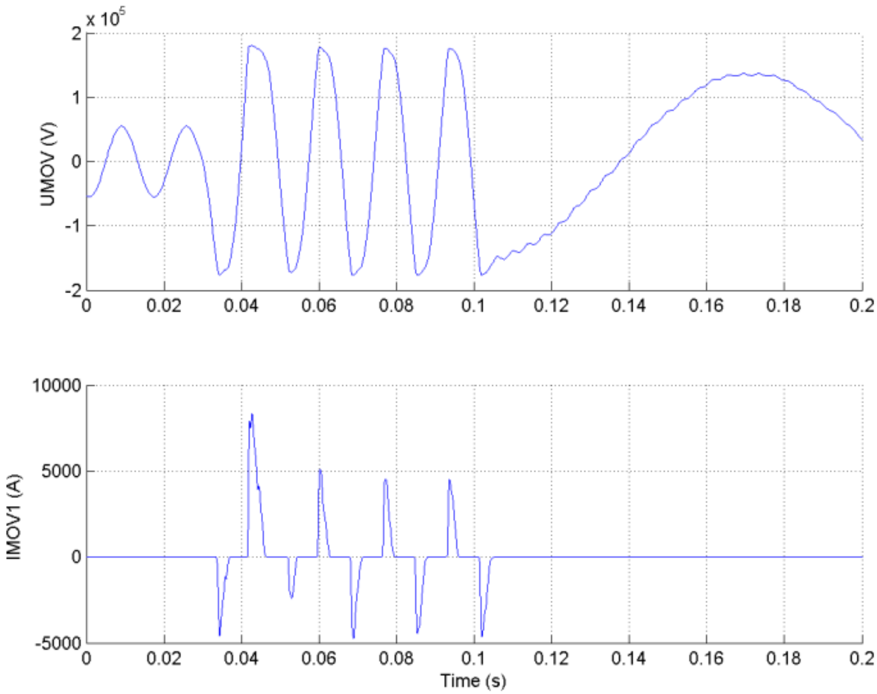
Example

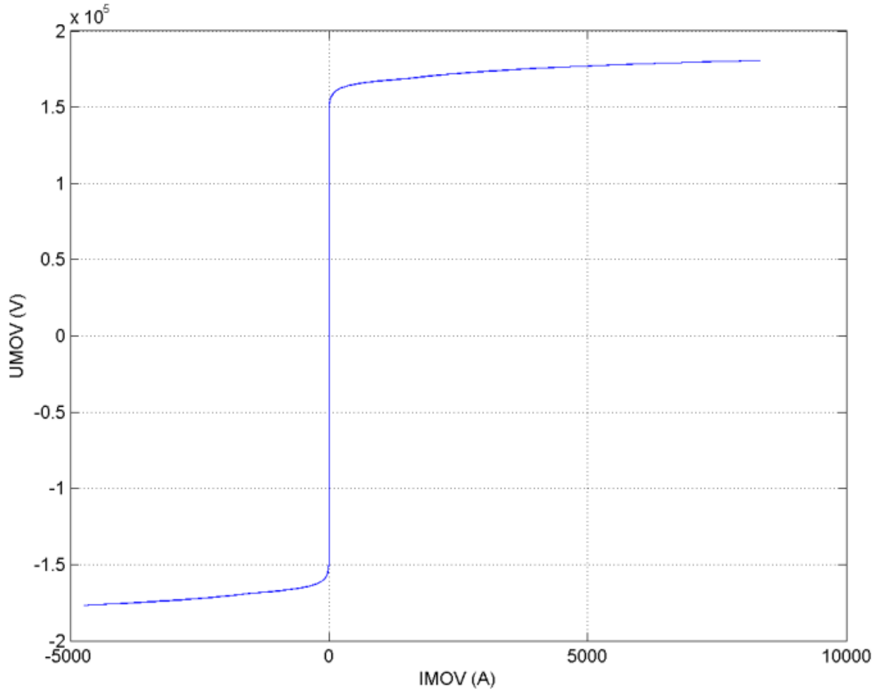
The `power_arrester` demo illustrates the use of metal-oxide varistors (MOV) on a 735 kV series-compensated network. Only one phase of the network is represented. The capacitor connected in series with the line is protected by a 30 column arrester. At t = 0.03 seconds, a fault is applied at the load terminals.

The current increases in the series capacitor and produces an overvoltage that is limited by the Surge Arrester block. Then the fault is cleared at t = 0.1 seconds.



At fault application, the resulting overvoltage makes the MOV conduct. The waveforms displayed by Umov and Imov measurements as well as the V-I characteristic plotted by the X-Y scope are shown below:


x 10^5
2
1
0
-1
-2
UMOV (V)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
10000
5000
0
-5000
IMOV1 (A)
Time (s)



See Also Multimeter

Purpose

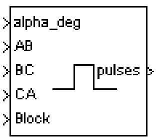
Implement a synchronized pulse generator to fire the thyristors of a six-pulse converter

Library

Extras/Control Block

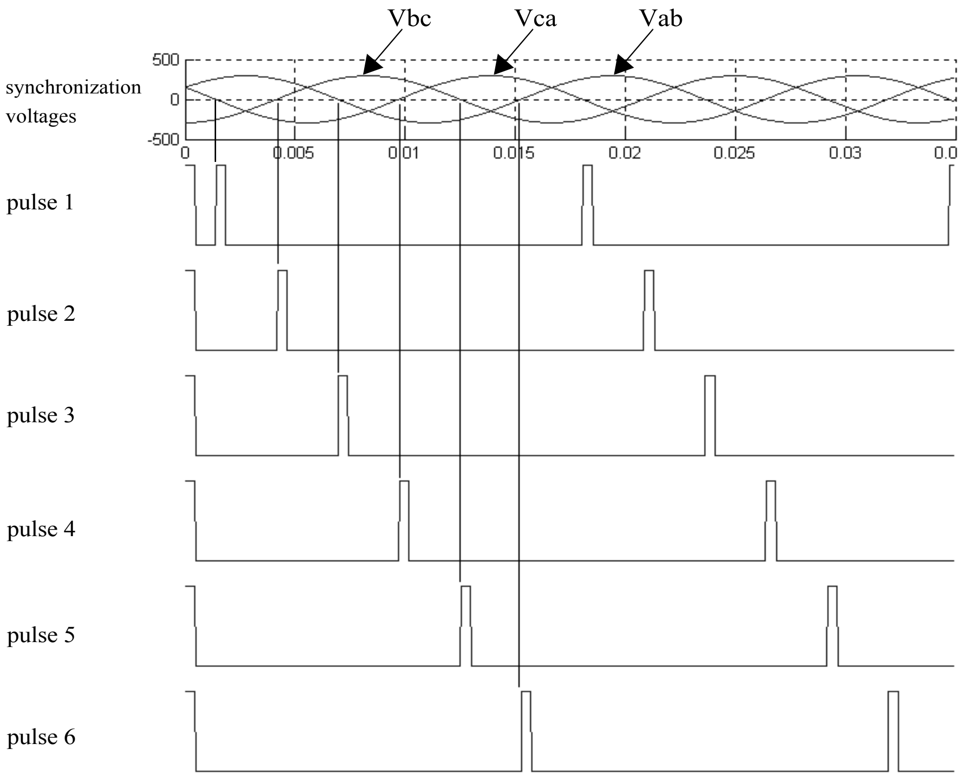
A discrete version of this block is available in the Extras/Discrete Control Blocks library.

Description



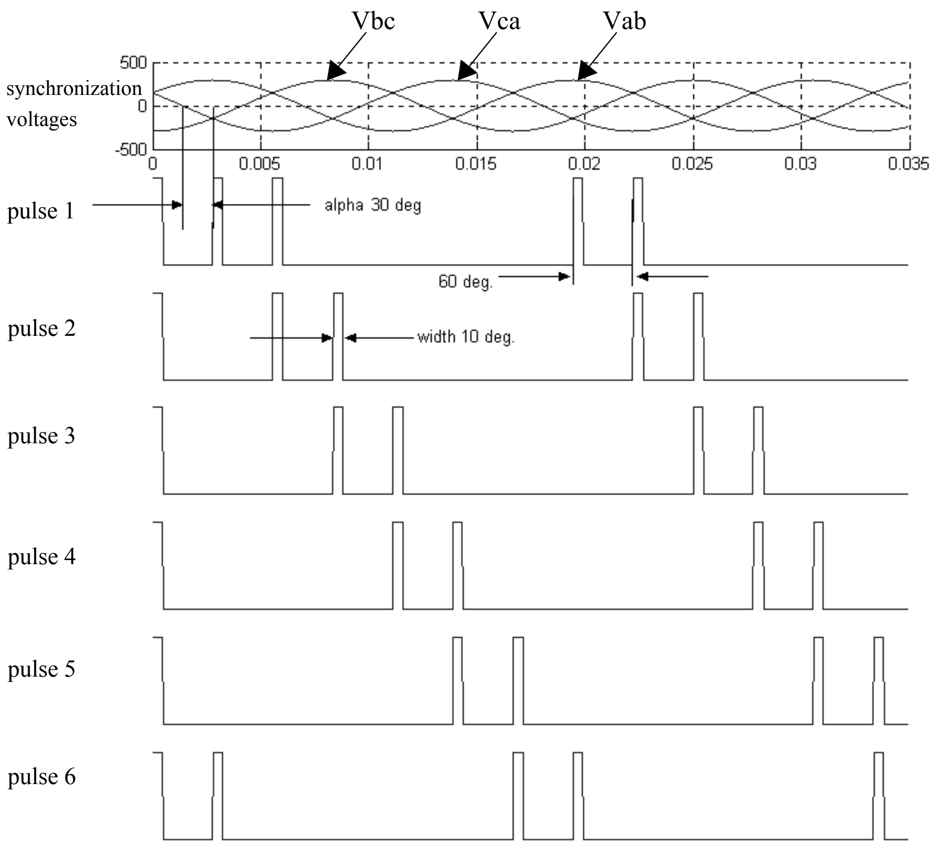
The Synchronized 6-Pulse Generator block can be used to fire the six thyristors of a six-pulse converter. The output of the block is a vector of six pulses individually synchronized on the six thyristor voltages. The pulses are generated alpha degrees after the increasing zero crossings of the thyristor commutation voltages.

The figures below display the synchronization of the six pulses for an alpha angle of 0 degrees. The pulses are generated exactly at the zero crossings of the three line-to-line synchronization voltages.

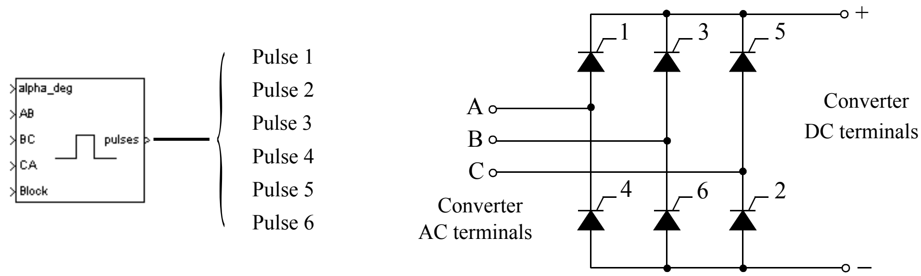


The Synchronized 6-Pulse Generator block can be configured to work in double-pulsing mode. In this mode two pulses are sent to each thyristor: a first pulse when the alpha angle is reached, then a second pulse 60 degrees later, when the next thyristor is fired.

The figures below display the synchronization of the six pulses for an alpha angle of 30 degrees and with double-pulsing mode. Notice that the pulses are generated 30 degrees after the zero crossings of the line-to-line.

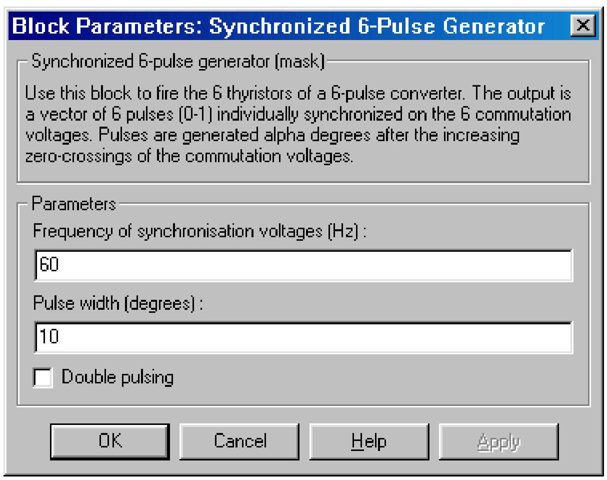


The pulse ordering at the output of the block corresponds to the natural order of commutation of a three-phase thyristor bridge. When you connect the Synchronized 6-Pulse Generator block to the pulses input of the Universal Bridge block (with the thyristors as the power electronic device), the pulses are sent to the thyristors in the following order:



When you build your own three-phase thyristor bridge with single thyristor blocks, you need to connect the pulse signals of the Synchronized 6-Pulse Generator block to the gate inputs of the corresponding thyristors.

Dialog Box and Parameters



Frequency of synchronization voltages

The frequency, in hertz, of the synchronization voltages. It usually corresponds to the frequency of the network.

Pulse width

The width of the pulses, in degrees.

Double pulsing

If selected, the generator sends to each thyristor a first pulse when the alpha angle is reached, and then a second pulse 60 degrees later when the next thyristor in the sequence is fired.

Inputs and Outputs

`alpha_deg`

Input 1 is the alpha firing signal, in degrees. This input can be connected to a Constant block, or it can be connected to a controller system to control the pulses of the generator.

`AB, BC, CA`

Inputs 2, 3, and 4 are the phase-to-phase synchronization voltages Vab, Vbc, and Vca. The synchronization voltages should be in phase with the three phase-phase voltages at the converter AC terminals. Synchronization voltages are normally derived at the primary windings of the converter transformer. If the converter is connected to the delta winding of a Wye/Delta transformer, the synchronization voltages should be the phase-to-ground voltages of the primary windings.

`Freq`

Available only with the discrete version of the Synchronized 6-Pulse Generator. This input should be connected to a constant block containing the fundamental frequency, in hertz, or to a PLL tracking the frequency of the system.

`block`

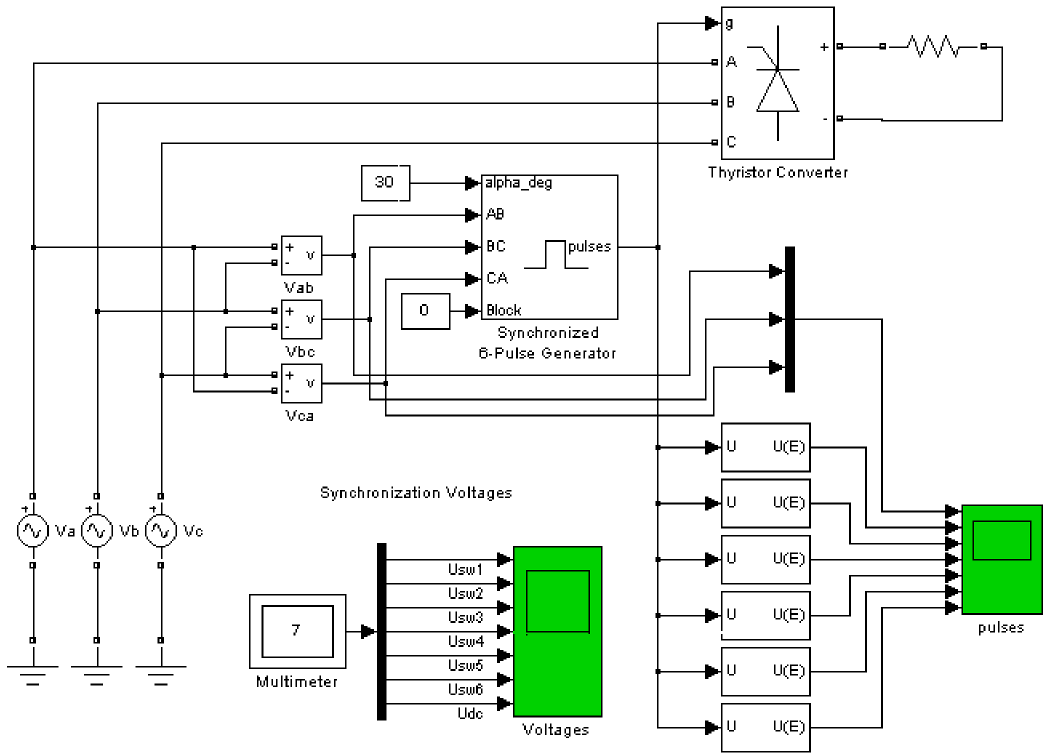
Input 5 allows you to block the operation of the generator. The pulses are disabled when the applied signal is greater than zero.

`pulses`

The output contains the six pulse signals.

Example

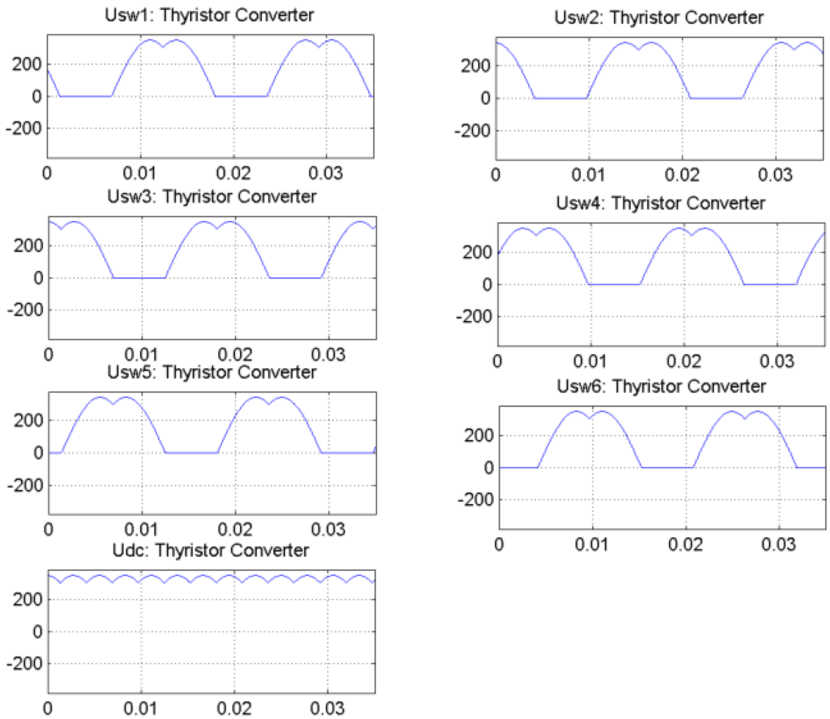
The `power_sixpulses` demo uses a Synchronized 6-Pulse Generator block to fire the thyristors of a six-pulse thyristor bridge. The bridge is fed by a three-phase voltage source (200 V peak line-to-ground or 245 V RMS line-to-line) and it is connected to a resistive load.



A first simulation is performed with an alpha angle of 0 degrees. Open the Constant block connected at input 1 of the Synchronized 6-Pulse Generator block and set its value to 0. Start the simulation. The average voltage is

$$V_{dc} = \frac{3\sqrt{2}}{\pi}E = \frac{3\sqrt{2}}{\pi}245 = 331 \text{ volts}$$

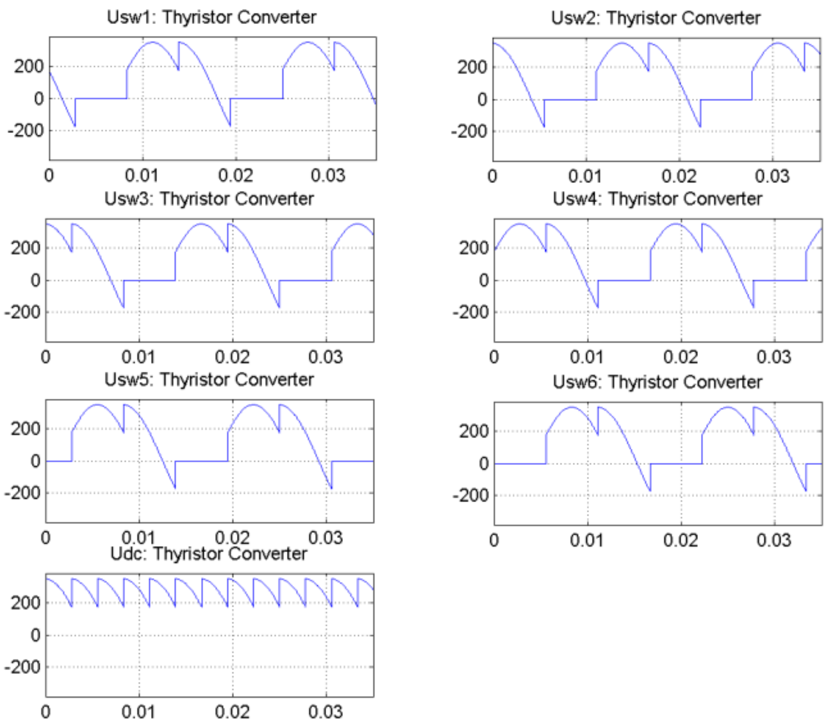
The six thyristor voltages are displayed in the next figure. The resulting DC voltage at the output of the rectifier is also displayed (average value of 331 V).



Now change the value of the alpha angle to 30 degrees and start the simulation. Notice that the waveforms of the thyristor voltages look different from the previous case. The thyristors start conducting 30 degrees after their commutation voltage becomes positive and the resulting DC voltage at the output of the rectifier is lower. Its average value is now

$$V_{dc} = \frac{3\sqrt{2}}{\pi}E\cos(\alpha) = \frac{3\sqrt{2}}{\pi}245\cos(30°) = 286 \text{ volts}$$

The thyristor voltages and DC voltage for alpha = 30 degrees are



The figures show that the mean value of the DC voltage can be controlled by the alpha angle applied to the Synchronized 6-Pulse Generator block.

See Also

The `power_hvdc` demo illustrates the use of the Discrete Synchronized 6-Pulse Generator block.

Synchronized 12-Pulse Generator

Purpose

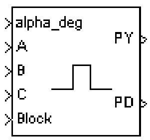
Implement a synchronized pulse generator to fire the thyristors of a twelve-pulse converter

Library

Extras/Control Blocks

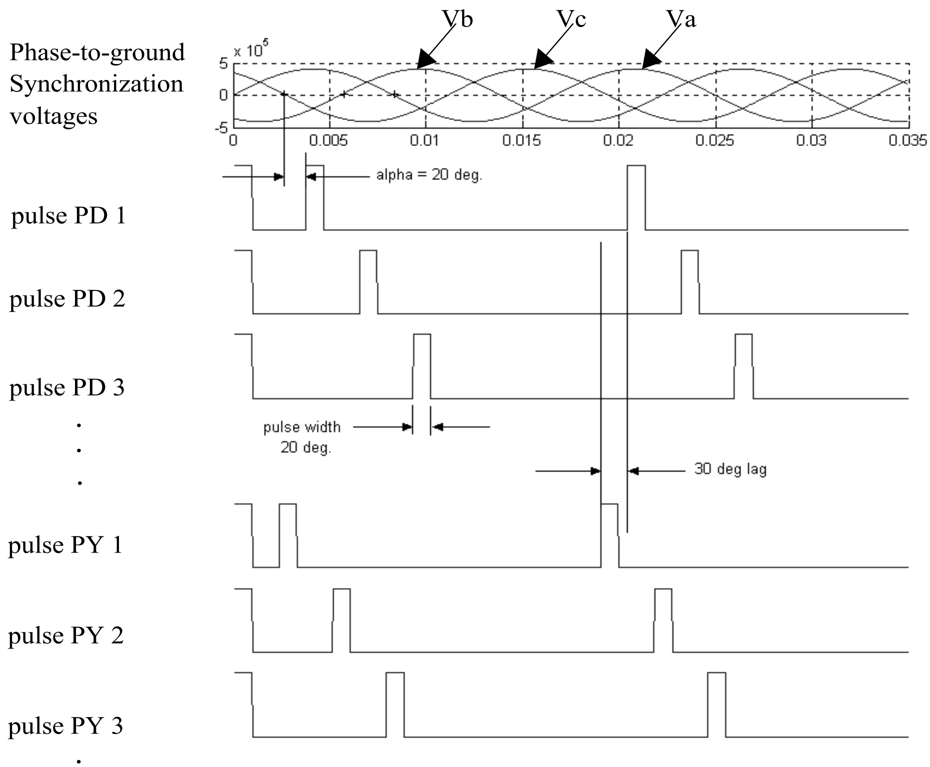
A discrete version of this block is available in the Extras/Discrete Control Blocks library.

Description



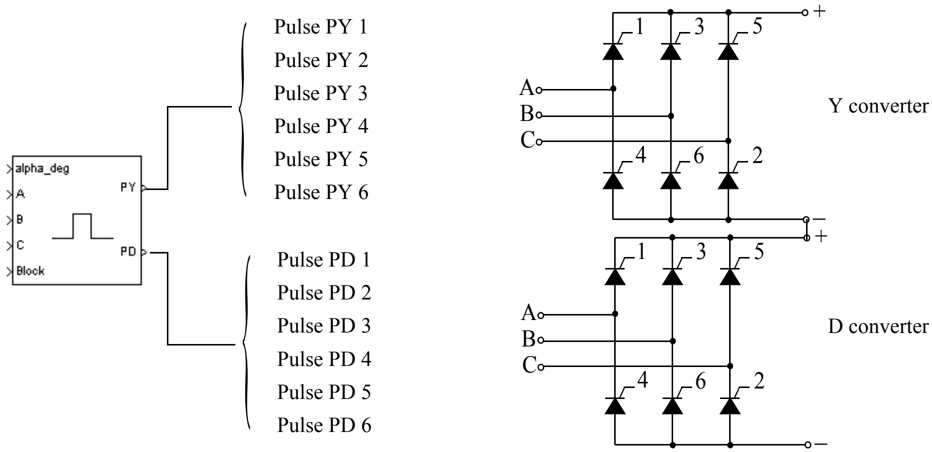
The Synchronized 12-Pulse Generator block generates two vectors of six pulses synchronized on the twelve thyristor commutation voltages. The first set of pulses, denoted PY, is sent to the six-pulse bridge connected to the wye secondary winding of the Y/Y/Delta converter transformer. It is generated alpha degrees after the zero crossing of the phase-to-phase synchronization voltages. The second set of pulses, denoted PD, is sent to the six-pulse bridge connected to the delta secondary winding of the converter transformer. It lags the PY pulses by 30 degrees.

The figure below shows the three synchronization voltages and the first three pulses of the two output vectors.The synchronization voltages are the three phase-to-ground voltages Va, Vb, Vc measured on the primary side (Y) of the Y/Y/Delta converter transformer.

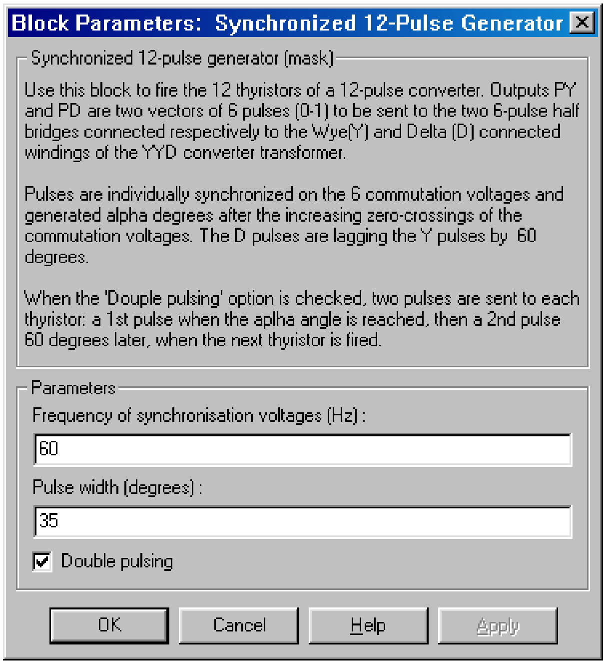


The phase-to-ground A,B, and C voltages are provided to the generator, and the two sets of phase-to-phase synchronization voltages required by the two six-pulse bridges are generated internally.

The ordering of the pulses in the two outputs of the block corresponds to the natural order of commutation of a three-phase thyristor bridge. When you connect the Synchronized 12-Pulse Generator block outputs to the pulse inputs of the Universal Bridge blocks (with the thyristor device), the pulses are sent to the thyristors in the following way:



Dialog Box and Parameters



Frequency of synchronization voltages

The frequency, in hertz, of the synchronization voltages. It usually corresponds to the frequency of the network.

Pulse width

The width of the pulses, in degrees.

Double pulsing

If selected, the generator sends to each thyristor a first pulse when the alpha angle is reached, and then a second pulse 60 degrees later when the next thyristor in the sequence is fired. The double pulsing is applied separately on the two vectors of pulses.

Inputs and Outputs

`alpha_deg`

Input 1 is the alpha firing signal, in degrees. This input can be connected to a Constant block, or it can be connected to a controller system to control the pulses of the generator.

`A, B, C`

Inputs 2, 3, and 4 are the phase-to-ground synchronization voltages Va, Vb, and Vc. The synchronization voltages should be measured at the primary side of the converter transformer.

`Freq`

Available only with the discrete version of the Synchronized 6-Pulse Generator. This input should be connected to a constant block containing the fundamental frequency, in hertz, or to a PLL tracking the frequency of the system.

`block`

Input 5 allows you to block the operation of the generator. The pulses are disabled when the applied signal is greater than zero.

`PY`

Output 1 contains the six-pulse signals to be sent to the six-pulse thyristor converter connected to the Y secondary winding of the converter transformer.

`PD`

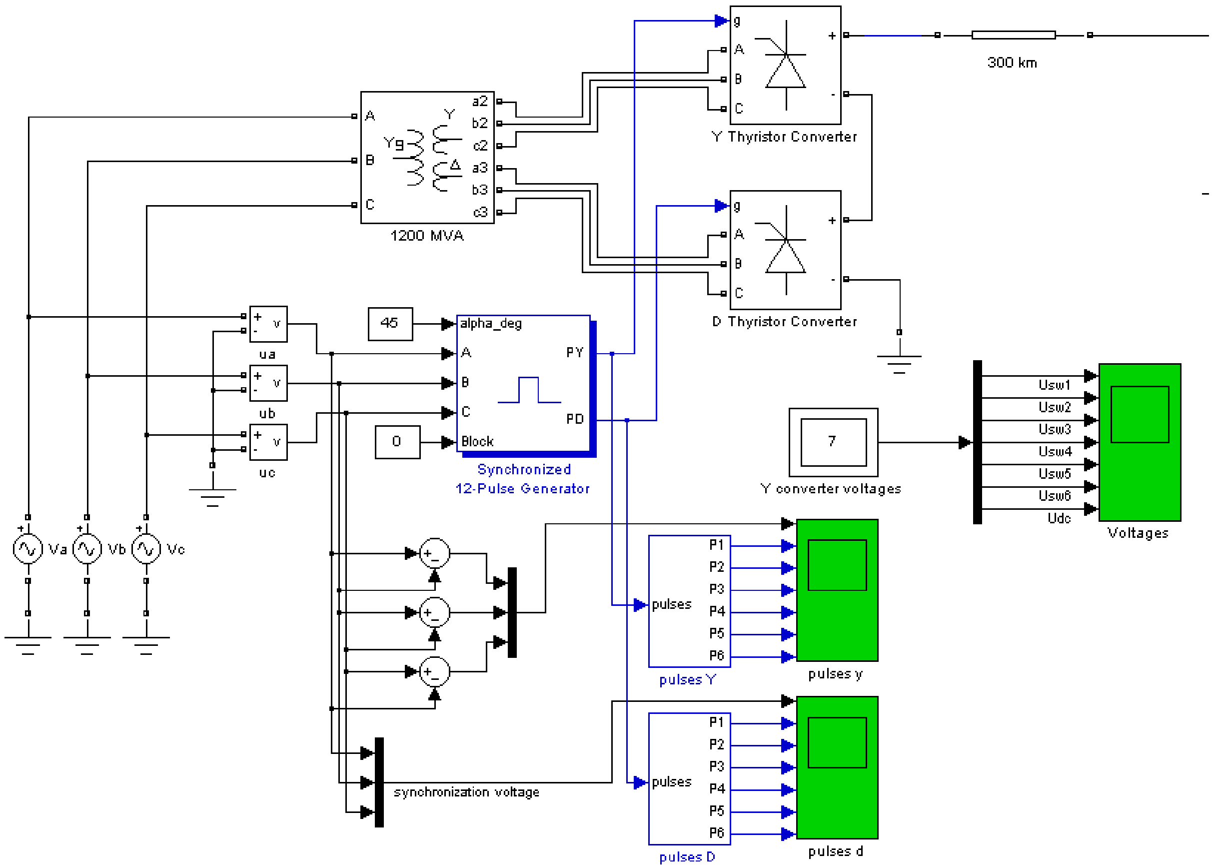
Output 2 contains the six-pulse signals to be sent to the six-pulse thyristor converter connected to the Delta (D) secondary winding of the converter transformer.

Example

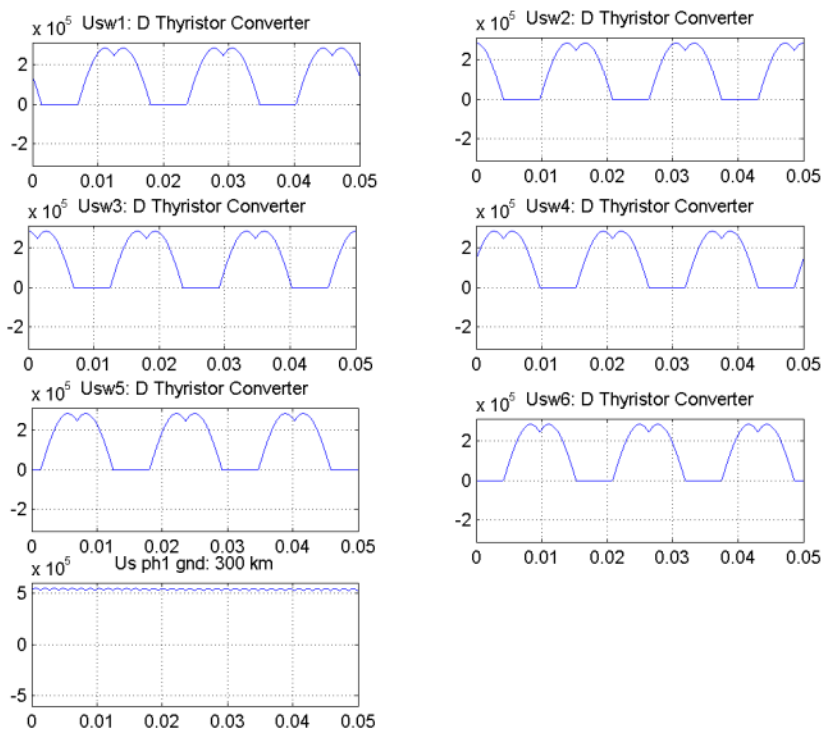
In the `power_twelvepulses` demo a Synchronized 12-Pulse Generator block is used to fire the thyristors of a twelve-pulse thyristor bridge built with two six-pulse bridges. The bridge is fed by a three-winding three-phase transformer (500 kV / 200 kV / 200 kV). The Y-connected secondary feeds the first six-pulse bridge. The Delta secondary feeds the second bridge. The transformer is assumed to be ideal (no leakage reactances, no resistance). The expected DC voltage obtained for alpha = 0 is

$$V_{dc} = 2\frac{3\sqrt{2}}{\pi}200\ \text{kV} = 540\ \text{kV}$$

The two bridge rectifiers are connected in series and a 300 km DC line is connected to the rectifier.



A first simulation is performed with an alpha angle of 0 degrees. Open the Constant block connected at input 1 of the Synchronized 12-Pulse Generator block and set its value to 0. Start the simulation. The voltages of the thyristors of the D thyristor Converter block are displayed in the next figure. The resulting DC voltage at the input terminal of the transmission line is also displayed (average value = 540 kV).



Compare the DC voltage generated by the Synchronized 12-Pulse Generator with the DC voltage you obtained with the Synchronized 6-Pulse Generator. Notice that the ripple in the DC voltage waveform is lower. The rectifier voltage contains the harmonics 12*k (k = 1,2,...).

See Also

The `power_hvdc12pulse` demo illustrates the use of the Discrete Synchronized 12-Pulse Generator block.

Synchronized 6-Pulse Generator

Synchronous Machine

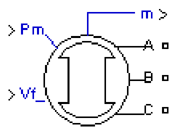
Purpose

Model the dynamics of a three-phase round-rotor or salient-pole synchronous machine

Library

Machines

Description

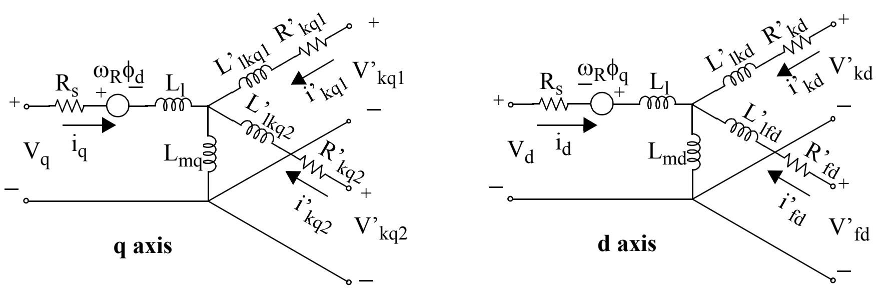


The Synchronous Machine block operates in generator or motor modes. The operating mode is dictated by the sign of the mechanical power (positive for generator mode, negative for motor mode). The electrical part of the machine is represented by a sixth-order state-space model and the mechanical part is the same as in the Simplified Synchronous Machine block.

The model takes into account the dynamics of the stator, field, and damper windings. The equivalent circuit of the model is represented in the rotor reference frame (qd frame). All rotor parameters and electrical quantities are viewed from the stator. They are identified by primed variables. The subscripts used are defined as follows:

- *d,q*: d and q axis quantity
- *R,s*: Rotor and stator quantity
- *l,m*: Leakage and magnetizing inductance
- *f,k*: Field and damper winding quantity

The electrical model of the machine is



with the following equations.

$$V_d = R_s i_d + \frac{d}{dt}\varphi_d - \omega_R \varphi_q$$

$$V_q = R_s i_q + \frac{d}{dt}\varphi_q + \omega_R \varphi_d$$

$$V'_{fd} = R'_{fd} i'_{fd} + \frac{d}{dt}\varphi'_{fd}$$

$$V'_{kd} = R'_{kd} i'_{kd} + \frac{d}{dt}\varphi'_{kd}$$

$$V'_{kq1} = R'_{kq1} i'_{kq1} + \frac{d}{dt}\varphi'_{kq1}$$

$$V'_{kq2} = R'_{kq2} i'_{kq2} + \frac{d}{dt}\varphi'_{kq2}$$

$$\varphi_d = L_d i_d + L_{md}(i'_{fd} + i'_{kd})$$

$$\varphi_q = L_q i_q + L_{mq} i'_{kq}$$

$$\varphi'_{fd} = L'_{fd} i'_{fd} + L_{md}(i_d + i'_{kd})$$

$$\varphi'_{kd} = L'_{kd} i'_{kd} + L_{md}(i_d + i'_{fd})$$

$$\varphi'_{kq1} = L'_{kq1} i'_{kq1} + L_{mq} i_q$$

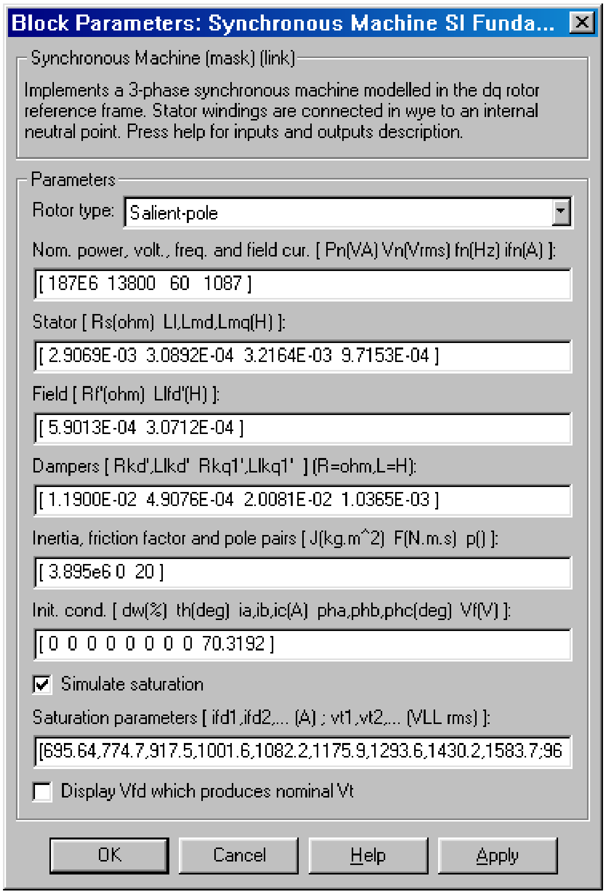
$$\varphi'_{kq2} = L'_{kq2} i'_{kq2} + L_{mq} i_q$$

Note that this model assumes currents flowing into the stator windings. The measured stator currents returned by the Synchronous Machine block (Ia, Ib, Ic, Id, Iq) are the currents flowing out of the machine.

Dialog Box and Parameters

In the **powerlib** library you can choose between three Synchronous Machine blocks to specify the parameters of the model.

Fundamental Parameters in SI Units



Rotor type

Specify rotor type: `Salient-pole` or `Round` (cylindrical). This choice affects the number of rotor circuits in the q-axis (damper windings).

Nominal power, voltage, frequency, and field current

The total three-phase apparent power Pn (VA), RMS line-to-line voltage Vn (V), frequency fn (Hz), and field current ifn (A).

The nominal field current is the current that produces nominal terminal voltage under no-load conditions. This model was developed with all quantities viewed from the stator. The nominal field current makes it possible to compute the transformation ratio of the machine, which allows you to apply the field voltage viewed from the rotor, as in real life. This also allows the field current, which is a variable in the output vector of the model, to be viewed from the rotor. If the value of the nominal field current is not known, you must enter 0 or leave it blank. Since the transformation ratio cannot be determined in this case, you have to apply the field voltage as viewed from the stator and the field current in the output vector is also viewed from the stator.

Stator

The resistance Rs (Ω), leakage inductance Lls (H), and d-axis and q-axis magnetizing inductances Lmd (H) and Lmq (H).

Field

The field resistance Rf' (Ω) and leakage inductance Llfd' (H), both referred to the stator.

Dampers

The d-axis resistance Rkd' (Ω) and leakage inductance Llkd' (H), the q-axis resistance Rkq1' (Ω) and leakage inductance Llkq1' (H), and (only if round rotor) the q-axis resistance Rkq2' (Ω) and leakage inductance Llkq2' (H). All these values are referred to the stator.

Inertia, friction factor, and pole pairs

The inertia coefficient J (kg.m^2), damping coefficient D (N.m.s./rad), and number of pole pairs p.

Initial conditions

The initial speed deviation Δω (% of nominal speed), electrical angle of the rotor θe (degrees), line current magnitudes ia, ib, ic (A) and phase angles pha, phb, phc (degrees), and the initial field voltage Vf (V).

You can specify the initial field voltage in one of two ways. If you know the nominal field current (first line, last parameter), enter in the dialog box the initial field voltage in volts DC referred to the rotor. Otherwise, enter a zero as nominal field current, as explained earlier, and specify the initial field voltage in volts DC referred to the stator. You can determine the nominal field voltage viewed from the stator by selecting the **Display Vfd which produces a nominal Vt** check box at the bottom of the dialog box.

Simulate saturation

Specifies whether magnetic saturation of rotor and stator iron is to be simulated or not.

Saturation parameters

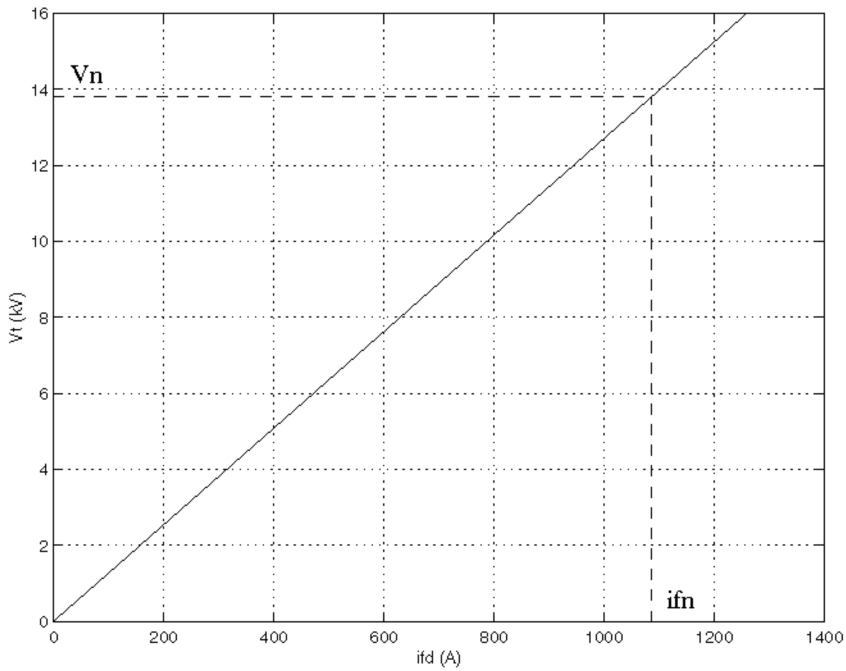
The no-load saturation curve parameters. Magnetic saturation of stator and rotor iron is modeled by a nonlinear function (in this case a polynomial) using points on the no-load saturation curve. You must enter a 2-by-n matrix, where n is the number of points taken from the saturation curve. The first row of this matrix contains the values of field currents, while the second row contains values of corresponding terminal voltages. The first point (first column of the matrix) must correspond to the point where the effect of saturation begins.

You must select the **Simulate saturation** check box to simulate saturation. This check box allows you to enter the matrix of parameters for simulating the saturation. If you do not want to model saturation in your simulation, do not select the **Simulate saturation** check box. In this case the relationship between ifd and Vt obtained is linear (no saturation).

Display Vfd which produces a nominal Vt

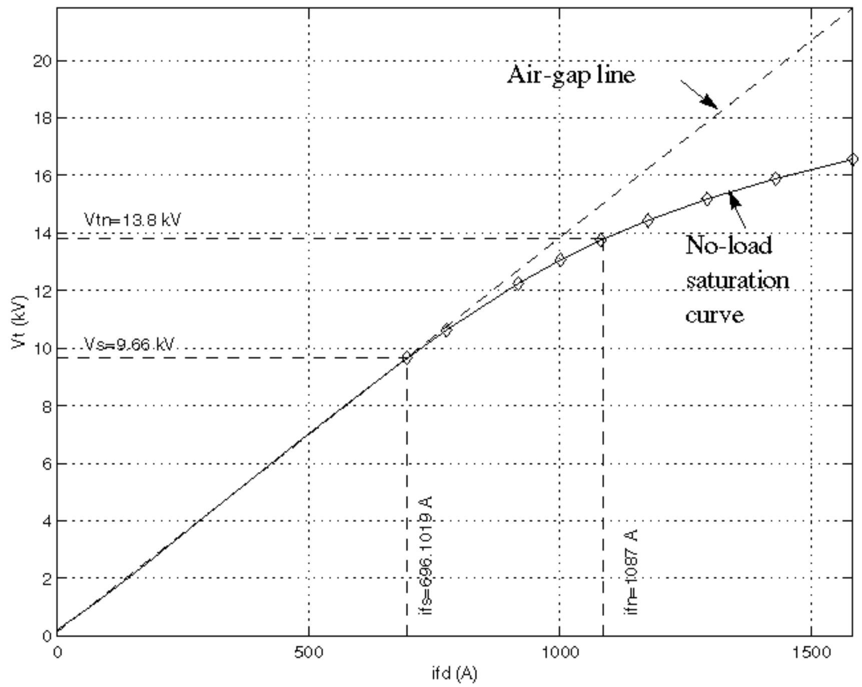
Select to determine the nominal field voltage viewed from the stator.

As an example, without saturation, a typical curve might be as shown below. Here ifn is 1087 A and Vn is 13800 V RMS line-to-line, which is also 11268 V peak line-to-neutral.



Saturation is modeled by fitting a polynomial to the curve corresponding to the matrix of points you enter. The more points you enter, the better the fit to the original curve.

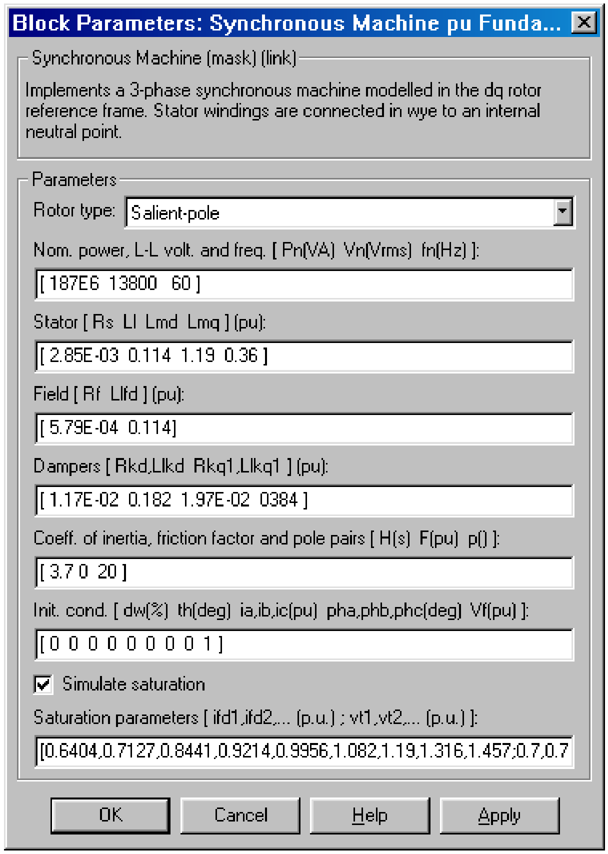
The next figure illustrates goodness of fit graphically (the diamonds are the actual points entered in the dialog box).



In this particular case, the following values were used:

ifn	1087 A
ifd	[695.64, 774.7, 917.5, 1001.6, 1082.2, 1175.9, 1293.6, 1430.2, 1583.7] A
Vt	[9660, 10623, 12243, 13063, 13757, 14437, 15180, 15890, 16567] V

Fundamental Parameters in p.u.



Rotor type

Specifies rotor type: `Salient-pole` or `Round` (cylindrical).

Nominal power, L-L voltage, and frequency

Total three-phase apparent power (VA), RMS line-to-line voltage (V), frequency (Hz), and field current (A).

This line is identical to the first line of the fundamental parameters in SI dialog box, except that you do not specify a nominal field current. This value is not required here because we do not need the transformation ratio.

Since rotor quantities are viewed from the stator, they are converted to p.u. using the stator base quantities derived from the preceding three nominal parameters.

Stator; Field; Dampers

Contain exactly the same parameters as in the previous dialog box, but they are expressed here in p.u. instead of SI units.

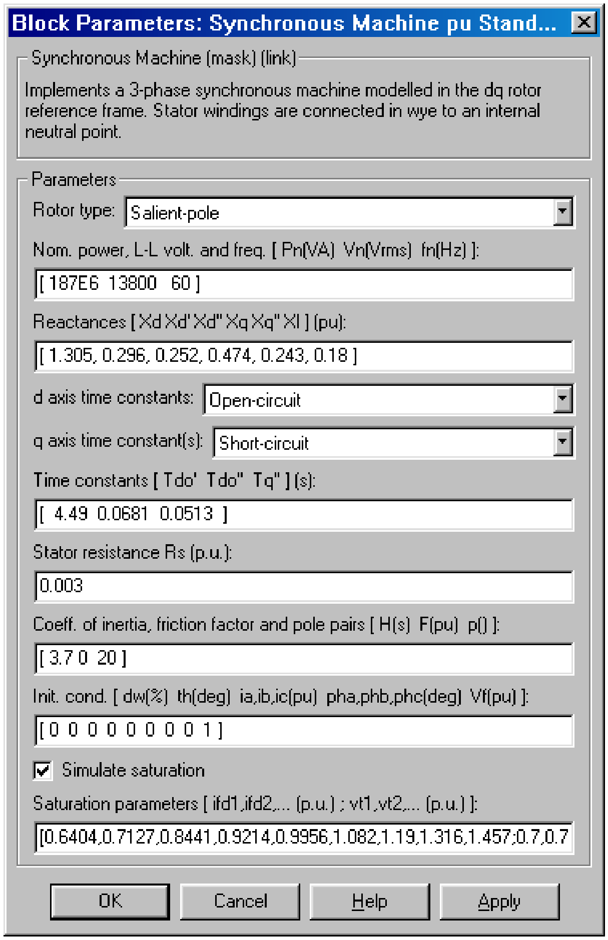
Coefficient of inertia, friction factor, and pole pairs

The inertia constant H (s), where H is the ratio of energy stored in the rotor at nominal speed over the nominal power of the machine, the damping coefficient D (p.u. torque/p.u. speed deviation), and the number of pole pairs p.

Initial conditions; Simulate saturation; Saturation parameters

The same initial conditions and saturation parameters as in the S.I. units dialog box, but all values are expressed in p.u. instead of SI units. For saturation, the nominal field current multiplied by the d-axis magnetizing inductance and nominal RMS line-to-line voltage are the base values for the field current and terminal voltage, respectively.

Standard Parameters in p.u.



Rotor type; Nominal power, L-L voltage, and frequency

The same parameters as the fundamental p.u. dialog box.

Reactances

The d-axis synchronous reactance Xd, transient reactance Xd', and subtransient reactance Xd'', the q-axis synchronous reactance Xq, transient reactance Xq' (only if round rotor), and subtransient reactance Xq'', and finally the leakage reactance Xl (all in p.u.).

d-axis time constants; q-axis time constant(s)

Specify the time constants you supply for each axis: either open-circuit or short-circuit.

Time constants

The d-axis and q-axis time constants (all in s). These values must be consistent with choices made on the two previous lines: d-axis transient open-circuit (Tdo') or short-circuit (Td') time constant, d-axis subtransient open-circuit (Tdo'') or short-circuit (Td'') time constant, q-axis transient open-circuit (Tqo') or short-circuit (Tq') time constant (only if round rotor), q-axis subtransient open-circuit (Tqo'') or short-circuit (Tq'') time constant.

Stator resistance

The stator resistance Rs (p.u.).

Coefficient of inertia, friction factor, and pole pairs; Initial conditions; Simulate saturation; Saturation parameters

The same parameters as the fundamental parameters in p.u. dialog box.

Note These three blocks simulate exactly the same synchronous machine model; the only difference is the way of entering the parameter units.

Inputs and Outputs

The units of inputs and outputs vary according to which dialog box was used to enter the block parameters. For the nonelectrical connections, there are two possibilities. If the first dialog box (fundamental parameters in SI units) is used, the inputs and outputs are in SI units (except for dw in the vector of internal variables, which is always in p.u., and angle θ, which is always in rad). If the second or third dialog boxes is used, the inputs and outputs are in p.u.

The first input is the mechanical power at the machine's shaft. In generating mode, this input can be a positive constant or function or the output of a prime mover block (see the Hydraulic Turbine and Governor or Steam Turbine and Governor blocks). In motoring mode, this input is usually a negative constant or function.

The second input of the block is the field voltage. This voltage can be supplied by a voltage regulator in generator mode (see the Excitation System block). It is usually a constant in motor mode.

If you use the model in SI fundamental units, the field voltage Vf should be entered in volts DC if nominal field current Ifn is specified or in volts referred to stator if Ifn is not specified. To obtain the Vfd producing nominal voltage, select the last check box of the dialog box. If you use the model in p.u. Standard or in p.u. Fundamental units, Vf should be entered in p.u. (1 p.u. of field voltage producing 1 p.u. of terminal voltage at no load).

The first three outputs are the electrical terminals of the stator. The last output of the block is a vector containing 21 signals. They are, in order:

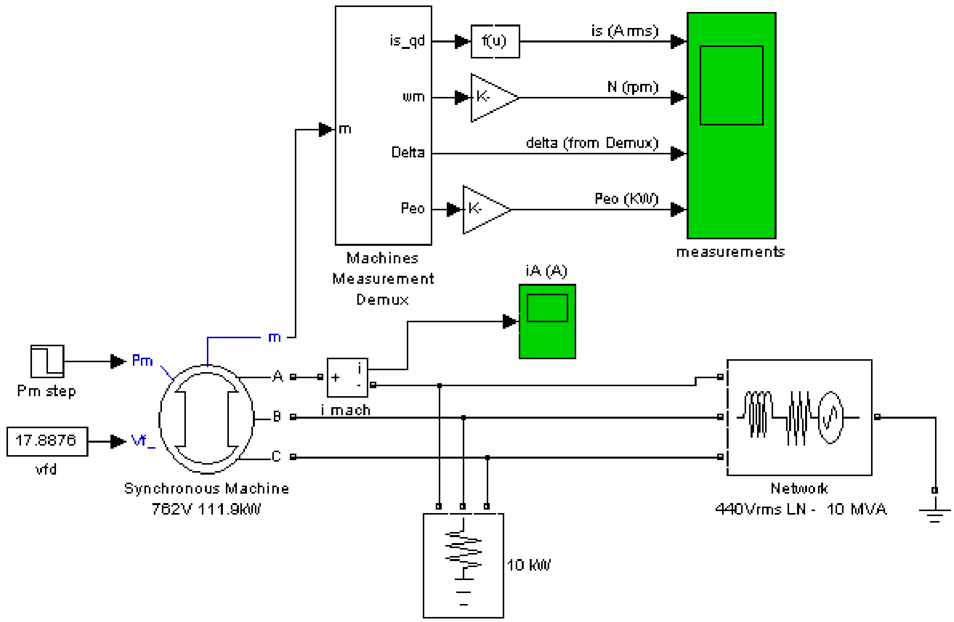
Signal	Definition
1 - 3	Stator currents (flowing out of machine) i_{sa}, i_{sb}, and i_{sc}
4 - 5	q- and d-axis stator currents (flowing out of machine) i_q, i_d
6 - 8	Field and damper winding currents (flowing into machine) i_{fd}, i_{kq}, and i_{kd}
9 - 10	q- and d-axis magnetizing fluxes φ_{mq}, φ_{md}
11 - 12	q- and d-axis stator voltages v_q, v_d
13	Rotor angle deviation $\Delta\theta$ with respect to a synchronous rotating frame
14	Rotor speed ω_r
15	Total electrical power Pe, including losses in stator, field, and damper windings
16	Rotor speed deviation $d\omega$
17	Rotor mechanical angle θ (degrees)
18	Electromagnetic torque Te

Signal	Definition (Continued)
19	Load angle δ (electrical degrees)
20	Output active power Peo
21	Output reactive power Qeo

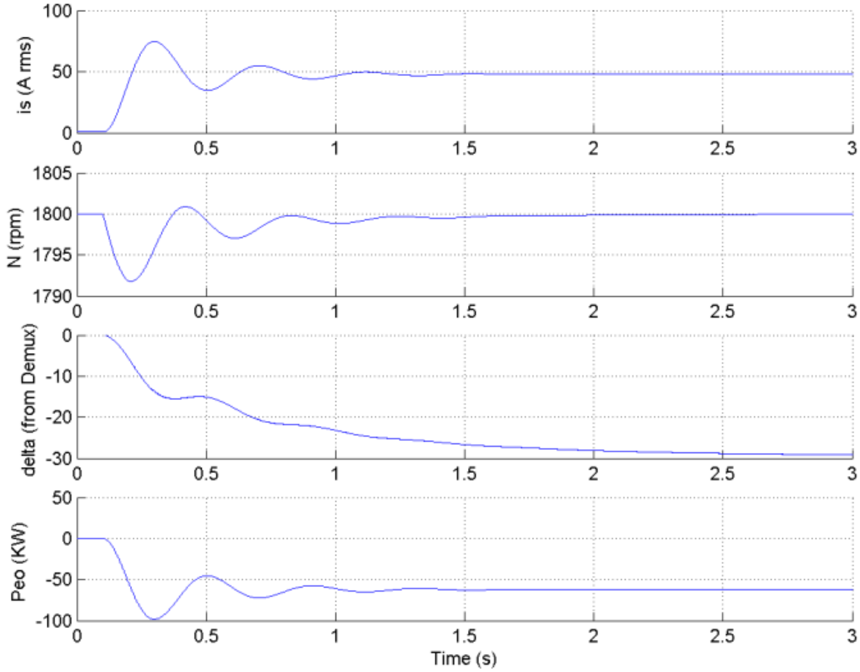
You can demultiplex these signals by using the Machine Measurement Demux block provided in the Machines library.

Example

The `power_syncmachine` demo illustrates the use of the Synchronous Machine block in motor mode. The simulated system consists of an industrial grade synchronous motor (150 HP (112 kVA), 762 V) connected to a network with a 10 MVA short-circuit level. In order to start simulation in steady state, the machine is initialized using the **Load Flow and Machine Initialization** option of the Powergui. The machine is initialized for an output electrical power of –50 kW (negative value for motor mode), corresponding to a mechanical power of –48.9 kW. The corresponding values of mechanical power and field voltage have been automatically entered by the **Load Flow** analysis into the Pm Step block and in the Vf Constant block. The Pm Step block has been programmed in order to apply a sudden increase of mechanical power from –48.9 kW to –60 kW at time t = 0.1 s.



Run the simulation and observe the RMS current, RMS voltage, speed, load angle δ and output electrical power of the motor.



Since this is a four-pole machine, the nominal speed is 1800 rpm. The initial speed is 1800 rpm as prescribed. After the load has increased from 48.9 kW to 100 kW at t = 0.1 s, the machine speed oscillates before stabilizing to 1800 rpm. The load angle (angle between terminal voltage and internal voltage) increases from –21 degrees to –53 degrees.

References

[1] Krause, P.C., *Analysis of Electric Machinery*, McGraw-Hill, 1986, Section 12.5.

[2] Kamwa, I., et al., “Experience with Computer-Aided Graphical Analysis of Sudden-Short-Circuit Oscillograms of Large Synchronous Machines,” *IEEE Transactions on Energy Conversion*, Vol. 10, No. 3, September 1995.

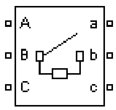
See Also

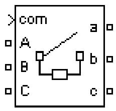
Excitation System, Hydraulic Turbine and Governor, Machine Measurement Demux, Powergui, Simplified Synchronous Machine, Steam Turbine and Governor

Purpose Implement a three-phase circuit breaker opening at the current zero crossing

Library Elements

Description





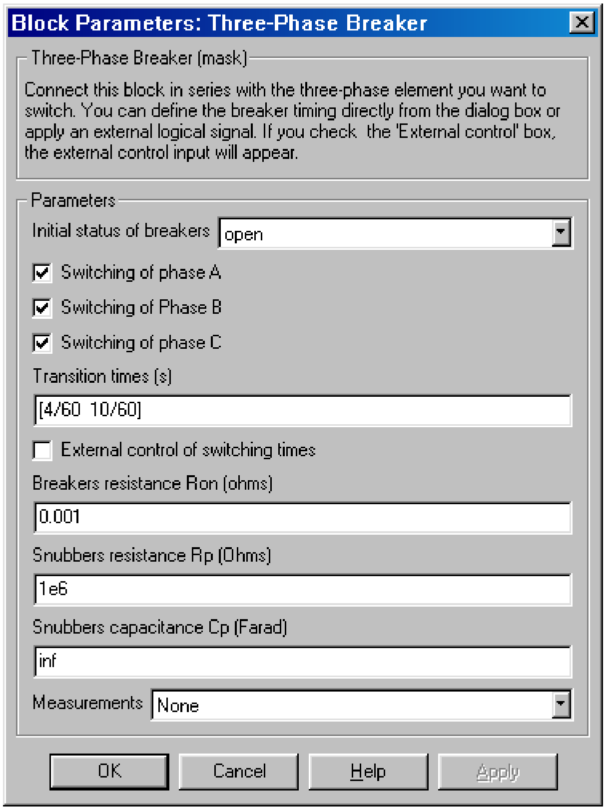
The Three-Phase Breaker block implements a three-phase circuit breaker where the opening and closing times can be controlled either from an external Simulink signal (external control mode), or from an internal control timer (internal control mode).

The Three-Phase Breaker block uses three Breaker blocks connected between the inputs and the outputs of the block. You can use this block in series with the three-phase element you want to switch. See the Breaker block reference pages for details on the modeling of the single-phase breakers.

If the Three-Phase Breaker block is set in external control mode, a control input appears in the block icon. The control signal connected to this input must be either 0 or 1, 0 to open the breakers, 1 to close them. If the Three-Phase Breaker block is set in internal control mode, the switching times are specified in the dialog box of the block. The three individual breakers are controlled with the same signal.

Series Rs-Cs snubber circuit are included in the model. They can be optionally connected to the three individual breakers. If the Three-Phase Breaker block happens to be in series with an inductive circuit, an open circuit or a current source, you must use the snubbers.

Dialog Box and Parameters



Initial status of breakers

The initial status of the breakers. The initial status is the same for the three breakers. Depending on the initial status, the icon shows a closed contact or an open contact.

Switching of phase A

If selected, the switching of phase A is activated. If not selected, the breaker of phase A stays in its initial status specified in the **Initial status of breakers** parameter.

Switching of Phase B

If selected, the switching of phase B is activated. If not selected, the breaker of phase B stays in its initial status specified in the **Initial status of breakers** parameter.

Switching of phase C

If selected, the switching of phase C is activated. If not selected, the breaker of phase C stays in its initial status specified in the **Initial status of breakers** parameter.

Transition times(s)

The **Transition times(s)** parameter is not visible in the dialog box if the **External control of switching times** parameter is selected.

Specify the vector of switching times when using the Three-Phase Breaker block in internal control mode. At each transition time the selected breakers opens or closes depending to their initial state.

External control of switching times

If selected, adds a fourth input port to the Three-Phase Breaker block for an external control of the switching times of the breakers. The switching times are defined by a Simulink signal (`0-1 sequence`).

Breakers resistance Ron

The internal breaker resistances, in ohms (Ω). The **Breaker resistance Ron** parameter cannot be set to `0`.

Snubbers resistance Rp

The snubber resistances, in ohms (Ω). Set the **Snubber resistance Rp** parameter to `inf` to eliminate the snubbers from the model.

Snubbers capacitance Cp

The snubber capacitances, in farads (F). Set the **Snubber capacitance Cp** parameter to `0` to eliminate the snubbers, or to `inf` to get resistive snubbers.

Measurements

Select `Breaker voltages` to measure the voltage across the three internal breaker terminals.

Select `Breaker currents` to measure the current flowing through the three internal breakers. If the snubber devices are connected, the measured currents are the ones flowing through the breakers contacts only.

Select `Breaker voltages and currents` to measure the breaker voltages and the breaker currents.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements is identified by a label followed by the block name and the phase:

Measurement	Label
Breaker voltages	`Ub <block name> /Breaker A: Ub <block name> /Breaker B:` `Ub <block name> /Breaker C.`
Breaker currents	`Ib <block name> /Breaker A: Ib <block name> /Breaker B:` `Ib <block name> Breaker C.`

Inputs and Outputs

The three inputs `A`, `B`, and `C` and the three outputs `a`, `b`, and `c` are the breaker terminals. Breaker A is connected between input 1 and output 1, breaker B is connected between input 2 and output 2, and breaker C is connected between input 3 and output 3. If the Three-Phase Breaker block is set in external control mode, the Simulink input 4 appears and it is used to control the opening and closing of the three internal breakers.

Example

See the `power_3phlinereclose` and `power_3phseriescomp` demos for circuits using the Three-Phase Breaker block.

See Also

Breaker, Multimeter, Three-Phase Fault

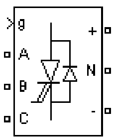
Purpose

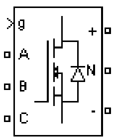
Implement a three-level neutral point clamped (NPC) power converter with selectable topologies and power switching devices

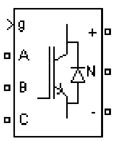
Library

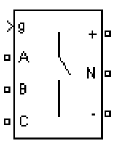
Power Electronics

Description

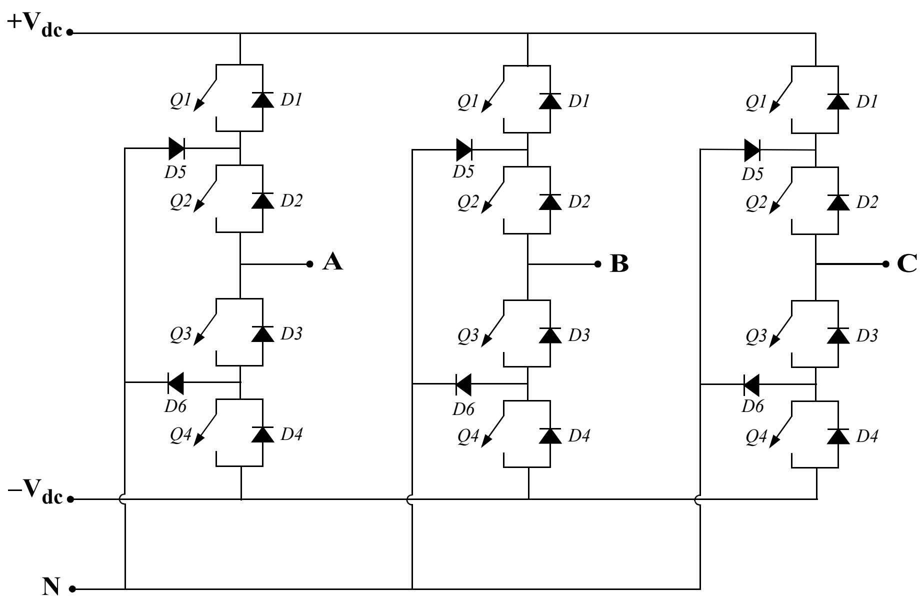






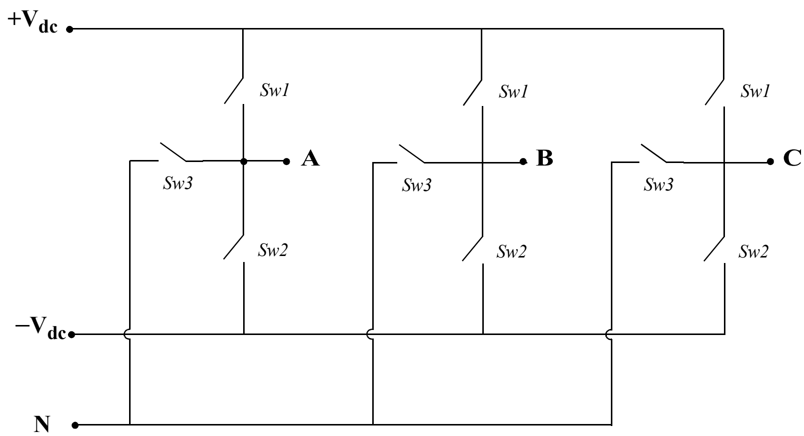


The Three-Level Bridge block implements a three-level power converter that consists of one, two, or three arms of power switching devices. Each arm consists of four switching devices (Q1 to Q4) along with their antiparallel diodes (D1 to D4) and two neutral clamping diodes (D5 and D6) as shown.

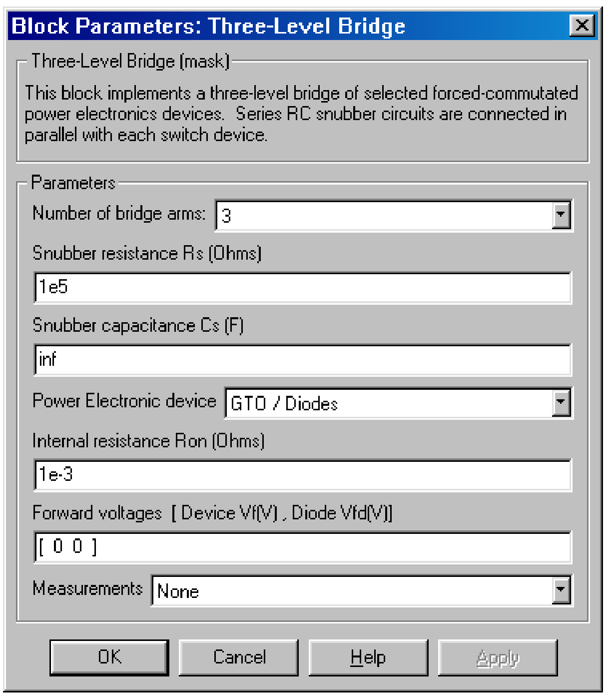


The type of power switching device (IGBT, GTO, MOSFET, or ideal switch) and the number of arms (one, two, or three) are selectable from the dialog box. When the ideal switch is used as the switching device, the Three-Level Bridge block implements an ideal switch bridge having a three-level topology as shown.

Three-Level Bridge



Dialog Box and Parameters



Number of bridge arms

Determine the bridge topology: one, two, or three arms.

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubbers from the model.

Snubber capacitance Cs

The snubber capacitance, in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubbers, or to `inf` to get a resistive snubber.

For forced-commutated devices (GTO, IGBT, or MOSFET) the Three-Level Bridge block operates satisfactorily with resistive snubbers as long as the firing pulses are sent to the switching devices.

If the firing pulses to forced-commutated devices are blocked, the bridge operates as a diode rectifier. In this condition, you must use appropriate values of Rs and Cs. If the model is discretized, you can use the following formulas to compute approximate values of Rs and Cs:

$$Rs > 2\frac{Ts}{Cs}$$

$$Cs < \frac{Pn}{1000(2\pi f)Vn^2}$$

where

P_n = Nominal power of single- or three-phase converter (VA)

Vn = Nominal line-to-line AC voltage (Vrms)

f = Fundamental frequency (Hz)

T_s = Sample Time (s)

These Rs and Cs values are derived from the following two criteria:

- The snubber leakage current at fundamental frequency is less than 0.1% of nominal current when power electronic devices are not conducting.

- The RC time constant of snubbers is higher than two times the sample time Ts.

Note that the Rs and Cs values that guarantee numerical stability of the discretized bridge can be different from actual values used in the physical circuit.

Power electronic device

Select the type of power electronic device to use in the bridge.

Internal resistance Ron

Internal resistance of the selected devices and diodes, in ohms (Ω).

Forward voltages [Device Vf, Diode Vfd]

The forward voltage of the selected devices (for GTO or IGBT only) and of the antiparallel and clamping diodes, in volts.

Measurements

Select `All Device currents` to measure the current flowing through all the components (Q1 to Q4 and D1 to D6). If the snubber devices are defined, the measured currents are those flowing through the power electronic devices only.

Select `Phase-to-neutral and DC voltages` to measure the terminal voltages (AC and DC) of the Three-Level Bridge block.

Select `All voltages and currents` to measure all voltages and currents defined for the Three-Level Bridge block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurement** list box of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Device currents for MOSFET, IGBT, and GTO Device currents for Ideal Switch, where x = a, b, or c	`IQ1x:, IQ2x:, IQ3x:, IQ4x:, ID1x:, ID2x:, ID3x:, ID4x:, ID5x:, ID6x:` or ISw1x:, ISw2x:, ISw3x:
Terminal voltages	Uan:, Ubn:, Ucn:, Udc+:, Udc-:

Inputs and Outputs

The Pulses input accepts a Simulink-compatible vectorized gating signal containing four pulses (Q1 to Q4) for each arm of the converter. For instance, if a three-arm topology is selected, the input vector must contain twelve pulses and the ordering must be as follows: Q1 of leg A, Q2 of leg A, …, Q4 of leg C.

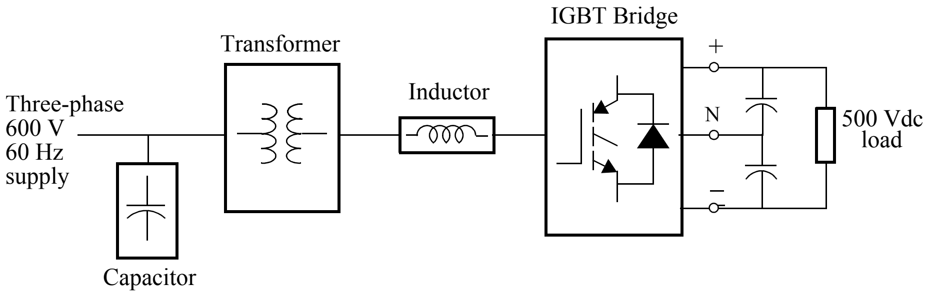
Note In the case of the ideal switch converter, the Q1 pulse is sent to Sw1, the Q4 pulse to Sw2, and a logical AND operation is performed on the Q2 and Q3 pulses and the result sent to Sw3.

Assumptions and Limitations

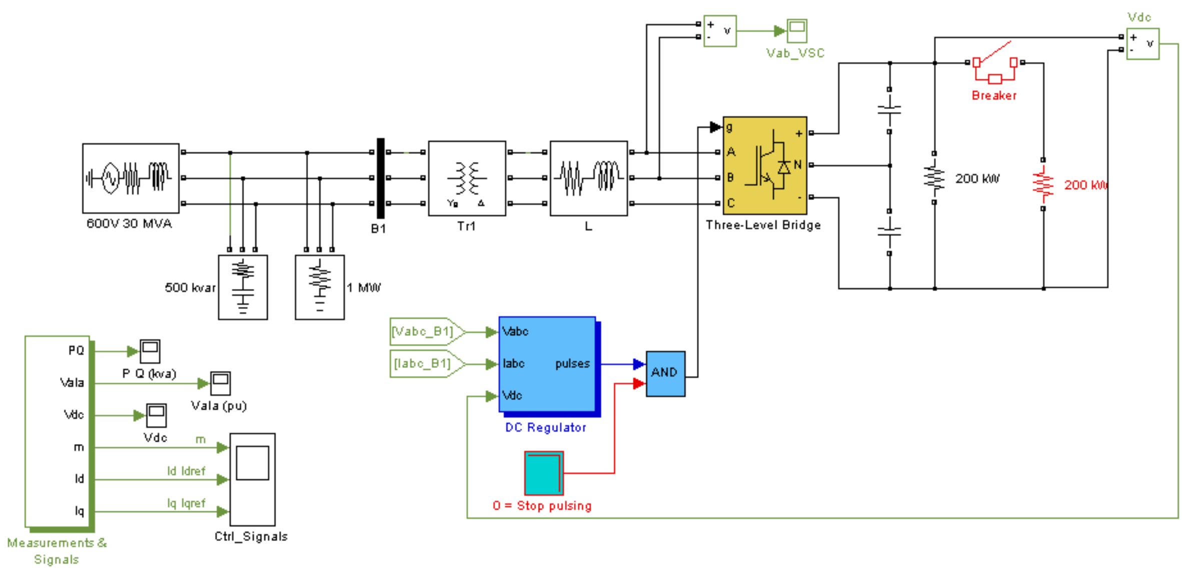
Turn-on and turn-off times (Fall time, Tail time) of power switching devices are not modeled in the Three-Level Bridge block.

Example

The `power_3levelVSC` demo illustrates the use of the Three-Level Bridge block in an AC-DC converter consisting of a three-phase IGBT-based voltage sourced converter (VSC). The converter is pulse-width modulated (PWM) to produce a 500 V DC voltage (+/– 250V). In this example the converter chopping frequency is 1620 Hz and the power system frequency is 60 Hz.



The VSC is controlled in a closed loop by two PI regulators in order to maintain a DC voltage of 500 V at the load while maintaining a unity input power factor for the AC supply.



The initial conditions for a steady-state simulation are generated by running an initial simulation to steady state for an integer number of cycles of 60 Hz. The final states (both SimPowerSystems and Simulink controller states) are saved in a vector called `xInitial`. This vector, as well as the sample times (`Ts_Power` and `Ts_Control`) are saved in the `power_3levelVSC_xinit.mat` file.

When you open this model, the initial condition vector `xInitial` and the sample times saved in the MAT file are automatically loaded in the workspace. Start the simulation. The monitored signals start in steady state.

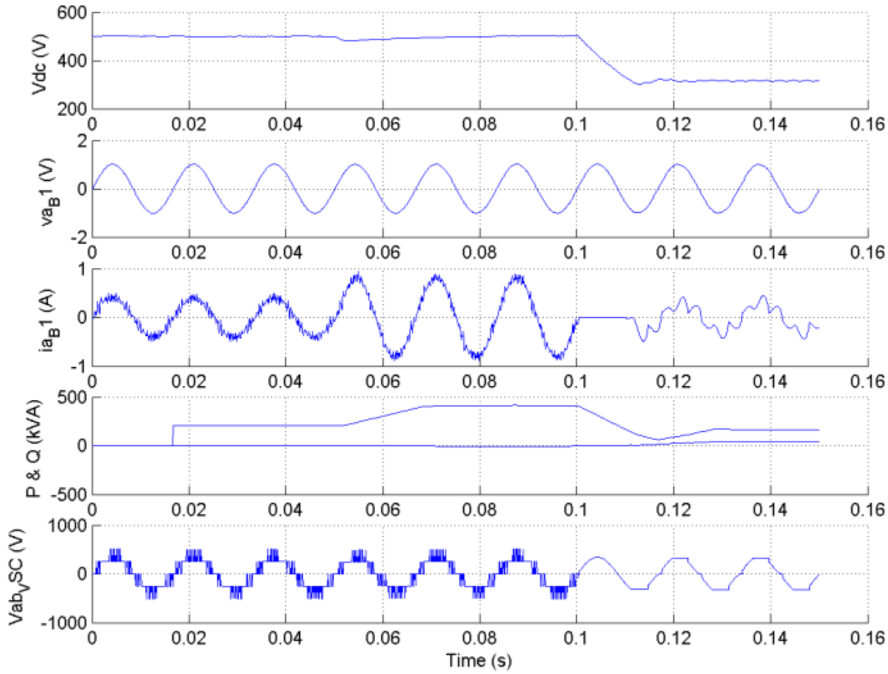
Observe the following signals:

- The DC voltage (Vdc Scope block)
- The primary voltage and current of phase A of the AC supply (VaIa Scope block)
- The device currents of leg A of the IGBT bridge (Ia_Devices Scope block inside the Measurements & Signals subsystem)
- The line-to-line terminal voltage of the VSC (Vab_VSC Scope block)

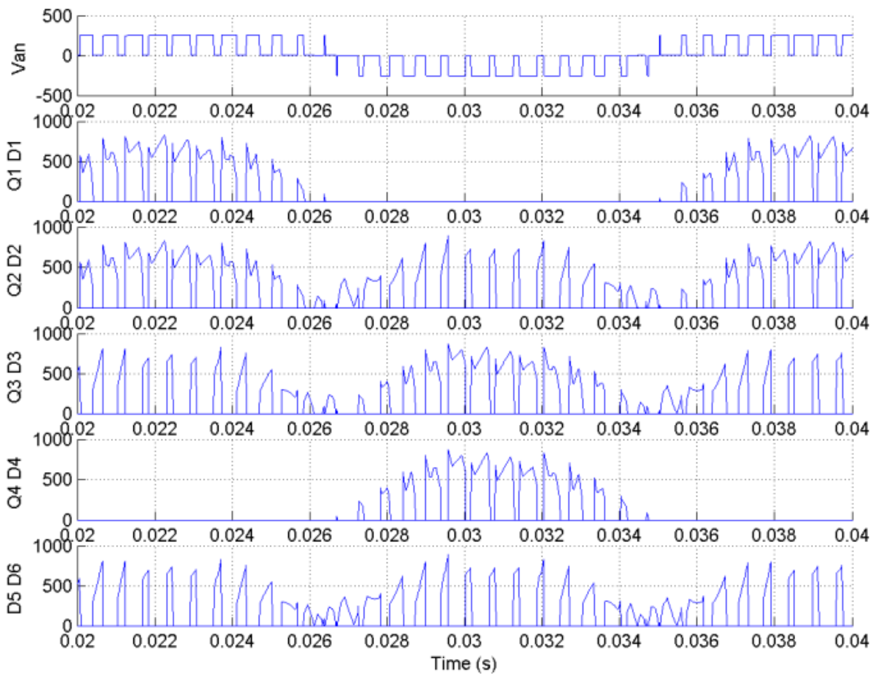
At 50 ms, a 200 kW load is switched in. You can see that the dynamic response of the DC regulator to the sudden load variation from 200 kW to 400 kW is satisfactory. The DC voltage reverts to 500 V within 2 cycles and the unity power factor on the AC side is maintained.

At 100 ms, a *stop-pulsing* signal is activated and the pulses normally sent to the converter are blocked. You can see that the DC voltage drops to 315 V. A drastic change in the primary current waveform can also be observed. When the pulses are blocked, the Three-Level Bridge block operation becomes similar to a three-phase diode bridge.

The following two figures summarize the results of the simulation. The first figure shows the operation of the AC-DC converter during the load variation and when the pulses are blocked.



The second figure shows the current flowing in the various devices of the IGBT bridge when the converter is feeding 500 Vdc to a 200-kW load.



See Also

Multimeter

Three-Phase Dynamic Load

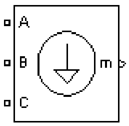
Purpose

Implements a three-phase dynamic load with active power and reactive power as a function of voltage or controlled from an external input

Library

Elements

Description



The Three-Phase Dynamic Load block implements a three-phase, three-wire dynamic load whose active power P and reactive power Q vary as function of positive-sequence voltage. Negative- and zero-sequence currents are not simulated. The three load currents are therefore balanced, even under unbalanced load voltage conditions.

The load impedance is kept constant if the terminal voltage V of the load is lower than a specified value Vmin. When the terminal voltage is greater than the Vmin value, the active power P and reactive power Q of the load vary as follows:

$$P(s) = P_o\left(\frac{V}{V_o}\right)^{n_p}\frac{(1+T_{p1}s)}{(1+T_{p2}s)}$$

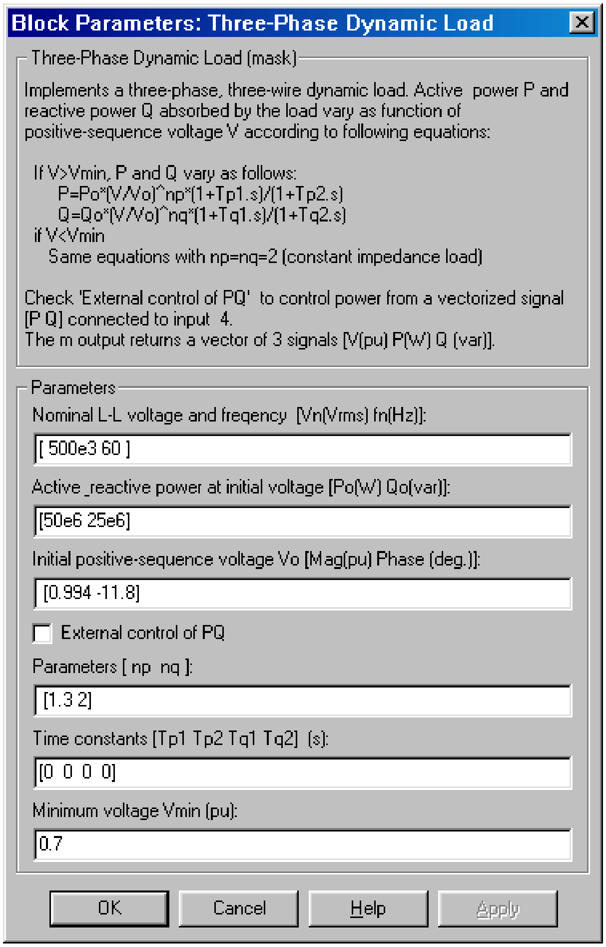
$$Q(s) = Q_o\left(\frac{V}{V_o}\right)^{n_q}\frac{(1+T_{q1}s)}{(1+T_{q2}s)}$$

where

- V_o is the initial positive sequence voltage.
- P_o and Q_o are the initial active and reactive powers at the initial voltage V_o.
- V is the positive-sequence voltage.
- n_p and n_q are exponents (usually between 1 and 3) controlling the nature of the load.
- T_{p1} and T_{p2} are time constants controlling the dynamics of the active power P.
- T_{q1} and T_{q2} are time constants controlling the dynamics of the reactive power Q.

For a constant current load, for example, you set n_p to 1 and n_q to 1, and for constant impedance load you set n_p to 2 and n_q to 2.

Dialog Box and Parameters



Nominal L-L voltage and frequency

Specifies the nominal phase-to-phase voltage, in volts RMS, and nominal frequency, in hertz, of the load.

Active and reactive power at initial voltage

Specifies the initial active power Po, in watts, and initial reactive power Qo, in vars, at the initial voltage Vo. If the load flow utility of the Powergui is used to initialize the dynamic load and start simulation in steady state, these parameters are automatically updated according to P and Q set points specified for the load.

Initial positive-sequence voltage Vo

Specifies the magnitude and phase of the initial positive-sequence voltage of the load. If the load flow utility of the Powergui is used to initialize the dynamic load and start simulation in steady state, these two parameters are automatically updated according to values computed by the load flow.

External control of PQ

If selected, the active power and reactive power of the load are defined by an external Simulink vector of two signals.

Parameters [np nq]

Specifies the **np** and **nq** parameters that define the nature of the load.

Time constants [Tp1 Tp2 Tq1 Tq2]

Specifies the time constants controlling the dynamics of the active power and the reactive power.

Minimum voltage Vmin

Specifies the minimum voltage at which the load dynamics commences. The load impedance is constant below this value.

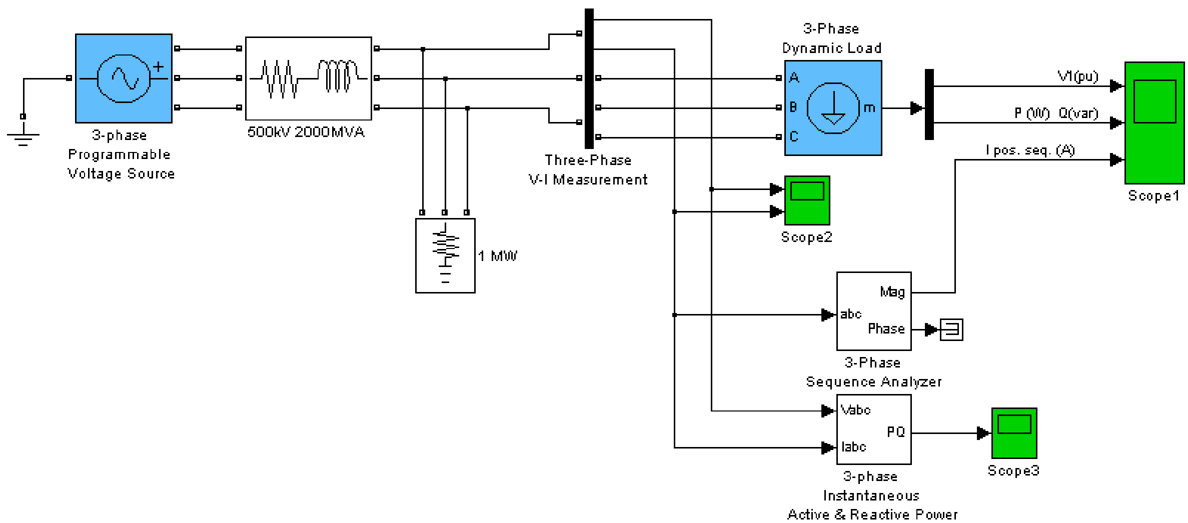
Inputs and Outputs

Inputs `A`, `B`, and `C` are the three terminals of the load. If **External control of PQ** is selected, a fourth input, labeled `PQ`, appears. This Simulink input is used to control the active and reactive powers of the load from a vector of two signals [P Q].

The m output is a vector containing the following three signals: positive-sequence voltage (p.u.); active power P (W); and reactive power Q (vars).

Example

The `power_dynamicload` model uses a Three-Phase Dynamic Load block connected on a 500 kV, 60 Hz power network. The network is simulated by its Thevenin equivalent (voltage source behind a R-L impedance corresponding to a three-phase short-circuit level of 2000 MVA). The source internal voltage is modulated in order to simulate voltage variation during a power swing. As the dynamic load is a nonlinear model simulated by current sources, it cannot be connected to an inductive network (R-L in series). Therefore, a small resistive load (1 MW) has been added in parallel with the dynamic load.



In order to start the simulation in steady state, you must specify the correct initial positive-sequence voltage Vo (magnitude and phase) corresponding to the desired Po and Qo values. You use the load flow utility to find this voltage and initialize the dynamic load. Open the Powergui and select **Load Flow and Machine Initialization**. Specify the desired active power and reactive powers for the dynamic load (50 MW, 25 Mvar):

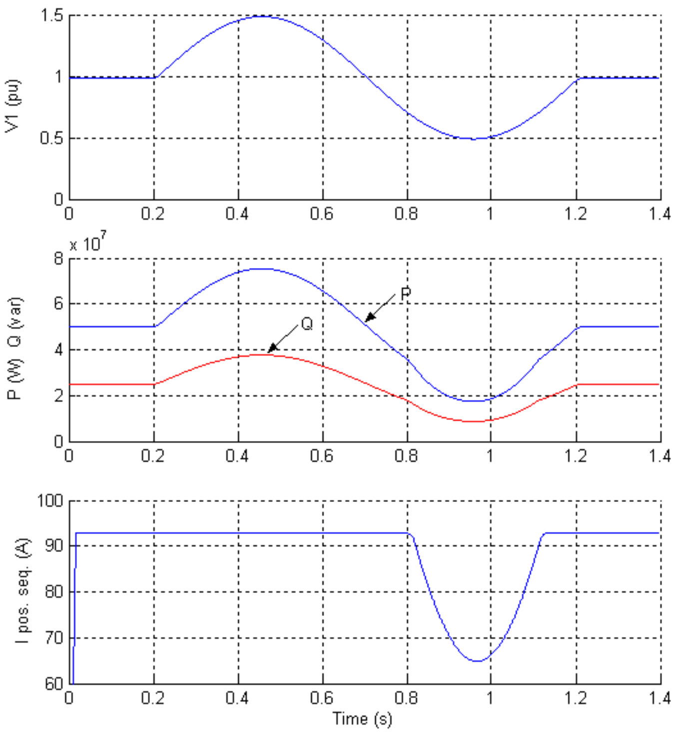
Active Power = 50e6; Reactive Power = 25e6.

Then press the **Update Load Flow** button.Once the load flow has been solved the three phase-to-phase voltages of the dynamic load (0.9844 p.u.) as well as its line currents are displayed. The phase angle of the phase-to-neutral load voltage Uan is also displayed (–1.41 degrees). This angle corresponds to the angle of the positive-sequence voltage. If you now open the Three-Phase Dynamic Load dialog box, notice that the values of Po, Qo, and Vo have been updated.

Start the simulation and observe load voltage, P&Q powers, and current on Scope1. Observe that simulation starts in steady state. At t = 0.2 s, when voltage modulation is initiated, P and Q start to increase (trace 2), but, as np and nq are set to 1, the load current (trace 3) stays constant. When voltage falls below 0.7 p.u. the load behaves as a constant impedance. Therefore load current follows this voltage variation.

Observe on Scope2 variations of instantaneous voltages and currents. Also, notice that computed P and Q displayed on Scope3 are the same as P and Q internal signals returned by the Dynamic Load measurement output.

The signals displayed on the Scope1 block are shown below.



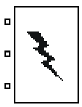
Purpose

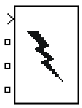
Implement a programmable phase-to-phase and phase-to-ground fault breaker system

Library

Elements

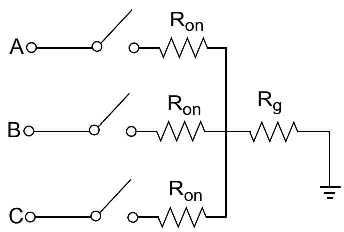
Description





The Three-Phase Fault block implements a three-phase circuit breaker where the opening and closing times can be controlled either from an external Simulink signal (external control mode), or from an internal control timer (internal control mode).

The Three-Phase Fault block uses three Breaker blocks that can be individually switched on and off to program phase-to-phase faults, phase-to-ground faults, or a combination of phase-to-phase and ground faults.

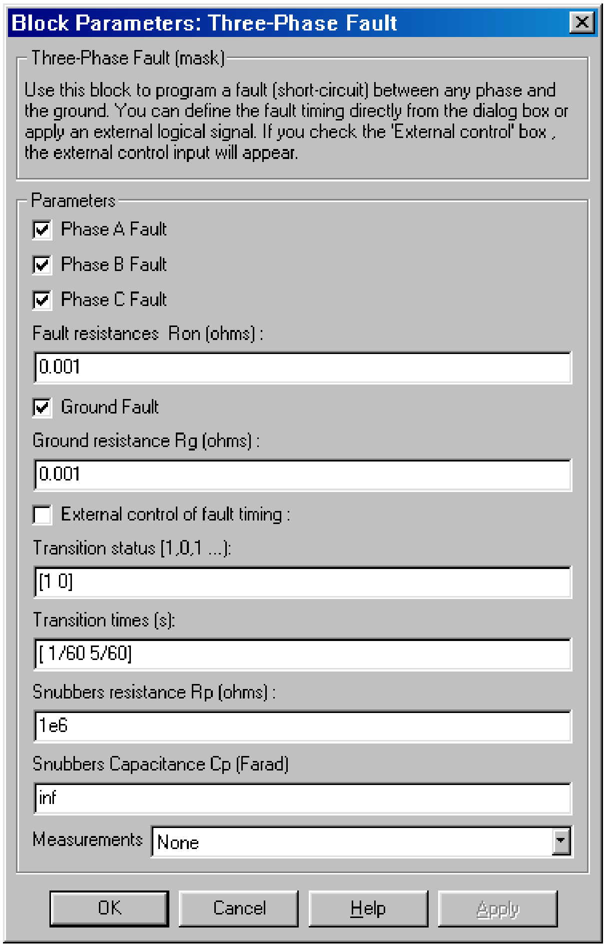


The ground resistance Rg is automatically set to 10^6 ohms when the ground fault option is not programmed. For example, to program a fault between the phases A and B you need to select the **Phase A Fault** and **Phase B Fault** block parameters only. To program a fault between the phase A and the ground, you need to select the **Phase A Fault** and **Ground Fault** parameters and specify a small value for the ground resistance.

If the Three-Phase Fault block is set in external control mode, a control input appears in the block icon. The control signal connected to the fourth input must be either 0 or 1, 0 to open the breakers, 1 to close them. If the Three-Phase Fault block is set in internal control mode, the switching times and status are specified in the dialog box of the block.

Series Rp-Cp snubber circuits are included in the model. They can be optionally connected to the fault breakers. If the Three-Phase Fault block is in series with an inductive circuit, an open circuit or a current source, you must use the snubbers.

Dialog Box and Parameters



Phase A Fault

If selected, the fault switching of phase A is activated. If not selected, the breaker of phase A stays in its initial status. The initial status of the phase A breaker corresponds to the complement of the first value specified in the vector of **Transition status**. The initial status of the fault breaker is usually 0 (open). However, it is possible to start a simulation in steady state with the fault initially applied on the system. For example, if the first value in the **Transition status** vector is 0, the phase A breaker is initially closed. It opens at the first time specified in the **Transition time(s)** vector.

Phase B Fault

If selected, the fault switching of phase B is activated. If not selected, the breaker of phase B stays in its initial status. The initial status of the phase B breaker corresponds to the complement of the first value specified in the vector of **Transition status**.

Phase C Fault

If selected, the fault switching of phase C is activated. If not selected, the breaker of phase C stays in its initial status. The initial status of the phase C breaker corresponds to the complement of the first value specified in the vector of **Transition status**.

Fault resistances Ron

The internal resistance, in ohms (Ω), of the phase fault breakers. The **Fault resistances Ron** parameter cannot be set to 0.

Ground Fault

If selected, the fault switching to the ground is activated. A fault to the ground can be programed for the activated phases. For example, if the **Phase C Fault** and **Ground Fault** parameters are selected, a fault to the ground is applied to the phase C. The ground resistance is set internally to 1e6 ohms when the **Ground Fault** parameter is not selected.

Ground resistance Rg

The **Ground resistance Rg (ohms)** parameter is not visible if the **Ground Fault** parameter is not selected. The ground resistance, in ohms (Ω). The **Ground resistance Rg (ohms)** parameter cannot be set to 0.

External control of fault timing

If selected, adds a fourth input port to the Three-Phase Fault block for an external control of the switching times of the fault breakers. The switching times are defined by a Simulink signal (`0` or `1`) connected to the fourth input port of the block.

Transition status

Specify the vector of switching status when using the Three-Phase Breaker block in internal control mode. The selected fault breakers open (0) or close

(1) at each transition time according to the **Transition status** parameter values.

The initial status of the breakers corresponds to the complement of the first value specified in the vector of switching status.

Transition times(s)

Specify the vector of switching times when using the Three-Phase Breaker block in internal control mode. At each transition time the selected fault breakers opens or closes depending to the initial state. The **Transition times (s)** parameter is not visible in the dialog box if the **External control of switching times** parameter is selected.

Snubbers resistance Rp

The snubber resistances, in ohms (Ω). Set the **Snubbers resistance Rp** parameter to `inf` to eliminate the snubbers from the model.

Snubbers capacitance Cp

The snubber capacitances, in farads (F). Set the **Snubbers capacitance Cp** parameter to `0` to eliminate the snubbers, or to `inf` to get resistive snubbers.

Measurements

Select `Fault voltages` to measure the voltage across the three internal fault breaker terminals.

Select `Fault currents` to measure the current flowing through the three internal breakers. If the snubber devices are connected, the measured currents are the ones flowing through the breakers contacts only.

Select `Fault voltages and currents` to measure the breaker voltages and the breaker currents.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name and the phase:

Measurement	Label
Fault voltages	`Ub <block name> /Fault A: Ub <block name> /Fault B: Ub <block name> /Fault C.`
Fault currents	`Ib <block name> /Fault A: Ib <block name> /Fault B: Ib <block name> /Fault C.`

Inputs and Outputs

The inputs 1, 2, and 3 are the fault breaker terminals connected respectively to phases `A`, `B`, and `C`. The breakers are connected between inputs 1, 2, and 3 and the internal ground resistor. If the Three-Phase Fault block is set in external control mode, Simulink input 4 appears and is used to control the opening and closing of the three internal breakers.

Example

See the `power_3phseriescomp` demo for a circuit using the Three-Phase Fault block.

See Also

Breaker, Multimeter, Three-Phase Breaker

Three-Phase Mutual Inductance Z1-Z0

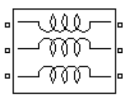
Purpose

Implement a three-phase impedance with mutual coupling among phases

Library

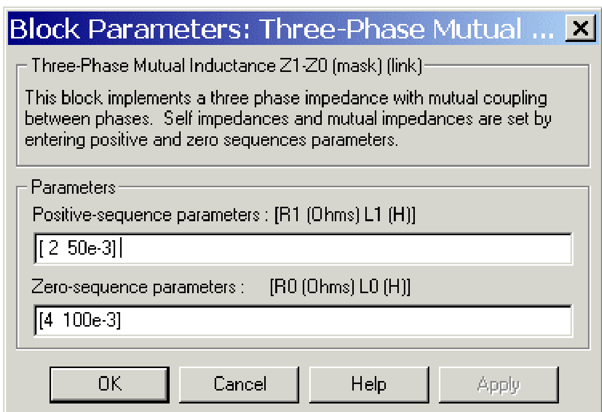
Elements

Description



The Three-Phase Mutual Inductance Z1-Z0 block implements a three-phase balanced inductive and resistive impedance with mutual coupling between phases. This block performs the same function as the three-winding Mutual Inductance block. For three-phase balanced power systems, it provides a more convenient way of entering system parameters in terms of positive- and zero-sequence resistances and inductances than the self- and mutual resistances and inductances.

Dialog Box and Parameters



Positive-sequence parameters

The positive-sequence resistance R1, in ohms (Ω), and the positive-sequence inductance L1, in henries (H).

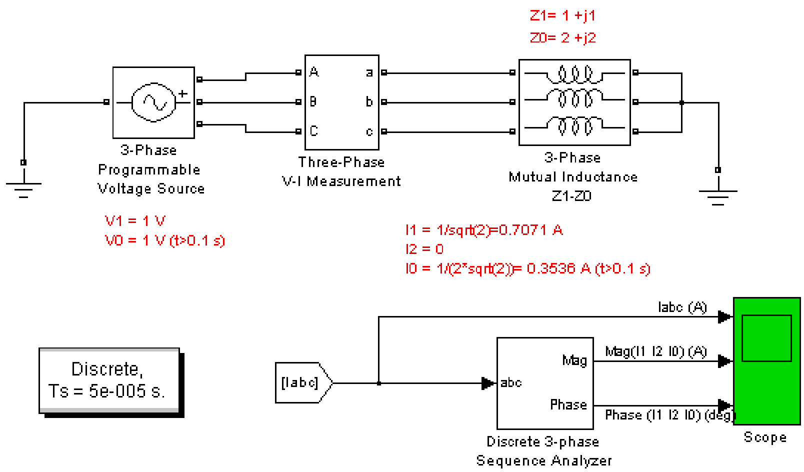
Zero-sequence parameters

The zero-sequence resistance R0, in ohms (Ω), and the zero-sequence inductance L0, in henries (H).

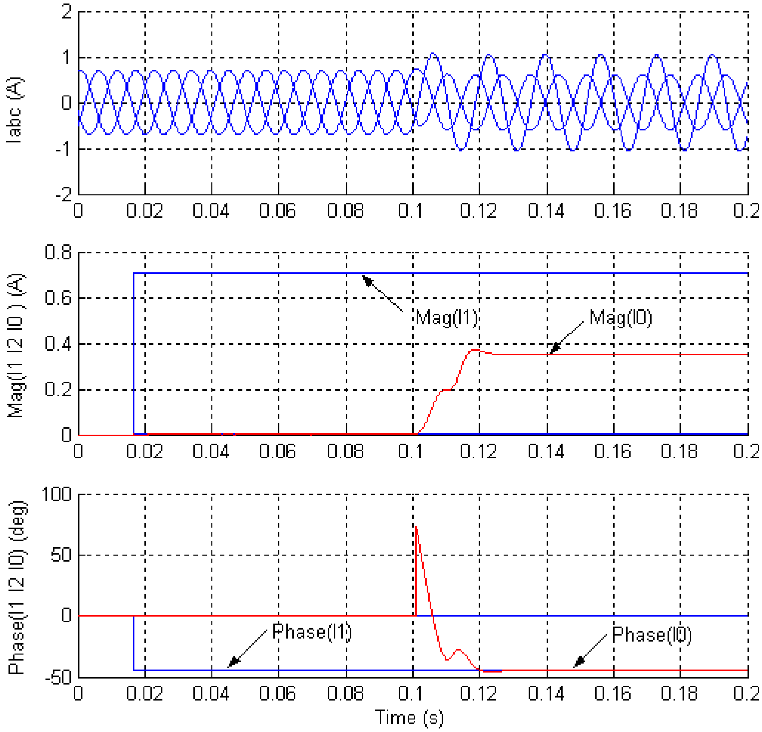
Example

The `power_3phmutseq10` demo illustrates the use of the Three-Phase Mutual Inductance Z1-Z0 block to build a three-phase inductive source with different values for the positive-sequence impedance Z1 and the zero-sequence impedance Z0. The programmed impedance values are Z1 = 1+j1 Ω and Z0 =

2+j2 Ω. The Three-Phase Programmable Voltage Source block is used to generate a 1-volt, 0-degree, positive-sequence internal voltage. At t = 0.1 s, a 1-volt, 0-degree, zero-sequence voltage is added to the positive-sequence voltage. The three source terminals are short-circuited to ground and the resulting positive-, negative-, and zero-sequence currents are measured using the Discrete 3-Phase Sequence Analyzer block.



The current waveforms and their sequence components (magnitude and phase) are displayed on the Scope block. The resulting waveforms are shown on the following figure.



The polar impedance values are $Z1 = \sqrt{2}\angle 45°\ \Omega$ and $Z0 = 2\sqrt{2}\angle 45°\ \Omega$

Therefore, the positive- and zero-sequence currents displayed on the scope are

$$I1 = V1/Z1 = 1/(\sqrt{2}\angle 45°) = 0.7071\ \text{A}\angle{-45°}$$

$$I0 = V0/Z0 = 1/(2\sqrt{2}\angle 45°) = 0.3536\ \text{A}\angle{-45°}$$

The transients observed on the magnitude and the phase angle of the zero-sequence current when the zero-sequence voltage is added (at t = 0.1 s) are due to the Fourier measurement technique used by the Discrete 3-Phase Sequence Analyzer block. As the Fourier analysis uses a running average window of one cycle, it takes one cycle for the magnitude and phase to stabilize.

See Also

Mutual Inductance

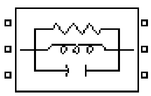
Purpose

Implement a three-phase parallel RLC branch

Library

Elements

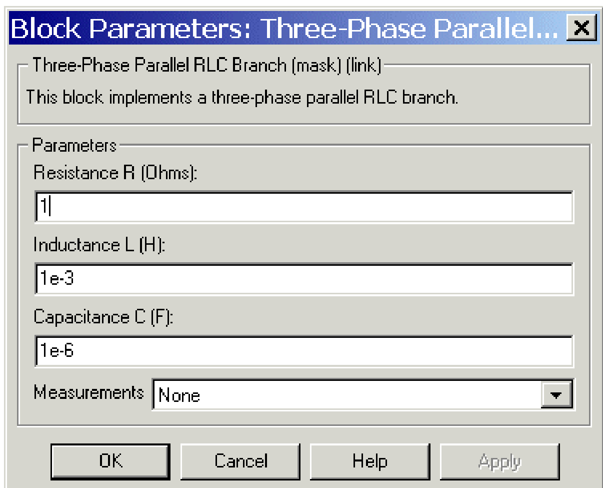
Description



The Three-Phase Parallel RLC Branch block implements three balanced branches consisting each of a resistor, an inductor, a capacitor, or a parallel combination of these. To eliminate either the resistance, inductance, or capacitance of each branch, the R, L, and C values must be set respectively to infinity (`inf`), infinity (`inf`), and `0`. Only existing elements are displayed in the block icon.

Negative values are allowed for resistance, inductance, and capacitance.

Dialog Box and Parameters



Resistance R

The branch resistances, in ohms (Ω).

Inductance L

The branch inductances, in henries (H).

Capacitance C

The branch capacitances, in farads (F).

Measurements

Select `Branch voltages` to measure the three voltages across the Three-Phase Parallel RLC Branch block terminals.

Select `Branch currents` to measure the three total currents (sum of R, L, C currents) flowing through the Three-Phase Parallel RLC Branch block.

Select `Branch voltages and currents` to measure the three voltages and the three currents of the Three-Phase Parallel RLC Branch block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement	Label
Branch voltages of phases A, B, and C	`Ub1: , Ub2: , Ub3:`
Branch currents of phases A, B, and C	`Ib1: , Ib2: , Ib3:`

See Also

Multimeter, Three-Phase Parallel RLC Load, Three-Phase Series RLC Branch, Three-Phase Series RLC Load

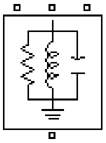
Purpose

Implement a three-phase parallel RLC load with selectable connection

Library

Elements

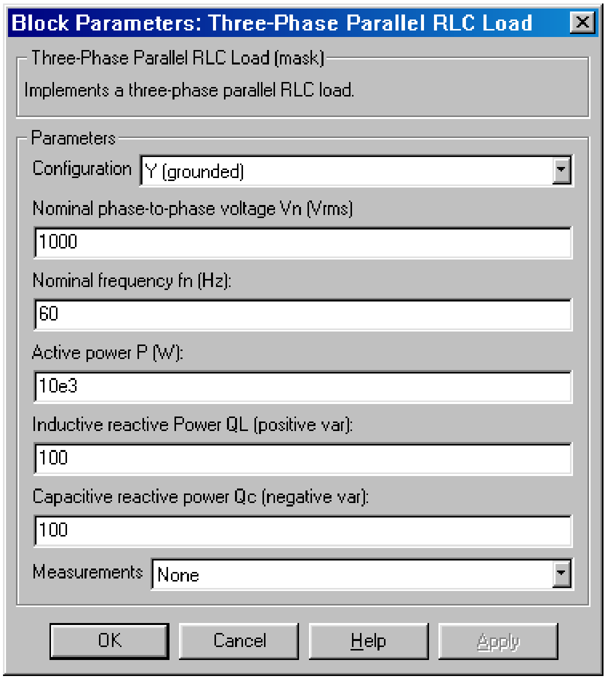
Description



The Three-Phase Parallel RLC Load block implements a three-phase balanced load as a parallel combination of RLC elements. At the specified frequency, the load exhibits a constant impedance. The active and reactive powers absorbed by the load are proportional to the square of the applied voltage.

Only elements associated with nonzero powers are displayed in the block icon.

Dialog Box and Parameters



Configuration

The connection of the three phases. Select one of the following four connections:

`Y(grounded)`	Neutral is grounded.
`Y(floating)`	Neutral is not accessible.
`Y(neutral)`	Neutral is made accessible through a fourth connector.
`Delta`	Three phases connected in delta

The block icon is updated according to the load connection.

Nominal phase-to-phase voltage Vn

The nominal phase-to-phase voltage of the load, in volts RMS (Vrms).

Nominal frequency fn

The nominal frequency, in hertz (Hz).

Active power P

The three-phase active power of the load, in watts (W).

Inductive reactive power QL

The three-phase inductive reactive power QL, in vars. Specify a positive value, or 0.

Capacitive reactive power QC

The three-phase capacitive reactive power QC, in vars. Specify a positive value, or 0.

Measurements

Select `Branch voltages` to measure the three voltages across each phase of the Three-Phase Parallel RLC Load block terminals. For a Y connection, these voltages are the phase-to-ground or phase-to-neutral voltages. For a delta connection, these voltages are the phase-to-phase voltages.

Select `Branch currents` to measure the three total currents (sum of R, L, C currents) flowing through each phase of the Three-Phase Parallel RLC Load block. For a delta connection, these currents are the currents flowing in each branch of the delta.

Select **Branch voltages and currents** to measure the three voltages and the three currents of the Three-Phase Parallel RLC Load block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement		Label
Branch voltages	Y(grounded): Uag, Ubg, Ucg	`Ub1: , Ub2: , Ub3:`
	Y(floating): Uan, Ubn, Ucn	
	Y(neutral): Uan, Ubn, Ucn	
	Delta: Uab, Ubc, Uca	
Branch currents	Y(grounded): Ia, Ib, Ic	`Ib1: , Ib2: , Ib3:`
	Y(floating): Ia, Ib, Ic	
	Y(neutral): Ia, Ib, Ic	
	Delta: Iab, Ibc, Ica	

See Also

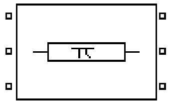
Multimeter, Three-Phase Dynamic Load, Three-Phase Parallel RLC Branch, Three-Phase Series RLC Branch, Three-Phase Series RLC Load

Three-Phase PI Section Line

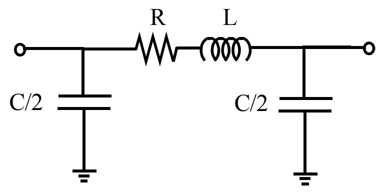
Purpose Implement a three-phase transmission line section with lumped parameters

Library Elements

Description The Three-Phase PI Section Line block implements a balanced three-phase transmission line model with parameters lumped in a PI section.



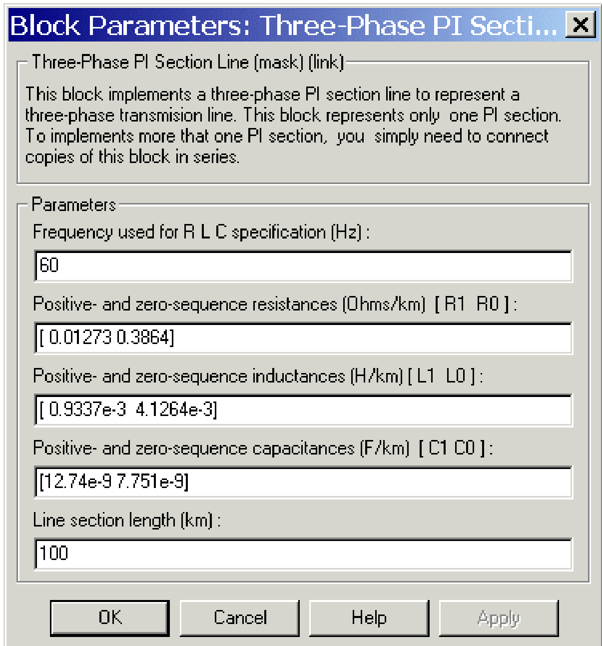
Contrary to the Distributed Parameter Line model where the resistance, inductance, and capacitance are uniformly distributed along the line, the Three-Phase PI Section Line block lumps the line parameters in a single PI section as shown in the figure below where only one phase is represented.



The line parameters R, L, and C are specified as positive- and zero-sequence parameters that take into account the inductive and capacitive couplings between the three phase conductors as well as the ground parameters. This method of specifying line parameters assumes that the three phases are balanced.

Using a single PI section model is appropriate for modeling short transmission lines or when the frequency range of interest is limited around the fundamental frequency. You can obtain a more accurate model by cascading several identical blocks. See the PI Section Line documentation for explanations of the maximum frequency range that can be achieved by a PI line model.

Dialog Box and Parameters



Frequency used for R L C specification

The frequency used for specification of line parameters, in hertz (Hz). This is usually the nominal system frequency (50 Hz or 60 Hz).

Positive- and zero-sequence resistances

The positive- and zero-sequence resistances in ohms/kilometer (Ω/km).

Positive- and zero-sequence inductances

The positive- and zero-sequence inductances in henries/kilometer (H/km).

Positive- and zero-sequence capacitances

The positive- and zero-sequence capacitances in farads/kilometer (F/km).

Line section length

The line section length in kilometers (km).

Example

The `power_triphaseline` demo illustrates voltage transients at the receiving end of a 200 km line when only phase A is energized. Voltages obtained with two line models are compared: 1) the Distributed Parameters Line block and 2) a PI line model using two Three-Phase PI Section Line blocks.

See Also

Distributed Parameter Line, PI Section Line

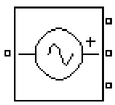
Purpose

Implement a three-phase voltage source with programmable time variation of amplitude, phase, frequency, and harmonics

Library

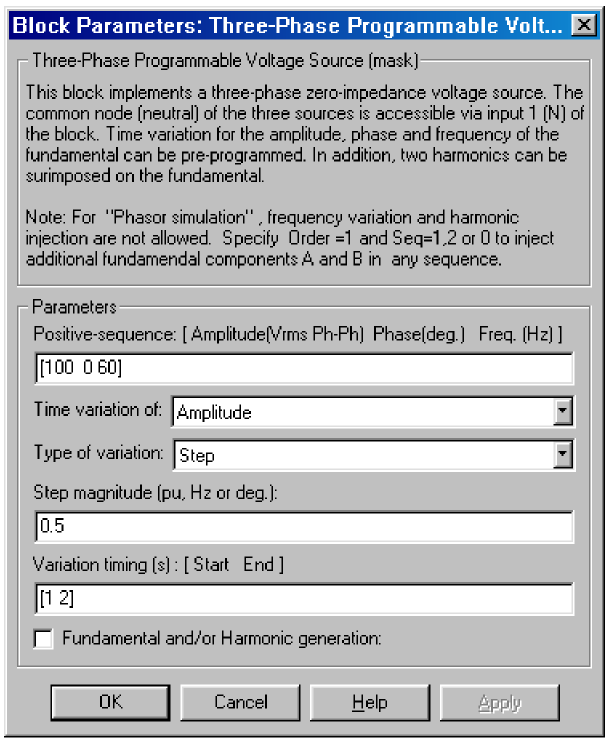
Electrical Sources

Description



Use this block to generate a three-phase sinusoidal voltage with time-varying parameters. You can program the time variation for the amplitude, phase, or frequency of the fundamental component of the source. In addition, two harmonics can be programmed and superimposed on the fundamental signal.

Dialog Box and Parameters



Positive-sequence

The amplitude in volts RMS phase-to-phase, the phase in degrees, and the frequency in hertz of the positive-sequence component of the three voltages.

Time variation of

Specify the parameter for which you want to program the time variation. Select `None` if you do not want to program the time variation of the source parameters. Select `Amplitude` if you want to program the time variation of the amplitude. Select `Phase` if you want to program the time variation of the phase. Select `Frequency` if you want to program the time variation of the frequency.

Note that the time variation applies on the three phases of the source except when the **Type of variation** parameter is set to `Table of amplitude-pairs`, in which case you can apply a variation on phase A only.

Type of variation

Specify the type of variation that is applied on the parameter specified by the **Time variation of** parameter. Select `Step` to program a step variation. Select `Ramp` to program a ramp variation. Select `Modulation` to program a modulated variation. Select `Table of amplitude-pairs` to program a series of step changes of amplitudes at specific times.

Step magnitude

This parameter is only visible if the **Type of Variation** parameter is set to `Step`.

Specify the amplitude of the step change. The variation of amplitude is specified in p.u. of the positive-sequence amplitude.

Rate of change

This parameter is only visible if the **Type of Variation** parameter is set to `Ramp`.

Specify the rate of change, in volt/seconds. The rate of change of voltage is specified in (p.u of the positive-sequence voltage)/second.

Amplitude of the modulation

This parameter is only visible if the **Type of variation** parameter is set to `Modulation`.

Specify the amplitude of the modulation for the source parameter specified in the **Time variation of** parameter. When the varying quantity is the voltage amplitude, the amplitude of the modulation is specified in p.u. of the positive-sequence amplitude.

Frequency of the modulation

This parameter is only visible if the **Type of variation** parameter is set to `Modulation`.

Specify the frequency of the modulation for the source parameter specified in the **Time variation of** parameter.

Variation timing(s)

Specify the time, in seconds, when the programmed time variation takes effect and the time when it stops.

Fundamental and/or Harmonic generation

If selected, two harmonics can be programmed to be superimposed on the fundamental voltage of the source.

A: [Order Amplitude Phase Seq]

This parameter is only visible if the **Fundamental and/or Harmonic generation** check box is selected.

Specify the order, amplitude, phase, and the type of sequence (1 = positive-sequence; 2 = negative-sequence; 0 = zero-sequence) of the first harmonic to be superimposed on the fundamental signal. The voltage of the harmonic is specified in p.u. of the positive-sequence voltage.

Specify `1` for the harmonic order and `0` or `2` for the sequence to produce a voltage imbalance without harmonics.

B: [Order Amplitude Phase Seq]

This parameter is only visible if the **Fundamental and/or Harmonic generation** check box is selected.

Specify the order, amplitude, phase, and the type of sequence (0 = zero sequence, 1 = positive sequence, 2 = negative sequence) of the second harmonic to be superimposed on the fundamental signal. The voltage of the harmonic is specified in p.u. of the positive-sequence voltage.

Specify `1` for the harmonic order and `0` or `2` for the sequence to produce a voltage unbalance without harmonics.

Variation Timing(s)

This parameter is only visible if the **Fundamental and/or Harmonic generation** check box is selected.

Specify the time, in seconds, when the harmonic generation is superimposed on the fundamental signal and the time when it stops.

Inputs and Outputs

Output connectors 1, 2, and 3 are the three source terminals of phases A, B, and C. The input connector is the neutral point. This neutral can be left open (ungrounded wye connection) or grounded (grounded wye connection).

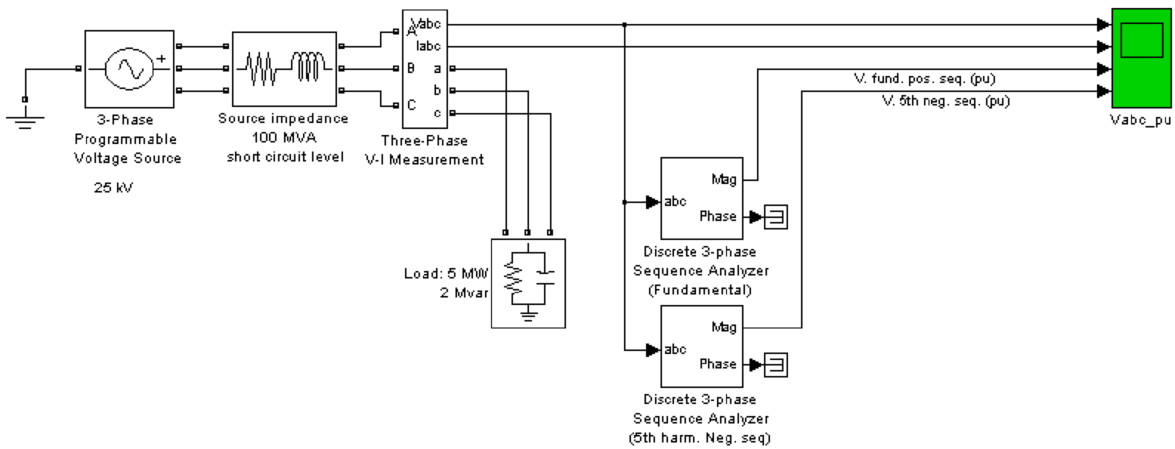
Example

The `power_3phsignalseq` circuit illustrates the use of the Three-Phase Programmable Voltage Source block to produce a step variation of the positive-sequence voltage and to inject harmonics into the circuit.

A 25 kV, 100 MVA short-circuit level, equivalent network feeds a 5 MW, 2 Mvar capacitive load. The internal voltage of the source is controlled by the Discrete 3-phase Programmable Voltage Source block.

A positive sequence of 1.0 p.u., 0 degrees is specified for the fundamental signal. At t = 0.05 s a step of 0.5 p.u. is applied on the positive-sequence voltage magnitude, then at t = 0.1 s, 0.08 p.u. of fifth harmonic in negative sequence is added to the 1.5 p.u. voltage.

The three-phase voltage and current are measured at the output of the source impedance. Two Discrete Sequence Analyzer blocks are used to measure the positive-sequence fundamental component and the negative-sequence fifth harmonic of the three-phase voltage.



See Also

Three-Phase Source

Three-Phase Sequence Analyzer

Purpose

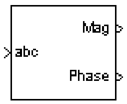
Measure the positive-, negative-, and zero-sequence components of a three-phase signal

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library

Description



The Three-Phase Sequence Analyzer block outputs the magnitude and phase of the positive- (denoted by the index 1), negative- (index 2), and zero-sequence (index 0) components of a set of three balanced or unbalanced signals. The signals can contain harmonics or not. The three sequence components of a three-phase signal (voltages V1 V2 V0 or currents I1 I2 I0) are computed as follows:

$$V_1 = \frac{1}{3}(V_a + a \cdot V_b + a^2 \cdot V_c)$$

$$V_2 = \frac{1}{3}(V_a + a^2 \cdot V_b + a \cdot V_c)$$

$$V_0 = \frac{1}{3}(V_a + V_b + V_c)$$

where

$$V_a, V_b, V_c = \text{three voltage phasors at specified frequency}$$

$$a = e^{j2\pi/3} = 1\angle 120 \text{degree complex operator}$$

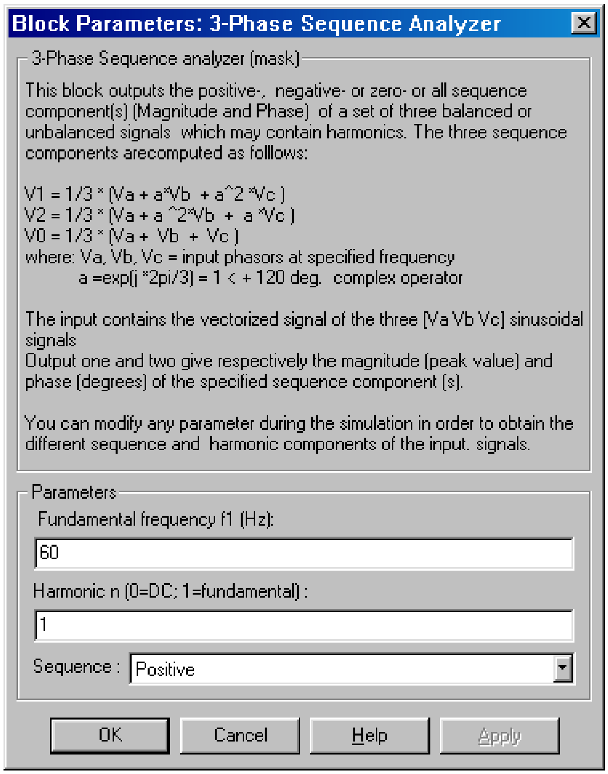
A Fourier analysis over a sliding window of one cycle of the specified frequency is first applied to the three input signals. It evaluates the phasor values Va, Vb, and Vc at the specified fundamental or harmonic frequency. Then the transformation is applied to obtain the positive sequence, negative sequence, and zero sequence.

The Three-Phase Sequence Analyzer block is not sensitive to harmonics or imbalances. However, as this block uses a running average window to perform the Fourier analysis, one cycle of simulation has to be completed before the outputs give the correct magnitude and angle. For example, its response to a step change of V1 is a one-cycle ramp.

The discrete version of this block allows you to specify the initial magnitude and phase of the output signal. For the first cycle of simulation the outputs are held to the values specified by the initial input parameter.

You can modify any parameter during the simulation in order to obtain the different sequence and harmonic components of the input signals.

Dialog Box and Parameters



Fundamental frequency f1

The fundamental frequency, in hertz, of the three-phase input signal.

Harmonic n

Specify the harmonic component from which you want to evaluate the sequences. For DC, enter `0`. For fundamental, enter `1`.

Sequence

Specify which sequence component the block outputs. Select `Positive` to calculate the positive sequence, select `Negative` to calculate the negative sequence, select `0` to compute the zero sequence of the fundamental or specified harmonic of the three-phase input signal. Select `Positive Negative Zero` to get all the sequences.

Inputs and Outputs

`abc`

Connect to the input the vectorized signal of the three [a b c] sinusoidal signals.

`Mag`

The first output gives the magnitude (peak value) of the specified sequence component.

`Phase`

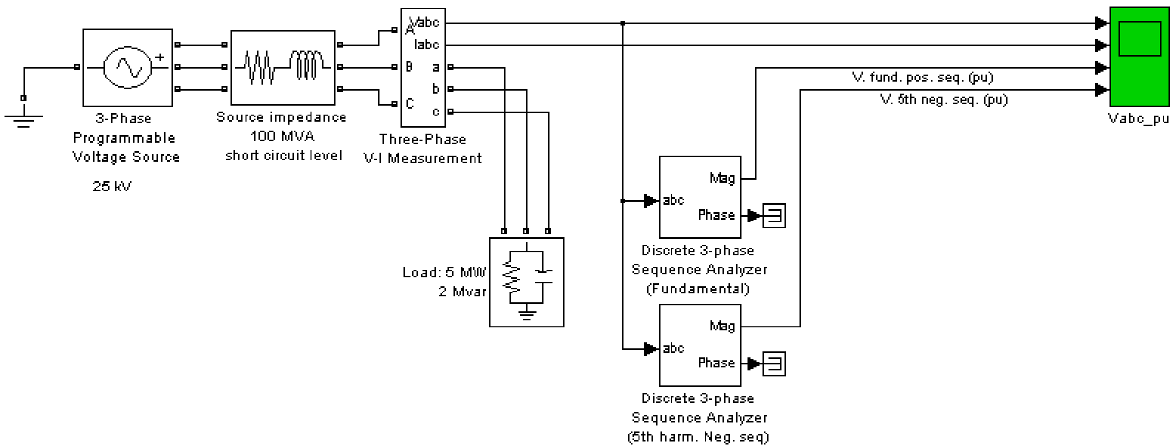
The second output gives the phase in degrees of the specified component(s).

Example

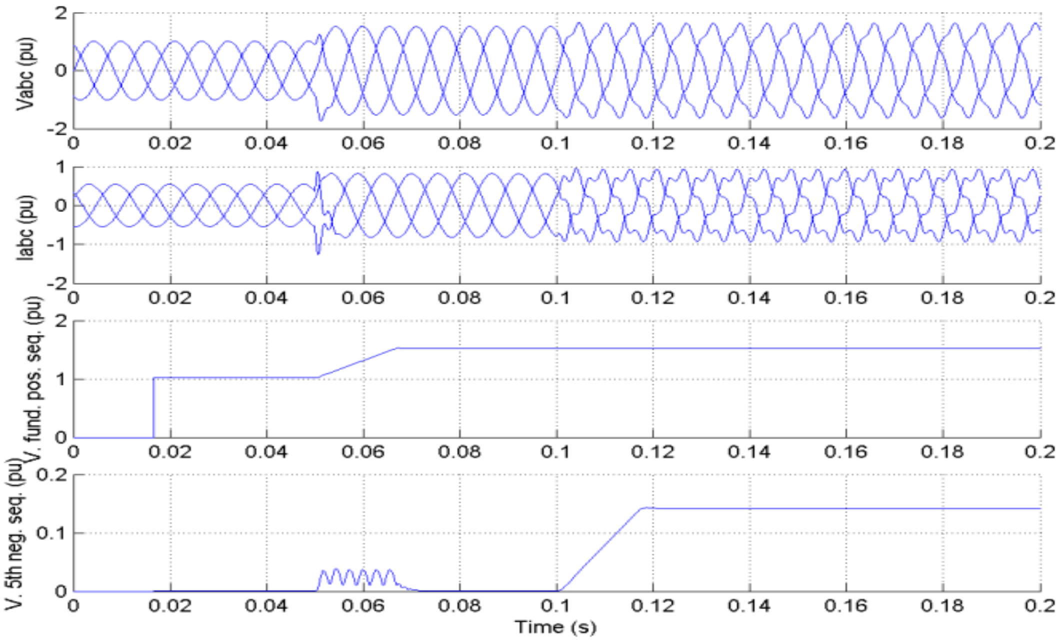
The `power_3phsignalseq` demo illustrates the use of the Discrete Sequence Analyzer block to measure the fundamental and harmonic components of a three-phase voltage. A 25kV, 100 MVA short-circuit level, equivalent network feeds a 5 MW, 2 Mvar capacitive load. The internal voltage of the source is controlled by the Discrete 3-phase Programmable Voltage Source block.

A positive sequence of 1.0 p.u., 0 degrees is specified for the fundamental signal. At t = 0.05 s a step of 0.5 p.u. is applied on the positive-sequence voltage magnitude, then at t = 0.1 s, 0.08 p.u. of fifth harmonic in negative sequence is added to the 1.5 p.u. voltage.

Two Discrete Three-Phase Sequence Analyzer blocks are used to measure the positive-sequence fundamental component and the negative-sequence fifth harmonic of the three-phase voltage.



As the Three-Phase Sequence Analyzer blocks use Fourier analysis, their response time is delayed by one cycle of the fundamental frequency.


Vabc (pu)
Iabc (pu)
V. fund. pos. seq. (pu)
V. 5th neg. seq. (pu)
Time (s)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2

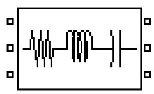
Purpose

Implement a three-phase series RLC branch

Library

Elements

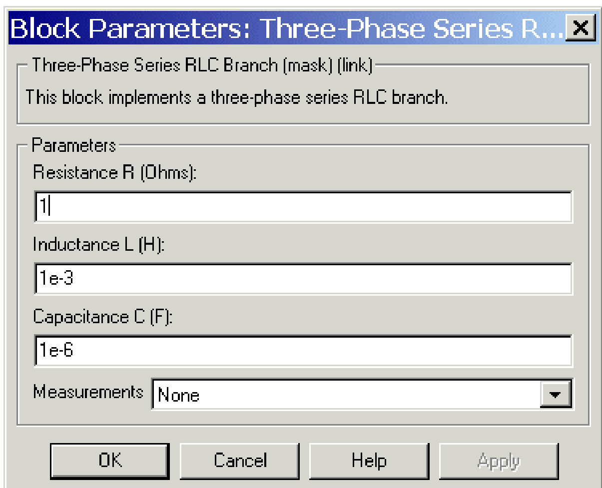
Description



The Three-Phase Series RLC Branch block implements three balanced branches consisting each of a resistor, an inductor, or a capacitor or a series combination of these. To eliminate either the resistance, inductance, or capacitance of each branch, the R, L, and C values must be set respectively to 0, 0, and infinity (inf). Only existing elements are displayed in the block icon.

Negative values are allowed for resistance, inductance, and capacitance.

Dialog Box and Parameters



Resistance R

The branch resistances, in ohms (Ω).

Inductance L

The branch inductances, in henries (H).

Capacitance C

The branch capacitances, in farads (F).

Measurements

Select `Branch voltages` to measure the three voltages across the Three-Phase Series RLC Branch block terminals.

Select `Branch currents` to measure the three currents flowing through the Three-Phase Series RLC Branch block.

Select `Branch voltages and currents` to measure the three voltages and the three currents of the Three-Phase Series RLC Branch block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement	Label
Branch voltages of phases A, B, and C	`Ub1: , Ub2: , Ub3:`
Branch currents of phases A, B, and C	`Ib1: , Ib2: , Ib3:`

See Also

Multimeter, Three-Phase Parallel RLC Branch,Three-Phase Parallel RLC Load, Three-Phase Series RLC Load

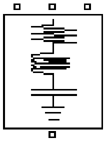
Purpose

Implement a three-phase series RLC load with selectable connection

Library

Elements

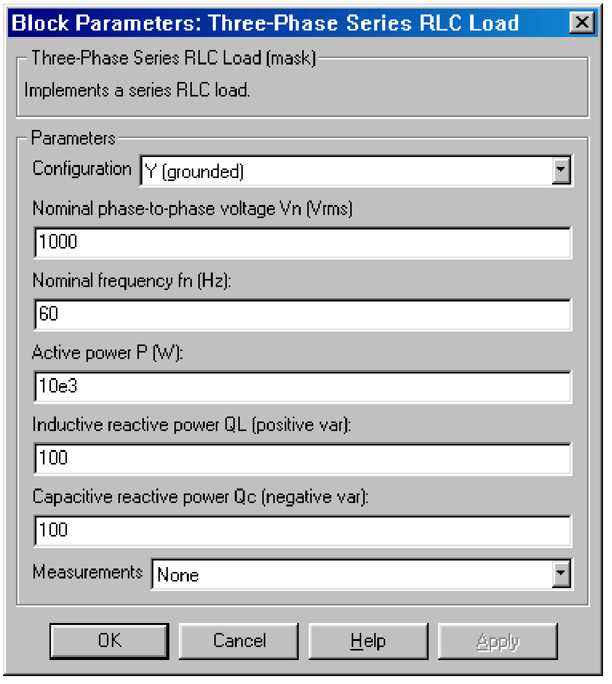
Description



The Three-Phase Series RLC Load block implements a three-phase balanced load as a series combination of RLC elements. At the specified frequency, the load exhibits a constant impedance. The active and reactive powers absorbed by the load are proportional to the square of the applied voltage.

Only elements associated with nonzero powers are displayed in the block icon.

Dialog Box and Parameters



Configuration

The connection of the three phases. Select one of the following four connections:

`Y(grounded)`	Neutral is grounded.
`Y(floating)`	Neutral is not accessible.
`Y(neutral)`	Neutral is made accessible through a fourth connector.
`Delta`	Three phases connected in delta

The block icon is updated according to the load connection.

Nominal phase-to-phase voltage Vn

The nominal phase-to-phase voltage of the load, in volts RMS (Vrms).

Nominal frequency fn

The nominal frequency, in hertz (Hz).

Active power P

The three-phase active power of the load, in watts (W).

Inductive reactive power QL

The three-phase inductive reactive power QL, in vars. Specify a positive value, or 0.

Capacitive reactive power Qc

The three-phase capacitive reactive power QC, in vars. Specify a positive value, or 0.

Measurements

Select `Branch voltages` to measure the three voltages across each phase of the Three-Phase Series RLC Load block terminals. For a Y connection, these voltages are the phase-to-ground or phase-to-neutral voltages. For a delta connection, these voltages are the phase-to-phase voltages.

Select `Branch currents` to measure the three total currents (sum of R, L, C currents) flowing through each phase of the Three-Phase Series RLC Load block. For a delta connection, these currents are the currents flowing in each branch of the delta.

Select `Branch voltages and currents` to measure the three voltages and the three currents of the Three-Phase Series RLC Load block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

Measurement		Label
Branch voltages	Y(grounded): Uag, Ubg, Ucg	`Ub1: , Ub2: , Ub3:`
	Y(floating): Uan, Ubn, Ucn	
	Y(neutral): Uan, Ubn, Ucn	
	Delta: Uab, Ubc, Uca	
Branch currents	Y(grounded): Ia, Ib, Ic	`Ib1: , Ib2: , Ib3:`
	Y(floating): Ia, Ib, Ic	
	Y(neutral): Ia, Ib, Ic	
	Delta: Iab, Ibc, Ica	

See Also

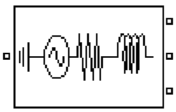
Multimeter, Three-Phase Dynamic Load, Three-Phase Parallel RLC Branch, Three-Phase Parallel RLC Load, Three-Phase Series RLC Branch

Three-Phase Source

Purpose Implement a three-phase source with internal R-L impedance

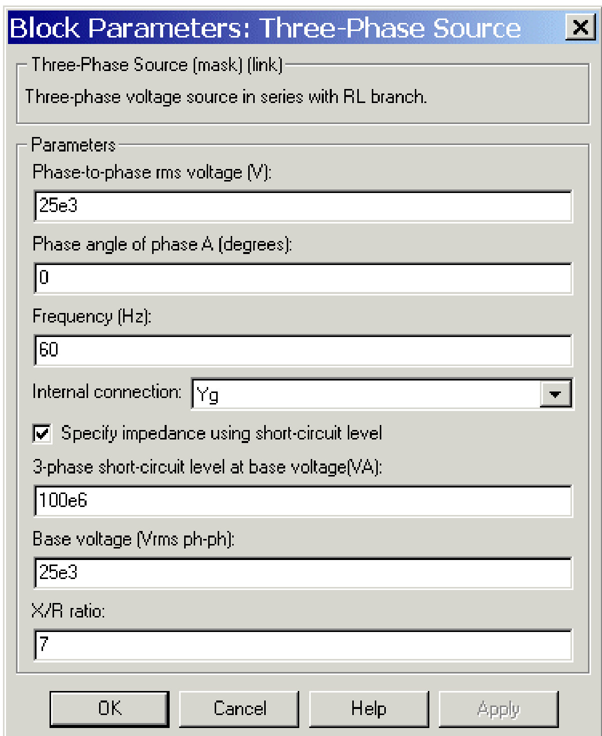
Library Electrical Sources

Description



The Three-Phase Source block implements a balanced three-phase voltage source with an internal R-L impedance. The three voltage sources are connected in Y with a neutral connection that can be internally grounded or made accessible. You can specify the source internal resistance and inductance either directly by entering R and L values or indirectly by specifying the source inductive short-circuit level and X/R ratio.

Dialog Box and Parameters



Phase-to-phase rms voltage

The internal phase-to-phase voltage in volts RMS (Vrms)

Phase angle of phase A

The phase angle of the internal voltage generated by phase A, in degrees. The three voltages are generated in positive sequence. Thus, phase B and phase C internal voltages are lagging phase A respectively by 120 degrees and 240 degrees.

Frequency

The source frequency in hertz (Hz).

Internal connection

The internal connection of the three internal voltage sources. The block icon is updated according to the source connection.

Select one of the following three connections:

`Y`	The three voltage sources are connected in Y to an internal floating neutral.
`Yn`	The three voltage sources are connected in Y to a neutral connection which is made accessible through a fourth terminal.
`Yg`	The three voltage sources are connected in Y to an internally grounded neutral.

Specify impedance using short-circuit level

Select to specify internal impedance using the inductive short-circuit level and X/R ratio.

3-phase short-circuit level at base voltage

The three-phase inductive short-circuit power, in volts-amperes (VA), at specified base voltage, used to compute the internal inductance L. This parameter is available only if **Specify impedance using short-circuit level** is selected.

The internal inductance L (in H) is computed from the inductive three-phase short-circuit power Psc (in VA), base voltage Vbase (in Vrms phase-to-phase), and source frequency f (in Hz) as follows:

$$L = \frac{(Vbase)^2}{Psc} \cdot \frac{1}{2\pi f}$$

Base voltage

The phase-to-phase base voltage, in volts RMS, used to specify the three-phase short-circuit level. The base voltage is usually the nominal source voltage. This parameter is available only if **Specify impedance using short-circuit level** is selected.

X/R ratio

The X/R ratio at nominal source frequency or quality factor of the internal source impedance. This parameter is available only if **Specify impedance using short-circuit level** is selected.

The internal resistance R (in Ω) is computed from the source reactance X (in Ω) at specified frequency, and X/R ratio as follows:

$$R = \frac{X}{(X/R)} = \frac{2\pi f L}{(X/R)}$$

Source resistance

This parameter is available only if **Specify impedance using short-circuit level** is not selected.

The source internal resistance in ohms (Ω).

Source inductance

This parameter is available only if **Specify impedance using short-circuit level** is not selected.

The source internal inductance in henries (H).

Note Either resistance or inductance of the source can be set to zero, but not both at the same time. The block icon is updated accordingly.

Example

See the `power_3phseriescomp` demo, which uses a Three-Phase Source block to model a portion of a 735 kV system with a simplified R-L source. The source impedance is specified by using the three-phase short-circuit level (30,000 MVA) and X/R ratio (X/R = 10).

See Also

Three-Phase Programmable Voltage Source

Three-Phase Transformer 12 Terminals

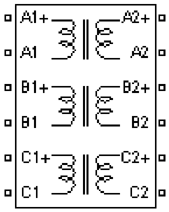
Purpose

Implement three single-phase, two-winding transformers where all terminals are accessible

Library

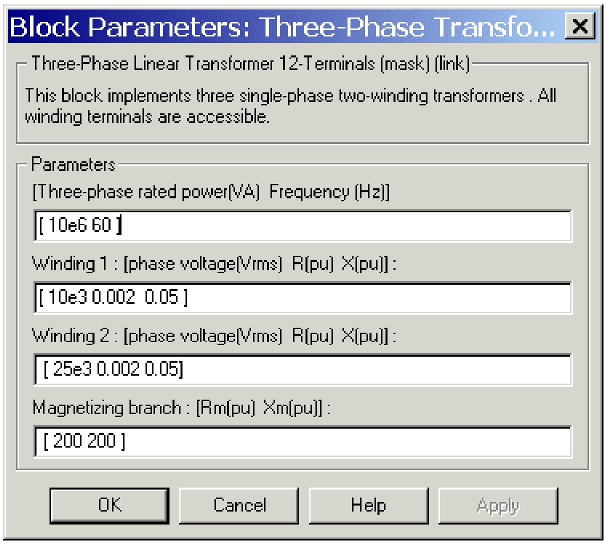
Elements

Description



The Three-Phase Transformer 12 Terminals block implements three single-phase, two-winding linear transformers where all the twelve winding connectors are accessible. The block can be used in place of the Three-Phase Transformer (Two Windings) block to implement a three-phase transformer when primary and secondary are not necessarily connected in Y or Delta.

Dialog Box and Parameters



[Three-phase rated power Frequency]

The total nominal power of the three phases, in volt-amperes (VA), and the nominal frequency, in hertz (Hz).

Winding 1: [phase voltage R X]

The nominal voltage of the three primary windings (labeled 1) in volts RMS (Vrms), the winding resistances, in p.u., and the winding leakage reactances, in p.u.

Winding 2: [phase voltage R X]

The nominal voltage of the three secondary windings (labeled 2) in volts RMS (Vrms), the winding resistances, in p.u., and the winding leakage reactances, in p.u.

Magnetizing branch: [Rm Xm]

The resistance and reactance simulating the core active and reactive losses, both in p.u. For example, to specify 0.2% of active and reactive core losses, at nominal voltage, use Rm = 500 p.u. and Lm = 500 p.u. Lm can be set to `inf` (no reactive core losses), but Rm must have a finite value.

Note Refer to the Linear Transformer block documentation for explanations on the per unit system.

Example

See the `power_3phPWM` demo for an example of use of the Three-Phase Transformer 12 Terminals block in a three-phase double-bridge voltage-sourced converter.

See Also

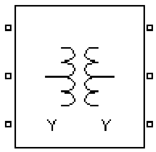
Linear Transformer, Three-Phase Transformer (Two Windings)

Three-Phase Transformer (Two Windings)

Purpose Implement a three-phase transformer with configurable winding connections

Library Elements

Description The Three-Phase Transformer (Two Windings) block implements a three-phase transformer using three single-phase transformers. You can simulate the saturable core or not simply by setting the appropriate check box in the parameter menu of the block. See the Linear Transformer block and Saturable Transformer block sections for a detailed description of the electrical model of a single-phase transformer.



The two windings of the transformer can be connected in the following manner:

- Y
- Y with accessible neutral
- Grounded Y
- Delta (D1), delta lagging Y by 30 degrees
- Delta (D11), delta leading Y by 30 degrees

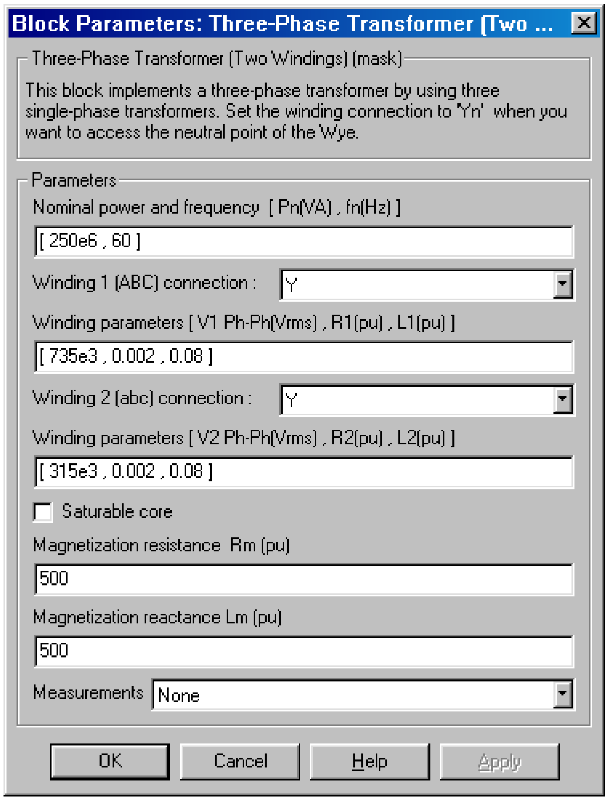
Note The D1 and D11 notations refer to the following clock convention. It assumes that the reference Y voltage phasor is at noon (12) on a clock display. D1 and D11 refer respectively to 1 PM (delta voltages lagging Y voltages by –30 degrees) and 11 AM (delta voltages leading Y voltages by +30 degrees).

The block takes into account the connection type you have selected, and the icon of the block is automatically updated. An input port labeled N is added to the block if you select the Y connection with accessible neutral for winding 1. If you ask for an accessible neutral on winding 2, an extra output port labeled n is generated.

The saturation characteristic, when activated, is the same as the one described for the Saturable Transformer block, and the icon of the block is automatically updated. If the fluxes are not specified, the initial values are automatically adjusted so that the simulation starts in steady state.

The leakage inductance and resistance of each winding are given in p.u. based on the transformer nominal power `Pn` and on the nominal voltage of the winding (V1 or V2). For an explanation of per units, refer to the Linear Transformer and Saturable Transformer block reference pages.

Dialog Box and Parameters



Nominal power and frequency

The nominal power rating, in volt-amperes (VA), and nominal frequency, in hertz (Hz), of the transformer.

Winding 1 (ABC) connection

The winding connections for winding 1.

Winding parameters

The phase-to-phase nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 1.

Winding 2 (abc) connection
The winding connections for winding 2.

Winding parameters
The phase-to-phase nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 2.

Saturable core
If selected, implements a saturable three-phase transformer.

Magnetization resistance Rm
The magnetization resistance Rm, in p.u.

Magnetization reactance Lm
The **Magnetization reactance Lm** parameter is not visible in the dialog box if the **Saturable core** parameter is selected.

The magnetization inductance Lm, in p.u., for a nonsaturable core.

Saturation characteristic
This parameter is visible only if the **Saturable core** parameter is selected.

The saturation characteristic for the saturable core. Specify a series of current/ flux pairs (in p.u.) starting with the pair (0,0).

Simulate hysteresis
Select to model a saturation characteristic including hysteresis instead of a single-valued saturation curve.

Hysteresis data MAT file
This parameter is visible only if the **Simulate hysteresis** parameter is selected.

Specify a `.mat` file containing the data to be used for the hysteresis model. When you open the **Hysteresis Design tool** of the Powergui, the default hysteresis loop and parameters saved in the hysteresis `.mat` file are displayed. Use the **File —> Load a model** menu of the **Hysteresis Design tool** to load another `.mat` file. Use the **File —> Save this model** menu of the **Hysteresis Design tool** to save your model in a new `.mat` file.

Specify initial fluxes

If selected, the initial fluxes are defined by the `[phi0A phi0B phi0C]` parameter.

[phi0A phi0B phi0C]

Specify initial fluxes for each phase of the transformer. This parameter is visible only if the **Specify initial fluxes** and **Saturable core** parameters are selected.

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals of the Three-Phase Transformer block.

Select `Winding currents` to measure the current flowing through the windings of the Three-Phase Transformer block.

Select `Fluxes and excitation currents (Im + IRm)` to measure the flux linkage, in volt seconds (V.s), and the total excitation current including iron losses modeled by Rm (for saturable transformers only).

Select `Fluxes and magnetization currents (Im)` to measure the flux linkage, in volt seconds (V.s), and the magnetization current, in amperes (A), not including iron losses modeled by Rm (for saturable transformers only).

Select `All measurements (V, I, Flux)` to measure the winding voltages, currents, magnetization currents, and the flux linkages.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

If the **Winding 1 ABC connection** parameter is set to `Y`, `Yn`, or `Yg`, the labels are as follows).

Measurement	Label

Winding 1 voltages	`Uan_w1:, Ubn_w1:, Ucn_w1:`
	or
	`Uag_w1:, Ubg_w1:, Ucg_w1:`

Winding 1 currents	`Ian_w1:, Ibn_w1:, Icn_w1:`
	or
	`Iag_w1:, Ibg_w1:, Icg_w1:`

Fluxes	Flux_A:, Flux_B:, Flux_C:
Magnetization currents	Imag_A:, Imag_B:, Imag_C:
Excitation currents	Iexc_A:, Iexc_B:, Iexc_C:

The same labels apply for winding 2, except that `1` is replaced by `2` in the labels.

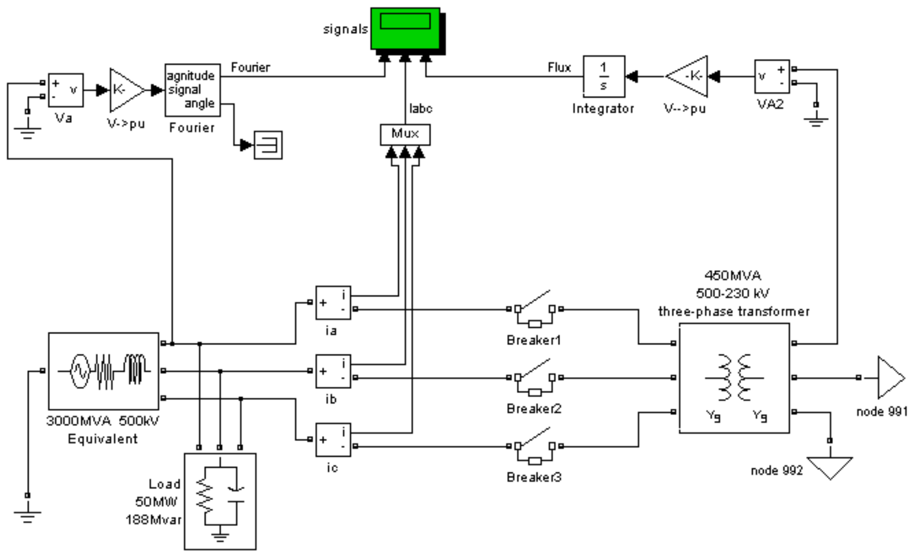
If the **Winding 1 ABC connection** parameter is set to `Delta (D11)` or `Delta (D1)`, the labels are as follows.

Measurement	Label
Winding 1 voltages	Uab_w1:, Ubc_w1:, Uca_w1:
Winding 1 currents	Iab_w1:, Ibc_w1:, Ica_w1:
Flux linkages	Flux_A:, Flux_B:, Flux_C:
Magnetization currents	Imag_A:, Imag_B:, Imag_C:
Excitation currents	Iexc_A:, Iexc_B:, Iexc_C:

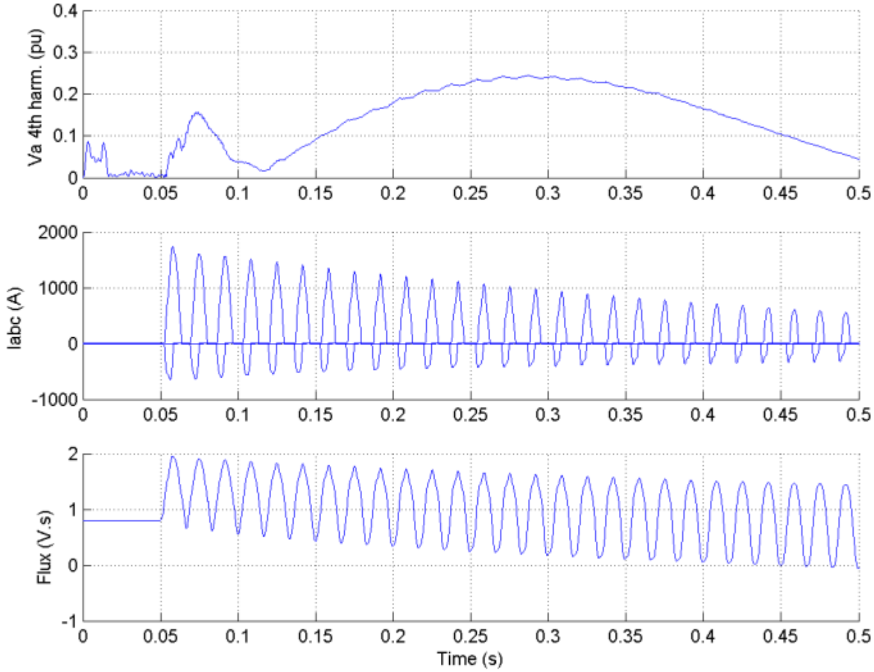
Example

The `power_transfo3ph` circuit uses the Three-Phase Transformer block where the saturable core is simulated. Both windings are connected in a Y grounded configuration. Note that the neutral points of the two windings are internally connected to the ground.

The 500 kV/ 230 kV saturable transformer is energized on the 500 kV system. Remanent fluxes of 0.8 p.u., –0.4 p.u., and 0.4 p.u. have been specified respectively for phases A, B, and C.



Run the simulation and observe inrush currents due to core saturation.



See Also

Linear Transformer, Multimeter, Saturable Transformer, Three-Phase Transformer (Three Windings)

Three-Phase Transformer (Three Windings)

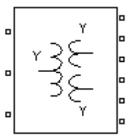
Purpose

Implement a three-phase transformer with configurable winding connections

Library

Elements

Description



This block implements a three-phase transformer by using three single-phase transformers with three windings. You can simulate the saturable core or not simply by setting the appropriate check box in the parameter menu of the block. See the Linear Transformer and Saturable Transformer block sections for a detailed description of the electrical model of a single-phase transformer.

The three windings of the transformer can be connected in the following manner:

- Y
- Y with accessible neutral (for windings 1 and 3 only)
- Grounded Y
- Delta (D1), delta lagging Y by 30 degrees
- Delta (D11), delta leading Y by 30 degrees

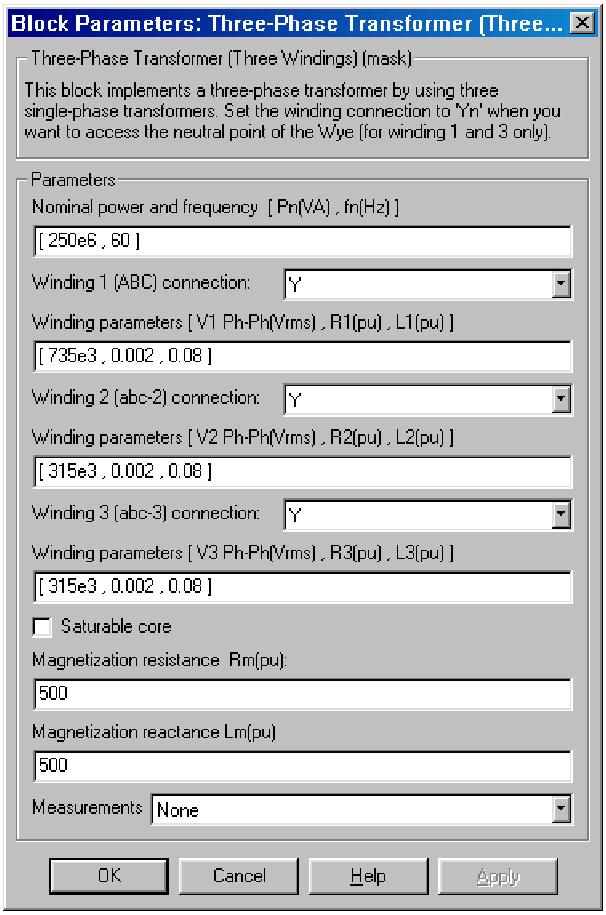
Note The D1 and D11 notations refer to the following clock convention. It assumes that the reference Y voltage phasor is at noon (12) on a clock display. D1 and D11 refer respectively to 1 PM (delta voltages lagging Y voltages by –30 degrees) and 11 AM (delta voltages leading Y voltages by +30 degrees).

The block takes into account the connection type you select, and the icon of the block is automatically updated. An input port labeled `N` is added to the block if you select the Y connection with accessible neutral for winding 1. If you ask for an accessible neutral on winding 3, an extra outport port labeled `n3` is generated.

The saturation characteristic, when activated, is the same as the one described for the Saturable Transformer block, and the icon of the block is automatically updated. If the fluxes are not specified, the initial values are automatically adjusted so that the simulation starts in steady state.

The leakage inductances and resistance of each winding are given in p.u. based on the transformer nominal power `Pn` and on the nominal voltage of the winding (V1, V2, or V3). For an explanation of per units, refer to the Linear Transformer and Saturable Transformer block reference pages.

Dialog Box and Parameters



Nominal power and frequency

The nominal power rating, in volt-amperes (VA), and nominal frequency, in hertz (Hz), of the transformer.

Winding 1 (ABC) connection

The winding connection for winding 1.

Winding parameters

The phase-to-phase nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 1.

Winding 2 (abc2) connection

The winding connection for winding 2.

Winding parameters

The phase-to-phase nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 2.

Winding 3 (abc3) connection

The winding connection for winding 3.

Winding parameters

The phase-to-phase nominal voltage in volts RMS, resistance, and leakage inductance in p.u. for winding 3.

Saturable core

If selected, implements a saturable three-phase transformer.

Magnetization resistance Rm

The magnetization resistance Rm, in p.u.

Magnetization reactance Lm

The magnetization inductance Lm, in p.u., for a nonsaturable core. The **Magnetization reactance Lm** parameter is not visible in the dialog box if the **Saturable core** parameter is selected.

Saturation characteristic

This parameter is visible only if the **Saturable core** parameter is selected.

The saturation characteristic for the saturable core. Specify a series of current/ flux pairs (in p.u.) starting with the pair (0,0).

Simulate hysteresis

Select to model a saturation characteristic including hysteresis instead of a single-valued saturation curve.

Hysteresis data MAT file

This parameter is visible only if the **Simulate hysteresis** parameter is selected.

Specify a `.mat` file containing the data to be used for the hysteresis model. When you open the **Hysteresis Design tool** of the Powergui, the default hysteresis loop and parameters saved in the hysteresis `.mat` file are displayed. Use the **File —> Load a model** menu of the **Hysteresis Design tool** to load another .mat file. Use the **File —> Save this model** menu of the **Hysteresis Design tool** to save your model in a new `.mat` file.

Specify initial fluxes

If selected, the initial fluxes are defined by the `[phi0A phi0B phi0C]` parameter.

[phi0A phi0B phi0C]

Specifies initial fluxes for each phase of the transformer. This parameter is visible only if the **Specify initial fluxes** and **Saturable core** parameters are selected.

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals of the Three-Phase Transformer block.

Select `Winding currents` to measure the current flowing through the windings of the Three-Phase Transformer block.

Select `Fluxes and excitation currents (Im + IRm)` to measure the flux linkage, in volt seconds (V.s), and the total excitation current including iron losses modeled by Rm (for saturable transformers only).

Select `Fluxes and magnetization currents (Im)` to measure the flux linkage, in volt seconds (V.s), and the magnetization current, in amperes (A), not including iron losses modeled by Rm (for saturable transformers only).

Select `All measurements (V, I, Flux)` to measure the winding voltages, currents, magnetization currents, and the flux linkages.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

If the **Winding 1 ABC connection** parameter is set to Y, Yn, or Yg, the labels are as follows.

Measurement	Label
Winding 1 voltages	Uan_w1:, Ubn_w1:, Ucn_w1: or Uag_w1:, Ubg_w1:, Ucg_w1:
Winding 1 currents	Ian_w1:, Ibn_w1:, Icn_w1: or Iag_w1:, Ibg_w1:, Icg_w1:
Flux linkages	Flux_A:, Flux_B:, Flux_C:
Magnetization currents	Imag_A:, Imag_B:, Imag_C:
Excitation currents	Iexc_A:, Iexc_B:, Iexc_C:

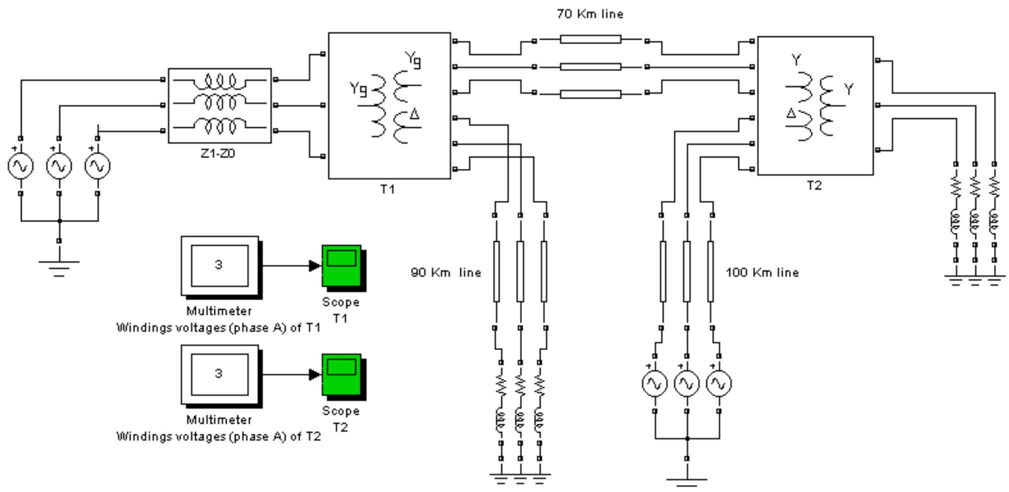
The same labels apply for winding 2 and winding 3, except that the 1 is replaced by 2 or by 3 in the labels.

If the **Winding 1 ABC connection** parameter is set to Delta (D11) or Delta (D1), the labels are as follows.

Measurement	Label
Winding 1 voltages	Uab_w1:, Ubc_w1:, Uca_w1:
Winding 1 currents	Iab_w1:, Ibc_w1:, Ica_w1:
Flux linkages	Flux_A:, Flux_B:, Flux_C:
Magnetization currents	Imag_A:, Imag_B:, Imag_C:
Excitation currents	Iexc_A:, Iexc_B:, Iexc_C:

Example

The power_transfo3wdn circuit uses two Three-Phase Transformer blocks. Two Multimeter blocks are used to measure the phase A voltage (or AB for delta connections) of each winding.



See Also

Linear Transformer, Multimeter, Saturable Transformer, Three-Phase Transformer (Two Windings)

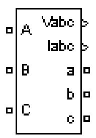
Purpose

Measure three-phase currents and voltages in a circuit

Library

Measurements

Description



The Three-Phase V-I Measurement block is used to measure three-phase voltages and currents in a circuit. When connected in series with three-phase elements, it returns the three phase-to-ground or phase-to-phase voltages and the three line currents.

The block can output the voltages and currents in per unit (p.u.) values or in volts and amperes. If you choose to measure the voltages and currents in p.u., the Three-Phase V-I Measurement block does the following conversions:

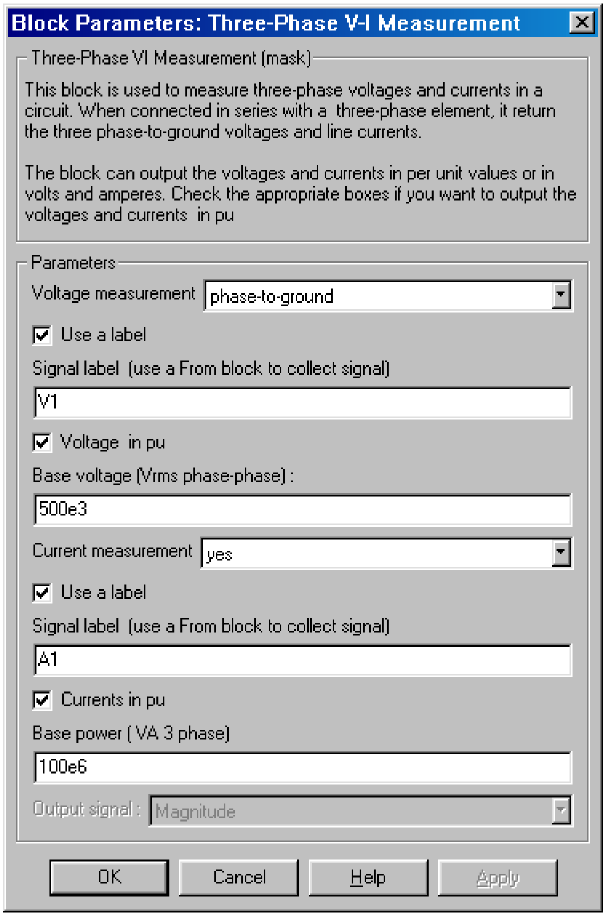
$$V_{abc}(\text{p.u.}) = \frac{V_{abc}(\text{volts})}{\left(\frac{V_{baseLL}}{\sqrt{3}} \cdot \sqrt{2}\right)}$$

$$I_{abc}(\text{p.u.}) = \frac{I_{abc}(\text{amperes})}{\left(\frac{P_{base}}{V_{baseLL} \cdot \sqrt{3}} \cdot \sqrt{2}\right)}$$

where V_{baseLL} is the base line-to-line voltage in volts RMS and P_{base} is the three-phase base power in volts-amperes. The two base values V_{baseLL} and P_{base} are specified in the Three-Phase Measurement block menu.

The steady-state voltage and current phasors measured by the Three-Phase V-I Measurement block can be obtained from the Powergui block by selecting **Steady-State Voltages and Current**s. The phasor magnitudes displayed in the Powergui stay in peak or RMS values even if the output signals are converted to p.u.

Dialog Box and Parameters



Voltage measurement

Select no if you do not want to measure three-phase voltage. Select `phase-to-ground` if you want to measure the phase-to-ground voltages. Select `phase-to-phase` if you want to measure the phase-to-phase voltages.

Use a label

If selected, the voltage measurements are sent to a labeled signal. Use a From block to read the voltages. The Goto tag of the From block must correspond to the label specified by the **Signal label** parameter. If not selected, the voltage measurements are available via the Vabc output of the block.

Signal label

Specifies a label tag for the voltage measurements.

Voltages in p.u.

If selected, the three-phase voltages are measured in p.u. Otherwise they are measured in volts.

Base voltage (Vrms phase-phase)

The base voltage, in volts RMS, used to convert the measured voltages in p.u. The **Base voltage (Vrms phase-phase)** parameter is not visible in the dialog box if `Voltages in p.u.` is not selected.

Current measurement

Select `yes` if you want to measure the three-phase currents that flow through the block.

Use a label

If selected, the current measurements are sent to a labeled signal. Use a From block to read the currents. The Goto tag of the From block must correspond to the label specified by the **Signal label** parameter. If not selected, the current measurements are available via the Iabc output of the block.

Signal label

Specifies a label tag for the current measurements.

Currents in p.u.

If selected, the three-phase currents are measured in p.u. Otherwise they are measured in amperes.

Base power (VA 3 phase)

The three-phase base power, in volt-ampere (VA), used to convert the measured currents in p.u. The **Base power (VA 3 phase)** parameter is not visible in the dialog box if `Currents in p.u.` is not selected.

Output signal

Specifies the format of the measured signals when the block is used in a phasor simulation. The **Output signal** parameter is disabled when the block is not used in a phasor simulation. The phasor simulation is activated by a Powergui block placed in the model.

Set to `Complex` to output the measured voltages and currents as complex values. The outputs are complex signals.

Set to `Real-Imag` to output the real and imaginary parts of the measured voltages and currents.

Set to `Magnitude-Angle` to output the magnitudes and angles of the measured voltages and currents.

Set to `Magnitude` to output the magnitudes of the measured voltages and currents. The output is a scalar value.

Inputs and Outputs

`A, B, C  a, b, c`

Inputs `A`, `B`, `C` and outputs `a`, `b`, `c` are the phase connectors of the measurement block. Connect the Three-Phase V-I Measurement block in series with other three-phase electrical blocks.

`Vabc`

Simulink port Output 1 is a vector containing the three measured phase-to-ground or phase-to-phase voltages. The Vabc output disappears when the **Use a label** parameter is selected or when the **Voltage measurement** menu is set to `no`.

`Iabc`

Simulink port Output 2 is a vector containing the three measured line currents. The Iabc output disappears when the **Use a label** parameter is selected or when the **Current measurement** menu is set to `no`.

Example

See the `power_3phseriescomp` demo for a circuit using the Three-Phase V-I Measurement block.

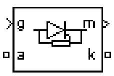
Purpose

Implement a thyristor model

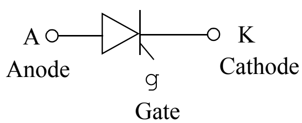
Library

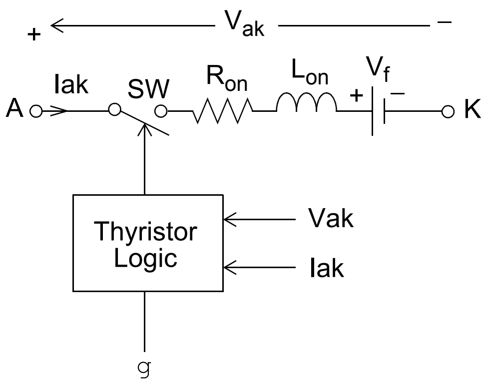
Power Electronics

Description



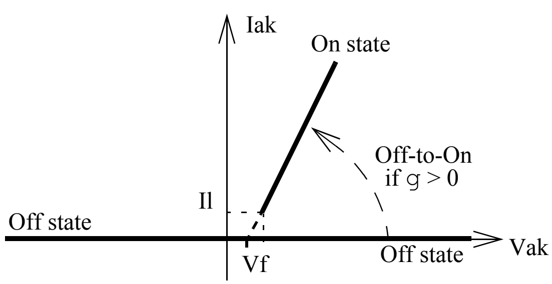
The thyristor is a semiconductor device that can be turned on via a gate signal. The thyristor model is simulated as a resistor Ron, an inductor Lon, and a DC voltage source Vf, connected in series with a switch. The switch is controlled by a logical signal depending on the voltage Vak, the current Iak, and the gate signal g.





The Thyristor block also contains a series Rs-Cs snubber circuit that can be connected in parallel with the thyristor device.

The static VI characteristic of this model is shown below.



The thyristor device turns on when the anode-cathode V_{ak} voltage is greater than Vf and a positive pulse signal is applied at the gate input (g > 0). The pulse height must be greater than 0 and last long enough to allow the thyristor anode current to become larger than the latching current *Il*.

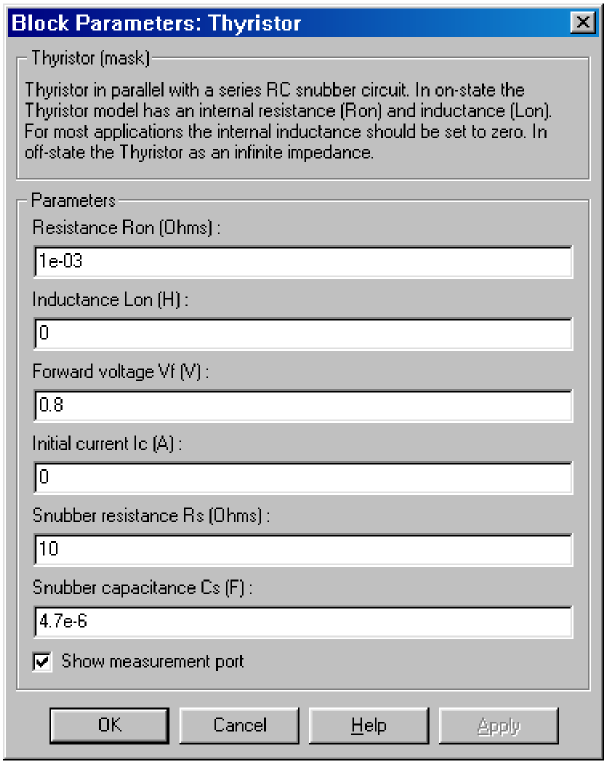
The thyristor device turns off when the current flowing in the device becomes 0 (Iak = 0) and a negative voltage appears across the anode and cathode for at least a period of time equal to the turnoff time Tq. If the voltage across the device becomes positive within a period of time less than Tq, the device turns on automatically even if the gate signal is low (g = 0) and the anode current is less than the latching current. Furthermore, if during turn-on, the device current amplitude stays below the latching current level specified in the dialog box, the device turns off after the gate signal level becomes low (g = 0).

The turnoff time Tq represents the carrier recovery time: it is the time interval between the instant the anode current has decreased to 0 and the instant when the thyristor is capable of withstanding positive voltage Vak without turning on again.

Dialog Boxes and Parameters

Thyristor Model and Detailed Thyristor Model

In order to optimize simulation speed, two models of thyristors are available: the thyristor model and the detailed thyristor model. For the thyristor model, the latching current *Il* and recovery time *Tq* are assumed to be 0.



Resistance Ron

The thyristor internal resistance Ron, in ohms (Ω). The **Resistance Ron** parameter cannot be set to `0` when the **Inductance Lon** parameter is set to `0`.

Inductance Lon

The thyristor internal inductance Lon, in henries (H). The **Inductance Lon** parameter cannot be set to `0` when the **Resistance Ron** parameter is set to `0`.

Forward voltage Vf

The forward voltage of the thyristor, in volts (V).

Initial current Ic

When the **Inductance Lon** parameter is greater than 0, you can specify an initial current flowing in the thyristor. It is usually set to 0 in order to start the simulation with the thyristor blocked.

You can specify an **Initial current Ic** value corresponding to a particular state of the circuit. In such a case all states of the linear circuit must be set accordingly. Initializing all states of a power electronic converter is a complex task. Therefore, this option is useful only with simple circuits.

Snubber resistance Rs

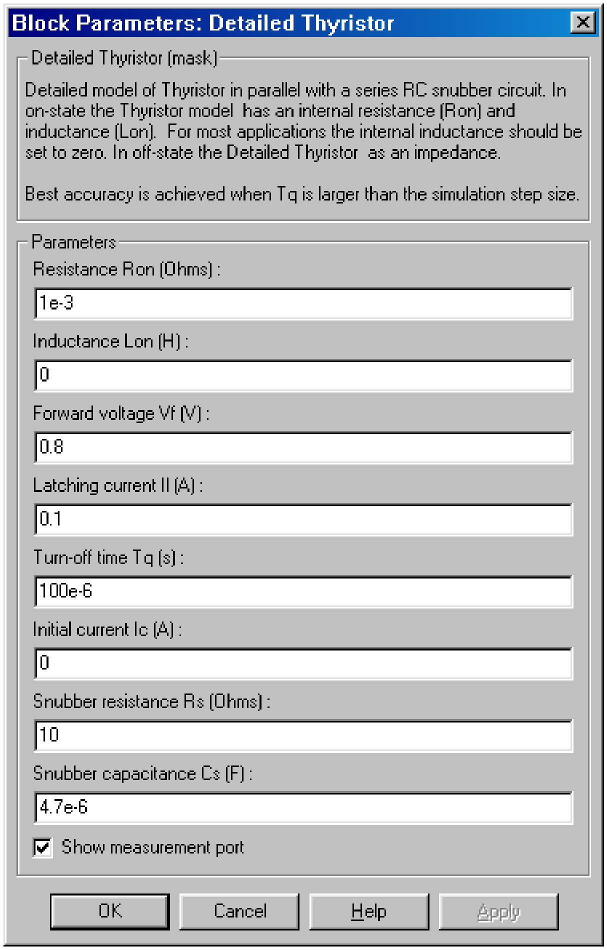
The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubber from the model.

Snubber capacitance Cs

The snubber capacitance in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubber, or to `inf` to get a resistive snubber.

Show measurement port

If selected, add a Simulink output to the block returning the thyristor current and voltage.



Latching current Il

The latching current of the detailed thyristor model, in amperes (A).

Turn-off time Tq

The turnoff time Tq of the detailed thyristor model, in amperes (A).

Inputs and Outputs

The Thyristor block consists of two inputs and two outputs. The first input and output are the thyristor terminals connected respectively to anode (a) and cathode (k). The second input (g) is a Simulink logical signal applied to the gate. The second output (m) is a Simulink signal output vector [Iak Vak] returning the thyristor current and voltage.

Assumptions and Limitations

The Thyristor block implements a macromodel of the real thyristor. It does not take into account either the geometry of the device or complex physical processes that model the behavior of the device [1, 2]. The forward breakover voltage and the critical value of the derivative of the reapplied anode-cathode voltage are not considered by the model.

Depending on the value of **Inductance Lon**, the Thyristor block is modeled either as a current source (Lon > 0) or as a variable topology circuit (Lon = 0). See the "Improving Simulation Performance" chapter for more details.

As the Thyristor block is modeled as a current source, it cannot be connected in series with an inductor, a current source, or an open circuit, unless a snubber circuit is used.

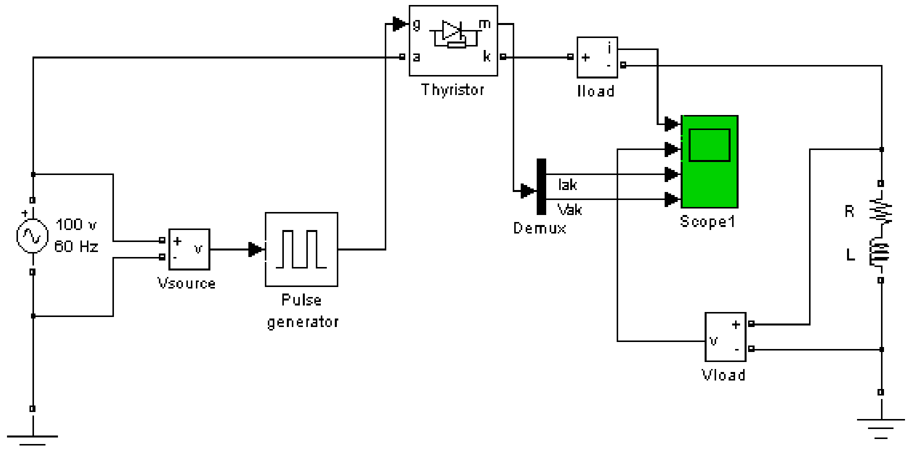
When simulating a continuous model, you must use a stiff integrator algorithm to simulate circuits containing thyristors. `ode23tb` or `ode15s` with default parameters usually gives the best simulation speed.

The inductance Lon is forced to 0 if you choose to discretize your circuit.

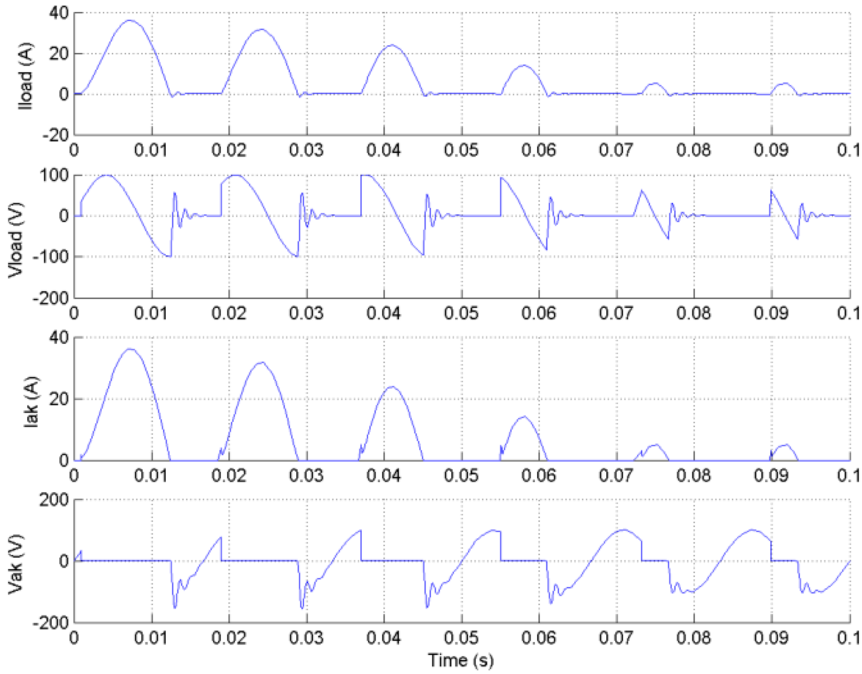
Example

In the `power_thyristor` demo a single-pulse thyristor rectifier is used to feed an RL load. The gate pulses are obtained from a pulse generator synchronized on the source voltage. The following parameters are used:

R		`1 Ω`
L		`10 mH`
Thyristor block:	**Ron**	`0.001 W`
	Lon	`0 H`
	Vf	`0.8 V`
	Rs	`20 Ω`
	Cs	`4e-6 F`



The firing angle is varied by a pulse generator synchronized on the voltage source. Run the simulation and observe the load current and load voltage, as well as the thyristor current and voltage.



References

[1] Rajagopalan, V., *Computer-Aided Analysis of Power Electronic Systems*, Marcel Dekker, Inc., New York, 1987.

[2] Mohan, N., T.M. Undeland, and W.P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, Inc., New York, 1995.

See Also

Diode, Universal Bridge

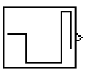
Purpose

Generate a signal changing at specified transition times

Library

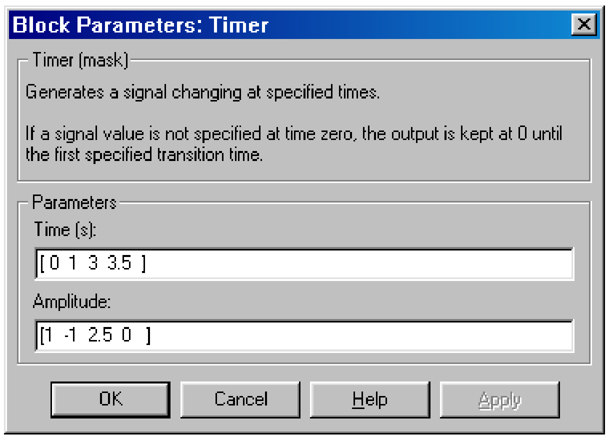
Extras/Control Blocks, Extras/Discrete Control Blocks

Description



The Timer block generates a signal changing at specified transition times. Use this block to generate a logical signal (0 or 1 amplitudes) and control the opening and closing times of power switches like the Breaker block and the Ideal Switch block. You can also use this block to generate a signal whose amplitude changes by steps at specified transition times.

Dialog Box and Parameters



Time(s)

The transition times, in seconds, when the output of the block changes its value as defined by the **Amplitude** parameter. The **Time(s)** parameter must be a vector of the same length as the vector defined in the **Amplitude** parameter. The definition of the time 0 is optional. If a signal is not specified at time 0, the output is kept at zero until the first transition time specified in the **Amplitude** vector.

Amplitude

The vector of amplitudes of signal to be generated by the Timer block. The amplitude is kept constant between transition times defined in the **Time(s)** vector.

Inputs and Outputs

The output is a signal changing by steps at specified transition times.

Example

See the power_breaker model for a circuit using the Timer block to control a circuit breaker.

Purpose

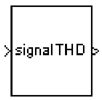
Measure the total harmonic distortion (THD) of a signal

Library

Extras/Measurements

A discrete version of this block is available in the Extras/Discrete Measurements library.

Description



The Total Harmonic Distortion block measures the total harmonic distortion (THD) of a periodic distorted signal. The signal can be a measured voltage or current.

The THD is defined as the root mean square (RMS) value of the total harmonics of the signal, divided by the RMS value of its fundamental signal. For example, for currents, the THD is defined as

$$\text{total harmonic distortion (THD)} = \frac{I_H}{I_F}$$

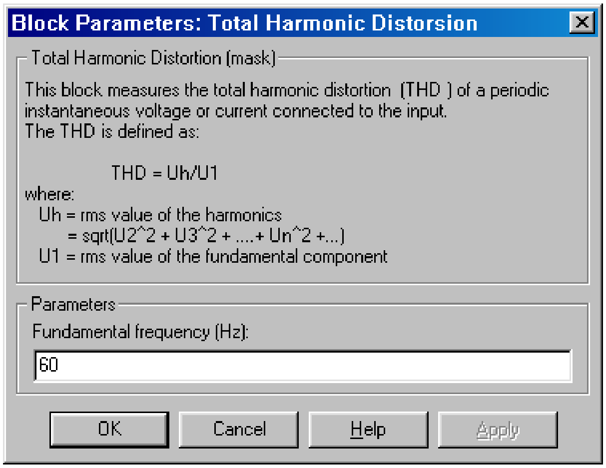
where

$$I_H = \sqrt{I_2^{\;2} + I_3^{\;2} + \ldots + I_n^{\;2}} \qquad I_n\text{: RMS value of the harmonic } n$$

I_F : RMS value of the fundamental current

It follows that the THD has a value comprised between zero and 1. It is null for a pure sinusoidal voltage or current.

Dialog Box and Parameters



Fundamental frequency

The frequency, in hertz, of the fundamental signal.

Inputs and Outputs

Connect to the first input the voltage or current you want to measure the total harmonic distortion. The output returns the THD of the input signal.

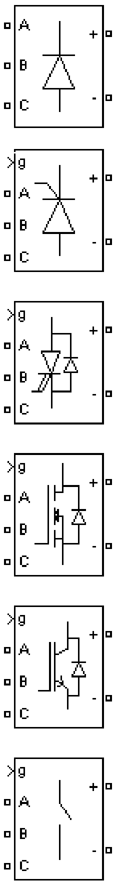
Purpose

Implement a universal power converter with selectable topologies and power electronic devices

Library

Power Electronics

Description

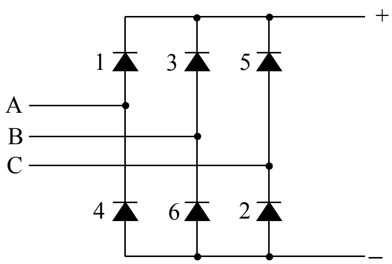


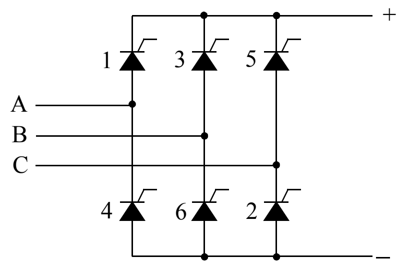
The Universal Bridge block implements a universal three-phase power converter that consists of up to six power switches connected in a bridge configuration. The type of power switch and converter configuration are selectable from the dialog box.

The Universal Bridge block allows simulation of converters using both naturally commutated (or line-commutated) power electronic devices (diodes or thyristors) and forced-commutated devices (GTO, IGBT, MOSFET).

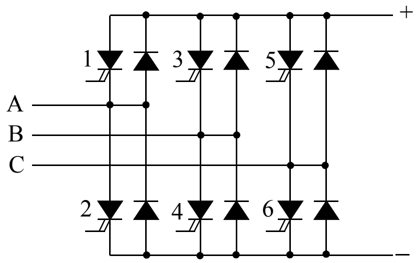
The Universal Bridge block is the basic block for building two-level voltage-sourced converters (VSC).

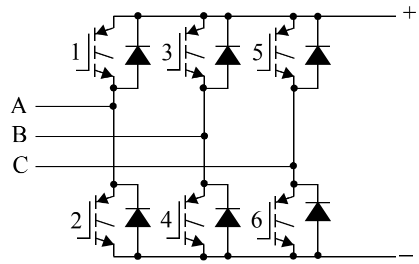
Diode and Thyristor bridges:



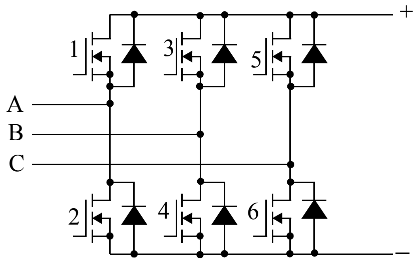


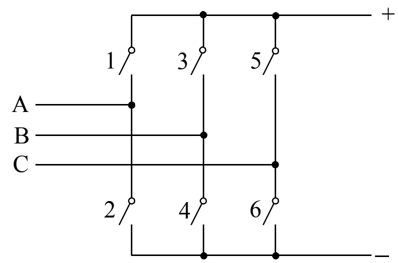
GTO-Diode and IGBT-Diode bridges:





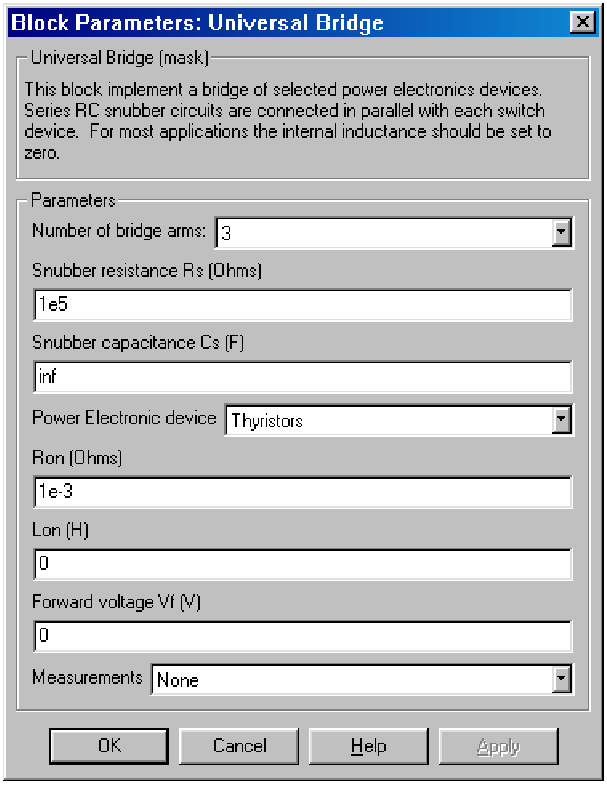
MOSFET-Diode and Ideal Switch bridges:





Note The device numbering is different if the power electronic devices are naturally commutated or forced-commutated. For a naturally commutated converter, numbering follows the natural order of commutation.

Dialog Box and Parameters



Number of bridge arms

Set to 1 or 2 to get a single-phase converter (two or four switching devices). Set to 3 to get a three-phase converter connected in Graetz bridge configuration (six switching devices).

Snubber resistance Rs

The snubber resistance, in ohms (Ω). Set the **Snubber resistance Rs** parameter to `inf` to eliminate the snubbers from the model.

Snubber capacitance Cs

The snubber capacitance, in farads (F). Set the **Snubber capacitance Cs** parameter to `0` to eliminate the snubbers, or to `inf` to get a resistive snubber.

In order to avoid numerical oscillations when your system is discretized, you need to specify Rs and Cs snubber values for diode and thyristor bridges. For forced-commutated devices (GTO, IGBT, or MOSFET), the bridge operates satisfactorily with purely resistive snubbers as long as firing pulses are sent to switching devices.

If firing pulses to forced-commutated devices are blocked, only antiparallel diodes operate, and the bridge operates as a diode rectifier. In this condition appropriate values of Rs and Cs must also be used.

When the system is discretized, use the following formulas to compute approximate values of Rs and Cs:

$$Rs > 2\frac{Ts}{Cs}$$

$$Cs < \frac{Pn}{1000(2\pi f)Vn^2}$$

where

P_n = Nominal power of single or three phase converter (VA)

Vn = Nominal line-to-line AC voltage (Vrms)

f = Fundamental frequency (Hz)

T_s = Sample Time (s)

These Rs and Cs values are derived from the following two criteria:

- The snubber leakage current at fundamental frequency is less than 0.1% of nominal current when power electronic devices are not conducting.
- The RC time constant of snubbers is higher than two times the sample time Ts.

These Rs and Cs values that guarantee numerical stability of the discretized bridge can be different from actual values used in a physical circuit.

Power electronic device

Select the type of power electronic device to use in the bridge.

Ron

Internal resistance of the selected device, in ohms (Ω).

Lon

Internal inductance, in henries (H), for the diode or the thyristor device. When the bridge is discretized, the Lon parameter must be set to zero.

Forward voltage Vf

This parameter is available only when the selected **Power electronic device** is `Diodes` or `Thyristors`.

Forward voltage, in volts (V), across the device when it is conducting.

Forward voltages [Device Vf, Diode Vfd]

This parameter is available when the selected **Power electronic device** is `GTO/Diodes` or `IGBT/Diodes`.

Forward voltages, in volts (V), of the forced-commutated devices (GTO, MOSFET, or IGBT) and of the antiparallel diodes.

[Tf (s) Tt (s)]

Fall time Tf and tail time Tt, in seconds (s), for the GTO or the IGBT devices.

Measurements

Select `Device voltages` to measure the voltages across the six power electronic device terminals.

Select `Device currents` to measure the currents flowing through the six power electronic devices. If antiparallel diodes are used, the measured current is the total current in the forced-commutated device (GTO, MOSFET, or IGBT) and in the antiparallel diode. A positive current therefore indicates a current flowing in the forced-commutated device and a negative current indicates a current flowing in the diode. If snubber devices are defined, the measured currents are the ones flowing through the power electronic devices only.

Select `UAB UBC UCA UDC voltages` to measure the terminal voltages (AC and DC) of the Universal Bridge block.

Select `All voltages and currents` to measure all voltages and currents defined for the Universal Bridge block.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** menu of the Multimeter block, the measurement is identified by a label followed by the block name.

Measurement	Label
Device voltages	`Usw1:, Usw2:,Usw3:,Usw4:,Usw5:,Usw6:`
Branch current	`Isw1:, Isw2:, Isw3:, Isw4:, Isw5:, Isw6:`
Terminal voltages	`Uab:, Ubc):, Uca:, Udc:`

Inputs and Outputs

`A B C +`

The three AC connectors and the two DC connectors of the bridge.

`g`

The gate input, except for the case of a diode bridge. The `g` input accepts a Simulink vector gating signal containing two, four, or six pulse trains, depending on the number of bridge arms (1, 2, or 3). The gating signals are sent to the power switches according to the number shown in the diagrams above.

Note The pulse ordering in the vector of the gate signals corresponds to the switch number indicated in the six circuits shown in the Description section. For the diode and thyristor bridges, the pulse ordering corresponds to the natural order of commutation. For all other forced-commutated switches, pulses are sent to upper and lower switches of phases A, B, and C with the following order: [A upper A lower B upper B lower C upper C lower].

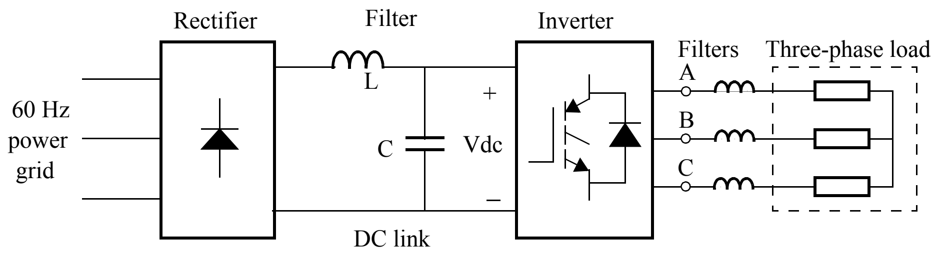
Assumptions and Limitations

Universal Bridge blocks can be discretized for use in a discrete time step simulation. In this case, the internal commutation logic of the Universal Bridge takes care of the commutation between the power switches and the diodes in the converter arms.

Note In a converter built with individual forced-commutated power components (GTOs, MOSFETs, IGBTs), discretization of the model is not available. See the “Improving Simulation Performance” chapter for more details.

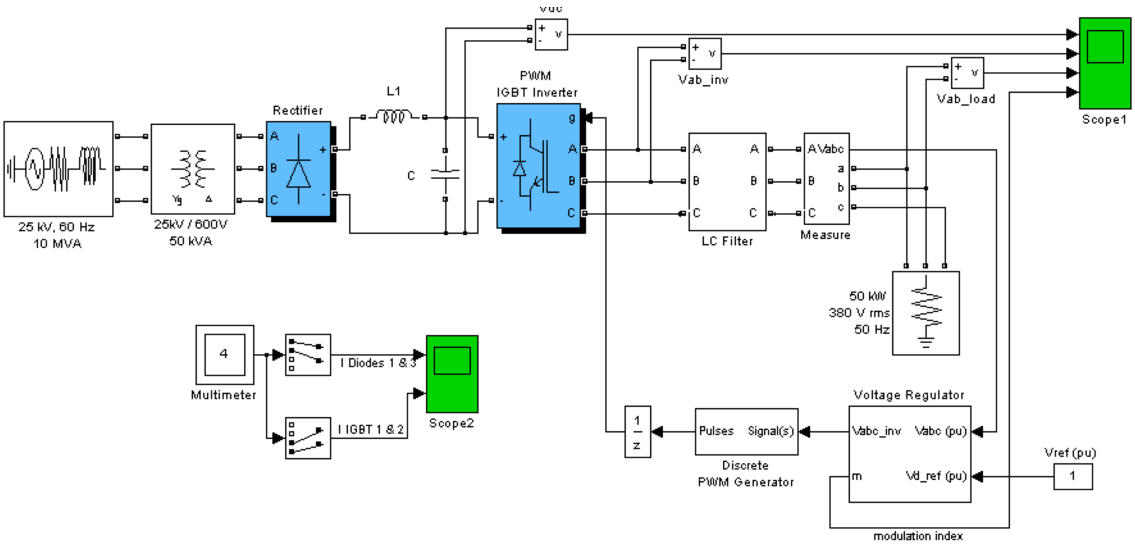
Example

The `power_bridges` demo illustrates the use of two Universal Bridge blocks in an ac/dc/ac converter consisting of a rectifier feeding an IGBT inverter through a DC link. The inverter is pulse-width modulated (PWM) to produce a three-phase 50 Hz sinusoidal voltage to the load. In this example the inverter chopping frequency is 2000 Hz.



The IGBT inverter is controlled with a PI regulator in order to maintain a 1 p.u. voltage (380 Vrms, 50 Hz) at the load terminals.

A Multimeter block is used to observe commutation of currents between diodes 1 and 3 in the diode bridge and between IGBT/Diodes switches 1 and 2 in the IGBT bridge.

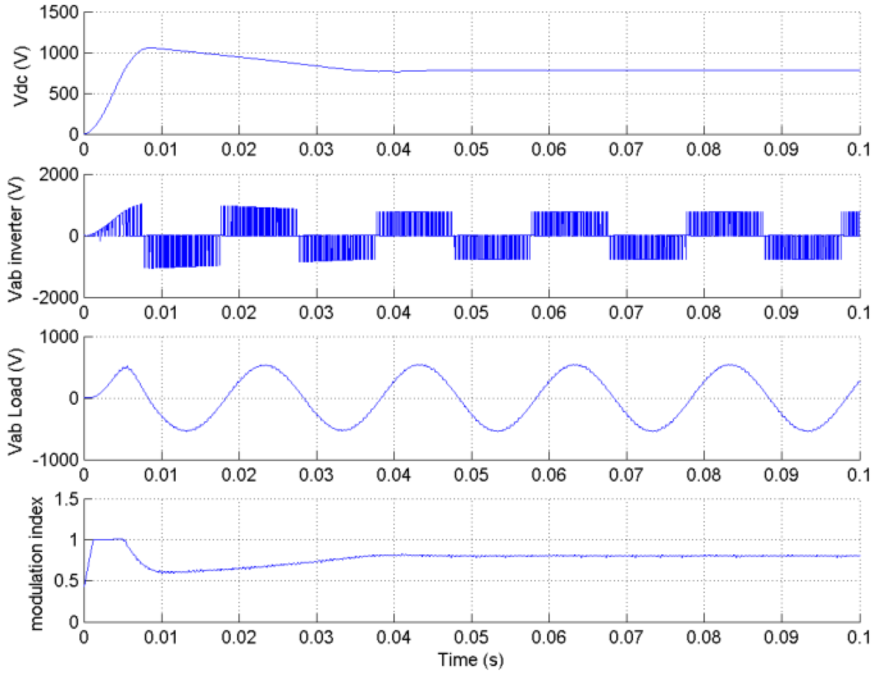


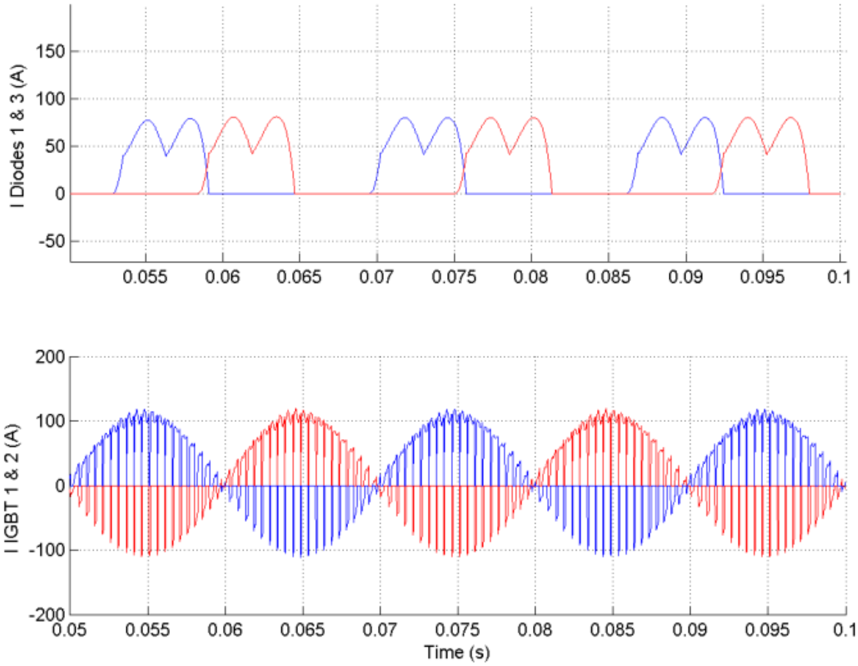
Start simulation. After a transient period of approximately 40 ms, the system reaches a steady state. Observe voltage waveforms at DC bus, inverter output, and load on Scope1. The harmonics generated by the inverter around multiples of 2 kHz are filtered by the LC filter. As expected the peak value of the load voltage is 537 V (380 V RMS).

In steady state the mean value of the modulation index is m = 0.77, and the mean value of the DC voltage is 780 V. The fundamental component of 50 Hz voltage buried in the chopped inverter voltage is therefore

Vab = 780 V * 0.612 * 0.80 = 382 V RMS

Observe diode currents on trace 1 of Scope2, showing commutation from diode 1 to diode 3. Also observe on trace 2 currents in switches 1 and 2 of the IGBT/Diode bridge (upper and lower switches connected to phase A). These two currents are complementary. A positive current indicates a current flowing in the IGBT, whereas a negative current indicates a current flowing in the antiparallel diode.


Vdc (V)
1500
1000
500
0
Vab inverter (V)
2000
0
-2000
Vab Load (V)
1000
0
-1000
modulation index
1.5
1
0.5
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time (s)



See Also

Diode, GTO, Ideal Switch, IGBT, MOSFET, Multimeter,Three-Level Bridge, Thyristor

Purpose

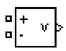
Measure a voltage in a circuit

Library

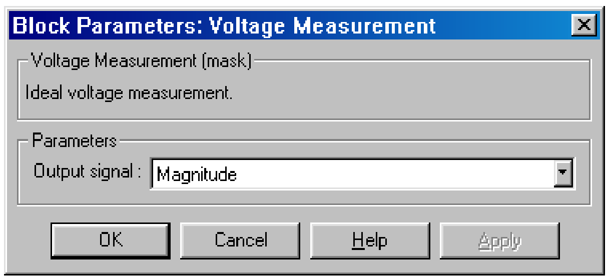
Measurements

Description

The Voltage Measurement block measures the instantaneous voltage between two electric nodes. The output provides a Simulink signal that can be used by other Simulink blocks.



Dialog Box and Parameters



Output signal

Specifies the format of the output signal when the block is used in a phasor simulation. The **Output signal** parameter is disabled when the block is not used in a phasor simulation. The phasor simulation is activated by a Powergui block placed in the model.

Set to `Complex` to output the measured current as a complex value. The output is a complex signal.

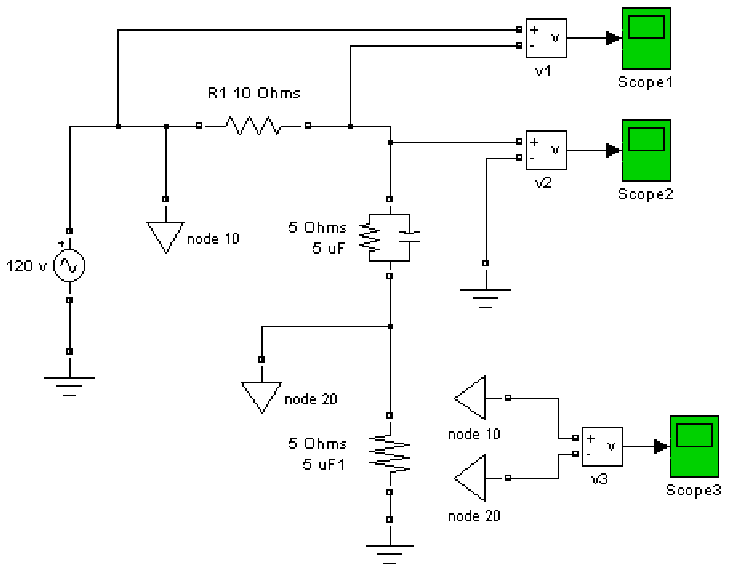
Set to `Real-Imag` to output the real and imaginary parts of the measured current. The output is a vector of two elements.

Set to `Magnitude-Angle` to output the magnitude and angle of the measured current. The output is a vector of two elements.

Set to `Magnitude` to output the magnitude of the measured current. The output is a scalar value.

Example

The `power_voltmeasure` demo uses three Voltage Measurement blocks to read voltages.



See Also

Current Measurement, Powergui, Three-Phase V-I Measurement

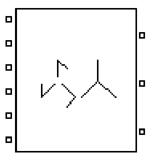
Purpose

Implement a zigzag phase-shifting transformer with a configurable secondary winding connection

Library

Elements

Description



The Zigzag Phase-Shifting Transformer block implements a three-phase transformer with a primary winding connected in a zigzag configuration and a configurable secondary winding. The model uses three single-phase, three-winding transformers. The primary winding connects the windings 1 and 2 of the single-phase transformers in a zigzag configuration. The secondary winding uses the windings 3 of the single phase transformers, and they can be connected in one of the following ways:

- Y
- Y with accessible neutral
- Grounded Y
- Delta (D1), delta lagging Y by 30 degrees
- Delta (D11), delta leading Y by 30 degrees

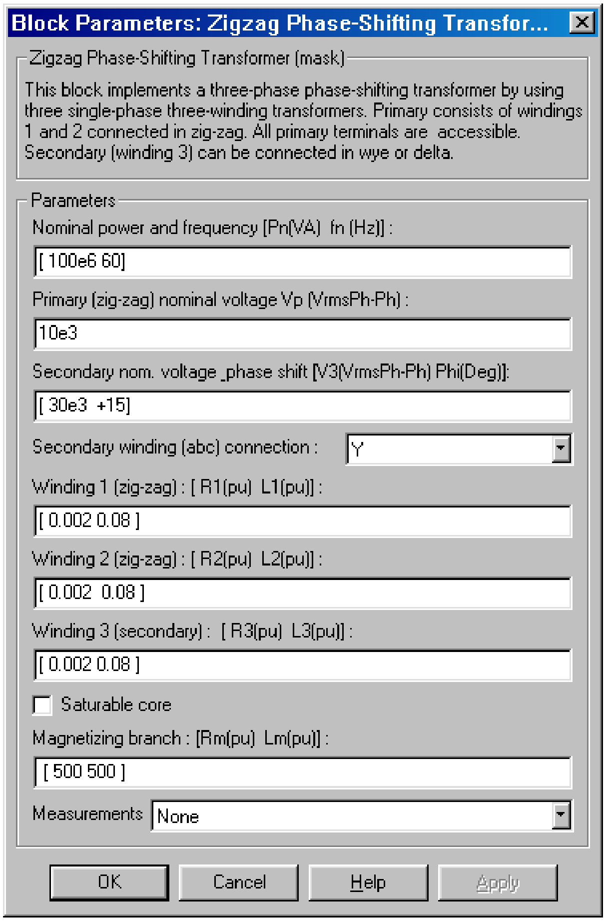
Note The D1 and D11 notations refer to the following clock convention. It assumes that the reference Y voltage phasor is at noon (12) on a clock display. D1 and D11 refer respectively to 1 PM (lagging Y by –30 degrees) and 11 AM (leading Y by +30 degrees).

If the secondary winding is connected in Y, the secondary phase voltages are leading or lagging the primary voltages by the Phi phase angle specified in the parameters of the block. If the secondary winding is connected in delta (D11), an additional phase shift of +30 degrees is added to the phase angle. If the secondary winding is connected in delta (D1), a phase shift of –30 degrees is added to the phase angle.

The block takes into account the connection type you have selected and the icon of the block is automatically updated. An output port labeled N is added to the block if you select the Y connection with accessible neutral for the secondary winding.

The saturation characteristic, when activated, is the same as the one described for the Saturable Transformer block.

Dialog Box and Parameters



Nominal power and frequency

The nominal power rating, in volt-amperes (VA), and nominal frequency, in hertz (Hz), of the transformer.

Primary (zigzag) nominal voltage Vp

The phase-to-phase nominal voltage in volts RMS, for the primary winding of the transformer.

Secondary nominal voltage and phase shift

The phase-to-phase nominal voltage, in volts RMS, and the phase shift, in degrees, for the secondary winding of the transformer.

Secondary winding (abc) connection

The winding connection for the secondary winding.

Winding 1 (zigzag): [R1 L1]

The resistance and leakage inductance of the windings 1 of the single-phase transformers used to implement the primary winding of the Zigzag Phase-Shifting Transformer.

Winding 2 (zigzag): [R2 L2]

The resistance and leakage inductance of the windings 2 of the single-phase transformers used to implement the primary winding of the Zigzag Phase-Shifting Transformer.

Winding 3 (secondary): [R1 L1]

The resistance and leakage inductance of the windings 3 of the single-phase transformers used to implement the secondary winding of the Zigzag Phase-Shifting Transformer.

Saturable core

If selected, implements a saturable core.

Magnetization resistance Rm

This parameter is visible only if the **Saturable core** check box is selected.

The magnetization resistance Rm, in p.u, when the saturation is simulated.

Magnetizing branch: [Rm(p.u.) Lm(p.u.)]

The **Magnetizing branch** parameter is not visible in the dialog box if the **Saturable core** check box is selected.

The magnetization resistance Rm and inductance Lm, in p.u., when the saturation is not simulated.

Saturation characteristic

This parameter is visible only if the **Saturable core** check box is selected.

The saturation characteristic for the saturable core. Specify a series of current/ flux pairs (in p.u.) starting with the pair (0,0).

Measurements

Select `Winding voltages` to measure the voltage across the winding terminals of the Three-Phase Transformer block.

Select `Winding currents` to measure the current flowing through the windings of the Three-Phase Transformer block.

Select `Fluxes and excitation currents (Im + IRm)` to measure the flux linkage, in volt-seconds (V.s), and the total excitation current including iron losses modeled by Rm (for saturable transformers only).

Select `Fluxes and magnetization currents (Im)` to measure the flux linkage, in volt-seconds (V.s), and the magnetization current, in amperes (A), not including iron losses modeled by Rm (for saturable transformers only).

Select `All measurements (V, I, Flux)` to measure the winding voltages, currents, magnetization currents, and the flux linkages.

Place a Multimeter block in your model to display the selected measurements during the simulation. In the **Available Measurements** list box of the Multimeter block, the measurements are identified by a label followed by the block name.

The labels used in the Multimeter are as follows.

Measurement	Label
Phase voltages of primary (zigzag)	`UA:, UB:, UC:`
Phase currents of primary (zigzag)	`IA:, IB:, IC:`

Phase voltages of secondary (Y, Yn or Yg)	`Uan:, Ubn:, Ucn:`
	or
	`Uag:, Ubg:, Ucg:`

Phase voltages of secondary (delta)	`Uab:, Ubc:, Uca:`

Measurement (Continued)	Label (Continued)
Phase currents of secondary (Y, Yn, or Yg)	`Ian:, Ibn:, Icn:` or `Iag:, Ibg:, Icg:`
Phase currents of secondary (delta)	`Iab:, Ibc:, Ica:`
Fluxes	`Flux_a:, Flux_b:, Flux_c:`
Excitation currents	`Iexc_a:, Iexc_b:, Iexc_c:`
Magnetization currents	`Imag_a:, Imag_b:, Imag_c:`

Example

See the help text of the `power_48pulsegtoconverter` demo.

In this model, a 48-pulse GTO converter is built with four Three-Level Bridge blocks and four Zigzag Phase-Shifting Transformer blocks. Harmonic neutralization is obtained by use of appropriate phase shifts introduced by the Zigzag connections (+7.5/–7.5 degrees) and of secondary winding connections (Y or Delta).

See Also

Multimeter, Three-Phase Transformer (Three Windings)

6

SimPowerSystems Command Reference

This table indicates the tasks performed by the commands described in this chapter.

Command	Purpose
power_analyze	Analyze an electric circuit built with SimPowerSystems
power_init	Set the initial states values of an electrical circuit
power_statespace	Compute the linear state-space model of an electrical circuit

power_analyze

Purpose

Analyze an electric circuit built with SimPowerSystems

Syntax

```
sps = power_analyze('sys','structure')
sps = power_analyze('sys','sort')
sps = power_analyze('sys','ss')
[A,B,C,D,x0,states,inputs,outputs,uss,xss,yss,freqyss,Hlin]=
   power_analyze( sys );
power_analyze('sys','net')
```

Description

The `power_analyze` command computes the equivalent state-space model of the specified electrical model built with SimPowerSystems. It evaluates the *A, B, C, D* standard matrices of the state-space system described by the equations

$$\begin{aligned}\dot{x} &= Ax + Bu\\ y &= Cx + Du\end{aligned}$$

where the state variables contained in the *x* vector are the inductor currents and capacitor voltages. Nonlinear elements are simulated by current sources driven by the voltages across the nonlinear elements.

The inputs of the system contained in the *u* vector are the voltage and current sources plus the current sources simulating the nonlinear elements. The outputs of the system contained in the *y* vector are the voltage and current measurements plus the voltages across the nonlinear elements.

The following conventions are used for inputs:

- Source current flowing in the arrow direction is positive.
- Positive source voltage is indicated by a + sign on the icon.

The sign conventions used for voltages and currents (state variables *x* and measurement outputs *y*) are explained below.

Sign Conventions for Voltages and Currents

When you measure a current using a Current Measurement block, the positive direction of current is indicated on the block icon (positive current flowing from + terminal to – terminal). Similarly, when you measure a voltage using a Voltage Measurement block, the measured voltage is the voltage of the + terminal with respect to the – terminal. However, when voltages and currents of blocks from the Elements library are measured using the Multimeter block, the voltage and current polarities are not so evident because blocks might have been rotated and there are no signs indicating polarities on the block icons.

Unlike the Simulink signal lines and input and output ports, the Physical Modeling connection lines and terminal ports □ of SimPowerSystems lack an intrinsic directionality. The voltage and current polarities are determined instead by the block orientation. To find out the block orientation, first click the block to select it. Then enter the following command:

```
get_param(gcb, Orientation )
```

The following table gives the polarities of the currents and voltages measured with the Multimeter block for single-phase and three-phase RLC elements (branches or loads), surge arresters, and single-phase and three-phase breakers. The table also indicates the polarities of the corresponding state variables (inductor currents and capacitor voltages).

Block Orientation	**Positive Current Direction**	**Measured Voltage**
`right`	left —> right	Vleft – Vright
`left`	right —> left	Vright – Vleft
`down`	top —> bottom	Vtop – Vbottom
`up`	bottom —> top	Vbottom – Vtop

The natural orientation of the blocks (that is, their orientation in the Element library) is *right* for horizontal blocks and *down* for vertical blocks.

For single-phase transformers (linear or saturable), with the winding ports appearing on the left and right sides, the winding voltages are the voltages of the top terminal port with respect to the bottom terminal port, whatever the block orientation (*right* or *left*). The winding currents are the currents entering the top port.

For three-phase transformers, the voltage polarities and positive current directions are indicated by the signal labels used in the Multimeter block. For example, Uan_w2 = phase A-to-neutral voltage of the Y connected winding #2, Iab_w1= winding current flowing from A to B in the delta connected winding #1.

Output Arguments: Structure

`sps = power_analyze('sys','structure')` creates a structure array sps with fields and values describing the model sys.

The fields of the structure array are defined in the following order.

Field	Description
circuit	Name of the model
states	char array of state variable names
inputs	char array of system input names
outputs	char array of system output names
A	nstates-by-nstates state-space A matrix
B	nstates-by-ninput state-space B matrix
C	noutput-by-nstates state-space C matrix
D	noutput-by-ninput state-space D matrix
x0	nstates-by-1 vector of initial conditions
Aswitch	A matrix including closed switches
Bswitch	B matrix including closed switches
Cswitch	C matrix including closed switches
Dswitch	D matrix including closed switches
x0switch	Vector of initial values of switch currents
uss	ninput-by-nfreq steady-state values of inputs
xss	nstates-by-nfreq steady-state values of states
yss	noutput-by-nfreq steady-state values of outputs

Field (Continued)	Description (Continued)
Hlin	nfreq-by-noutput-by-ninput transfer function of impedances
frequencies	1-by-nfreq vector of input source frequencies
LoadFlow	Load flow information for circuits with machines
OscillatoryModes	Oscillatory modes of linear part of the system

The table uses the following conventions:

- nstates is the number of states.
- ninput is the number of inputs.
- noutput is the number of outputs.
- nfreq is the number of input source frequencies.

states is a string matrix containing names of the state variables. Each line of states begins with a prefix Uc_ for capacitor voltages or Il_ for inductor currents, followed by the name of the block in which the element (C or L) is found. See “Sign Conventions for Voltages and Currents” on page 6-22 for inductor current directions and capacitor voltage polarities. A string is added to this prefix for blocks containing more than two inductances or capacitors. For example, the Linear Transformer blocks produce four lines in the states matrix, one for each of the three leakage inductances, with prefixes Il_winding_x:, where x is the winding number of the transformer, and one line for the magnetization inductance with the prefix Il_Lm:.

inputs is a string matrix containing names of the inputs of the system. Each line of inputs begins with a prefix U_ for voltage sources or I_ for current sources, followed by the name of the source block. A string can be added to the prefix for blocks containing more than one source. For example, the Synchronous Machine block produces two current inputs with prefixes I_A: and I_B: (phase A and phase B machine currents).

outputs is a string matrix containing names of the outputs of the state-space system (vector y). Each line of outputs begins with a prefix U_ for voltage outputs or I_ for current outputs, followed by the name of the block that produces the output. A string can be added to the prefix for blocks containing more than one source. For example, the Synchronous Machine block produces two voltage outputs with prefixes U_AB: and U_BC: (two machine phase-to-phase voltages). See "Sign Conventions for Voltages and Currents" on page 6-22 for current directions and voltage polarities.

A,B,C,D are the state-space matrices of the linear part of the model.

x0 is a vector containing the initial conditions of the state variables listed in the states variable.

uss, xss, and yss are complex matrices containing the steady-state values of inputs, states and outputs. If voltage and current sources all generate the same frequency, these are column vectors. If sources with different frequencies are used, each column of the matrices corresponds to a frequency contained in the frequencies vector.

frequencies is a row vector containing the input source frequencies ordered by increasing values.

Hlin is the complex transfer impedance three-dimensional array (nfreq-by-noutput-by ninput) of the linear system corresponding to the frequencies contained in the frequencies vector. For a particular frequency, Hlin is defined by

```
yss(:,ifreq) = Hlin(ifreq,:,:) * uss(:,ifreq)
```

Output Arguments: Sort

sps = power_analyze('sys','**sort**') returns a structure array sps with the following fields related to the interconnection of SimPowerSystems blocks in a model. The fields are defined in the following order.

Field	Description
circuit	Name of the model
SampleTime	Sample time for discrete systems
RlcBranch	rlc matrix in the power_statespace format
RlcBranchNames	List of blocks containing the state variable
SourceBranch	Source matrix in the power_statespace format
SourceBranchNames	Names of the blocks defined as sources

Field (Continued)	Description (Continued)
InputNames	Names of the inputs of the system
OutputNames	Names of the outputs of the system
OutputExpressions	Output expression in the power_statespace format
OutputMatrix	Output expression in matrix format (internal)
MeasurementBlocks	Names of the voltage and current measurement blocks

[A,B,C,D,x0,states,inputs,outputs,uss,xss,yss,frequencies,Hlin] = power_analyze('sys') returns the state-space calculations in separate variables.

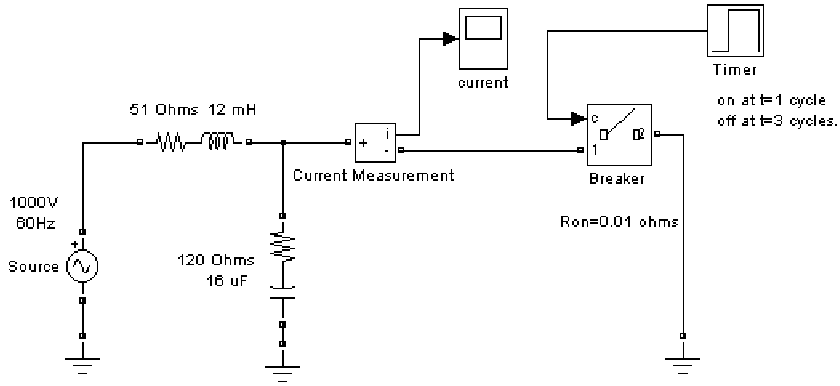
sps = power_analyze('sys','**ss**') creates a continuous state-space model of the model sys with matrices *A, B, C, D*. You must have Control System Toolbox installed for this option. The output is a state-space object.

Output Arguments: Net

power_analyze('sys','**net**') generates a netlist stored in a file, sys.net. The file contains the node numbers automatically generated by power_analyze, as well as parameter values of all linear elements. See the formats described in the power_statespace reference page.

Example

Obtain the state-space matrices and steady-state voltages and currents for the power_netsim2 circuit.



The command

```
sps = power_analyze('power_netsim2','structure')
```

returns the state-space model in the sps structure variable.

```
sps.A =
   1.0e+04 *
         0    6.2500
   -0.0083 -1.4250

sps.uss =
   0
   1000

sps.xss =
   1.0e+02 *
   4.8392 - 5.1314i
   0.0310 + 0.0292i

sps.yss =
   1.0e+02 *
   8.5535 - 1.6287i
    0

sps.inputs =
```

```
      I_Breaker
      U_Source

   sps.outputs =
      U_Breaker
      I_Current Measurement
```

The inductor current of the 51 ohms-12 mH block and the capacitor voltage of the 120 ohms-16 uF block are the two state variables in this circuit. The Breaker block is a nonlinear element that is represented by a current source (the first input) driven by the voltage across its terminals (the first output).

See Also

`power_statespace`, `power_init`, Powergui

power_init

Purpose Set the initial state values of a model built with SimPowerSystems

Syntax

```
power_init('sys','look')
power_init('sys','reset')
power_init('sys','steady')
power_init('sys','set',X0)
power_init('sys','setb','StateVariableName',Value)
```

Description

`power_init('sys','look')` displays the current initial states for the specified system.

`power_init('sys','reset')` resets to zero the initial states of the specified system.

`power_init('sys','steady')` sets the initial states of the specified system in order to start the simulation in steady state.

`power_init('sys','set',X0)` sets the initial state values of the model `sys` to the specified vector `X0`. The ordering of the `states` variable is given by the `power_init('sys','look')` command.

`power_init('sys','setb','StateVariableName',Value)` sets the initial state of the variable specified in '`StateVariableName`' to `Value`. The names of the variables `states` are given by the `power_init('sys','look')` command.

Example

The following commands reset to zero the initial state values of the `power_filter` demo.

```
power_filter
power_init('power_filter','reset')
```

This command returns the names of the states and their current values.

```
power_init('power_filter','look')

Initial states for a particular case:

Il_5th Harm. Filter = 0
Uc_5th Harm. Filter = 0
Il_Zsource          = 0
```

See Also `power_analyze`, Powergui

Purpose

Compute the state-space model of a linear electrical circuit

Synopsis

You must call power_statespace with a minimum of seven input arguments.

```
[A,B,C,D,states,x0,x0sw,rlsw,u,x,y,freq,Asw,Bsw,Csw,Dsw,Hlin] =
power_statespace(rlc,switches,source,line_dist,yout,y_type,unit)
```

You can also specify optional arguments. To use these optional arguments, the number of input arguments must be 12, 13, 14 or 16.

```
[A,B,C,D,states,x0,x0sw,rlsw,u,x,y,freq,Asw,Bsw,Csw,Dsw,Hlin] =
power_statespace(rlc,switches,source,line_dist,yout,y_type,unit,
net_arg1,net_arg2,net_arg3,...,netsim_flag,fid_outfile,
freq_sys,ref_node,vary_name,vary_val)
```

Description

The power_statespace command computes the state-space model of a linear electrical circuit expressed as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

where x is the vector of state-space variables (inductor currents and capacitor voltages), u is the vector of voltage and current inputs, and y is the vector of voltage and current outputs.

When you build a circuit from SimPowerSystems blocks of the **powerlib** library, power_statespace is automatically called by the power_analyze command. power_statespace is also available as a stand-alone command for expert users. This allows you to generate state-space models without using the SimPowerSystems block modeling interface and to access options that are not available through **powerlib**. For example, using power_statespace, you can model transformers and mutual inductances with more than three windings.

The linear circuit can contain any combination of voltage and current sources, RLC branches, multiwinding transformers, mutually coupled inductances, and switches. The state variables are inductor currents and capacitor voltages.

The state-space representation (matrices A,B,C,D, and vector x0) computed by power_statespace can then be used in a Simulink system, via a State-Space block, to perform simulation of the electrical circuit (see the “Example” on page 6-22). Nonlinear elements (mechanical or power electronic switches, transformer saturation, machines, distributed parameter lines, etc.) can be connected to the linear circuit.

These Simulink models are interfaced with the linear circuit through voltage outputs and current inputs of the state-space model. You can find the models of the nonlinear elements provided with SimPowerSystems in the **powerlib_models** library (see the "Improving Simulation Performance" chapter).

Input Arguments

The number of input arguments must be 7, 12, 13, 14, or 16. Arguments 8 to 16 are optional. The first seven arguments that must be specified are

- `rlc`: Branch matrix specifying the network topology as well as the resistance R, inductance L, and capacitance C values. See format below.
- `switches`: Switch matrix. Specify an empty variable if no switches are used. See format below.
- `source`: Source matrix specifying the parameters of the electrical voltage and current sources. Specify an empty variable if no sources are used. See format below.
- `line_dist`: Distributed parameter line matrix. Specify an empty variable if no distributed lines are used. See format below.
- `yout`: String matrix of output expressions. See format below.
- `y_type`: Integer vector indicating output types (`0` for voltage output, `1` for current output).
- `unit`: String specifying the units to be used for R, L, and C values in the `rlc` matrix. If `unit = 'OHM'`, R L C values are specified in ohms Ω at the fundamental frequency specified by `freq_sys` (default value is 60 Hz). If `unit = 'OMU'`, R L C values are specified in ohms (Ω), millihenries (mH), and microfarads (μF).

The last nine arguments are optional. The first three are used to pass arguments from the `power_analyze` command. Hereafter, only the arguments to be specified when `power_statespace` is used as a stand-alone command are described:

- `net_arg1`, `net_arg2`, `net_arg3`: Used to pass arguments from `power_analyze`. Specify an empty variable `[]` for each of these arguments.

- netsim_flag: Integer controlling the messages displayed during the execution of power_statespace. Default value is 0.

  If netsim_flag = 0, the version number, number of states, inputs, outputs, and modes are displayed. Output values are displayed in polar form for each source frequency.

  If netsim_flag = 1, only version number, number of states, inputs, and outputs are displayed.

  If netsim_flag = 2, no message is displayed during execution.
- fid_outfile: File identifier of the power_statespace output file containing parameter values, node numbers, steady-state outputs, and special messages. Default value is 0.
- freq_sys: Fundamental frequency (Hz) considered for specification of X_L and X_C reactances if unit is set to 'OHM'. Default value is 60 Hz.
- ref_node: Reference node number used for ground of PI transmission lines. If -1 is specified, the user is prompted to specify a node number.
- vary_name: String matrix containing the symbolic variable names used in output expressions. These variables must be defined in your MATLAB workspace.
- vary_val: Vector containing the values of the variable names specified in vary_name.

Output Arguments

- A,B,C,D: state-space matrices of the linear circuit with all switches open.

  ```
  A(nstates, nstates) , B(nstates, ninput),
  C(noutput, nstates) , D(noutput, ninput),
  ```

  where nstates is the number of state variables, ninput is the number of inputs, and noutput is the number of outputs.

- states: String matrix containing the names of the state variables. Each string has the following format:

  Inductor currents: Il_bxx_nzz1_zz2

  Capacitor voltages: Uc_bxx_nzz1_zz2

  where

  xx = branch number
  zz1 = first node number of the branch
  zz2 = second node number of the branch

The last lines of the `states` matrix, which are followed by an asterisk, indicate inductor currents and capacitor voltages that are not considered as state variables. This situation arises when inductor currents or capacitor voltages are not independent (inductors forming a cut set – for example, inductors connected in series – or capacitors forming a loop). The currents and voltages followed by asterisks can be expressed as a linear combination of the other state variables:

- `x0`: Column vector of initial values of state variables considering the open or closed status of switches.
- `x0sw`: Vector of initial values of switch currents.
- `rlsw`: Matrix (`nswitch`,`2`) containing the R and L values of series switch impedances in ohms (Ω) and henries (H). `nswitch` is the number of switches in the circuit.
- `u,x,y`: Matrices `u`(`ninput`,`nfreq`), `x`(`nstates`,`nfreq`), and `y`(`noutput`,`nfreq`) containing the steady-state complex values of inputs, states, and outputs. `nfreq` is the length of the `freq` vector. Each column corresponds to a different source frequency, as specified by the next argument, `freq`.
- `freq`: Column vector containing the source frequencies ordered by increasing frequency.
- `Asw,Bsw,Csw,Dsw`: State-space matrices of the circuit including the closed switches. Each closed switch with an internal inductance adds one extra state to the circuit.
- `Hlin`: Three-dimensional array (`nfreq`, `noutput`, `ninput`) of the `nfreq` complex transfer impedance matrices of the linear system corresponding to each frequency of the `freq` vector.

Format of the RLC Input Matrix

Two formats are allowed:

- Six columns: Implicit branch numbering. Branch numbers correspond to the RLC line numbers.
- Seven columns: Explicit branch numbering. Branch number `Nobr` is assigned by the user.

Each line of the `RLC` matrix must be specified according to the following format.

`[node1, node2, type, R, L, C, Nobr]` for RLC branch or line branch

`[node1, node2, type, R, L, C, Nobr]` for transformer magnetizing branch [node1, node2, type, R, L, U, Nobr] for transformer winding

[node1, node2, type, R, L, U, Nobr] for mutual inductances

- node1: First node number of the branch. The node number must be positive or zero. Decimal node numbers are allowed.
- node2: Second node number of the branch. The node number must be positive or zero. Decimal node numbers are allowed.
- type: Integer indicating the type of connection of RLC elements, or, if negative, the transmission line length:

  type = 0: Series RLC element

  type = 1: Parallel RLC element

  type = 2: Transformer winding

  type = 3: Coupled (mutual) winding

  If type is negative, the transmission line is modeled by a PI section of length |type|. See details below.

For a mutual inductor or a transformer having N windings, N+1 consecutive lines must be specified in RLC matrix:

1. N lines with type = 2 or type = 3; (one line per winding). Each line specifies R/L/U or R/Xl/Xc where [R/L, R/Xl = winding resistance and leakage reactance for a transformers or winding resistance and self reactance for mutually coupled windings. U is the nominal voltage of transformer winding (specify 0 if type = 3).
2. One extra line with type = 1 for the magnetizing branch of a transformer (parallel Rm/Lm or Rm/Xm) or one line with type = 0 for a mutual impedance (series Rm/Lm or Rm/Xm).

For a transformer magnetizing branch or a mutual impedance, the first node number is an internal node located behind the leakage reactance of the first winding. The second node number must be the same as the second node number of the first winding.

To model a saturable transformer, you must use a nonlinear inductance instead of the linear inductance simulating the reactive losses. Set the Lm/Xm value to 0 (no linear inductance) and use the Saturable Transformer block, set with proper flux-current characteristics.

This block can be found in the **powerlib_models/Continuous** library. It must be connected to the linear part of the system (State-Space block or S-function) between a voltage output (voltage across the magnetizing branch) and a current input (current source injected into the transformer internal node). See the "Example" on page 6-22.

If `type` is negative, its absolute value specifies the length (km) of a transmission line simulated by a PI section. For a transmission line, the R/L/C or R/Xl/Xc values must be specified in Ω/km, mH/km, and μF/km, or in Ω/km.

Parameter	Description
`R`	Branch resistance (Ω)
`Xl`	Branch inductive reactance (Ω at `freq_sys`) or transformer winding leakage reactance (Ω at `freq_sys`)
`L`	Branch inductance (mH)
`Xc`	Branch capacitive reactance (Ω at `freq_sys`). The negative sign of `Xc` is optional.
`C`	Capacitance (μF)
`U`	Nominal voltage of transformer winding. The same units (volts or kV) must be used for each winding. For a mutual inductance (`type=3`), this value must be set to zero.

Zero value for `R`, `L` or `Xl`, `C` or `Xc` in a series or parallel branch indicates that the corresponding element does not exist.

The following restrictions apply for transformer winding R-L values. Null values are not allowed for secondary impedances if some transformer secondaries form loops (as in a three-phase delta connection). Specify a very low value for R or L or both (e.g., `1e-6` p.u. based on rated voltage and power) to simulate a quasi-ideal transformer. The resistive and inductive parts of the magnetizing branch can be set to infinite (no losses; specify `Xm = Rm = inf`).

Format of the Source Input Matrix

Three formats are allowed:

- Five columns: All sources are generating the same frequency specified by `freq_sys`.
- Six columns: The frequency of each source is specified in column 6.
- Seven columns: The seventh column is used to specify the type of nonlinear element modeled by the current source.

Each line of the source matrix must be specified according to the following format:

```
[ node1, node2, type, amp, phase, freq, model ]
```

- `node1`, `node2`: Node numbers corresponding to the source terminals. These are the polarity conventions:
  - Voltage source: `node1` is the positive terminal.
  - Current source: Positive current flowing from `node1` to `node2` inside the source.
- `type:` Integer indicating the type of source: `0` for voltage source, `1` for current source.
- `amp:` Amplitude of the AC or DC voltage or current (V or A).
- `phase`: Phase of the AC voltage or current (degree).
- `freq`: Frequency (Hz) of the generated voltage or current. Default value is 60 Hz. For a DC voltage or current source, specify `phase = 0` and `freq = 0`. `amp` can be set to a negative value. The generated signals are

  `amp * sin(2π*freq*t + phase)` for AC, `amp` for DC.
- `model:` Integer specifying the type of nonlinear element modeled by the current source (saturable inductance, thyristor, switch, ...). Used by `power_analyze` only.

Order in Which Sources Must Be Specified

The commands that compute the state-space representation of a system expect the sources in a certain order. You must respect this order in order to obtain correct results. You must be particularly careful if the system contains any switches. This is the proper ordering of sources:

1. The currents from all switches that have a null inductance (Lon = 0), if any.

2 The currents from all nonlinear models that have a finite inductance (switches with Lon > 0, the magnetizing inductance in saturable transformers, etc.), if any.
3 All other voltage and current sources in any order, if any.

Refer to the Example section below for an example illustrating proper ordering of sources for a system containing nonlinear elements.

Format of the Switches Input Matrix

Switches are nonlinear elements simulating mechanical or electronic devices such as circuit breakers, diodes, or thyristors. Like other nonlinear elements, they are simulated by current sources driven by the voltage appearing across their terminals. Therefore, they cannot have a null impedance. They are simulated as ideal switches in series with a series R-L circuit. Various models of switches (circuit breaker, ideal switch, and power electronic devices) are available in the **powerlib_models** library. They must be interconnected to the linear part of the system through appropriate voltage outputs and current inputs.

The switch parameters must be specified in a line of the switches matrix in seven different columns, according to the following format.

```
[ node1, node2, status, R, L/Xl, no_I , no_U ]
```

Parameter	Description
node1, node2	Node numbers corresponding to the switch terminals
status	Code indicating the initial status of the switch at t = 0: 0 = open; 1 = closed
R	Resistance of the switch when closed (Ω)
L/Xl	Inductance of the switch when closed (mH) or inductive reactance (Ω at freq_sys)

For these last two fields, you must use the same units as those specified for the RLC matrix. Either field can be set to 0, but not both.

The next two fields specify the current input number and the voltage output number to be used for interconnecting the switch model to the State-Space block. The output number corresponding to the voltage across a particular switch must be the same as the input number corresponding to the current from the same switch (see Example section below):

- `no_I:` Current input number coming from the output of the switch model
- `no_U`: Voltage output number driving the input of the switch model

Format of the Line_Dist Matrix

The distributed parameter line model contains two parts:

1. A linear part containing current sources and resistances that are connected at the line sending and receiving buses together with the linear circuit.
2. A nonlinear part available in the distributed_param_line block of the **powerlib_models/Continuous** library. This block performs the phase-to-mode transformations of voltage and currents and simulates the transmission delays for each mode. The distributed_param_line block must be connected to appropriate voltage outputs and current inputs of the linear part of the system. The line parameters have to be specified in the `line_dist` matrix and also in the distributed_param_line block.

Each row of the `line_dist` matrix is used to specify a distributed parameter transmission line. The number of columns of `line_dist` depends on the number of phases of the transmission line.

For an `nphase` line, the first `(4 + 3 * nphase + nphase^2)` columns are used. For example, for a three-phase line, 22 columns are used.

```
[nphase, no_I, no_U, length, L/Xl, Zc, Rm, speed, Ti]
```

Parameter	Description
nphase	Number of phases of the transmission line
no_I	Input number in the source matrix corresponding to the first current source Is_1 of the line model. Each line model uses 2*nphase current sources specified in the source matrix as follows: Is_1, Is_2, ..., Is_nphase for the sending end followed by Ir_1, Ir_2, ..., Ir_nphase for the receiving end.
nu_U	Output number of the state-space corresponding to the first voltage output Vs_1 feeding the line model. Each line model uses 2*nphase voltage outputs in the source matrix as follows: Vs_1, Vs_2, ..., Vs_nphase for the sending end followed by Vr_1, Vr_2, ..., Vr_nphase for the receiving end.
length	Length of the line (km)
Zc	Vector of the nphase modal characteristic impedances (Ω)
Rm	Vector of the nphase modal series resistances (Ω/km)
speed	Vector of the nphase modal propagation speeds (km/s)
Ti	Transformation matrix from mode to phase currents such that Iphase = Ti * Imod. The nphase * nphase matrix must be given in vector format, [col_1, col_2,... col_nphase].

Format of the Yout Matrix

The desired outputs are specified by a string matrix yout. Each line of the yout matrix must be an algebraic expression containing a linear combination of states and state derivatives, specified according to the following format:

Parameter	Description
Uc_bn	Capacitor voltage of branch *n*
Il_bn	Inductor current of branch *n*
dUc_bn	Derivative of Uc_bn or Il_bn
Un, In	Source voltage or current specified by line *n* of the source matrix
U_nx1_x2	Voltage between nodes x1 and x2 = Ux1 -Ux2
I_bn	Current in branch *n* flowing from node1 to node2 (See format of RLC matrix). For a parallel RLC branch, I_bn corresponds to the total current IR + IL + IC.
I_bn_nx	Current flowing into node x of a PI transmission line specified by line *n* of the RLC matrix. This current includes the series inductive branch current and the capacitive shunt current.

Each output expression is built from voltage and current variable names defined above, their derivatives, constants, other variable names, parentheses and operators (+ − * / ^), in order to form a valid MATLAB expression. For example:

```
yout =
char(['R1*I_b1+Uc_b3-L2*dIl_b2','U_n10_20','I2+3*I_b5']);
```

If variable names are used (R1 and L2 in the above example), their names and values must be specified by the two input arguments vary_name and vary_val.

Sign Conventions for Voltages and Currents

Parameter	Sign Convention
`I_bn, Il_bn, In`	Branch current, inductor current of branch *n*, or current of source #n is oriented from `node1` to `node2`
`I_bn_nx`	Current at one end (node x) of a PI transmission line. If x = `node1`, the current is entering the line. If x = `node2`, the current is leaving the line.
`Uc_bn, Un`	Voltage across capacitor or source voltage (`Unode1 - Unode2`)
`U_nx1_x2`	Voltage between nodes x1 and x2 = Ux1 – Ux2. Voltage of node x1 with respect to node x2.

Order in Which Outputs Must Be Specified

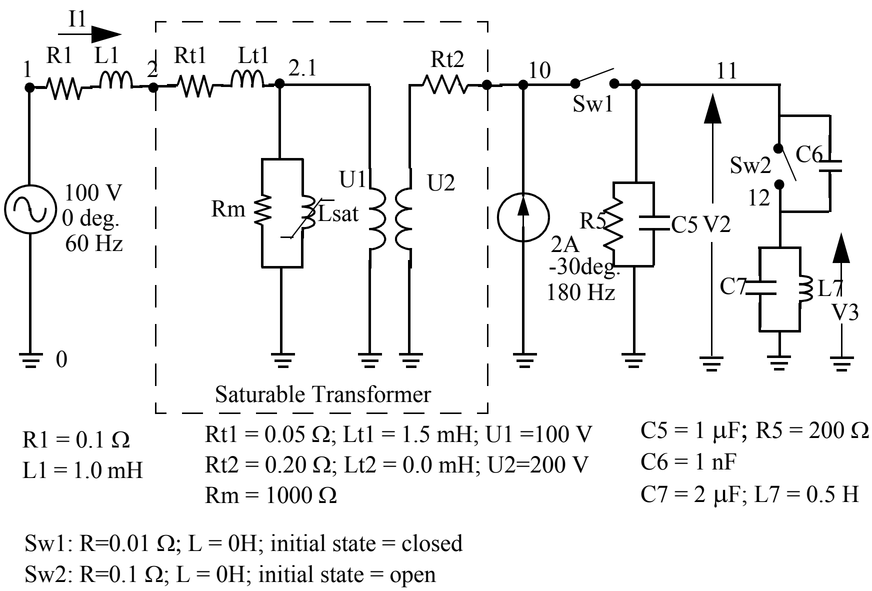
The commands that compute the state-space representation of a system expect the outputs to be in a certain order. You must respect this order in order to obtain correct results. You must be particularly careful if the system contains any switches. The following list gives the proper ordering of outputs:

1. The voltages across all switches that have a null inductance (Lon = 0), if any
2. The currents of all switches that have a null inductance (Lon = 0), if any, in the same order as the voltages above
3. The voltages across all nonlinear models that have a finite inductance (switches with Lon > 0, the magnetizing inductance in saturable transformers, etc.)
4. All other voltage and current measurements that you request, in any order

Refer to the Example section below for an example illustrating proper ordering of outputs for a system containing nonlinear elements.

Example

The following circuit consists of two sources (one voltage source and one current source), two series RLC branches (`R1-L1` and `C6`), two parallel RLC branches (`R5-C5` and `L7-C7`), one saturable transformer, and two switches (`Sw1` and `Sw2`). `Sw1` is initially closed whereas `Sw2` is initially open. Three measurement outputs are specified (I1, V2, and V3). This circuit has seven nodes numbered 0, 1, 2, 2.1, 10, 11, and 12. Node 0 is used for the ground. Node 2.1 is the internal node of the transformer where the magnetization branch is connected.



Linear state space. You can use the `power_statespace` command to find the state-space model of the linear part of the circuit. The nonlinear elements `Sw1`, `Sw2`, and `Lsat` must be modeled separately by means of current sources driven by the voltages appearing across their terminals. Therefore you must provide three additional current sources and three additional voltage outputs for interfacing the nonlinear elements to the linear circuit.

You can find the state-space model of the circuit by entering the following commands in a MATLAB script file. The example is available in the `power_circ2ss.m` file. Notice that an output text file named `power_circ2ss.net` containing information on the system is requested in the call to `power_statespace`.

```
unit='OMU'; % Units = ohms, mH, and uF

rlc=[
```

```
%N1N2  typeR   L   C(uF)/U(V)
1  2   0   0.1 1   0   %R1 L1
2  0   2   0.051.5 100 %transfo Wind.#1
10 0   2   0.200   200 %transfo Wind.#2
2.10   1   10000   0   %transfo mag. branch
11 0   1   200 0   1   %R5 C5
11 12  0   0   0   1e-3%C6
12 0   1   0   500 2   %L7 C7
];

source=[
%N1N2  typeU/I phasefreq
10 11  1   0   0   0   %Sw1
11 12  1   0   0   0   %Sw2
2.10   1   0   0   0   %Saturation
1  0   0   100 0   60  %Voltage source
0  10  1   2   -30 180 %Current source
];

switches=[
%N1N2  statusR(ohm)L(mH)I#U#  #
10 11  1   0.010   1   1   %Sw1
11 12  0   0.1 0   2   2   %Sw2
];

%outputs
%
% Both switches have Lon=0, so their voltages must be the first
outputs,
% immediately followed by their currents (in the same order as the
voltages).
% The voltage across all nonlinear models that don't have L=0
follow
% (in this case the saturable transformer's magnetizing inductor).
% The measurements that you request follow, in any order.
%
y_u1='U_n10_11';%U_Sw1= Voltage across Sw1
y_u2='U_n11_12';%U_Sw2= Voltage across Sw2
y_i3='I1'; %I1= Switch current Sw1
y_i4='I2'; %I2= Switch current Sw2
```

```
y_u5='U_n2.1_0';%U_sat= Voltage across saturable reactor
y_i6='I_b1';%I1 measurement
y_u7='U_n11_0';%V2 measurement
y_u8='U_n12_0';%V3 measurement

yout=char(y_u1,y_u2,y_i3,y_i4,y_u5,y_i6,y_u7,y_u8);% outputs
y_type=[0,0,1,1,0,1,0,0];%output types; 0=voltage 1=current

% Open file that contains power_statespace output information
fid=fopen('power_circ2ss.net','w');

[A,B,C,D,states,x0,x0sw,rlsw,u,x,y,freq,Asw,Bsw,Csw,Dsw,Hlin]=
power_statespace(rlc,switches,source,[],yout,y_type,unit,[],[],
[],0,fid);
```

Command line messages. While `power_statespace` is executing, the following messages are displayed.

```
Computing state space representation of linear electrical circuit
(V2.0)...
(4 states ; 5 inputs ; 7 outputs)

Oscillatory modes and damping factors:
F=159.115Hz zeta=4.80381e-08

Steady state outputs @ F=0 Hz :
y_u1= 0Volts
y_u2= 0Volts
y_i3= 0Amperes
y_i4= 0Amperes
y_u5= 0Volts
y_i6= 0Amperes
y_u7= 0Volts
y_u8= 0Volts
Steady state outputs @ F=60 Hz :
y_u1 = 0.009999 Volts < 3.168 deg.
y_u2 = 199.4 Volts < -1.148 deg.
y_i3 = 0.9999 Amperes < 3.168 deg.
y_i4 = 0 Amperes < 0 deg.
y_u5 = 99.81 Volts < -1.144 deg.
```

```
y_i6 = 2.099 Amperes < 2.963 deg.
y_u7 = 199.4 Volts < -1.148 deg.
y_u8 = 0.01652 Volts < 178.9 deg.

Steady state outputs @ F=180 Hz :
y_u1 = 0.00117 Volts < 65.23 deg.
y_u2 = 22.78 Volts < 52.47 deg.
y_i3 = 0.117 Amperes < 65.23 deg.
y_i4 = 0 Amperes < 0 deg.
y_u5 = 11.4 Volts < 53.48 deg.
y_i6 = 4.027 Amperes < 146.5 deg.
y_u7 = 22.83 Volts < 52.47 deg.
y_u8 = 0.0522 Volts < 52.47 deg.
```

State space output. The names of the state variables are returned in the `states` string matrix.

```
states
states =
Il_b2_n2_2.1
Uc_b5_n11_0
Uc_b6_n11_12
Il_b7_n12_0
Il_b1_n1_2*
Uc_b7_n12_0*
```

Although this circuit contains a total of six inductors and capacitors, there are only four state variables. The names of the state variables are given by the first four lines of the `states` matrix. The last two lines are followed by an asterisk indicating that these two variables are a linear combination of the state variables. The dependencies can be viewed in the output file `power_circ2ss.net`.

```
The following capacitor voltages are dependent:
Uc_b7_n12_0 =  + Uc_b5_n11_0  - Uc_b6_n11_12
The following inductor currents are dependent:
Il_b1_n1_2 =  + Il_b2_n2_0
```

The A,B,C,D matrices contain the state-space model of the circuit without nonlinear elements (all switches open). The `x0` vector contains the initial state values considering the switch Sw1 closed. The `Asw`, `Bsw`, `Csw`, and `Dsw` matrices contain the state-space model of the circuit considering the closed switch `Sw1`. The `x0sw` vector contains the initial current in the closed switch.

```
A
A =
-4.0006e+05         0         0         0
          0     -4995         0   -499.25
          0   -4992.5         04.9925e+05
          0         2        -2         0

Asw
Asw =
   -80.999   -199.99         0         0
4.9947e+05   -5244.7         0   -499.25
4.9922e+05   -5242.1         04.9925e+05
          0         2        -2         0
```

The system source frequencies are returned in the `freq` vector.

```
freq
freq =
     0    60   180
```

The corresponding steady-state complex outputs are returned in the (6-by-3) `y` matrix where each column corresponds to a different source frequency.

For example, you can obtain the magnitude of the six voltage and current outputs at 60 Hz as follows:

```
abs(y(:,2))
ans =
   0.0099987
   199.42
   0.99987
   0
   99.808
   2.0993
   199.41
   0.016519
```

The initial values of the four state variables are returned in the x0 vector. You must use this vector in the State-Space block to start the simulation in steady state.

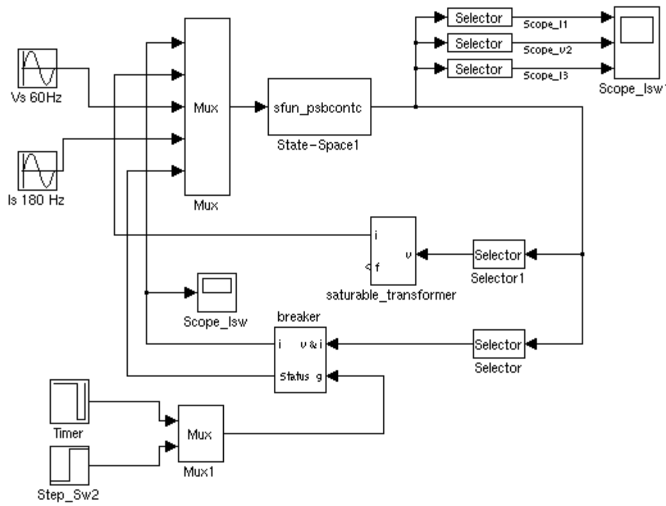
```
x0
x0 =
   2.3302
   14.111
   14.07
   3.1391e-05
```

The initial values of switch currents are returned in x0sw. To start the simulation in steady state, you must use these values as initial currents for the nonlinear model simulating the switches.

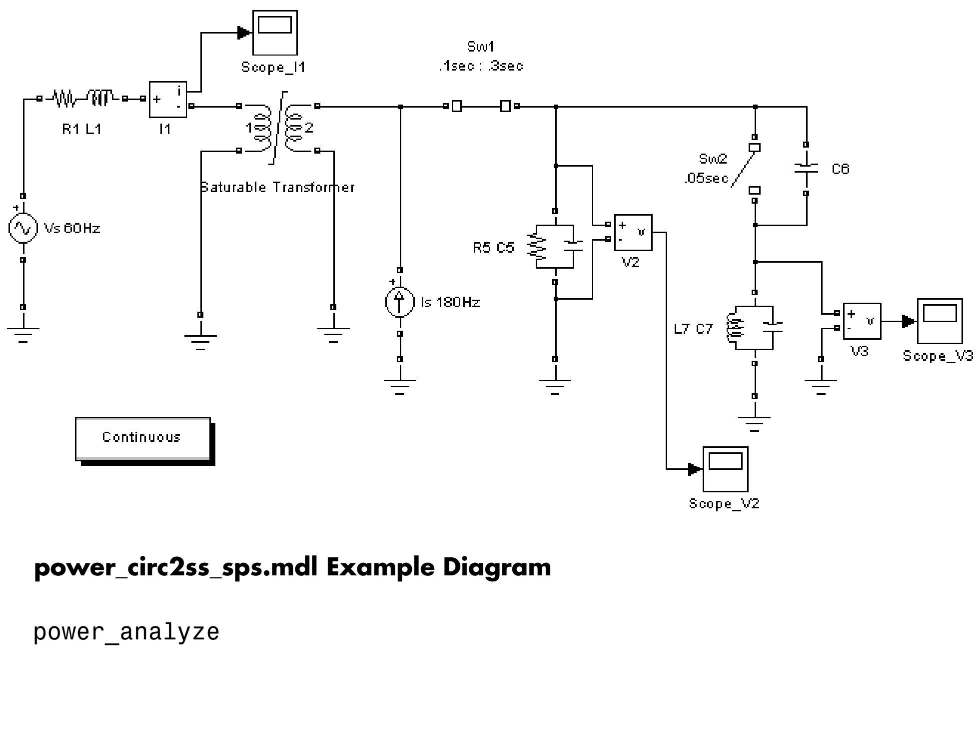
```
x0sw
x0sw =
   0.16155
   0
```

The Simulink diagram of the circuit shown in the following figure is available in the power_circ2ss_slk model. If no resistive switches had been used, the linear part of the circuit could have been simulated with the State-Space block of the Simulink/Continuous library. However, as resistive switches are used, the sfun_psbcontc S-function is used instead of the State-Space block. This S-function reevaluates the state-space matrices during simulation when the circuit topology is changing (after a switch is opened or closed). Appropriate inputs and outputs are used to connect the switch and saturable reactance models to the linear system. Notice that the status of each switch is fed back from the breaker to the S-function, after the inputs mentioned earlier. You can find the breaker and saturable_transformer blocks in the **powerlib_models/Continuous** library containing all the nonlinear continuous models used by SimPowerSystems. As the breaker model is vectorized, a single block is used to simulate the two switches Sw1 and Sw2.

If you use the **powerlib** library to build your circuit, the same Simulink system is generated automatically by the power_analyze command. The **powerlib** version of this system is also available in the power_circ2ss_sps model and is shown below.



power_circ2ss_slk.mdl Example Diagram



power_circ2ss_sps.mdl Example Diagram

See Also

`power_analyze`

Index

A

B

C

D

E