POWERSIM

PSIM®

User's Guide

Powersim Inc.

PSIM® User's Guide

Version 11.1

Release 2

November 2017

Disclaimer

Powersim Inc. ("Powersim") makes no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will Powersim or its direct or indirect suppliers be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the licence to the software and documentation.

Powersim Inc.

email: info@powersimtech.com
http://www.powersimtech.com

Contents

1 General Information

2 Circuit Schematic Design

3 Waveform Processing in SIMVIEW

4 Power Circuit Components

5 Control Circuit Components

1

General Information

1.1 Introduction

PSIM[1] is a simulation software specifically designed for power electronics, motor drives, and power conversion systems. With fast simulation speed and friendly user interface, PSIM provides a powerful simulation environment to meed your simulation and development needs.

PSIM includes the basic package, as well as the following add-on options:

Motor Drive Module: Built-in electric machine models and mechanical load models for motor drive system studies.

Digital Control Module: Discrete library elements such as zero-order hold, z-domain transfer function blocks, quantization blocks, digital filters, for digital control system analysis.

SimCoupler Module: Interface between PSIM and Matlab/Simulink[2] for co-simulation.

Thermal Module: Library elements and functions calculate semiconductor device losses and inductor losses.

Renewable Energy Module: Library elements, such as solar module, wind turbine, battery, and ultra-capacitor models, for renewable energy applications.

SPICE Module: SPICE engine, powered by CoolSPICE[8] from CoolCAD Electronics LLC., for SPICE simulation. It also includes the function to run LTspice[9].

SimCoder[3] Module: Function for automatic code generation.

F2833x Target: Library elements for automatic code generation for TI F2833x series DSP.

F2803x Target: Library elements for automatic code generation for TI F2803x series DSP.

F2802x Target: Library elements for automatic code generation for TI F2802x series DSP.

F2806x Target: Library elements for automatic code generation for TI F2806x series DSP.

PE-Expert4 Target: Library elements for automatic code generation for Myway PE-Expert4 hardware platform.

PIL Module: Interface between PSIM and TI DSP hardware boards for Processor-In-the-Loop (PIL) simulation. It also includes the function block to support TI's InstaSPIN motor control algorithm.

MagCoupler Module: Interface between PSIM and the electromagnetic field analysis software JMAG[4] for co-simulation.

MagCoupler-RT Module: Link between PSIM and JMAG-RT[4] data files.

[1] PSIM and SIMVIEW are registered trademarks of, and are copyright by, Powersim Inc., 2001-2017

[2] Matlab and Simulink are registered trademarks of the MathWorks, Inc.

[3] SimCoder is a trademark of Powersim Inc., and is copyright by Powersim Inc., 2008-2017

[4] JMAG and JMAG-RT are copyright by JSOL Corporation, 1997-2017

[5] ModCoupler and SmartCtrl are copyright by Universidad Carlos III de Madrid, 2011-2017

[6] ModelSim is a registered trademark of Mentor Graphics Co.

[7] CosiMate is copyright by ChiasTek Inc., 2010-2017

[8] CoolSPICE is copyright by CoolCAD Electronics LLC., 2011-2017

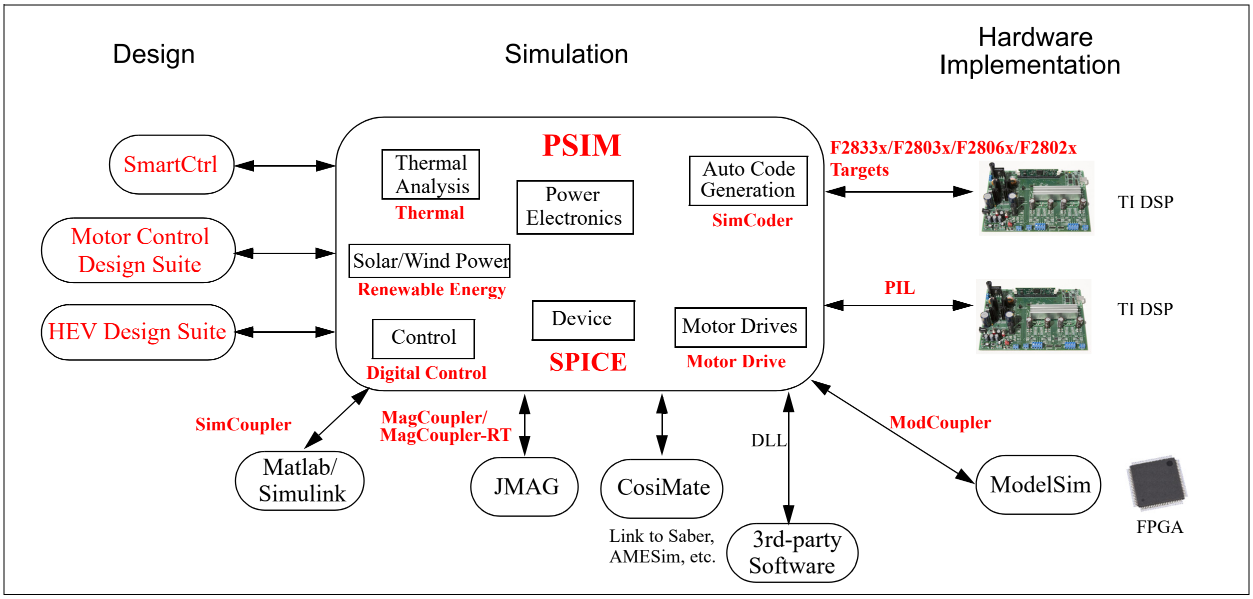
[9] LTspice is copyright by Linear Technology Co., 1998-2017

ModCoupler[5] Modules:	Interface between PSIM and ModelSim[6] for co-simulation. There are two versions of the interface: ModCoupler-VHDL that supports VHDL code, and ModCoupler-Verilog that supports Verilog code.
Motor Control Design Suite:	Pre-built templates for induction motor and linear/nonlinear PMSM drives.
HEV Design Suite:	Pre-built templates for hybrid electric vehicle (HEV) powertrain system design.

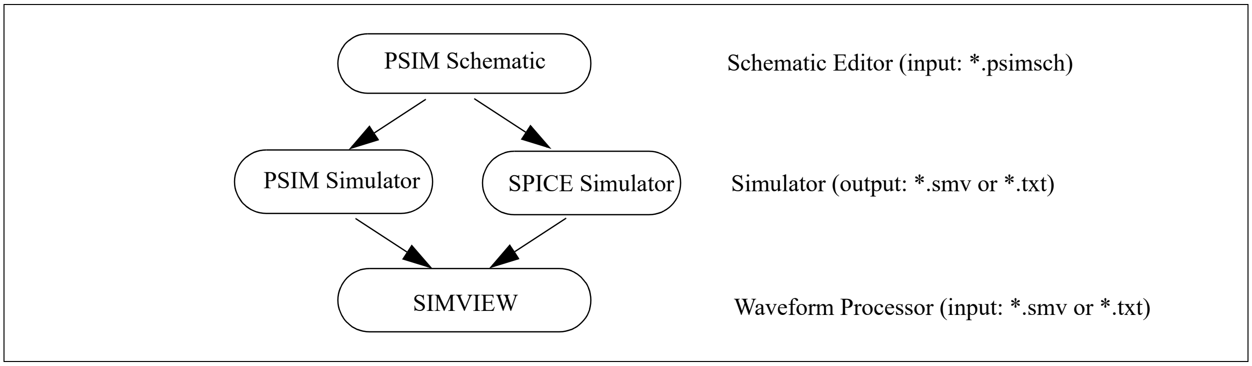
In addition, a link is provided between PSIM and CosiMate[7], a co-simulation framework that supports various software such as Matlab/Simulink, ModelSim, Saber (from Synopsys), Easy5 and Adams (from MSC Software), Inventor (from Autodesk), AMESim (from LMS), GT-Power (from Gamma Technologies), etc. Through CosiMate, PSIM can perform co-simulation with these software. For more information on CosiMate, visit www.chiastek.com.

Also, PSIM links with the software SmartCtrl[5] for control loop design. SmartCtrl is designed specifically for power converter applications. For more information on SmartCtrl, refer to *SmartCtrl User's Guide*.

With these product lineup, Powersim provides a complete platform from design to simulation, to hardware implementation. The overall environment is shown in the figure below.



The PSIM simulation environment consists of the schematic program PSIM, two simulator engines (one PSIM engine and the other SPICE engine), and the waveform processing program SIMVIEW[1]. The simulation process is illustrated as follows.


PSIM Schematic
Schematic Editor (input: *.psimsch)
PSIM Simulator
SPICE Simulator
Simulator (output: *.smv or *.txt)
SIMVIEW
Waveform Processor (input: *.smv or *.txt)

This manual covers both PSIM and all add-on Modules and subjects except the ones listed below which are covered by these documents respectively:

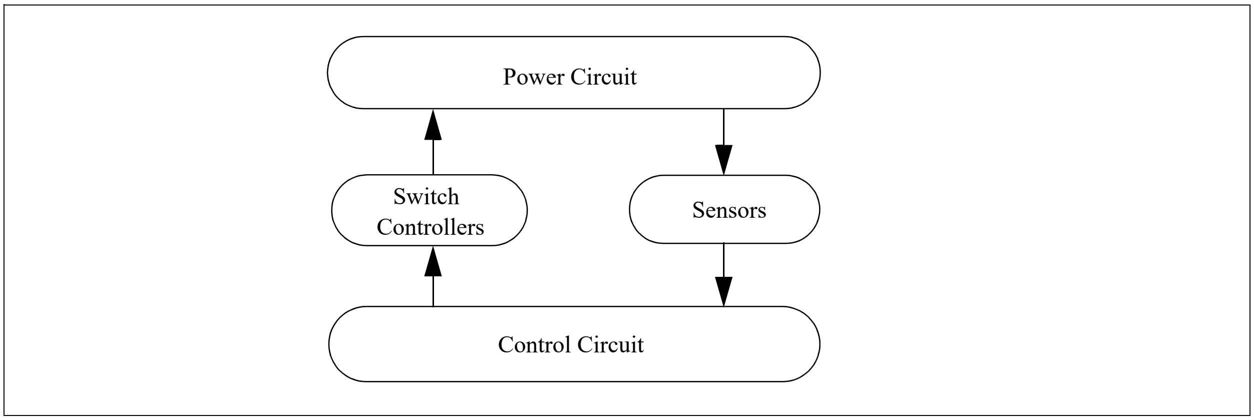
- For SimCoder and hardware Targets:
  SimCoder User Manual.pdf
- SPICE Module:
  SPICE User Manual.pdf
- PIL Module:
  Tutorial - Processor-In-the-Loop Simulation.pdf
- Motor Control Design Suite:
  Tutorial - Motor Control Design Suite.pdf
  Tutorial - PMSM drive with sensorless control.pdf
- HEV Design Suite:
  Tutorial - HEV Design Suite.pdf
- Motor idenfitication, InstaSPIN, and sensorless motor drives:
  Toturial - Motor parameter identification with PSIM.pdf
  Tutorial - Simulation and code generation of TI InstaSPIN using DRV8305 EVM.pdf

The organization of this manual is as follows:

Chapter 1:	PSIM circuit structure, software/hardware requirement, and parameter format.
Chapter 2:	PSIM environment and how to build a PSIM schematic.
Chapter 3:	Simulation result display and analysis with Simview.
Chapter 4 through 7:	Components in PSIM element library.
Chapter 8:	Error/warning messages

1.2 Circuit Structure

A circuit is represented in PSIM in four blocks: power circuit, control circuit, sensors, and switch controllers. The figure below shows the relationship between these blocks.



The power circuit consists of switching devices, RLC branches, transformers, and coupled inductors. The control circuit is represented in block diagram. Components in s-domain and z-domain, logic components (such as logic gates and flip flops), and nonlinear components (such as multipliers and dividers) are used in the control circuit. Sensors are used to measure power circuit quantities and pass them to the control circuit. Gating signals are then generated from the control circuit and sent back to the power circuit through switch controllers to control switches.

1.3 Software/Hardware Requirement

PSIM runs in Microsoft Windows 7/8/10 on personal computers. A minimum of 1GB RAM memory is needed.

1.4 Installing the Program

A quick installation guide is provided in the flier “PSIM - Quick Guide” and on the USB memory stick.

Some of the files in the PSIM directory are:

PSIM.exe	PSIM circuit schematic editor
SIMVIEW.exe	PSIM waveform processing program
PcdEditor.exe	Device database editor
SetSimPath.exe	Program to set up the SimCoupler Module

File extensions used in PSIM are:

*.psimsch	PSIM schematic file
*.psimpjt	PSIM project file
*.lib	PSIM library file
*.fra	PSIM ac analysis output file (text)
*.dev	Device database file
*.txt	Simulation output file in text format
*.smv	Simulation output file in binary format
*.net; *.cir	SPICE netlist files

1.5 Simulating a Circuit

To simulate the buck converter circuit “buck.psimsch” in "examples\dc-dc":

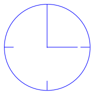
- Start PSIM. From the **File** menu, choose **Open** to load the file “buck.psimsch”.
- From the **Simulate** menu, choose **Run PSIM** to start the simulation. Simulation results will be saved to File “buck.smv”.
- By default, **Auto-run SIMVIEW** is selected in the **Options** menu. SIMVIEW will be launched automatically. In SIMVIEW, select curves for display. If this option is not selected, from the **Simulate** menu, choose **Run SIMVIEW** to start SIMVIEW.

1.6 Simulation Control

The Simulation Control element defines parameters and settings related to simulation.

To place the Simulation Control in the schematic, go to the **Simulate** menu, and select **Simulation Control**.

Image:



There are four tabs in the Simulation Control dialog:

PSIM: Define parameters for PSIM transient simulation.

SPICE: Define the analysis type and parameters for SPICE simulation. For more detailed information, please refer to *SPICE User’s Manual*.

SimCoder: Define the hardware for SimCoder simulation and automatic code generation. For more detailed information, please refer to *SimCoder User’s Manual*.

Color: Define the color of the Simulation Control image. The default color is blue.

PSIM Tab:

Time Step	Simulation time step, in sec.
Total Time	Total simulation time, in sec.
Free Run checkbox	When the *Free Run* checkbox is not checked, the simulation will run up to the *Total Time* and then stop. But when it is checked, the simulation will run in the free-run mode and it will keep on running until manually stopped. In the free-run mode, voltage/current scopes can be used to monitor and display voltages and currents in the middle of the simulation.
Print Time	Time from which simulation results are saved to the output file. No output is saved before this time.
Print Step	Print step. If it is set to 1, for example, every data point will be saved to the output file. If it is set to 10, only one out of 10 data points will be saved. This helps to reduce the output file size.
Load Flag	Flag for the LOAD function. If the flag is 1, the previous simulation values (saved by setting the *Save Flag*) will be loaded from a file (with the ".ssf" extension) as the initial conditions.
Save Flag	Flag for the SAVE function. If the flag is 1, values at the end of the current simulation will be saved to a file with the ".ssf" extension.

In PSIM, the simulation time step is fixed throughout the simulation. In order to ensure accurate simulation results, the time step must be chosen properly. The factors that limit the time step in a circuit include the switching period, widths of pulses or waveforms, and intervals of transients. It is recommended that the time step should be at least one magnitude smaller than the smallest of the above.

Also, an interpolation technique is implemented which will calculate the switching instants more accurately. With this technique, the error due to the misalignment of switching instants and discrete simulation points is significantly reduced. It is possible to simulate with a large time step while still maintaining accurate results.

The allowable maximum time step is automatically calculated in PSIM. It is compared with the time step set by the user, and the smaller value of the two will be used in the simulation.

With the SAVE and LOAD functions, the circuit voltages, currents and other quantities can be saved at the end of a simulation session, and loaded back as the initial conditions for the next simulation session. This provides the flexibility of running a long simulation in several shorter stages with different time steps and parameters. Components values and parameters of the circuit can be changed from one simulation session to the other. The circuit topology, however, must remain the same.

SPICE Tab:

There are 3 types of analysis in SPICE: *Transient*, *AC*, and *DC*. Simulation parameters of different analysis types are described below. For more information on SPICE simulation, please refer to the SPICE User Manual.

For Transient Analysis:

Parameters	The following parameters are defined: *Use Initial Conditions*: If this box is checked, element initial conditions will be used, and the "UIC" option will be added to the .tran command. *Time Step*: Suggested computing increment and plotting increment Tstep, in sec. It can be left undefined (blank). *Max Step*: The maximum step size Tmax that SPICE uses, in sec. By default, the program uses either Tstep or (Tend-Tstart)/50, whichever is smaller. Tmax is useful when one wishes to guarantee a computing interval not greater than Tmax. It can be left undefined. *Start Time*: The initial time Tstart, in sec. The transient analysis always begins at time zero. In the interval from zero to Tstart, the circuit is simulated, but no outputs are stored. This parameter is equivalent to *Print Time* in PSIM simulation parameters. It can be left undefined. *End Time*: The final simulation time Tend, in sec.
Integration Method	Three integration methods can be selected for SPICE simulation: *Trapezoidal*: Trapezoidal Rule method. *Modified Trapezoidal*: Modified Trapezoidal Rule method. *Gear*: Gear method. *Order*: The order of the integration method. For Trapezoidal, the order can be 1 or 2. For Gear, the order can be 2 to 6.

For AC Analysis:

Parameters	The following parameters are defined: *Start Freq*: Start frequency, in Hz *End Freq*: End frequency, in Hz *Dec*: Number of points per decade. If octive or linear variations are desired, one may use the SPICE Directive Block to write the analysis command.

For DC Analysis:

Parameters	Source 1 is used by default. Source 2 can be enabled by checking the checkbox. The following parameters are defined: *Voltage/Current*: Specify if the source is a voltage source or current source *Start*: Starting value, in V or A *Step*: Incremental value, in V or A *End*: End value, in V or A

The rest of the parameters for SPICE simulation are described below.

Operating Point	If the *Enable* box is checked, SPICE simulation will determine the dc operating point of a circuit with inductors shorted and capacitors opened.
Thermal Analysis (CoolSPICE only)	If the *Enable* box is checked, SPICE simulation will perform thermal analysis by including the heating effect in device models. All CoolSPICE SiC and GaN models support thermal analysis. With the thermal analysis, the extra node of SiC/GaN models will show the juction temperature of the devices in ^{o}C. Note that this option is for CoolSPICE simulation only.

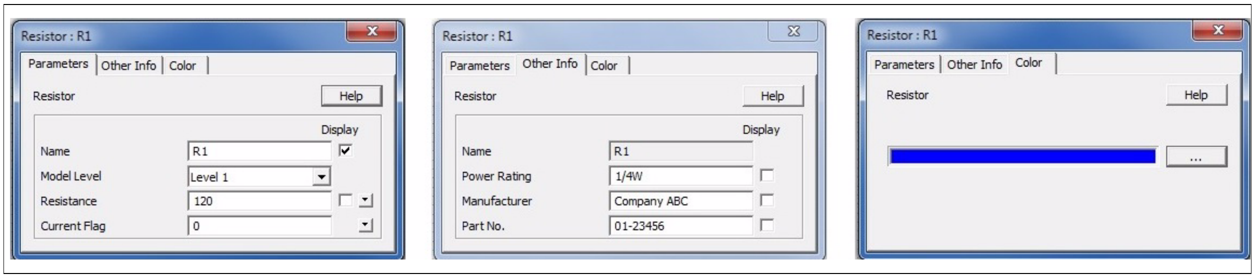
Step Run Option	If the *Enabled* box is checked, SPICE simulation will perform parameter sweep. The parameter sweep definition is as follows: *Parameter*: The name of the parameter to be swept *Start*: Starting value *Step*: Incremental value *End*: Final value
Error Tolerance Option	If the *Enabled* box is checked, error tolerances for SPICE simulation can be changed. Otherwise, default values will be used. Error tolerances are: *RELTOL*: Relative tolerance *TRTOL*: Transient tolerance *VNTOL*: Absolute voltage error tolerance *ABSTOL*: Absolute current error tolerance *CHGTOL*: Charge tolerance

SimCoder Tab:

Hardware Target	The hardware target can be one of the following: *None*: No hardware target in the circuit *F2833x*: F2833x Hardware Target for TI F2833x series DSP F2802x: F2802x Hardware Target for TI F2802x series DSP F2803x: F2803x Hardware Target for TI F2803x series DSP F2806x: F2806x Hardware Target for TI F2806x series DSP
Memory Map Options	Specify the memory map for compiler. For F2833x and F2803x hardware target: RAM Debug; RAM Release; Flash Release; and Flash RAM Release. For PE_Exper3 hardware target: PE-View9 PE-View8
CPU Version	Specify the CPU version. For F2833x: F28335, 28334, and 28332. For F2803x: F28035, 28034, 28033, 28032, 28031, and 28030 For F2802x: F28027, 28026, 28023, 28022, 28021, 28020, and 280200 For F2806x: F28069, 28068, 28067, 28066, 28065, 28064, 28063, and 28062.
InstaSPIN enabled	If the DSP is InstaSPIN enabled (for example F28069M), this box must be checked. Otherwise it should be unchecked.
Default Data Type	This parameter is for fixed-point DSPs, such as F2803x. The default data type options are: Integer, IQ0, IQ1, ... IQ30. If the box for Check Fixed-Point Range is checked, the SimCoder will check all the variables against the range and display the result.
DMC Library Version	SimCoder has function blocks of all the functions in TI’s DMC library for the following DMC versions: 4.0, 4.1, and 4.2.
Comments	Comments can be entered and these comments will be inserted at the beginning of the automatically generated code.

1.7 Component Parameter Specification and Format

The parameter dialog window of each component in PSIM has three tabs: **Parameters**, **Other Info**, and **Color,** as shown below.



Name	R1	☑
Model Level	Level 1	
Resistance	120	□
Current Flag	0	

Name	R1	
Power Rating	1/4W	□
Manufacturer	Company ABC	□
Part No.	01-23456	□

The parameters in the **Parameters** tab are used in the simulation. The information in the **Other Info** tab, on the other hand, is not used in the simulation. It is for reporting purposes only and will appear in the parts list in **View** >> **Element List** in PSIM. Information such as device rating, manufacturer, and part number can be stored under the **Other Info** tab.

The component color can be set in the **Color** tab.

Parameters under the **Parameters** tab can be a numerical value or a mathematical expression. A resistance, for example, can be specified in one of the following ways:

12.5
12.5k
12.5Ohm
12.5kOhm
25./2.Ohm
R1+R2
R1*0.5+(Vo+0.7)/Io

where R1, R2, Vo, and Io are symbols defined either in a parameter file (see Section 4.1), or in a main circuit if this resistor is in a subcircuit (see Section 6.3.4.1).

Power-of-ten suffix letters are allowed in PSIM. The following suffix letters are supported:

G	10^9
M	10^6
k or K	10^3
m	10^{-3}
u	10^{-6}
n	10^{-9}
p	10^{-12}

A mathematical expression can contain brackets and is not case sensitive. The following mathematical functions are allowed:

+	addition
-	subtraction
*	multiplication
/	division
^	to the power of [Example: 2^3 = 2*2*2]
sin(x)	// sine
cos(x)	// cosine
tan(x)	// tangent
asin(x)	// arcsine
acos(x)	// arccosine
atan(x)	// arctangent
atan2(y,x)	// arctangent with x and y defined
sinh(x)	// hyperbolic sine

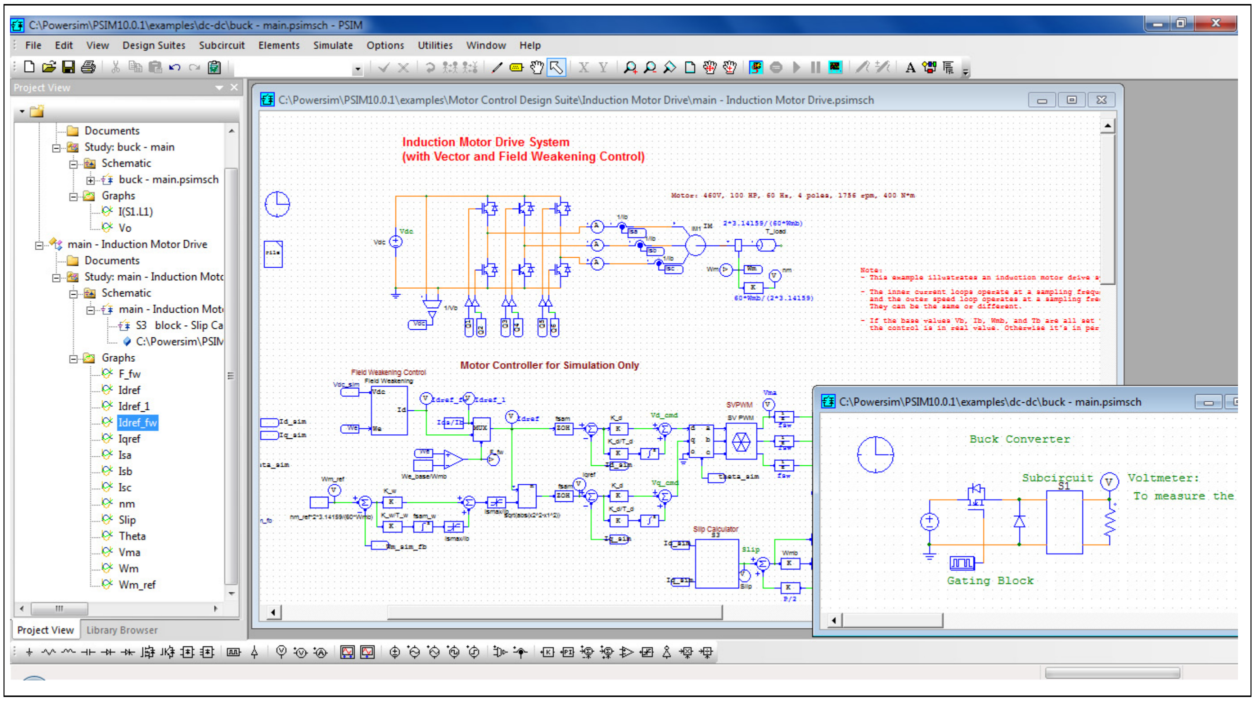
```
cosh(x)             // hyperbolic cosine
tanh(x)             // hyperbolic tangent
pow(x,y)            // x to the power of y
sqrt(x)             // square root
exp(x)              // base-e exponential of x, i.e. e^x
ln(x) (or log(x))   // natural logarithm of x (base e)
log10(x)            // common logarithm of x (base 10)
abs(x)              // absolute
sign(x)             // sign function that returns 1 if x > 0, -1 if x < 0, and 0 if x = 0
ceil(x)             // function that returns the integer larger than x
floor(x)            // function that returns the integer smaller than x
```

2
Circuit Schematic Design

PSIM's schematic program provides interactive and user-friendly interface for circuit schematic entry and editing. The PSIM user interface consists of an integrated set of windows, tools, menus, toolbars, and other elements that allow you to create, simulate, and refine your circuits in one places.

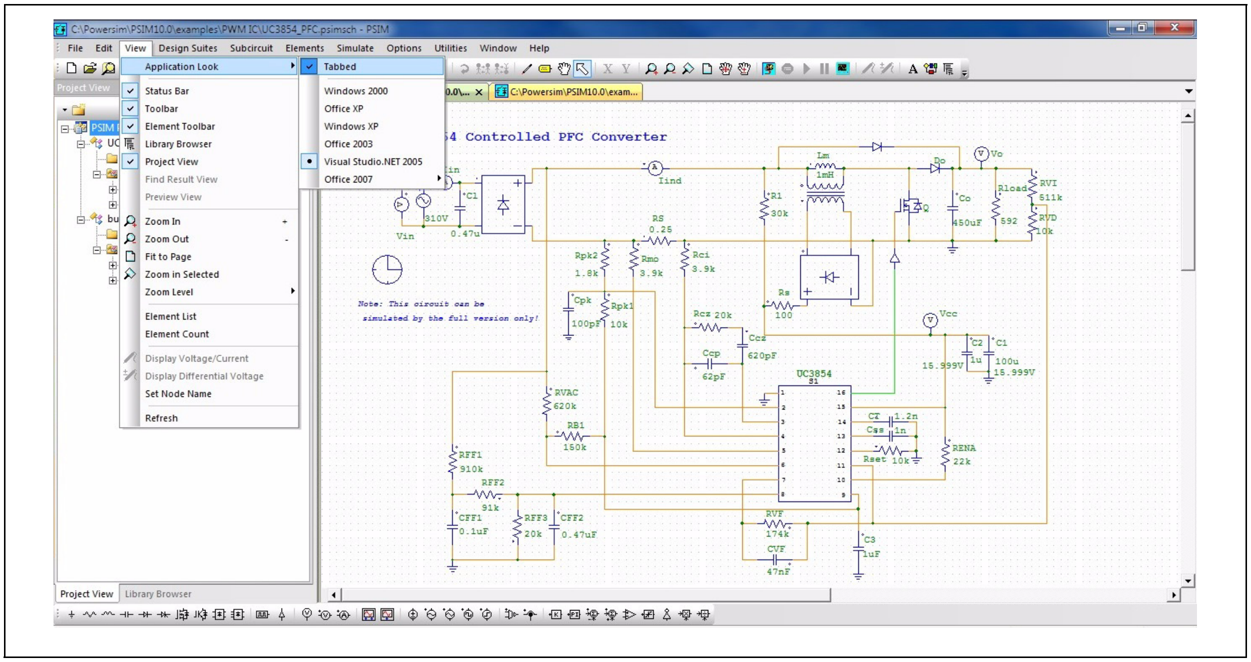
2.1 PSIM Environment

The following figure shows typical screen display of PSIM environment. In the figure, to illustrate as examples, two PSIM circuit files are open: a boost power factor correction circuit and a dc-dc buck converter circuit.


C:\Powersim\PSIM10.0.1\examples\dc-dc\buck - main.psimsch - PSIM
File
Edit
View
Design Suites
Subcircuit
Elements
Simulate
Options
Utilities
Window
Help
Project View
Documents
Study: buck - main
Schematic
buck - main.psimsch
Graphs
I(S1.L1)
Vo
main - Induction Motor Drive
Documents
Study: main - Induction Moto
Schematic
main - Induction Moto
S3 block - Slip Ca
C:\Powersim\PSIM
Graphs
F_fw
Idref
Idref_1
Idref_fw
Iqref
Isa
Isb
Isc
nm
Slip
Theta
Vma
Wm
Wm_ref
Project View
Library Browser
C:\Powersim\PSIM10.0.1\examples\Motor Control Design Suite\Induction Motor Drive\main - Induction Motor Drive.psimsch
Induction Motor Drive System
(with Vector and Field Weakening Control)
Motor: 460V, 100 HP, 60 Hz, 4 poles, 1736 rpm, 400 N*m
Motor Controller for Simulation Only
Field Weakening Control
Slip Calculator
SVPWM
C:\Powersim\PSIM10.0.1\examples\dc-dc\buck - main.psimsch
Buck Converter
Subcircuit
S1
Voltmeter:
To measure the
Gating Block

By default, the menu bar and the standard toolbar appear on top of the window, while the frequently used element bar appears at the bottom, and the Project View is on the left hand side.

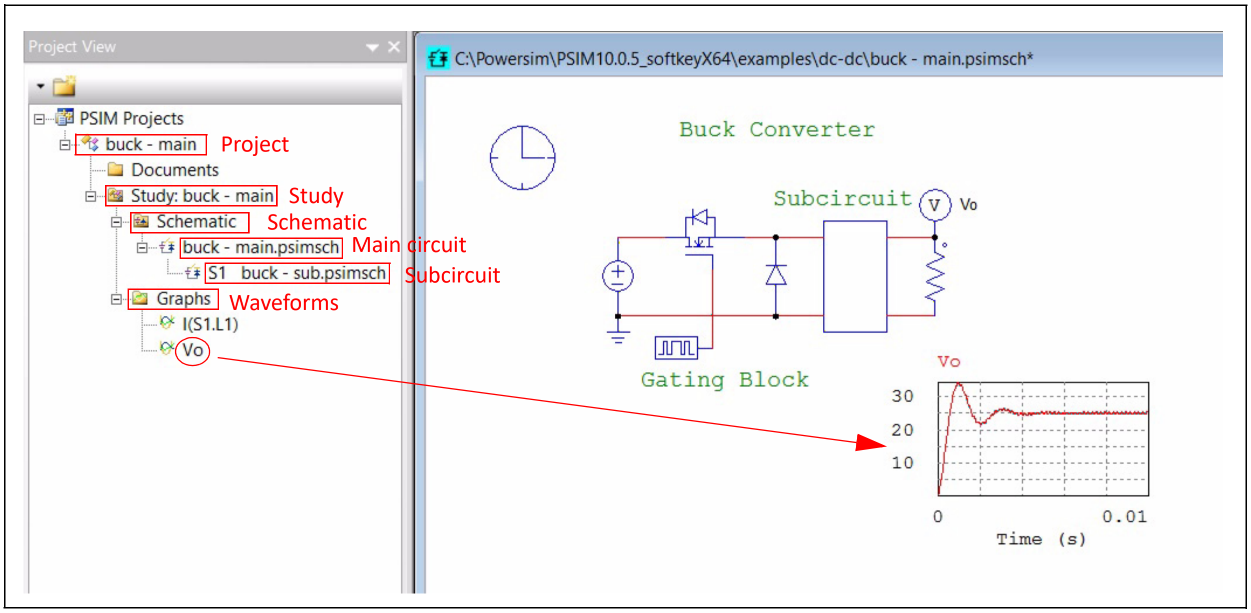
On the right hand side is the Design window. This is a graphic editor where users can build and editor their simulation circuit schematics. User may arrange schematics in the Design window in tiles (as shown above), or in tabs (as shown below).



The Project View window provides an organized tree view of user’s projects and their related files, as well as the simulation result graphs. Each PSIM circuit is treated as a project. The following content are displayed in stacking tiers in the project view:

Project Name:	Usually this is the same as the top level PSIM circuit file name.
Documents:	Any files related to the project, for example document, datasheet, etc.
Study:	Name of the study
Schematic:	The PSIM schematic files, top level and subcircuits.
Graphs:	All probes in the PSIM schematic are included in the graph list.

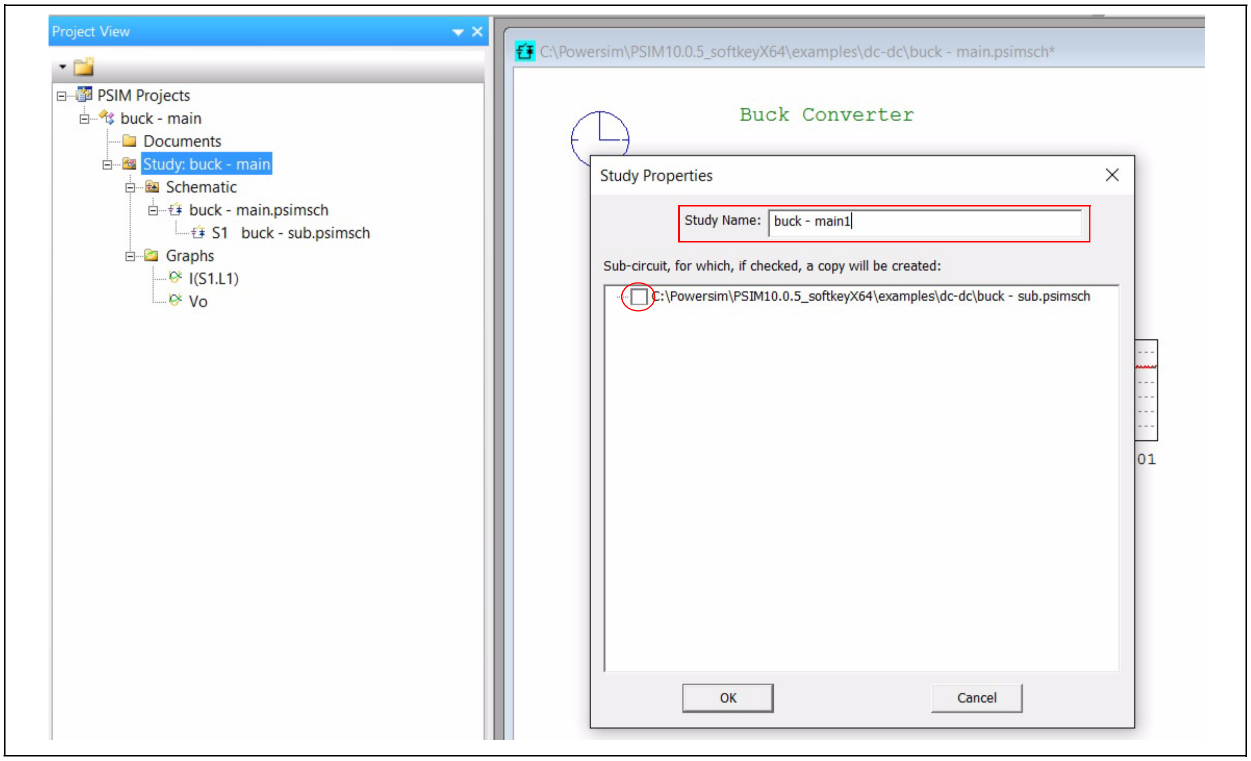
The following shows a buck converter example in the Project View.



In this example, the project is the buck converter. It contains one study. The main circuit is "buck - main.psimsch", and the subcircuit is "buck - sub.psimsch". There are two simulation waveforms: I(S1.L1) and Vo.

After the simulation is done, the waveforms I(S1.L1) and Vo can be loaded into Simview by simply double clicking on the waveform names in Project View. The waveforms can also be embedded into the schematic by dragging into the schematic. For example, the figure above shows the Vo waveform embedded in the schematic.

A project may contain multiple studies. For example, one may study the buck converter circuit with a different load filter or switching frequency. To create another study out of an existing study, right click on the existing study "buck - main" and select **Create Study copy**. A dialog window will appear as shown below to allow you to define the new study name and determine if you wish to make a copy of the subcircuit.



C:\Powersim\PSIM10.0.5_softkeyX64\examples\dc-dc\buck - main.psimsch*

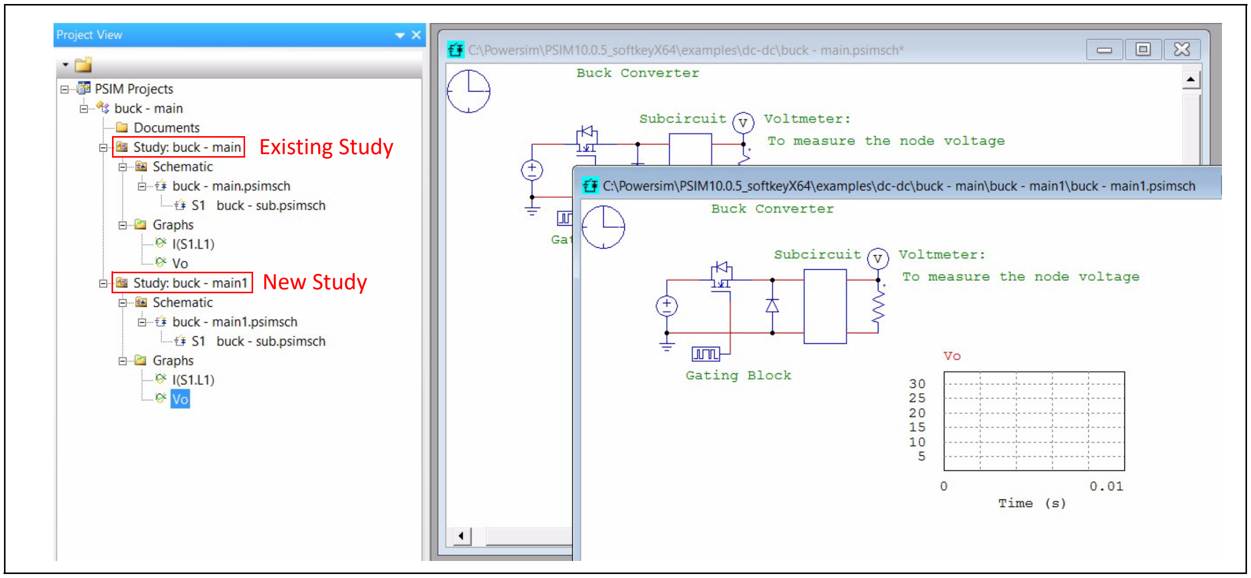
Study Properties

Study Name: buck - main1

Sub-circuit, for which, if checked, a copy will be created:

□ C:\Powersim\PSIM10.0.5_softkeyX64\examples\dc-dc\buck - sub.psimsch

In this example, we will create a new study called "buck - main1" and create a copy of the subcircuit. After confirmation, the new study is created as shown below. The new files "buck - main1.psimsch" and "buck - sub.psimsch" are placed in a newly created folder "buck - main\buck - main1".



C:\Powersim\PSIM10.0.5_softkeyX64\examples\dc-dc\buck - main.psimsch*

C:\Powersim\PSIM10.0.5_softkeyX64\examples\dc-dc\buck - main\buck - main1\buck - main1.psimsch

To add an existing schematic file into the project as another study, right click on the project "buck - main" and select **Add Study**. To save the project, right click on the project "buck - main" and select **Save Project**.

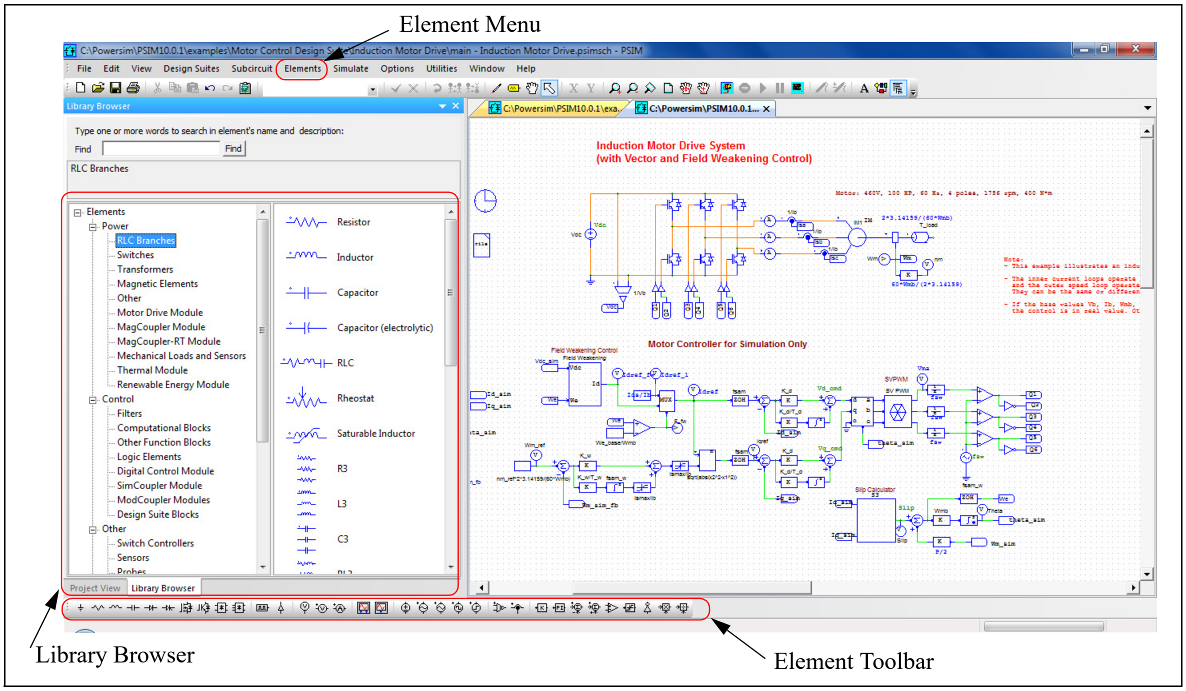
2.2 Creating a Circuit

The basic and most commonly used functions provided for circuit creation are:

Get Element There are several ways to get an element from the element library. One is to use the pull-down menu. Go to the **Elements** menu, and go into the submenu and highlight the element to be selected.

The most often used elements can be selected the **Element Toolbar.** It is located at the bottom of the PSIM screen by default.

Another way is to use the **Library Browser,** as shown below. The Library Browser provides a convenient way of navigating through the library. To launch the Library Browser, go to **View** >> **Library Browser**.



Place Once an element is selected from the menu, the image of the element will appear on the screen and move with the mouse. Click the left button of the mouse to place the element at desired location on schematic.

Select Element(s) To select an existing element on a schematic, click on the element. A rectangle will appear around the element. To select a section of a circuit, keep the left button of a mouse pressed and drag the mouse until the rectangle covers the selected area.

Rotate Before the element is placed, right click to rotate the element. After an element is selected, select **Edit** >> **Rotate** to rotate the element.

Wire To connect a wire between two nodes, select **Edit** >> **Wire**. The image of a pen will appear on the screen. To draw a wire, keep the left button of the mouse pressed and drag the mouse. A wire always starts from and end at a grid intersection.

For easy inspection, a floating node is displayed as a circle, and a junction node is displayed as a solid dot.

Label If two or more nodes are connected to the same label, they are connected. It is equivalent as though they were connected by wire. Using labels will reduce the cross-wiring and improve the schematic layout.

The text of a label can be moved. To select the text, left click on the label, then press the **Tab** key.

Assign To assign the parameters of an element, double click on the element. A dialog box will appear. Specify the values and hit the <Return> key or click on **OK**.

Move To move an element or a circuit block, select the element/circuit block and drag the mouse while keeping the left button pressed.

Pan Schematic To scroll schematic, right click and drag the mouse.

2.3 File Menu

The following functions are provided in the **File** menu for various file operations:

New Create a new schematic with a single worksheet with no size limit.

New (worksheet) Create a new schematic with a predefined worksheet size.

Open Open an existing schematic file.

Open Examples Open sample schematic files that come with PSIM.

Change Worksheet Size Change the worksheet size.

New SPICE Netlist File Create a new SPICE netlist file

Open SPICE Netlist File Open an existing SPICE netlist file

New Project Create a new PSIM project.

Open Project Open an existing PSIM project.

Save Project Save current PSIM project.

Close Close the current schematic file.

Close All Close all schematic files.

Save Save the current schematic file.

Save As... Save the current schematic file to a different name.

Save All Save all schematic files.

Save with Password Save a schematic file so that it is protected with a password. When a file is password protected, it can still be used in the simulation, but one needs to enter the correct password in order to see the schematic. The password protection is used in situations where the person who created the file needs to share it with someone else, but does not wish to reveal the details of the schematic.

Save in Package File Save a schematic file and all associated files to one single package file. This is especially useful if the main circuit calls multiple subcircuits, and one needs to send the files to someone else. Rather than finding and collecting all the subcircuit files, one can just create the package file and send out the single package file.

Save as Older Versions Save a file in the previous version format. Note that if the file uses elements that are unique in the current version, these elements will be omitted.

Print Print the schematic. Note that the schematic is printed as it appears on the screen. If you zoom in or out the schematic, the printout will be changed accordingly.

Print Preview Preview the printout.

Print Selected Print only a portion of the schematic selected.

Print Selected Preview Preview the printout of the portion of the schematic selected.

Print Page Setup Adjust the print page position and set the print page legend.

Printer Setup Set up the printer.

Exit Exit the PSIM schematic program.

2.4 Edit Menu

The following functions are provided in the **Edit** menu for circuit editing:

Undo Undo the previous change.

Redo Go back to the state before undoing the changes.

Cut Cut the selected circuit out of the schematic. The circuit that is cut can be pasted back.

To delete an element or a portion of the circuit, select the item and hit <Delete> key.

Copy Copy an element or a portion of a circuit into a buffer, which can then be pasted back.

Paste Paste back the copied element or circuit.

Select Matched Elements Select the elements which matches the specification.

Select All Select the entire schematic.

To select only a portion of the schematic, left click and drag the mouse.

Copy to Clipboard Copy the schematic image to the clipboard which can then be pasted back in another software. One can choose one of the three options: *Metafile Format*, *Color Bitmap*, or *Black and White* bitmap. The metafile format is vector based, and gives better image quality especially when the image is resized. The Black & White option will result in a smaller image file size as compared to the color bitmap.

Draw Draw images on the schematic for display purposes. The following images are provided: line, ellipse, rectangle, half-circles, bitmap images, and graph.

To draw a bitmap image: left click the mouse and drag the mouse to define the area that will contain the bitmap image. Then select the bitmap file.

To draw a graph, left click the mount and drag the mouse to define the area that will display the waveform of selected probe.

Change All Text Font Change the font for all the text in the opened PSIM file

Change all Text Link Font Change the font for all the text link in the opened PSIM file.

Place Text To place text on the screen, choose **Text**. Enter the text in the dialog box, and click the left button of the mouse to place it.

Place Wire Enter the wiring mode. The cursor will change to the shape of a pen.

Place Label Place a label on the schematic. When two nodes are connected to two labels of the same name, they are considered physically connected.

Edit Attributes When an element is selected, choose **Attributes** to bring out the property dialog window.

Add/Remove Current Scope Add or remove the current scope for an element that has the current flag. After this function is selected, click on top of the element, and select the branch current name to display the current scope. Select the branch current name again to remove the current scope.

Show/Hide Runtime Variables Show or hide the parameters of elements that can be changed at runtime in the middle of the simulation. After this is selected, the text of the parameter will appear. Double click on the text, and a small dialog window will appear. Enter the new value directly in the data field, and click Apply. Or alternatively, click on the up/down arrow keys on the keyboard to increase/decrease the value.

Disable Disable an element or part of a circuit. When the element or the circuit is disabled, it will be grayed out and will be treated as non-existent as far as the simulation is concerned. This function is useful if an element or circuit needs to be excluded but not deleted from the circuit.

Enable Enable a previously disabled element or circuit.

Rotate Rotate the selected element or a portion of the circuit by 90^{o} clockwise.

Flip Left/Right Flip the selected element horizontally.

Flip Top/Bottom Flip the selected element vertically.

Find Find a particular element based on type and name.

Find Next Repeat the previous **Find** operation.

Find in Files Fined a particular element in several files.

Edit Library Edit PSIM image libraries. For more details, please refer to Section 2.12.

Image Editor Launch the image editor. For more details, please refer to Section 2.13.

Escape Quit from any of the above editing modes by choosing **Escape**.

2.5 View Menu

The following additional functions are provided in the **View** menu for circuit editing:

Application Look Select the display style of the PSIM windows.

Status Bar Show/hide the status bar.

Toolbar Show/hide the toolbar.

Element Bar Show/hide the element bar.

Library Browser Launch the Library Browser. The Library Browser is another way of accessing the PSIM Element library.

Project View Launch the Project View. The project view organizes and manages the related files. The projects organizing structure has been illustrated in Section 2.1

Zoom In Zoom in the schematic.

Zoom Out Zoom out the schematic.

Fit to Page Adjust the zooming so that the entire schematic fits the screen.

Zoom In Selected Zoom in to the selected area.

Zoom Level Zoom the schematic to 10%, 20%, ..., 200%, and custom size.

Element List Generate the parts list of the circuit.

Element Count Count the number of elements. Voltage/current probes and meters are not included in the element count.

SPICE Model List List all the SPICE models defined by the .model statement in the files in all the folders under the SPICE Model Path.

SPICE Subcircuit List List all the SPICE models defined by the .subckt statement in the files in all the folders under the SPICE Model Path.

Display Voltage/Current If the option *Save all voltages and currents* (under **Options** >> **Settings** >> **General**) is checked, after the simulation is complete, choose this function to display any node voltages or branch currents.

Display Differential Voltage With the option *Save all voltages and currents* checked, after the simulation is complete, choose this function to display any voltages between two nodes.

Set Node Name With the option *Save all voltages and currents* checked, after the simulation is complete, choose this function to define the name of a node so that the displayed voltage waveform has the specified name.

Refresh Refresh the screen display.

2.6 Design Suites Menu

Functions are provided in the **Design Suites** menu to run two design suite templates which generate electrical circuit schematics automatically.

The following functions are provided:

Update Parameters Update the parameters in the circuit generated by design suite

Show Design File Display the system in design template form

Display Parameters Display the parameters of the system

HEV Design Suite Run the design templates in the HEV Design Suite. Four design templates are provided: HEV Powertrain System, PHEV (plug-in hybrid electric vehicle) Powertrain System, HEV Generator, and HEV Traction Motor. Each template has its linear and nonlinear version.

Motor Control Design Suite Run the design templates in the Motor Control Design Suite. Five design templates are provided: PMSM (IPM) Drive, PMSM(IPM) Drive (Nonlinear), PMSM(SPM) Drive, PMSM Drive, and Induction Motor Drive. For more information about how to use Motor Control Design Suite, refer to the document "Tutorial - Motor Control Design Suite.pdf".

2.7 Subcircuit Menu

Functions are provided in the **Subcircuit** menu for subcircuit editing and manipulation.

The following functions are to be performed in the parent circuit (outside the subcircuit):

New Subcircuit	Create a new subcircuit
Load Subcircuit	Load an existing subcircuit. The subcircuit will appear on the screen as a block.
Edit Subcircuit	Edit the size and file name of the subcircuit
Display Subcircuit Name	Display the name of a selected subcircuit in the main circuit
Show Subcircuit Ports	Display the port names of the subcircuit in the main circuit
Hide Subcircuit Ports	Hide the port names of the subcircuit in the main circuit
Subcircuit List	List the file names of the main circuit and the subcircuits

The following functions are to be performed inside the subcircuit:

Set Size	Set the size of the subcircuit
Place Bi-directional Port	Place a bi-directional connection port in the subcircuit
Place Input Signal Port	Place an input signal connection port in the subcircuit
Place Output Signal Port	Place an output signal connection port in the subcircuit
Display Port	Display the connection port of the subcircuit
Edit Default Variable List	Edit the default variable list of the subcircuit. This is done inside the subcircuit.
Edit Image	Edit the subcircuit image
One Page up	Go back to the main circuit. The subcircuit is automatically saved.
Top Page	Jump from a lower-level subcircuit to the top-level main circuit. This is useful for circuits with multiple layers of subcircuits.

If the functions **Set Size**, **Display Port**, **Edit Default Variable List**, and **Edit Image** are performed in the main circuit, they will be applied to the main circuit instead.

There are three types of subcircuit ports for signal interface with the main circuit: Bi-directional port for power circuit and mechanical systems, and input signal and output signal ports for control circuit.

Even though bi-directional ports also work for control circuit, it is strongly recommended to use input or output signal ports for control circuit for better clarity. Furthermore, if a subcircuit is involved in code generation, only input or output signal ports can be used.

Right click on top of the subcircuit block, and choose **Attributes** to display the subcircuit property dialog window. There are three tabs: *Subcircuit Info*, *Subcircuit Variables*, and *Color*.

In the *Subcircuit Info* Tab:

In this tab, the name of the subcircuit can be edited. click on the **Change Subcircuit File** button to change to a different subcircuit. The selected subcircuit file will be loaded instead.

In the *Subcircuit Variables* Tab:

In this tab, variables used in the subcircuit can be edited. For example, a resistor in the circuit has the resistance defined as "Rparasitic", and for better clarity, this resistance is referred to as "Parasitic Resistance". Also, the resistance has a value of 1mOhm. This variable will be entered as:

Variable Description:Parasitic Resistance
Variable Name:Rparasitic
Variable Value:1m

When the checkbox for "Parasitic Resistance" is checked, in the main circuit, this variable will be displayed as:

Parasitic Resistance = 1m

Since subcircuit variable list can be edited, the current variable list may be different from the default variable list. One can click on the **Set as Default Variables** button to set the current variable list as the default list, or click on the **Reload Default Variables** button to reload the default list if the default list has been modified.

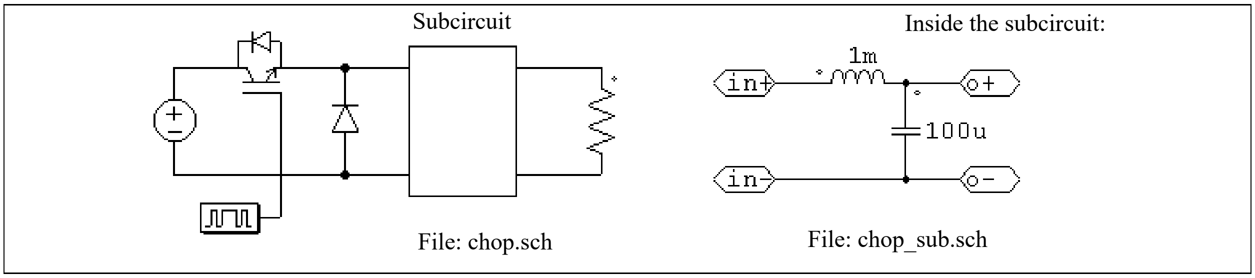
Two functions are provided at the bottom of the dialog for SimCoder for automatic code generation. Click on the **Generate Code** button to generate code for this subcircuit. If the checkbox **Replace subcircuit with generated code for simulation** is checked, the schematic inside the subcircuit will be replaced by the generated code for simulation.

In the *Color* Tab: In this tab, the subcircuit color can be changed.

Example: Use of Subcircuit

The circuit below illustrates the use of subcircuit. The circuit on the left is a buck converter, with the L-C filter inside a subcircuit. The content of the subcircuit is shown on the right.

In this example, there are two bi-directional ports ("in+" and "in-") on the left, and two bi-directional ports ("o+" and "o-") on the right.



2.7.1 Creating Subcircuit - In the Main Circuit

The following are the steps to create the subcircuit “chop_sub.sch” in the main circuit “chop.sch”.

- Open or create the main circuit “chop.sch”.
- If the file “chop_sub.sch” does not exist, go to the **Subcircuit** menu, and select **New Subcircuit**. If the file exists, select **Load Subcircuit** instead**.**
- A subcircuit block (rectangle) will appear on the screen. Place the subcircuit.

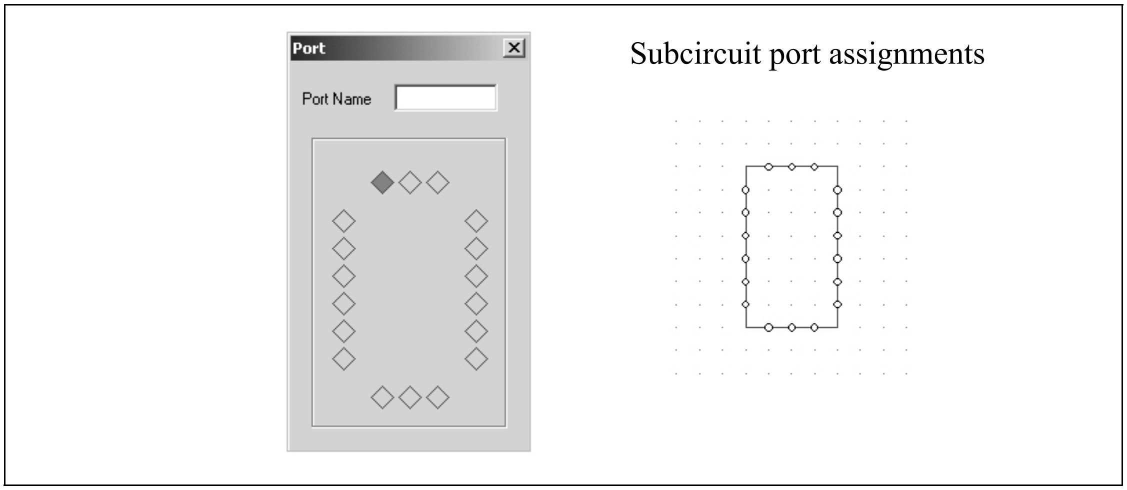
If the circuit that is to be converted into a subcircuit has already been created in the main circuit, a quick way of converting it into a subcircuit is to select the circuit, and then right click and choose **Create Subcircuit**. Specify the subcircuit file name as "chop_sub.sch", and the circuit will be converted into a subcircuit. Adjust the port location and wire connection if necessary.

2.7.2 Creating Subcircuit - Inside the Subcircuit

To enter the subcircuit, double click on the subcircuit block.

- Create/edit the content of the subcircuit circuit exactly the same way as in the main circuit.
- To specify the subcircuit size, select **Set Size** in the **Subcircuit** menu. In this example, the size is set to 4x7 (width of 4 divisions and height of 7 divisions). Note that the size of the subcircuit should be chosen such that it gives the proper appearance and allows easy wire connection in the main circuit.
- Once the subcircuit is complete, define ports to connect the subcircuit nodes with the corresponding nodes in the main circuit. Choosing **Place Port** in the **Subcircuit** menu, and a port image will appear.

After the port is placed in the circuit, a pop-up window (shown on the left below) will appear.



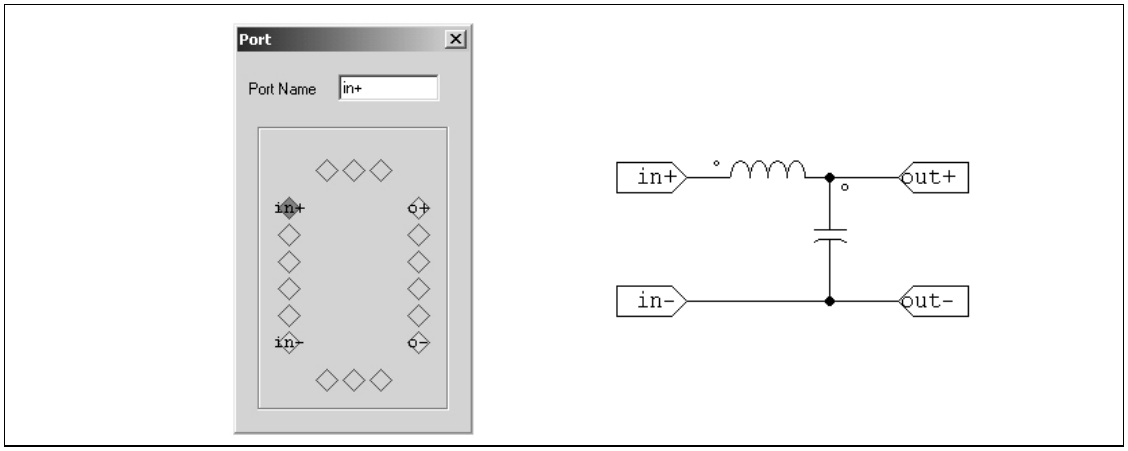
The diamonds on the four sides represent the connection nodes and the positions of the subcircuit. They correspond to the connection nodes of the subcircuit block on the right. There are no diamonds at the four corners since connections to the corners are not permitted.

When a diamond is selected, it is colored red. By default, the left diamond at the top is selected and marked with red color. Click on the desired diamond to select and to specify the port name.

In this example, in the main circuit “chop.sch”, there are four linking nodes, two on the left side and two on the right side of the subcircuit block. The relative position of the nodes are that the upper two nodes are 1 division below the top and the lower two nodes are 1 division above the bottom.

To specify the upper left linking node, click on the top diamond of the left side, and type “in+”. The text “in+” will be within that diamond box and a port labelled with “in+” will appear on the screen. Connect the port to the upper left node. The same procedure is repeated for the linking nodes “in-”, “out+”, and “out-”.

- After the four nodes are placed, the node assignment and the subcircuit appear as shown below.


Port Name: in+

The creation of the subcircuit is now complete. Save the subcircuit, and go back to the main circuit.

2.7.3 Connecting Subcircuit - In the Main Circuit

Once the subcircuit is created and connection ports are defined, complete the connection to the subcircuit block in the main circuit.

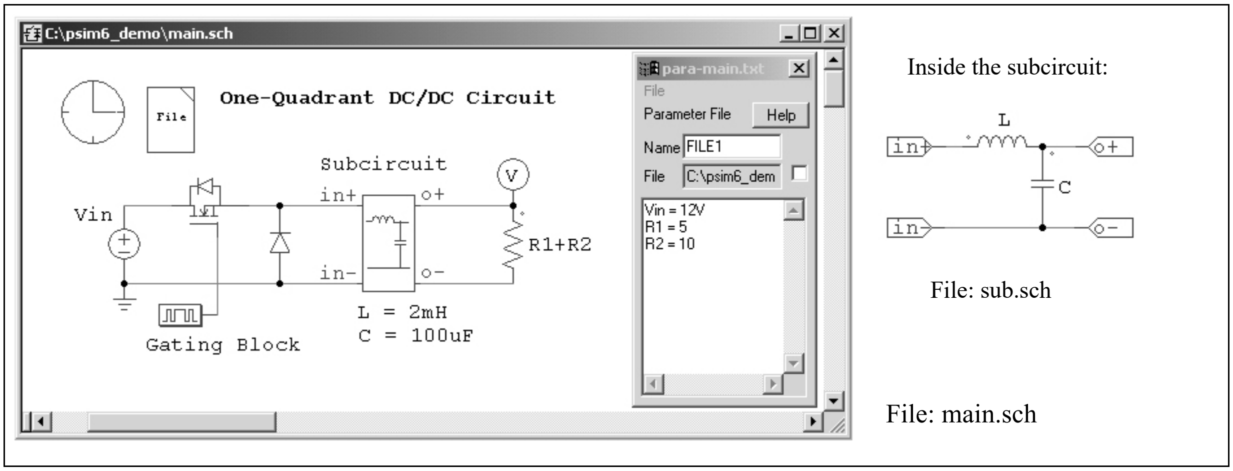
- In the main circuit, the connection points on the borders of the subcircuit block appear as hollow circles.
- Select the subcircuit block, and select **Show Subcircuit Ports** in the Subcircuit menu to display the

port names as defined inside the subcircuit.

- Connect the wires to the connection points accordingly.

2.7.4 Other Features of the Subcircuit

This section describes other features of the subcircuit through the example shown below.



```
Vin = 12V
R1 = 5
R2 = 10
```

2.7.4.1 Passing Variables from the Main Circuit to Subcircuit

In this example, the main circuit “main.sch” uses a subcircuit “sub.sch”. In the subcircuit, the inductance value is defined as “L” and the capacitance is defined as “C”. The default values of L and C can be set by selecting **Subcircuit | Set Default Variable List**. In this case, L is set to 5mH and C is set to 100uF.

When the subcircuit is loaded into the main circuit the first time, this default variable list will appear in the tab “Subcircuit Variables” in **Subcircuit | Edit Subcircuit** from the main circuit “main.sch”. New variables can be added here and variable values can be changed. In this case, L is changed to 2mH, and C is kept the same as the default value.

Note that the variables and the values are saved to the netlist file and used in simulation. The default variable list inside the subcircuit is not saved to the netlist and is not used for simulation.

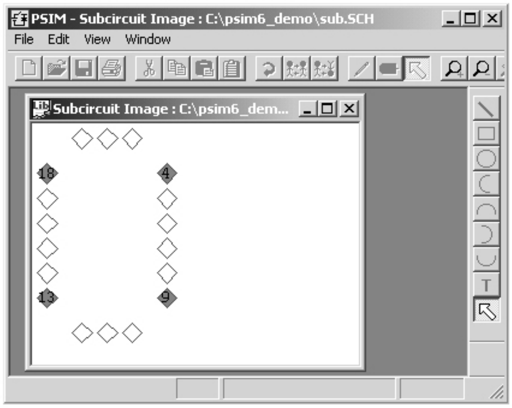
This feature allows the parameters of a subcircuit to be defined at the main circuit level. In the case where the same subcircuit is used several times in one main circuit, different parameters can be assigned to the same variable. For example, if the subcircuit “sub.sch” is used two times in above example, in one subcircuit L can be defined as 3mH, and in another subcircuit L can be defined as 1mH.

Note that this example also illustrates the feature that parameters can be defined as a variable (for example “Vin” for the input dc voltage source) or a mathematical expression (for example “R1+R2” for the load resistance). The variables “Vin”, “R1”, and “R2”, are defined in the parameter file “para-main.txt”. See Section 4.1 for more details.

2.7.4.2 Customizing the Subcircuit Image

The following are the procedures to customize the subcircuit image of “sub.sch”:

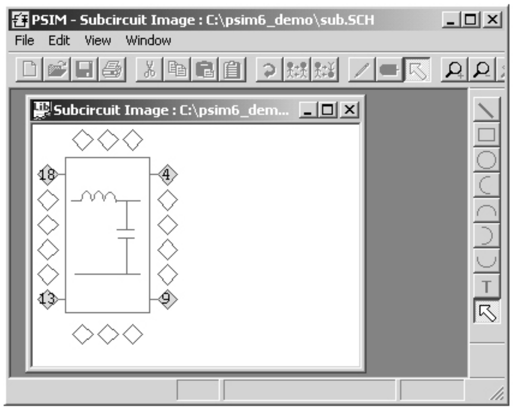
- In the subcircuit, select **Edit Image** in the **Subcircuit** menu. A window will pop-up, as shown below.


PSIM - Subcircuit Image : C:\psim6_demo\sub.SCH
File Edit View Window
Subcircuit Image : C:\psim6_dem...
18
4
13
9
T

In the window, the diamonds marked red are the connection nodes of the subcircuit block, in exactly the same positions as appearing in the main circuit.

- Use the drawing tool to create/edit the image for the subcircuit block. If the drawing tool is not already displayed, go to the **View** menu and check **Drawing Tools**. Click on **Zoom In** and **Zoom Out** icons on the toolbar to adjust the size of the image working area.

  After the image is created, the pop-out window will appear as follows.


PSIM - Subcircuit Image : C:\psim6_demo\sub.SCH
File Edit View Window
Subcircuit Image : C:\psim6_dem...
18
4
13
9
T

- Go back to the subcircuit window (“sub.sch” in this case), and save the subcircuit. The new subcircuit block image should appear in the main circuit.

2.7.4.3 Including Subcircuits in the PSIM Element List

If you create a directory called “User Defined” under the PSIM directory, and place subcircuits inside this directory. subcircuits will appear as items in the **Elements** menu, under **Elements | User Defined**, just like any other PSIM elements. You can also create subdirectories under the directory **User Defined**, and place subcircuits inside the subdirectories. For example, the **Elements** menu may look like this:

- **Power**
- **Control**
- **Other**
- **Sources**
- **Symbols**
- **User Defined**
  - Subcircuit 1

- **Project A**
    - Subcircuit 2
    - Subcircuit 3
- **Project B**
    - Subcircuit 4

In this way, common-used custom-built subcircuits can be grouped together and easily managed and accessed.

2.8 Elements Menu

In PSIM, all the elements are stored under the **Elements** menu. They are stored in the following menus:

Power Power circuit elements, such as R, L, C, switching devices, transformers, motor drive modules, and etc.

Control Control circuit elements, such as computational. logic, digital control elements and etc.

Other Elements interconnecting power and control circuit, such as switch controllers, voltage/current sensors, probes, and etc.

Sources Various voltage and current sources

Symbols Symbols for drawing purpose, not for simulation usage

Event Control Event control elements

SimCoder SimCoder blocks for automatic code generation

User Defined Users can create a subfolder called "User Defined" in the PSIM folder, and place common used files in this folder. The folder "User Defined" and files in the folder will appear in the **Elements** >> **User Defined** menu.

Page Place a page title block for printing

2.9 Simulate Menu

The following functions are provided in the **Simulate** menu:

Simulation Control Set the simulation parameters such as time step, total time, etc. When this is selected, the cursor will change to the image of a clock. Place this clock on the schematic, and double click to display the property window.

Run Simulation Run PSIM simulation

Run SPICE Simulation Run SPICE simulation using the built-in SPICE engine in PSIM

Run LTspice Simulation Run LTspice simulation

Cancel Simulation Cancel the simulation that is currently in progress

Pause Simulation Pause the simulation that is currently in progress

Restart Simulation Resume a paused simulation

Simulate Next Time Step Run the simulation to the next time step, and pause

Run SIMVIEW Launch the waveform display program SIMVIEW

Generate Netlist File Generate the PSIM netlist file

Generate Netlist File (xml) Generate the PSIM netlist file in xml format

View Netlist File View the generated PSIM netlist file

Generate SPICE Netlist (.net) Generate the SPICE netlist file from the schematic

Generate SPICE Netlist (.cir) Generate the SPICE netlist for LTspice simulation. The .cir file will be saved to the folder automatically.

Show Warning Show the warning messages, if any, from the simulation.

Show Fixed-Point Range Check Result Display the fixed-point range check result. This is used to show if fixed-point variables are within the range or not for SimCoder circuits for fixed-point DSP (such as F2803x/F2806x/F2802x)

Arrange SLINK Nodes Rearrange the SLINK nodes. This function is for the SimCoupler Module for co-simulation with Matlab/Simulink. Please refer to Section 3.6 for more details.

Check Multi-Level Elements Check and change the model level setting of multi-level elements in one place. This function can also show elements that are not compatible with PSIM simulation or with SPICE simulation.

Generate Code Generate code from the control schematic. This function is for SimCoder for automatic code generation. Please refer to *SimCoder User Manual* for more details.

Open Generated Code Folder Open the folder where the generated code are located.

Runtime Graphs Select waveforms to display in the middle of a simulation run.

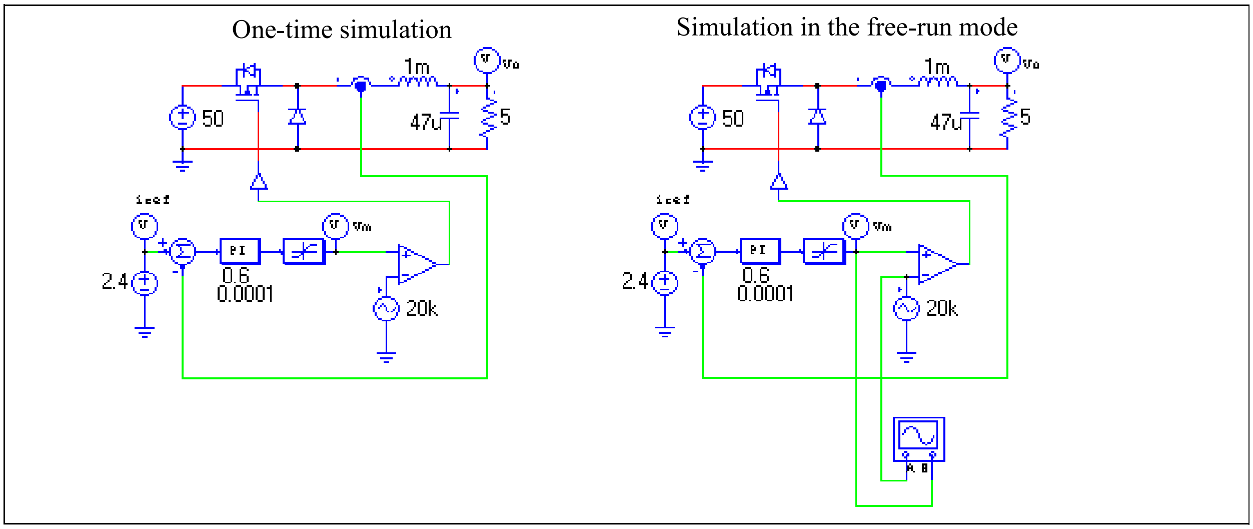
The dialog window of the **Runtime Graphs** function has two tabs: *Standard* and *Vector*. The Standard tab lists the variables for time-domain waveform display. The Vector tab defines vectors for vector plot. The real and imaginary parts of a vector comes from the same variable list as in the Standard tab.

To view the simulation waveforms of output variables in the middle of simulation, one can either go to **Simulate** >> **Runtime Graphs** and select the variables, or right click on top of a voltage probe or current probe and select *Runtime Graph Window* from the menu.

A runtime graph display the waveform in its entirety, from the beginning to the final study time. Because of this, the runtime graphs are disabled in the free-run mode as the final study time is undetermined.

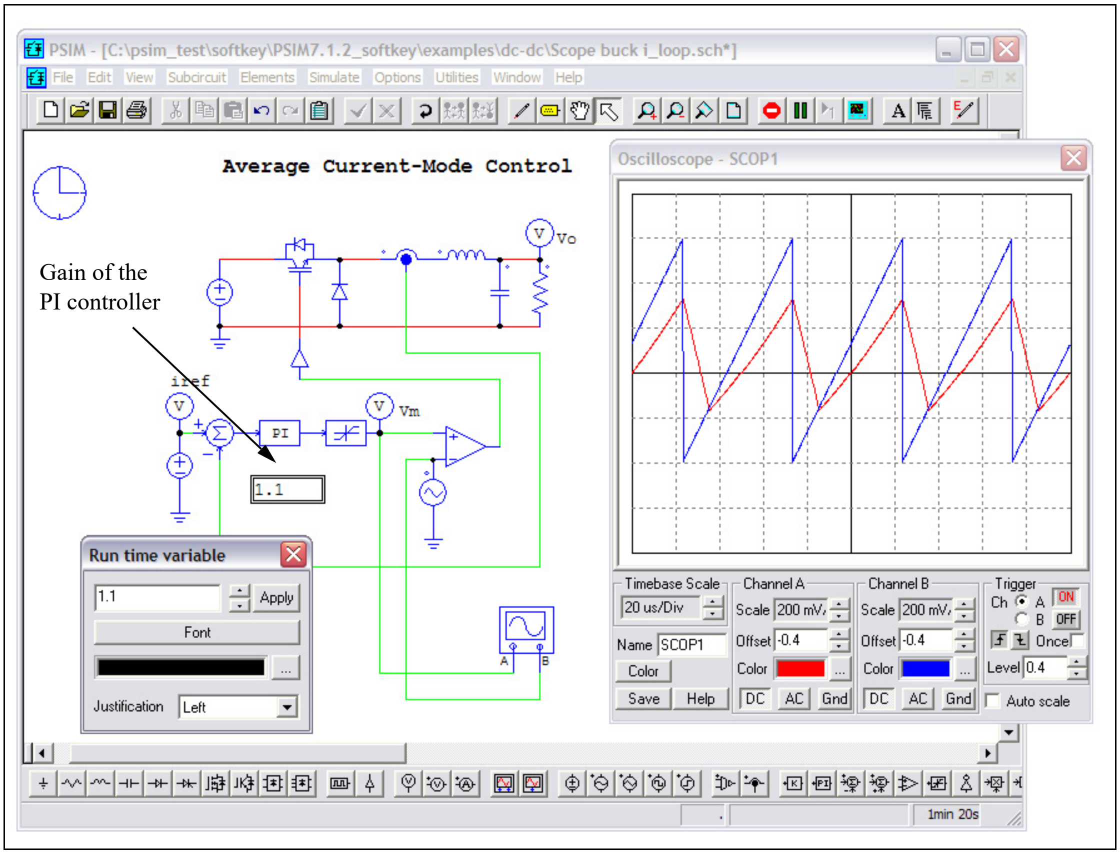
In the free-run mode, the majority of the element parameters can be changed during runtime in the middle of the simulation. This makes it possible to tune a circuit while inspecting key waveforms using voltage/current scopes, until desired performance is achieved.

To illustrate how to run a simulation in the free-run mode, a buck converter circuit shown below is used as an example. The circuit on the left was originally set up for the one-time simulation, with the total simulation time set to a specific value.


One-time simulation
Simulation in the free-run mode

To set up the simulation in the free-run mode:

- In Simulation Control, check the **Free Run** checkbox.
- Go to **Elements** >> **Other** >> **Scopes**, and select the 2-channel voltage scope. Connect the scope as shown above on the right.
- Double click on the scope, and the scope image will appear. Start the simulation, and the waveforms will appear and will be updated continuously in the scope. Change the scope settings as desired.
- Elements parameters can now be adjusted in the middle of the simulation. To adjust the gain of the PI controller, for example, right click on top of the PI block, and choose **Runtime Variables** >> **Gain**. The text of the gain "0.6" will be displayed if it has not been displayed already.
- Click on the text "0.6", and a small dialog window will appear. The screen should look as follows.



Run time variable: 1.1

Justification: Left

Timebase Scale: 20 us/Div

Name: SCOP1

Channel A: Scale 200 mV, Offset -0.4, Color 🟥

Channel B: Scale 200 mV, Offset -0.4, Color 🟦

Trigger: Ch ● A ○ B, Level 0.4

- Place the cursor inside the input field of the dialog window for the gain, and change the gain either by pressing on the upper/down arrow keys on the keyboard, or by entering a new value and then clicking on **Apply**. Watch how the waveforms change as the gain is changed.

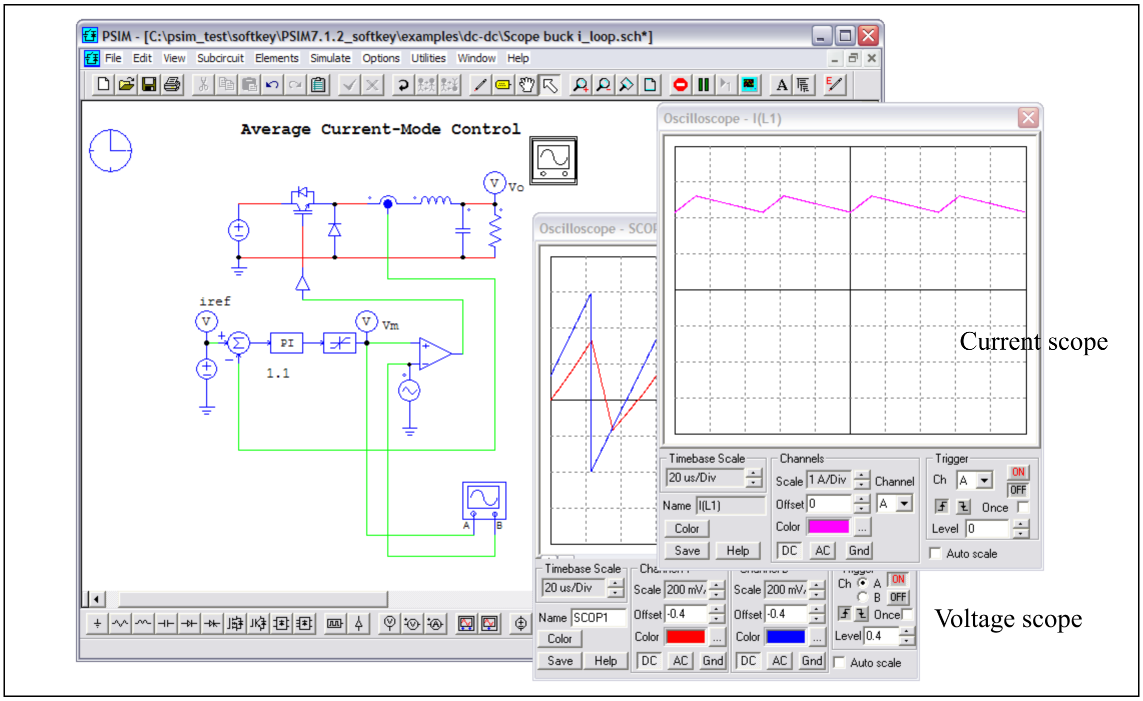
  Other parameters, such as current reference, dc input voltage, inductance, capacitance, and load resistance, can be changed in the similar way.

- Branch currents can also be displayed in the free-run mode. To display the inductor current, for example, right click on top of the inductor, and a menu will appear. Choose **Current Scopes** and the branch current name.

- An image of the current scope (similar to the voltage scope image, but without connection terminals) will appear. Double click on the scope to expand and view the inductor waveform.

  Below is how the window would look like with both the voltage scope and the current scope.

  Other branch currents, such as capacitor current, load current, diode current, or MOSFET switch current, can be displayed in the similar way.



Running Simulation in a Command Line:

Simulation can also be launched with the command line option by running the program PsimCmd.exe. For example, to simulate the circuit "buck.psimsch" which is stored in the folder "c:\Powersim\examples", go to the PSIM folder, and run the following command:

```
        PsimCmd  -i "c:\psim\examples\buck.psimsch" -o "c:\psim\examples\buck.smv"
```

The format of the command line is as follows:

```
        PsimCmd  -i "[input file]" -o "[output file]" -v "VarName1=VarValue1" -v "VarName2=VarValue2" -t
            "TotalTime" -s "TimeStep" -g
```

Note that the quotes around the parameter values must be present. The command-line parameters are:

-i: Input schematic file name

-o: Output file name (in either .txt or .smv extension)

-v: Variable name and value. This parameter can be used multiple times. For example, to define the resistance R1 as 1.5 and the inductance L1 as 0.001, we have:

```
-v "R1=1.5"  -v "L1=0.001"
```

-t: Total time of the simulation

-s: Time step of the simulation

-g Run SIMVIEW after the simulation is complete

Note that the number of variables that can be defined in a command line is limited to 30.

With the command-line option, one can run several circuits automatically in a batch run.

2.10 Options Menu

The following functions are provided in the **Options** menu for various settings:

Settings... Set the simulation parameters such as time step, total time, etc. When this is selected, the cursor will change to the image of a clock. Place this clock on the schematic, and double click to display the property window.

Languages Select different languages for PSIM display.

Auto-run SIMVIEW Automatically run SIMVIEW after the simulation is complete.

Set Path... Set the PSIM search paths and device file paths.

Enter Password Enter the password to view a schematic file that is password protected.

Disable Password Disable the protection of a schematic file that is password protected.

Customize... Create customized toolbars, keyboard, and application menu frame.

Save Custom Settings... Save the following custom settings to a file: Default element values; custom keyboard definition; and custom toolbar definition. This file can then be loaded on another computer using the **Load Custom Settings...** function. This is useful when you want to apply the same custom settings on another computer.

Load Custom Settings... Load custom setting files saved by the **Save Custom Settings...** function. This will apply the custom settings defined in the file to this computer.

Load Legacy Tool Bars Load legacy tool bars of previous PSIM versions.

Deactivate Deactivate the PSIM license. This is for softkey version only.

Change Softkey Password Change the password of login for running PSIM softkey version. This is for softkey version only.

Check for Software Update Check if any newer software updates are available on the Powersim server. This is for licenses that have the Annual Software Maintenance only.

2.10.1 Setting Option

The **Settings** menu has tabs: *General, Advanced, Colors,* and *Page Sizes*.

The *General* tab contains these sections: *Editing, Text Font, Printing,* and *Simulation*:

In the *Editing* section:

Display grid Check this option to display the grid in the PSIM schematic.

Zoom factor The zoom factor defined here is used when the schematic is zoomed in or out.

Enable rubber band When checked, an element or a portion of a circuit remains connected with the rest of the circuit when moved.

Show print page border When enabled, the border of the printout will be displayed.

In the *Text Font* section:

Default text font Set the default font for the text placed in the schematic.

Justification Define how the text will be aligned.

Default graph font Set the text font for the runtime graphs.

In the *Printing* section:

Line thickness Define the thickness of the line as it appears at the printout. It can be set to 1 (the thinnest) to 4 (the thickest). Note that the line thickness only affects the printout. It does not affect the display on the screen.

In the *Simulation* Section:

Simulation result format Simulation results can be saved in either binary format (default) or text format. The binary format will result in a smaller result file, and will be faster to load.Set the default font for the text placed in the schematic.

Output buffer size When checked, the simulation data will be written to the result file in segment. For example, if the buffer size is set to 20 MB, the simulation data will be first saved to the buffer, and when it reaches 20 MB, the whole 20-MB data will be written to the result file.

Please note that the runtime graph only plots the data in the buffer. Therefore, when the old data are saved to the file and the new data fills in the buffer, the runtime graph will only show the waveform of the new data, and the old waveform will be lost.

To retain all the waveforms in the runtime graph, one can either increase the buffer size, or un-check this option.

When this option is un-checked, however, PSIM will allocate all the required memory for the buffer at the very beginning. If the simulation time step is small and the total time is long, and if there are many output curves, a very large memory may be required, which will take some time to allocate, and may even fail if the computer does not have sufficient amount of memory.

Disable simulation warning messages When this option is checked, warning messages generated in the simulation are suppressed.

Save all voltages and current When this option is checked, all the voltages and currents of the circuit will be saved for display. To display a voltage or current, after the simulation is complete and after results are loaded into SIMVIEW, choose **View** >> **Display Voltage/Current** (or click on the corresponding icon). When the cursor is on top of a node or a branch, it will change to the image of a voltage probe or current clamp probe. Left click the mouse, and the corresponding voltage or current will appear in SIMVIEW.

To display a differential voltage, choose **View** >> **Display Differential Voltage**. Then click on the first node, and then the second node. The differential voltage waveform will appear in SIMVIEW.

Maximum number of points for oscilloscope It defines the maximum number of points that an oscilloscope will plot. Increase this number to display waveforms for a longer time interval.

The *Advanced* tab contains these sections: *Updates, Backup, Idle Time, Hardware Code Generation, SPICE, Alternate PSIM Help File Path,* and *Parameter File Variables*.

In the *Updates* section:

Check for software updates When this option is checked, if you have the valid software annual maintenance, PSIM will automatically check for new updates on the Powersim server. If a new update is available, you will be prompted to install the update.

In the *Backup* section:

Automatic backup When this option is checked, PSIM will create a backup of the file currently being edited in the time interval specified. In case of a program crash, the backup file will preserve the previous work. The backup file is deleted automatically when the file is closed normally from the PSIM environment.

In the *Idle Time* (*for network version*) section:

Idle time When this option is checked, PSIM will be timed out after the program is idle for the specified amount of time. Note that this applies to the PSIM network version only.

When PSIM is timed out, the license will be released and other users will be able to check it out. When PSIM is back from the idle state, it will try to log back in the License Manager if a license is still available. If there is no license available, an error message will be posted.

This feature will prevent users from holding up licenses unintentionally.

In the *Hardware Code Generation* section:

Show image next to elements ... This is for SimCoder only for automatic code generation. When this option is checked, in the PSIM library, an image will appear next to the elements that can be used for code generation. This is used to differentiate elements that can be used for code generation from the ones that can not.

In the *SPICE* section:

Show image next to elements that can be used for SPICE When this option is checked, in the PSIM library, an image Sp (for SPICE) or C/p (for code generation and SPICE) will appear next to the elements that can be used for SPICE simulation.

In the *Alternate PSIM Help File Path* section:

Use alternate PSIM help file path By default, PSIM reads the help file psim.chm from the same PSIM folder. When this option is checked, PSIM will read the help file from the specified file path instead.

In the *Parameter File Variables* section:

Significant digits It defines the number of significant digits for variables defined in a parameter file and used in a C block. For example, if k1 is defined as "k1 = 10/3" and the number of significant digits is defined as 8, a value of k1=3.3333333 will be used.

Delete SIMVIEW files on exit If this option is checked, PSIM will delete simulation output files of the current session on exit.

The *Colors* tab defines the colors for grids, elements, subcircuits, ports, wire, text, nodes, and labels.

2.10.2 Set Path Option

The Set Path option defines four types of search paths:

- PSIM Search Path:	Search path for DLL files
- Device File Path:	Search path for Thermal Module device files
- C Block Include Path:	Search path for C block include files
- SPICE Model Path:	Search path for SPICE model library

The PSIM search path allows users to define additional search paths when loading an external DLL file. For example, if a schematic file uses a DLL file, and this DLL file is placed in a directory other than the schematic directory or the PSIM directory, this directory can be included in PSIM by using the **Set Path** function.

Similarly, if a Thermal Module device is placed in a folder other than the default device folder, this device file can be used by PSIM schematic by including the folder in the Device File Path.

PSIM searches the DLL files in the following order:

- PSIM directory
- Schematic file directory
- Directories as defined in the **PSIM Search Path** section.

The first time that the DLL file is encountered, it will be loaded. For example, assume that the PSIM program files are in C:\PSIM, the schematic file is in C:\TEMP; and the directory as defined in the **Set Path** function is C:\TEMPDLL. The DLL file can be in one of the three places:

- C:\PSIM
- C:\TEMP
- C:\TEMPDLL

PSIM loads the Thermal Module device files in the following order:

- The "device" sub-folder in the PSIM directory
- Directories as defined in the **Device File Path** section.

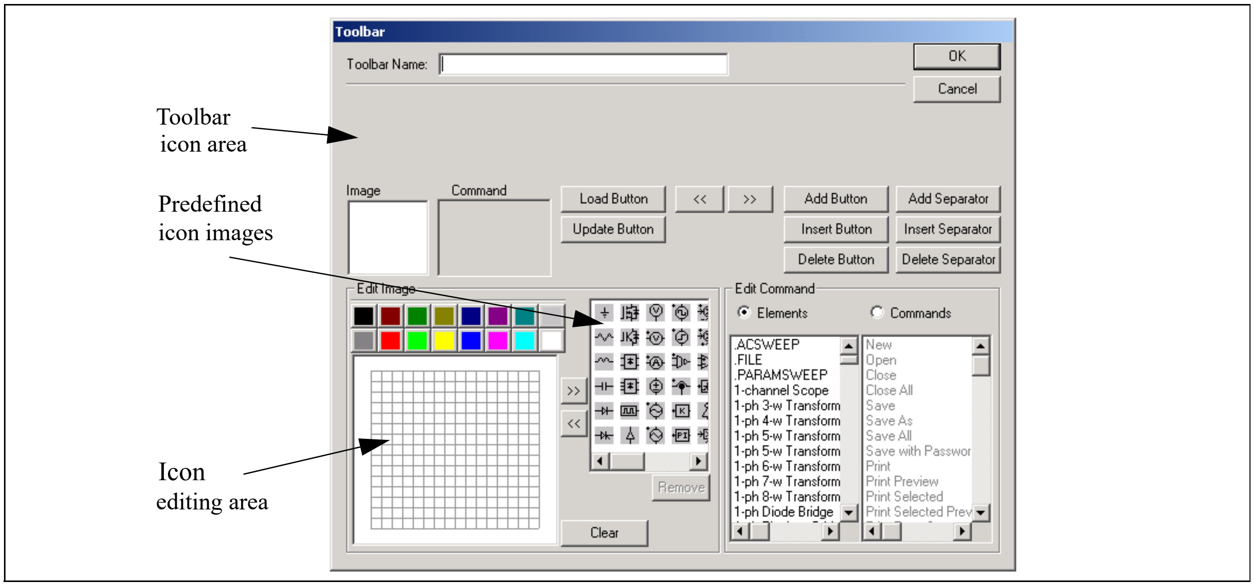
All the device files in the PSIM's device sub-folder and in the folders defined in the Device File Path will be loaded.

2.10.3 Customizing Toolbars and Keyboards

The procedures for create customized toolbars and to customize keyboards are described below.

Customizing Toolbars:

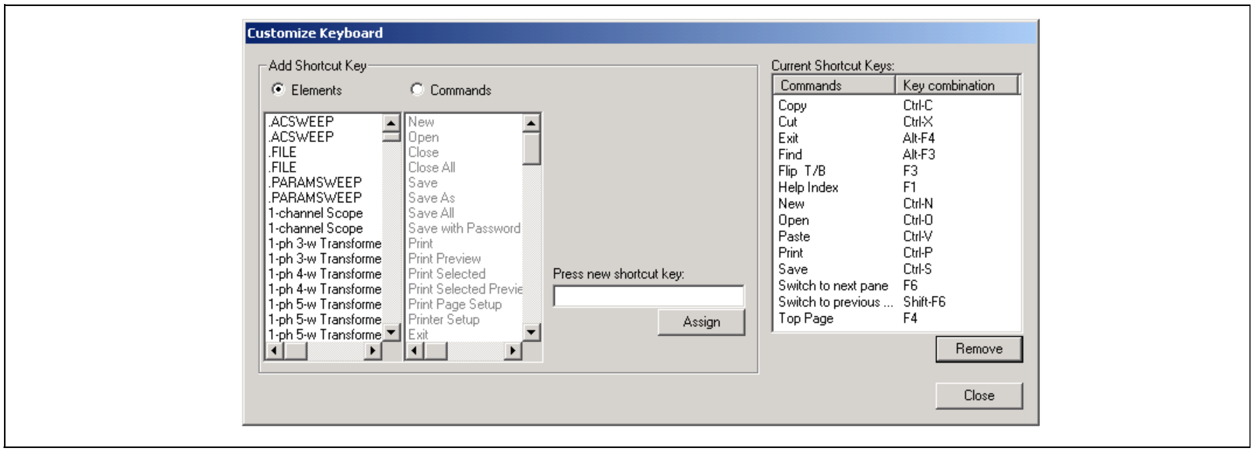
To create a toolbar called "new" and add the AND gate to the toolbar, for example, do the following:



- Choose **View** >> **Custom Toolbars**. The Custom Toolbars dialog window will appear. Choose **New Toolbar**, and the following window will appear.
- Specify the **Toolbar Name** as "new".
- Draw the AND gate icon in the icon editing area. Or if the icon is already available in the predefined icon images, select the icon and copy it to the icon editing area.
- Under the **Edit Command** section, with the option **Elements** selected, highlight "AND Gate". Then click on the **Add Button**. The icon will appear in the toolbar icon area. Click on **OK** to close the window, and go back to the Custom Toolbars window.
- Check "new" in the Custom Toolbars dialog window, and the new toolbar will appear. Uncheck the box will hide the toolbar.

Customizing Keyboard:

To define the key "r" for getting a resistor from the library, for example, do the following:



Commands	Key combination
Copy	Ctrl-C
Cut	Ctrl-X
Exit	Alt-F4
Find	Alt-F3
Flip T/B	F3
Help Index	F1
New	Ctrl-N
Open	Ctrl-O
Paste	Ctrl-V
Print	Ctrl-P
Save	Ctrl-S
Switch to next pane	F6
Switch to previous ...	Shift-F6
Top Page	F4

- Choose **View** >> **Custom Keyboard**. The Custom Toolbars dialog window will appear. Choose **New Toolbar**, and the following window will appear.
- In the section **Add Shortcut Key**, select the option **Elements**. Then find and highlight the element "Resistor".

- Move the cursor into the input field of **Press new shortcut key**, and press the key "r" on the keyboard. Then click on **Assign**.
- The key "r" will be assigned to the resistor, and the definition will appear in the **Current Shortcut Key** list.

2.11 Utilities Menu

Several utility programs are provided under the **Utilities** menu.

Parameter Tool It launches a parameter file window. One can load an existing parameter file, or enter expressions for computation purposes.

Script Tool The script tool allows one to run a script.

SPICE Netlist Check The tool checks if a SPICE netlist is supported by CoolSPICE, the SPICE engine used in PSIM. The tool will convert a non-compatible netlist to CoolSPICE-compatible netlist whenever possible, suggest how to make the change, or identify the netlist that is not supported.

s2z Converter This program converts a s-domain function to a z-domain function. This program is part of the Digital Control Module option.

SimCoupler Setup This will launch the program SetSimPath.exe that performs the setup for co-simulation between PSIM and Matlab/Simulink. This function is part of the SimCoupler Module.

Set Default PSIM Program Run this function to set this version of PSIM to be the default program to open .psimsch, .sch, and .smv files when double clicking on these files in Windows Explorer.

DSP Oscilloscope The DSP Oscilloscope function is available as part of the SimCoder Hardware Targets. It is used together with the SCI functions to display DSP waveforms in real time. For more information on how to use the DSP Oscilloscope, please refer to the document "Tutorial - Using SCI for Waveform Monitoring.pdf".

Device Database Editor The device database editor PcdEditor.exe for the Thermal Module.

Curve Capture Tool Tool to capture curves from manufacturer datasheet. The captured data can be plotted using SIMVIEW, or used in lookup tables. This tool can also be used to read the x/y values from a curve.

To start the capture process, click on the right arrow at the upper left corner of the dialog window.

B-H Curve This program plots the B-H curve of the saturable core element (under **Elements** >> **Power** >> **Magnetic Elements**).

Solar Module (physical model) This program plots the i-v curve of the solar module (physical model) element (under **Elements** >> **Power** >> **Renewable Energy**). This function is part of the Renewable Energy option.

Ultra-capacitor Model Tool The parameter extraction tool for ultra-capacitor models. For more information, please refer to the document "Tutorial - Ultra-capacitor Model.pdf".

Launch/Export to SmartCtrl Launch the SmartCtrl software, or if the ac sweep result is available, to export the ac sweep result to SmartCtrl. SmartCtrl is a separate software for control loop design. For more information, please refer to the Help menu in SmartCtrl.

Unit Converter This program performs unit conversion in length, area, weight, and temperature.

Calculator This will launch the Windows' calculator.

2.12 Managing the PSIM Library

A PSIM library element consists of two parts: the netlist part and the image part. The netlist part comes from the netlist library, and there is only one netlist library, psim.lib. The netlist library can not be edited.

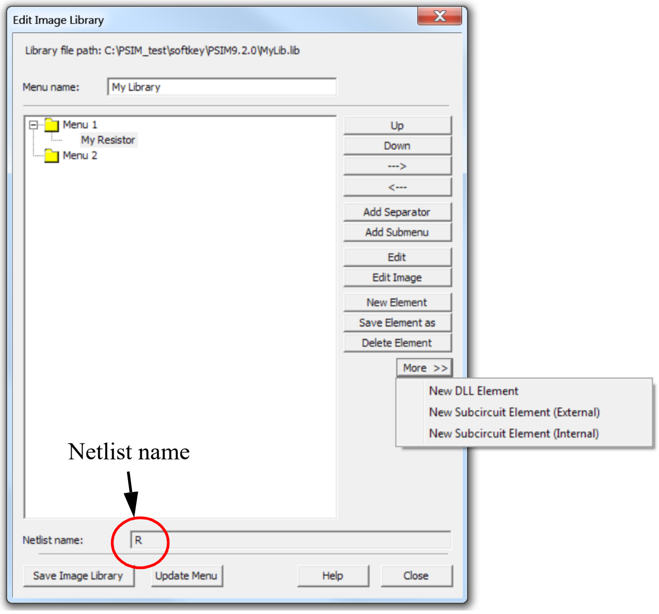
The image part comes from an image library. There can be multiple image libraries, and all the image libraries in the PSIM directory will be automatically loaded into PSIM. The standard image library provided by PSIM is psimimage.lib. This file also can not be edited. However, in order to facilitate users to copy images from the standard image library, the standard image library can be viewed by going to **Edit** >> **Edit Library** >> **Edit library files**, and choosing psimimage.lib.

Users can create their own custom image libraries. To create a new custom image library, go to **Edit** >> **Edit Library** >> **Edit library files**, and click on **New library**. Then define the library name as it appears in the PSIM Elements menu, and the library file name. This library file will be created and placed in the PSIM directory.

To edit an image library, go to **Edit** >> **Edit Library** >> **Edit library files**, and select the library file. The figure below shows the library editor dialog window. The dialog shows the menu tree of the library as well as various functions.

To create a new element in the custom image library, click on **New Element**, and select the netlist that this element corresponds to from the list. For example, the netlist name of the resistor is "R". To create a new element called "My Resistor", select the netlist "R". To create the image for this resistor, click on **Edit Image**.

The following functions are provided in the library editor:



Edit Image Library

Library file path: C:\PSIM_test\softkey\PSIM9.2.0\MyLib.lib

Menu name: My Library

Netlist name: R

Netlist name

Up Move an element up in the menu

Down Move an element down in the menu

---> Move an element one level lower in the menu

<--- Move an element one level higher in the menu

Add Separator Add a separator between elements

Add Submenu Add a submenu in the library

Edit Edit the name of an element

Edit Image Edit the image of an element

New Element Create a new element in the image library. This element will be linked to a netlist element in the netlist library.

Save Element as Save the existing element as a new element. The new element will have the image of the existing element as the default image.

Delete Element Delete an element from the library

New Element (DLL) Create a new element from a DLL file

New Subcircuit Element (External) Create a new element from a subcircuit, and the subcircuit file is stored outside the library file

New Subcircuit Element (Internal) Create a new element from a subcircuit, and the subcircuit file is stored inside the library file

2.12.1 Creating a Secondary Image

It is possible that some users may find certain element images in the standard PSIM image library psimimage.lib different from what they are used to use. In this case, users can create their own secondary images.

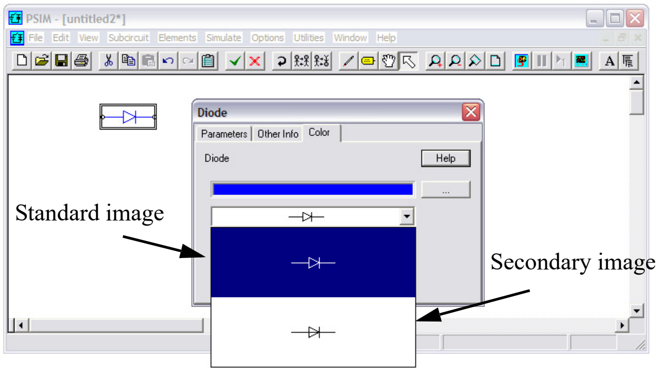
A secondary image can be created for an element in either standard image library psimimage.lib or users’ own custom image libraries. Secondary images are saved in a secondary image library with the .lib2 extension.

To illustrate the process, a secondary image will be created in the library "mylib.lib2" for the "Diode" element in the standard image library.

- Go to **Edit** >> **Edit Library** >> **Edit secondary image library files**, and click on **New library**. In the dialog, define the secondary image library name as "mylib.lib2".
- Then select "mylib.lib2" and click on **Edit selected library**. The dialog window for editing secondary image library will appear.
- Click on the **Add** button. From the PSIM library tree, navigate to **Power** >> **Switches**, and select "Diode". The element "Diode" will appear in the list of the secondary images. The text "PSIMIMAGE" in front of the "Diode" text shows that this is for the diode element in the standard image library psimimage.lib.
- Highlight "Diode", and click on the **Edit** button to create the image.
- If this image is to be used as the default image for this element, click on the "**Set as Default Image**".

After the secondary image is created, this image will be available for selection in the PSIM schematic. For example, if a diode is placed on the schematic, double click to display the property dialog of the diode, then click on the **Color** tab. Click on the pull-down arrow, and two images will be displayed. One from the standard image library, and the other from the custom image library mylib.lib.

If the secondary image is selected, all the images of the same element will be automatically changed to the secondary image. The selected image will also be set as the default image when a schematic is created or loaded the next time.



To share the secondary images that one creates with other people, one just has to send to them the secondary image library file (with the .lib2 extension).

2.12.2 Adding a New Subcircuit Element into the Library

There are two ways to add a custom model to the PSIM library list. One is to have the model in the form of a subcircuit, and then place the schematic file in a folder called **user defined** in the PSIM directory, or in one of the sub-folders of the **user defined** folder. Any schematic files and sub-folders under the **user defined** folder will appear in the PSIM library list.

Another way is to add the custom model directly to an image library. The advantage of this approach is that the custom element will have the same look-and-feel as the standard PSIM elements, giving it a better interface. It is also possible to associate a help file to the custom model.

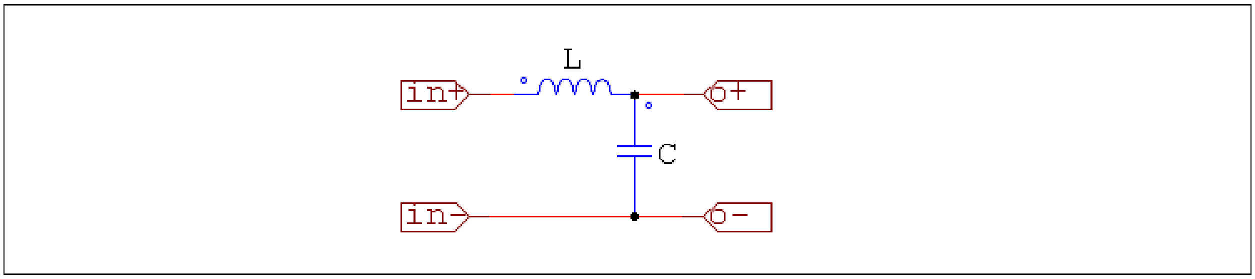
There are three main steps to add a new element, modeled in a subcircuit, into the library:

- Create the subcircuit model of the new element.
- Add this element to the PSIM library.
- Create an on-line help file for this new element.

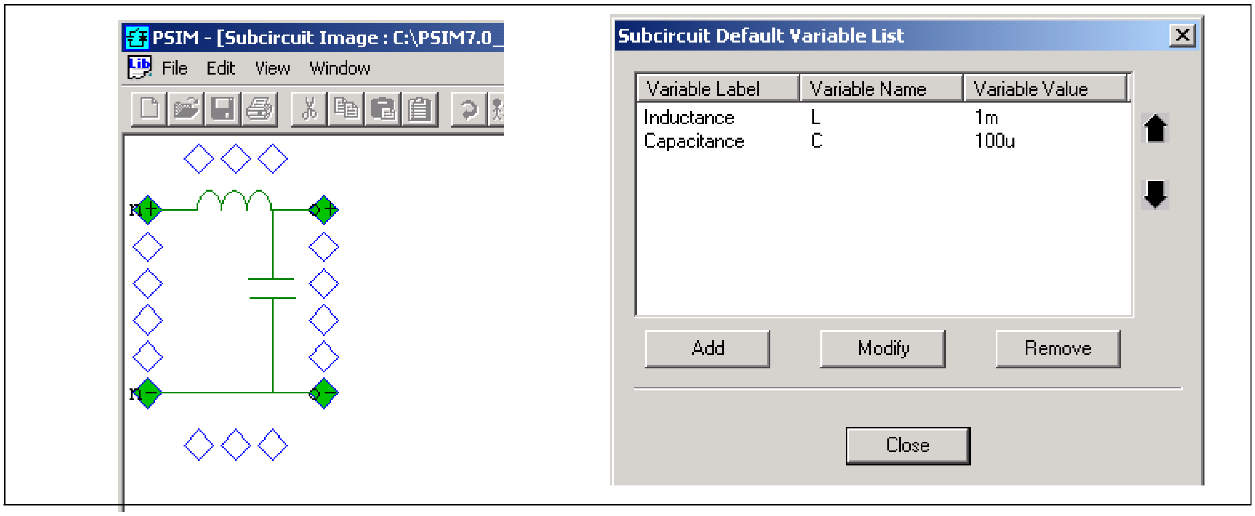
To illustrate this process, a LC-filter element is used as an example.

<u>**Step 1: Create Subcircuit**</u>

The first step is to create the subcircuit of the new element in the same way as if the subcircuit is to be called by another circuit. For example, the subcircuit of the 2nd-order LC filter, called "LC_filter.sch", and its image are shown below:



In this case, the inductance and capacitance values will be defined through the interface, and need to appear in the property window of the new LC filter element. Therefore, the parameter value for the inductance needs to be defined as a variable, in this case, *L*, and the value for the capacitance as *C*.



Subcircuit Default Variable List

Variable Label	Variable Name	Variable Value
Inductance	L	1m
Capacitance	C	100u

Then from **Subcircuit** >> **Edit Default Variable List**, add the variables *L* and *C* as the default variables. This step is necessary as the new element obtains the parameter information from the default variable list. The default variable list window should appears as follows.

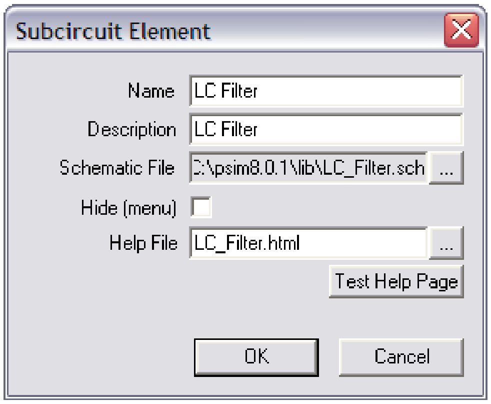
Here **Variable Label** is the text that describes the parameter, **Variable Name** is the variable that is used as the parameter value in the subcircuit, and **Variable Value** is the default value of the parameter. In the example, for the inductor *L*, the **Variable Label** is *Inductance*, the **Variable Name** is *L*, and the **Variable Value** is 1m. For the capacitor C, the **Variable Label** is Capacitance, the **Variable Name** is C, and the **Variable Value** is 100u.

After the file is created, place it in the "lib" sub-folder in the PSIM directory.

Step2: Add the New Element to the PSIM Library:

To add the subcircuit element into the PSIM library, follow these steps:

- -Go to **Edit** >> **Edit Library** >> **Edit Library Files**, and choose the custom image library for the new element. Click on **New Library** to create a new image library, or select an existing library and click on **Edit Selected Library**.
- -In the Library Editor, click on the button **New Element (Subcircuit)**. Enter the information to the dialog window as shown below:



Subcircuit Element

Name: LC Filter
Description: LC Filter
Schematic File: :\psim8.0.1\lib\LC_Filter.sch
Hide (menu): □
Help File: LC_Filter.html

Name: Name of the new element as it appears in the PSIM library
Description: Description of the new element
File Path: The location of the subcircuit schematic file "LC_filter.sch". The schematic file of the subcircuit must be placed in the "lib" sub-folder in the PSIM directory.
Input nodes: Number of input nodes.
Output nodes: Number of output nodes.
Hide (menu): Leave this box unchecked. If this box is checked, this element will not appear in the library.
Help File: On-line help file associated with this element. This file must be placed in the "help" sub-folder in the PSIM directory. When the **Help** button is clicked in the property dialog window, this file will be displayed. This file can be a text file (which can be opened by a text editor such as NotePad) or a HTML file.

- Click on the buttons **Save Image Library** and **Update Menu**. The new element will appear in the library and will be ready to use.

2.12.3 Adding a New DLL Element into the Library

Similar to that of a subcircuit element, there are three main steps to add a new element, modeled in a DLL, into the PSIM library:

- Create the model in the DLL file.
- Add this element to the PSIM library.
- Create an on-line help file for this new element.

To illustrate this process, an inductor is used as an example.

Creating the DLL:

The first step is to create the inductance model in DLL. Please refer to the relevant section on how to create a custom DLL.

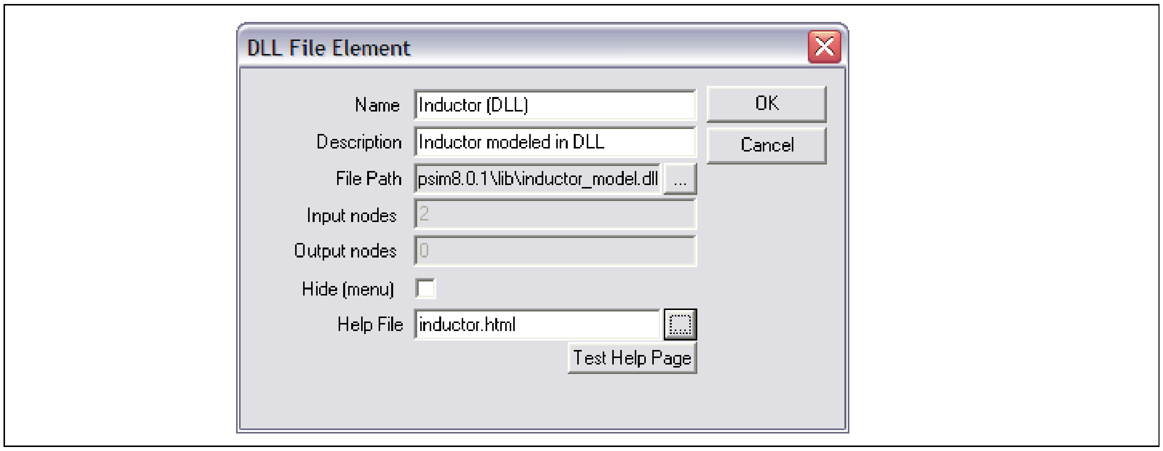
Here we assume that the DLL file, "inductor_model.dll", has already been created. It has one parameter called

"Inductance", and two connecting nodes. The file is placed in the "lib" sub-folder in the PSIM directory.

Adding the New Element to the PSIM Library:

To add the DLL element into the PSIM library, follow these steps:

- Go to **Edit** >> **Edit Library** >> **Edit Library Files**, and choose the library for the new element. Click on **New Library** to create a new image library, or select an existing library and click on **Edit Selected Library**.
- In the Library Editor, click on the button **New DLL File**. Enter the information to the dialog window as shown below:



DLL File Element

Name: Inductor (DLL)

Description: Inductor modeled in DLL

File Path: psim8.0.1\lib\inductor_model.dll

Input nodes: 2

Output nodes: 0

Hide (menu): □

Help File: inductor.html

Name: Name of the new inductor element as it appears in the PSIM library
Description: Description of the new inductor element
File Path: The location of the DLL file "inductor_model.dll" that models the new inductor element. The DLL file must be placed in the "lib" sub-folder in the PSIM directory.
Input Nodes: Number of input terminals of the new element.
Output Nodes: Number of output terminals of the new element.
Hide (menu): Leave this box unchecked. If this box is checked, this element will not appear in the library.
Help File: On-line help file associated with this element. This file must be placed in the "help" sub-folder in the PSIM directory. When the **Help** button is clicked in the property dialog window, this file will be displayed. This file can be a text file (which can be opened by a text editor such as NotePad) or a HTML file.

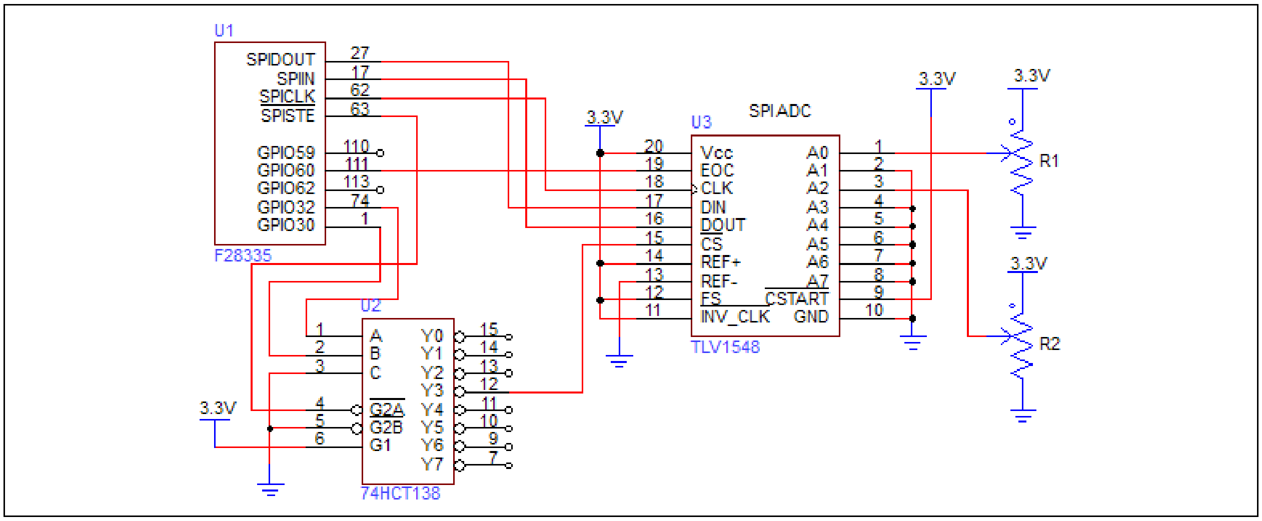
- In the next dialog window, set the new element size as: Width = 5, and Height = 2. Then create an image for this element, or accept the default image.
- Click on the buttons **Save Image Library** and **Update Menu**. The new element will appear in the library and will be ready to use.

The information regarding the number of parameters and the parameter description for the new inductor element is obtained from the DLL file automatically. In this case, the new element will have one parameter as "Inductance".

2.13 Creating a Symbol Library

With the Image Editor in the **Edit** menu, one can easily create good component images very quickly. These images can be used as secondary images of PSIM library elements, or images of subcircuits. One can also store these images in a symbol library for the purpose of circuit wiring diagrams. Note that such a schematic is solely for display purposes, and can not be simulated.

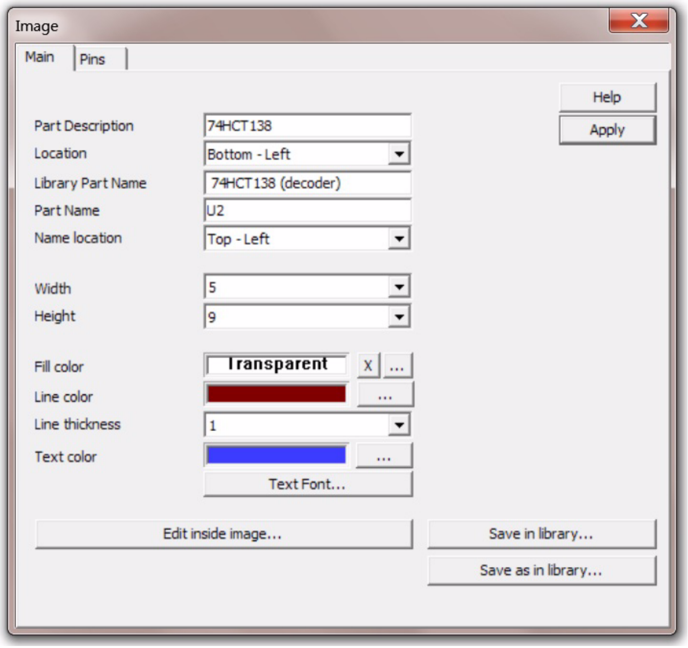
The figure below shows a hardware circuit diagram from a SPI A/D converter example.



The blocks F28335, 74HCT138, and TLV1548 are all created by the Image Editor.

To illustrate how to create an image using the Image Editor, and save the image as a symbol into a custom library, we will create the image for the decoder 74HCT138 as shown below:

- In PSIM, create a new schematic window. Then choose **Edit** >> **Image Editor**. A rectangle will appear. Place the rectangle on the schematic, and press the Esc key to escape.
- Double click on the rectangle, and a dialog window will appear. In the *Main* tab, enter the data field as shows on the screen below. The *Library Part Name* is the name of the element as it appears in the library. The size of the block is 5 in width and 9 in height, without taking into account the length of the leads.



Part Description: 74HCT138

Location: Bottom - Left

Library Part Name: 74HCT138 (decoder)

Part Name: U2

Name location: Top - Left

Width: 5

Height: 9

Fill color: Transparent

Line color: 🟥

Line thickness: 1

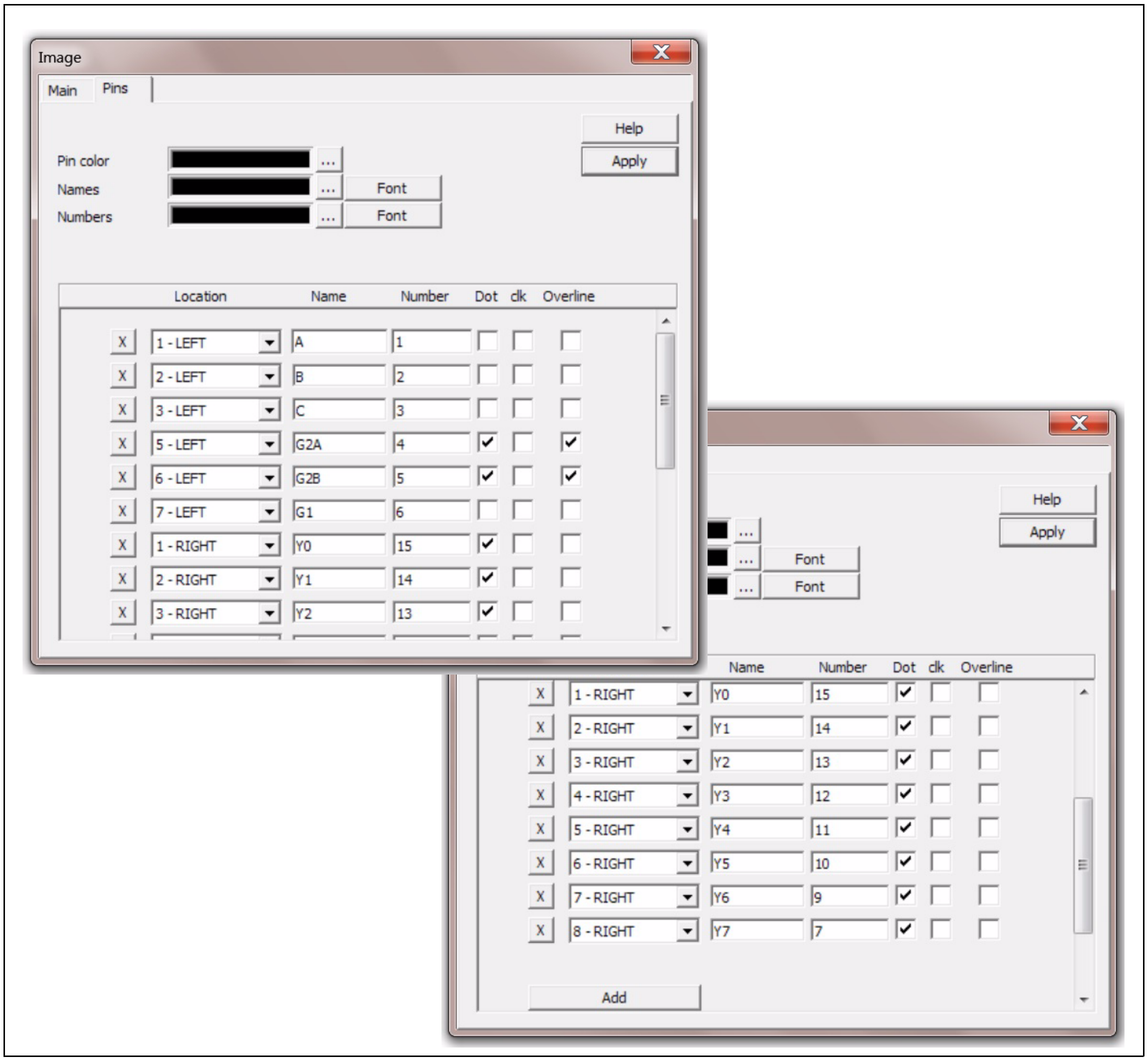
Text color: 🟦

- Click on the *Pins* tab. In this tab, the connection pins will be defined. Click on the Add button, and define *Location* as "1 - Left", *Name* as "A", and *Number* as "1". Leave the boxes for Dot, clk, and Overline unchecked.

  If the boxes are checked, for Dot, there will be a circle between the connection pin and the block; for clk, a symbol > will be added inside the block to indicate that it is a clock; and for Overline, an overline will be added to the name.

- Click on the Add button, and define *Location* as "2 - Left", *Name* as "B", and *Number* as "2". Repeat the same process for the rest of the pins. The dialog windows below show the definitions of all the connection pins.

- Once all the pins are added, go back to the *Main* tab. Click on **Save in library**, and select **[New Library]**. Enter the library name as "My Symbol", and specify the library file name as "My Symbol".The library "My Symbol" will appear under the **Elements** menu, and this library will contain the newly created element 74HCT138.



Pin color ■ ...
Names ■ ... Font
Numbers ■ ... Font

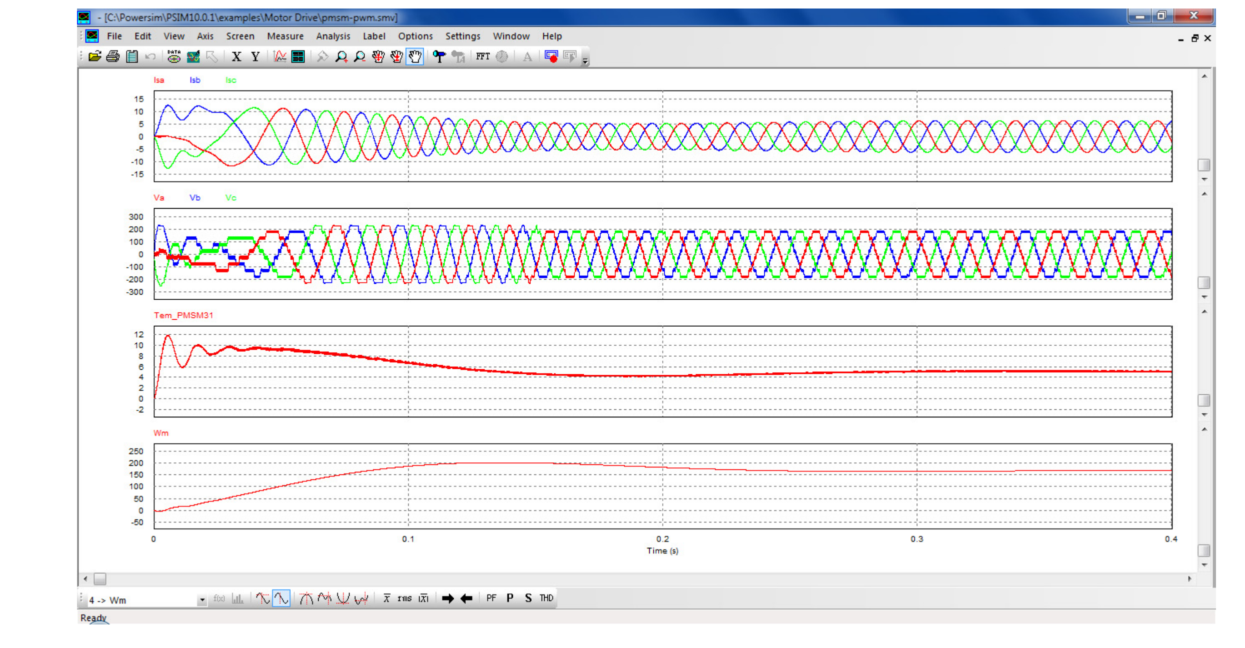
	Location	Name	Number	Dot	clk	Overline
X	1 - LEFT	A	1	□	□	□
X	2 - LEFT	B	2	□	□	□
X	3 - LEFT	C	3	□	□	□
X	5 - LEFT	G2A	4	☑	□	☑
X	6 - LEFT	G2B	5	☑	□	☑
X	7 - LEFT	G1	6	□	□	□
X	1 - RIGHT	Y0	15	☑	□	□
X	2 - RIGHT	Y1	14	☑	□	□
X	3 - RIGHT	Y2	13	☑	□	□

		Name	Number	Dot	clk	Overline
X	1 - RIGHT	Y0	15	☑	□	□
X	2 - RIGHT	Y1	14	☑	□	□
X	3 - RIGHT	Y2	13	☑	□	□
X	4 - RIGHT	Y3	12	☑	□	□
X	5 - RIGHT	Y4	11	☑	□	□
X	6 - RIGHT	Y5	10	☑	□	□
X	7 - RIGHT	Y6	9	☑	□	□
X	8 - RIGHT	Y7	7	☑	□	□

3

Waveform Processing in SIMVIEW

SIMVIEW is PSIM’s waveform display and post-processing program. The following shows simulation waveforms in the SIMVIEW environment.



SIMVIEW reads data in either ASCII text format or SIMVIEW binary format. The following shows a sample text data file:

```
   Time            Isa              Isc              Isb              Tem_IM
5.000000000E-006  0.000000000E+000  0.000000000E+000  0.000000000E+000  7.145888260E-048
1.000000000E-005  0.000000000E+000  0.000000000E+000  0.000000000E+000  1.082981714E-046
1.500000000E-005  0.000000000E+000  0.000000000E+000  0.000000000E+000  5.408644357E-046
2.000000000E-005  1.139566166E-001 -2.279132474E-001  1.139566166E-001  1.613605209E-017
2.500000000E-005  5.072914178E-001 -1.014582858E+000  5.072914178E-001  3.598226665E-015
... ... ... ...
```

Functions in each menu are explained in the following sections.

3.1 File Menu

The **File Menu** has the following functions:

Open Load a data file in ASCII text format (with .txt extension) or SIMVIEW binary format (with .smv extension)

Merge Merge another data file with the existing data file for display

Re-Load Data Re-load data from the same text file

Save As Save the waveforms to either binary data format or text format. When saving to the binary format, the current settings are also saved. In the FFT display, this will save the FFT results to a text file specified by the user.

Print Print the waveforms

Print Setup Set up the printer

Print Page Setup Set up the hardcopy printout size

Print Preview Preview the printout

Exit Quit SIMVIEW

When the data of a file are currently being displayed, if new data is available, by selecting **Re-Load Data**, new data will be loaded and waveforms will be re-drawn.

With the **Merge** function, data from multiple files can be merged together for display. For example, if one file contains the curves “I1” and “I2”, and another file contains the curves “V1” and “V2”, all four curves can be merged and displayed on one screen. If the second file also contains a curve with the same name “I1”, it will be modified to “I1_({*second_file_name*})” automatically where *second_file_name* is the name of the second file.

3.2 Edit Menu

The **Edit Menu** has the following functions:

Undo Go back to the previous X and Y axis settings

Copy to Clipboard Copy the waveforms to the clipboard either in metafile format or bitmap format.

View Data Points View the data points of the waveforms within the displayed range in a separate window. In this window, one can use the left mouse to highlight data points in rows or columns, then right mouse click and choose *Copy Selected* to copy the data to the clipboard. One could also just copy the row that the cursor is on by selecting *Copy Row*, or copy the entire data by selecting *Copy All*. One can then paste the copied data back in another program.

Note that the **Copy to Clipboard** function will copy the displayed waveforms on the screen to the clipboard. To save the memory and have the waveform image in black & white, first go to **Option** and de-select **Color** to have a black & white display, then copy the waveform to the clipboard.

3.3 Axis Menu

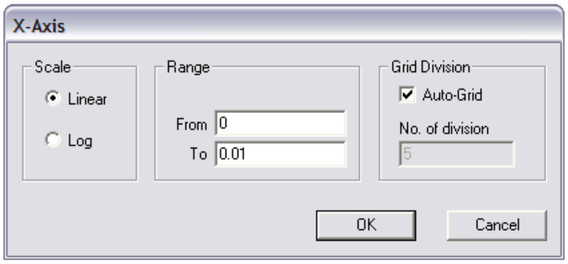
The **Axis Menu** has the following functions:

X Axis Change the settings of the X axis

Y Axis Change the settings of the Y axis

Choose X-Axis Variable By default, the first column of the data is selected as the X axis. However, other columns can also be selected as the X axis through this function.

The dialog box of the X/Y axis settings are shown below.



Scale: ● Linear ○ Log

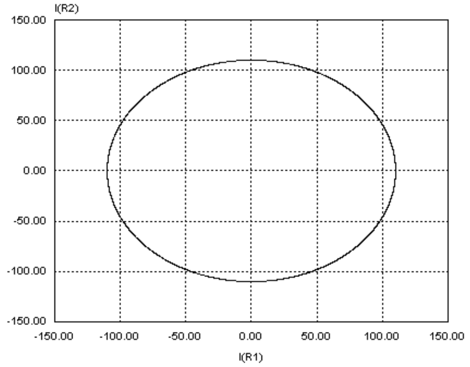
Range: From 0 To 0.01

Grid Division: ☑ Auto-Grid

No. of division 5

If the *Auto-Grid* box is checked, the number of axis divisions will be automatically determined. The data range and grid division, however, can be manually set.

The following figure shows a sine waveform is chosen as the X-axis versus a cosine waveform in the Y axis.



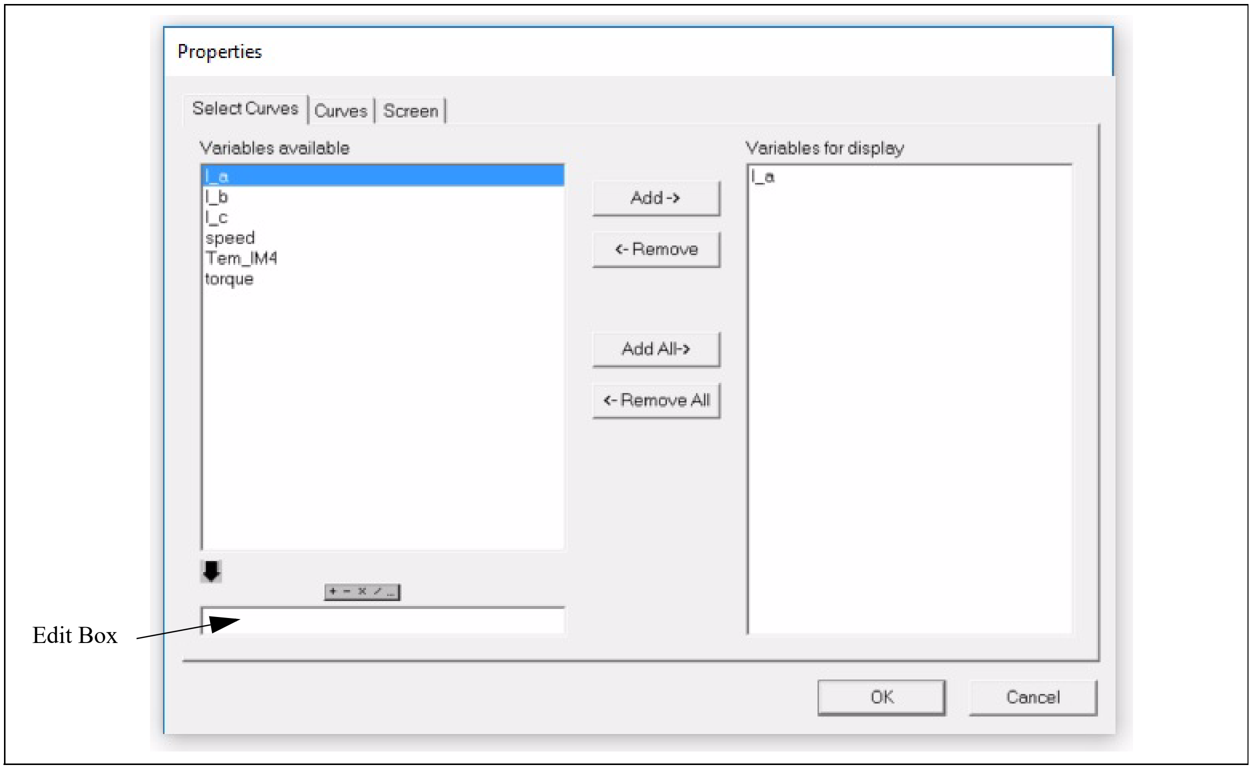
3.4 Screen Menu

The **Screen Menu** has the following functions:

New Window Create a duplicate window for the active window. This function is useful for viewing additional curves in a new window.
Add/Delete Curves Add or delete curves from the selected screen
Add Screen Add a new screen
Delete Screen Delete the selected screen
Plot Vector Diagram Plot the vector diagram. A vector is defined by its real and imaginary parts, and both come from the same variable list of the time-domain plot.
Display in Full Screen Expand the SIMVIEW window to full screen
Move Up Move the selected screen up
Move Down Move the selected screen down

A screen is selected by clicking the left mouse on top of the screen.

The property dialog window of curves is shown below.



All the variables available for display are in the *Variables Available* box, and the variables currently being displayed are in the *Variables for Display* box. After a variable is highlighted in the *Variables Available* box, it can be added to the *Variables for Display* box by clicking on “Add ->”. Similarly, a variable can be removed from display by highlighting the variable and clicking on “<- Remove”.

In the Edit Box, mathematical expressions can be specified. A mathematical expression can contain brackets and is not case sensitive. The following math functions are allowed:

+	addition
-	subtraction
*	multiplication
/	division
^	to the power of [Example: 2^3 = 2*2*2]
SQRT	square-root function
SIN	sine function
COS	cosine function
TAN	tangent function

ATAN inverse tangent function

EXP exponential (base e) [Example: EXP(x) = e^x]

LOG logarithmic function (base e) [Example: LOG(x) = ln (x)]

LOG10 logarithmic function (base 10)

ABS absolute function

SIGN sign function [Example: SIGN(1.2) = 1; SIGN(-1.2) = -1]

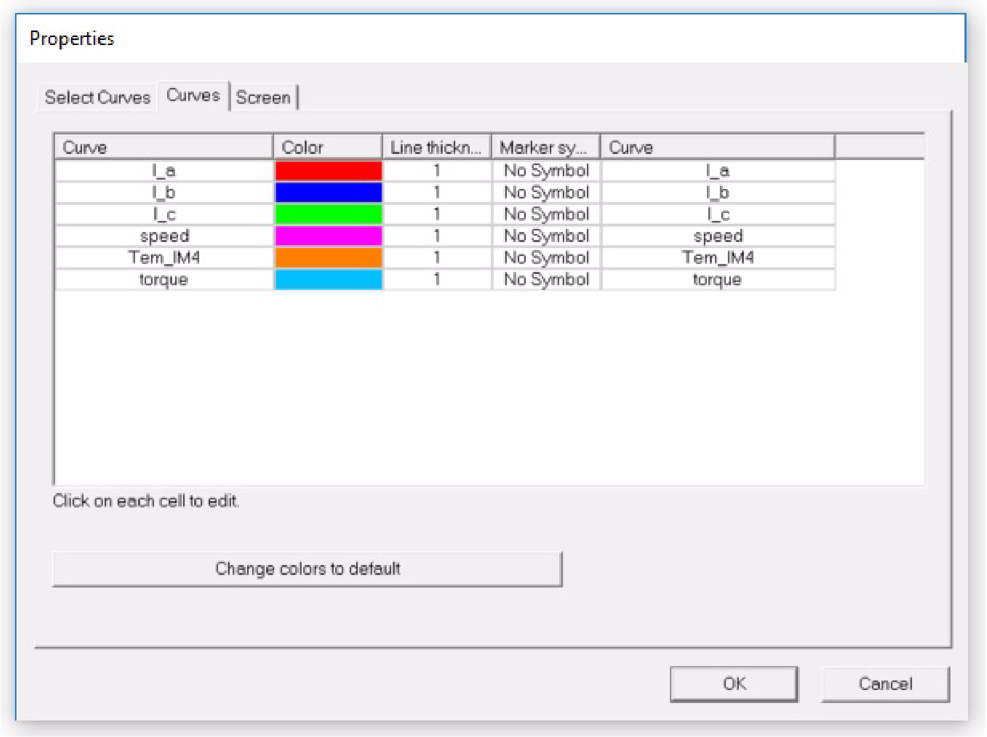
AVG moving average function that calculates the average of the curve up to the measured point.

AVGX periodic average function AVGX(y,T_p) where y is the curve name and T_p is the time interval where the average is calculated. Calculates the average of the curve y in each time segment with interval of the time segment as T_p. For example, AVGX(V1, 0.016667) will calculate the average of the curve V1 at the 60-Hz interval.

INT integration function

Type an expression in the Edit Box and click the "**Add ->**" button to add the calculated curve to the screen. Highlight the expression on the right and click the "**<- Remove**" button, the expression will be moved into the Edit Box for further editing.

In the **Curves** tab, the curve properties, such as color, line thickness, marker symbol, and label, can be defined. A dialog window is shown below.

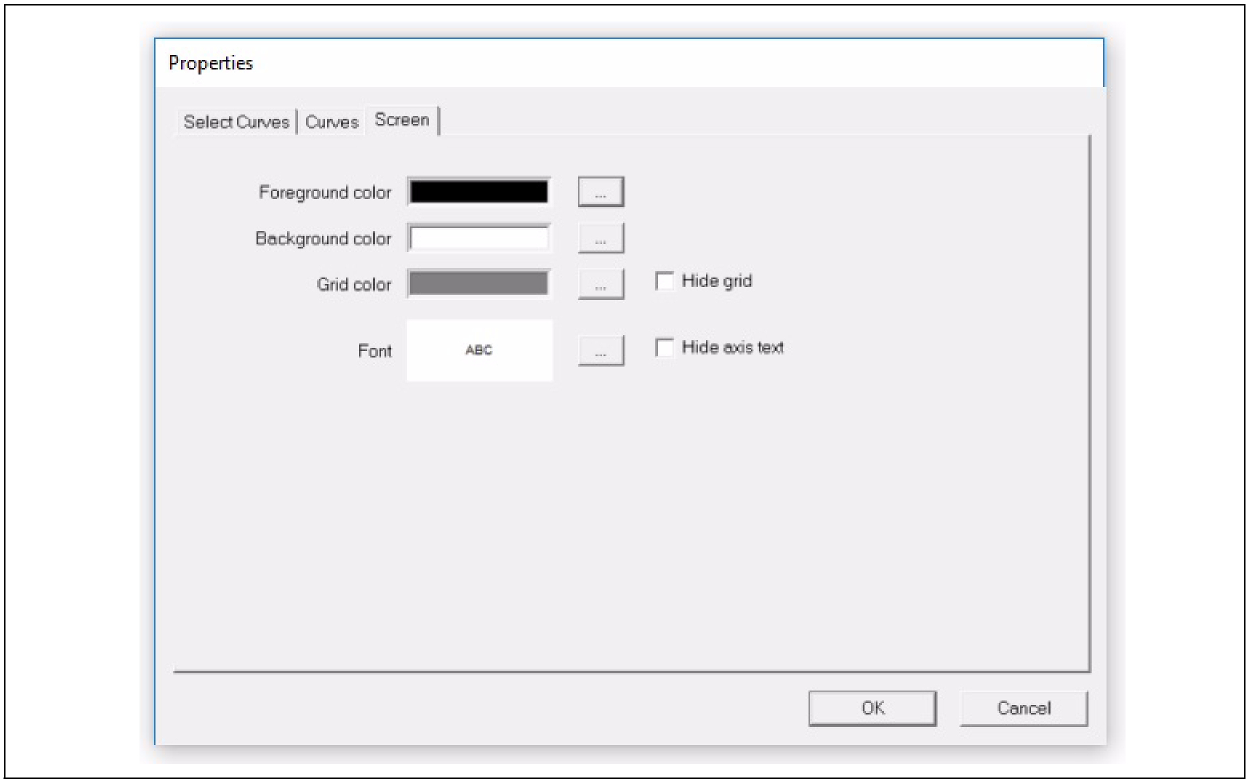


Curve	Color	Line thickn...	Marker sy...	Curve
I_a	🟥	1	No Symbol	I_a
I_b	🟦	1	No Symbol	I_b
I_c	🟩	1	No Symbol	I_c
speed	🟪	1	No Symbol	speed
Tem_IM4	🟧	1	No Symbol	Tem_IM4
torque	🟦	1	No Symbol	torque

Click on each cell to edit.

The default settings of the curve color, line thickness, and mark symbol are defined in **Options** >> **Settings**.

In the **Screen** tab, the screen properties, such as foreground/background colors, grid color, and font size/type, can be defined. A dialog window is shown below.


Properties

Select Curves | Curves | Screen

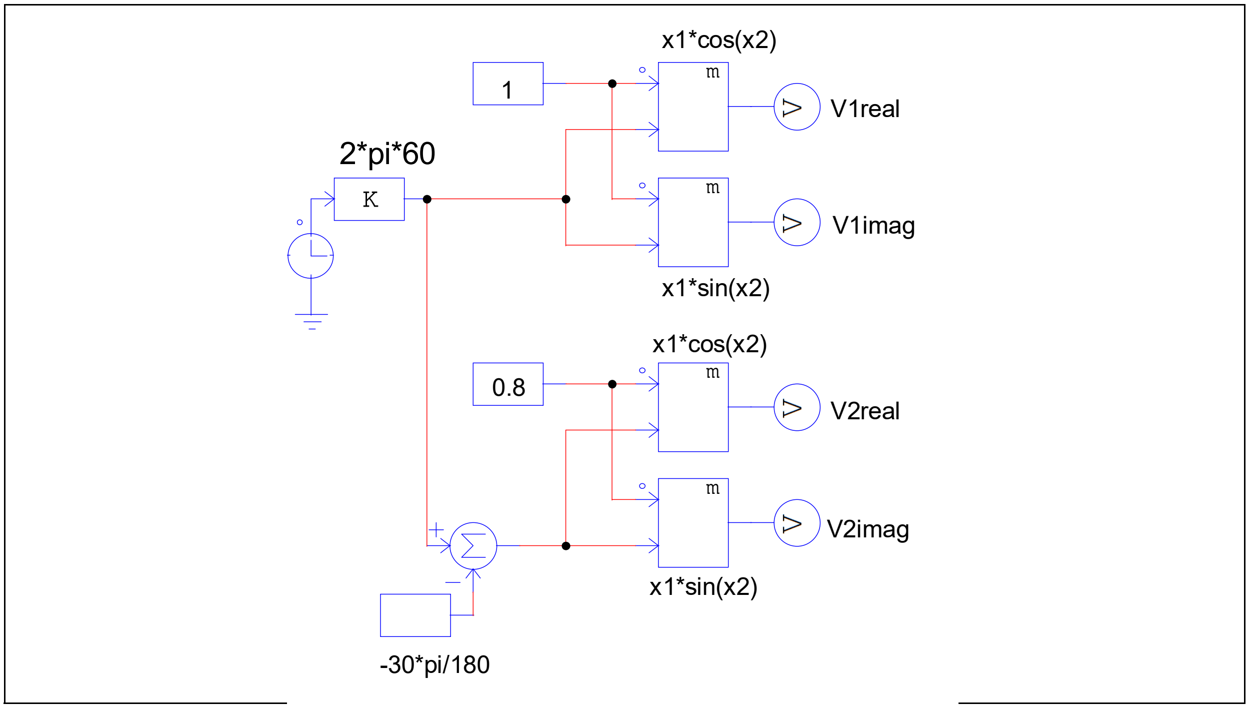
Foreground color ■ ...

Background color □ ...

Grid color ▨ ... □ Hide grid

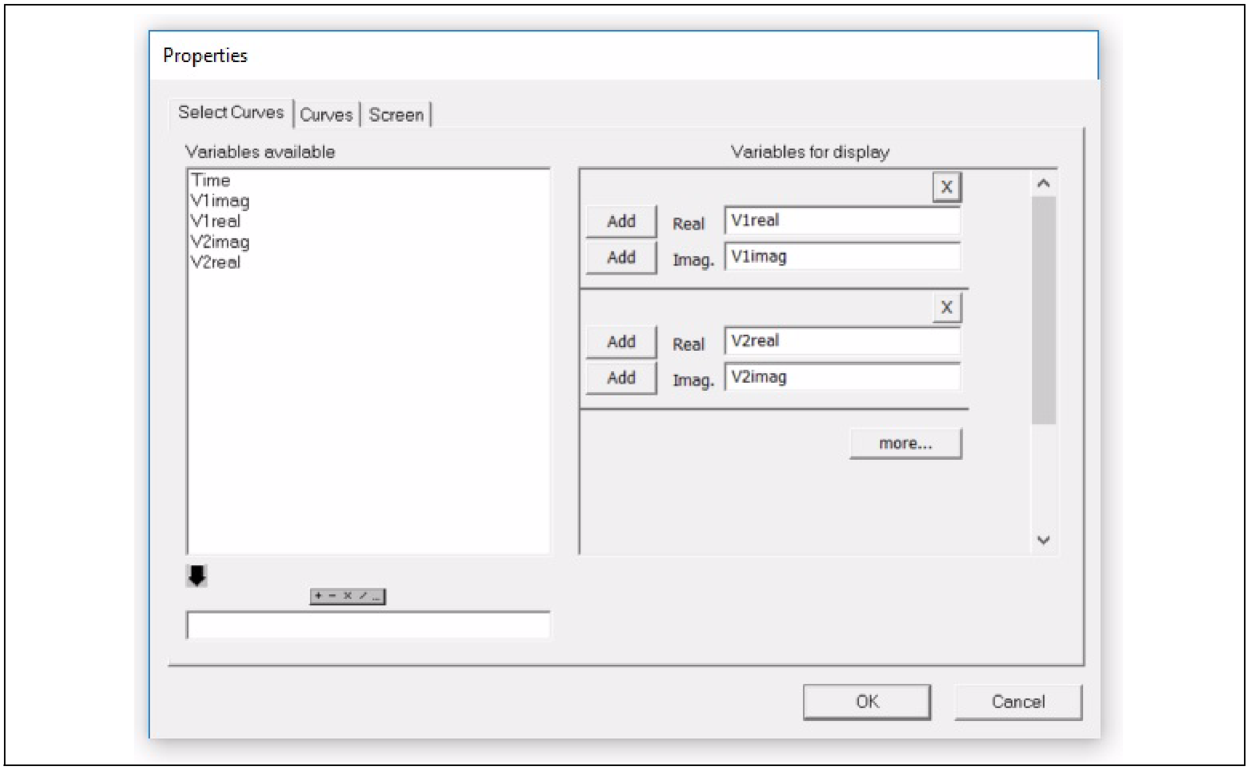
Font ABC ... □ Hide axis text

To plot a vector diagram, first perform the simulation. Then select **Plot Vector Diagram** and define the vectors. An example is used to illustrate this, as shown below.



In this circuit, the real and imaginary parts of two vectors V1 and V2 are created. The amplitude of V1 is 1 and the amplitude of V2 is 0.8. Vector V2 is leading V1 by 30 deg.

After simulation is completed, select **Screen** >> **Plot Vector Diagram**. In the dialog window, define the real and imaginary parts for vectors V1 and V2. The dialog window will appear as follows:



Properties

Variables available

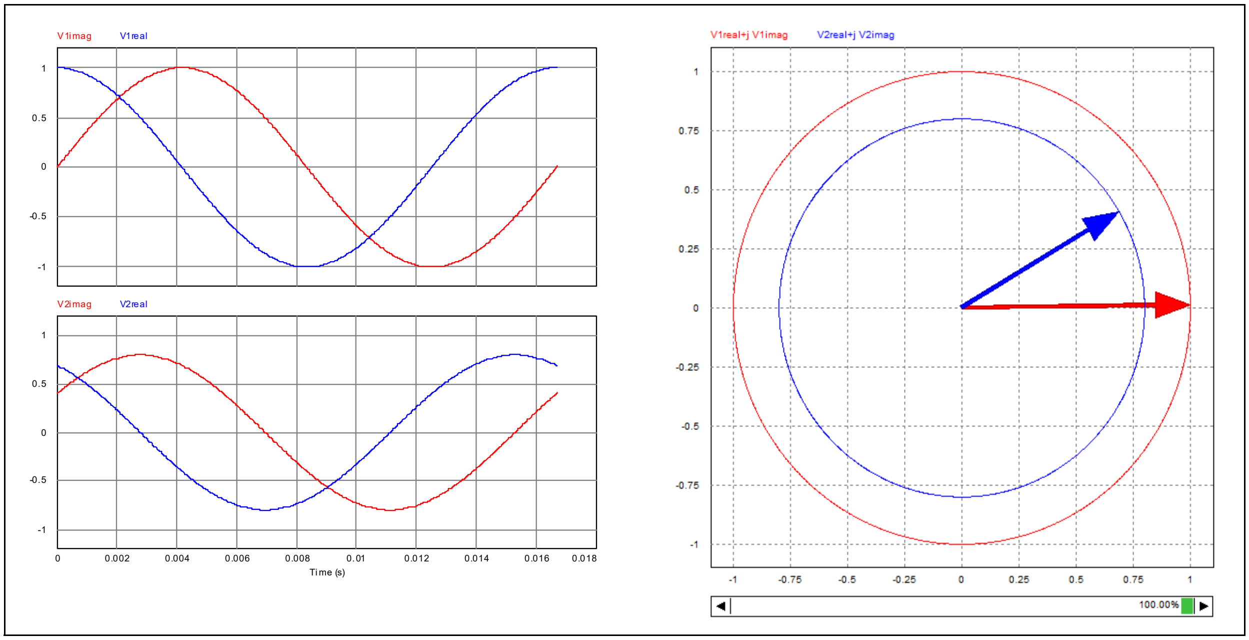
Time
V1imag
V1real
V2imag
V2real

Variables for display

Real: V1real
Imag.: V1imag

Real: V2real
Imag.: V2imag

Click on OK, and the vector plot will show as below on the right. The time-domain waveforms of the real and imaginary parts of V1 and V2 are shown below on the left.



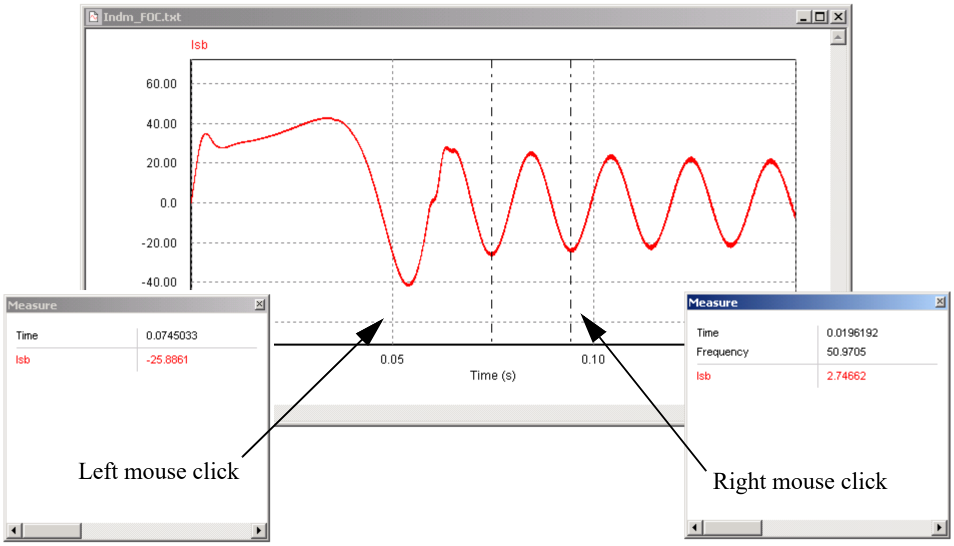
There is a sliding bar at the bottom of the vector plot. By sliding it with the left mouse, one can replay the drawing of the vector plot. The percentage shows the vector positions with respect to the final position of the drawing. For example, in this case, the time is from 0 to 0.0167 sec. A sliding bar of 0% corresponds to the moment at 0 sec., and 100% corresponds to the moment at 0.0167 sec.

3.5 Measure Menu

The **Measure Menu** has the following functions:

Measure Enter the measure mode.
Mark Data Point Mark the x and y coordinate values of the data point in a selected curve
Max Find the global maximum of a selected curve
Min Find the global minimum of a selected curve
Next Max Find the next local maximum of a selected curve
Next Min Find the next local minimum of a selected curve

The **Measure** function allows the measurement of waveforms. After **Measure** is selected, the measure dialog window will appear. By clicking the left mouse, a line will appear and the values of the waveforms will be displayed. By clicking the right mouse, another line will appear and the different between the current position and the previous position, which is marked by the left mouse, will be measured. A SIMVIEW window with the measure dialog windows in these two modes are shown below.



Time	0.0745033
Isb	-25.8861

Time	0.0196192
Frequency	50.9705
Isb	2.74662

Once **Measure** is selected, an individual curve can be selected by clicking on the pull-down menu Isb ▾ on the Measure toolbar. The functions, **Max**, **Min**, **Next Max**, **Next Min, Avg, and rms,** can be used to evaluate the curve. Note that these functions are only enabled in the **Measure** mode.

3.6 Analysis Menu

The **Analysis Menu** has the following functions:

Perform FFT Perform the FFT (Fast Fourier Transform) analysis of time-domain waveforms

Display in Time Domain Show the corresponding time-domain waveforms of FFT results

Avg Calculate the average value

Avg(|x|) Calculate the average of the absolute value

RMS Calculate the rms value

PF (power factor) Calculate the power factor of two waveforms on the screen. The screen must display two curves only. Assuming the first curve is a voltage and the second curve is a current, the power factor is defined as the real power P divided by the apparent power S produced by the voltage and current.

Note that the power factor is different from the displacement power factor, defined as cos(theta) where theta is the angle difference between the fundamental components of the voltage and current. When the voltage and current are pure sine waves without harmonics, the power factor is the same as the displacement power factor.

P (real power)	Calculate the real power of two waveforms on the screen. The screen must display two curves only.
S (apparent power)	Calculate the apparent power of two waveforms on the screen. The screen must display two curves only.
THD	Calculate the THD (total harmonic distortion)

All these functions apply to the time interval currently being displayed on the screen.

By selecting **Perform FFT**, the harmonic spectrum (both the amplitudes and angle) of the time-domain waveforms can be calculated and displayed. To display the angles of the FFT results, double click on top of the screen (or click on the **Add/Delete Curves** icon). In the *Select Curves* tab, click on the *Angle* tab, and select the angles. The name convention of angles is *Angle*(*D*)_{*name*} for the angle in deg., and *Angle*(*R*)_{*name*} for the angle in rad., and *name* is the curve name in the time domain.

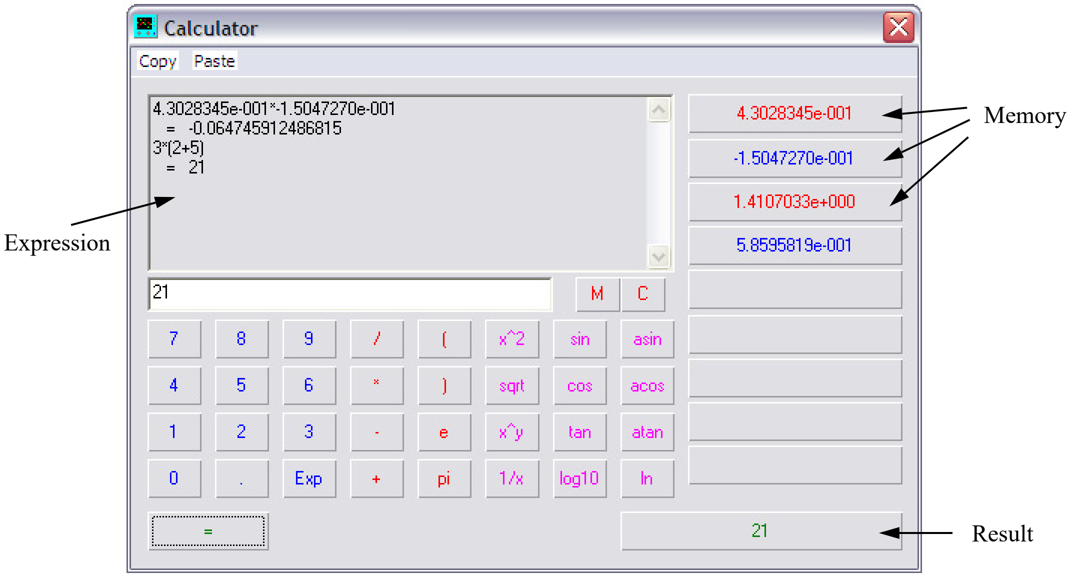
Note that, in order to obtain correct FFT results, the simulation must reach the steady state, and the data range must be restricted (using the manual range setting in the **X Axis** function) to have the integer number of the fundamental period. For example, for a 60-Hz fundamental frequency, the data length in display must be integer multiples of 1/60 sec.

3.7 View Menu

The **View Menu** has the following functions:

Zoom	Zoom into a selected region
Re-Draw	Re-draw the waveform using the auto-scale
Escape	Escape from the Zoom or Measure mode
Standard Toolbar	Enable/disable standard toolbar
Measure Toolbar	Enable/disable measure toolbar
Status Bar	Enable/disable status bar
Calculator	Launch the Calculator in Simview

The interface of the calculator is shown below.



```
4.3028345e-001*-1.5047270e-001
  =  -0.064745912486815
3*(2+5)
  =  21
```

```
21
```

```
4.3028345e-001
-1.5047270e-001
1.4107033e+000
5.8595819e-001
```

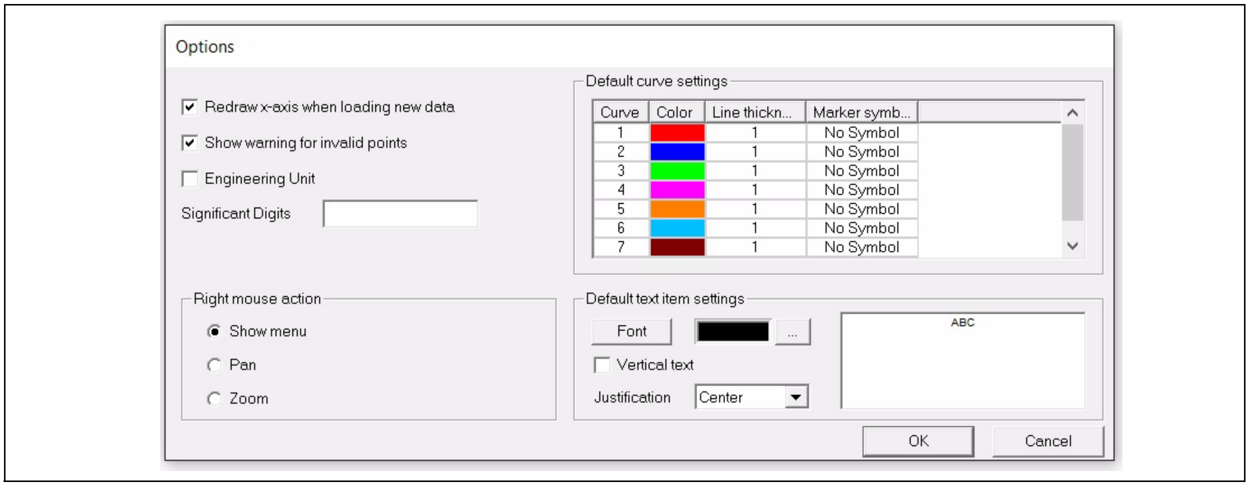
One key feature of the calculator is that it provides 9 memory spaces. By double clicking on a number in the Measure dialog window in Simview, the value will be automatically transferred to the calculator and stored in one of the memory spaces, starting from the top. In this way, data can be directly transferred to this calculator for calculation without the need to record them on a piece of paper.

3.8 Option Menu

The **Option Menu** has the following functions:

Options...	Various options can be set here.
Grid	Enable or disable the grid display
Color	Set the curves to be either Color (default) or Black and White

The **Options** dialog window is shown below.



Options

☑ Redraw x-axis when loading new data

☑ Show warning for invalid points

☐ Engineering Unit

Significant Digits

Default curve settings

Curve	Color	Line thickn...	Marker symb...
1	🟥	1	No Symbol
2	🟦	1	No Symbol
3	🟩	1	No Symbol
4	🟪	1	No Symbol
5	🟧	1	No Symbol
6	🟦	1	No Symbol
7	🟫	1	No Symbol

Right mouse action

● Show menu

○ Pan

○ Zoom

Default text item settings

Font ■ ...

☐ Vertical text

Justification: Center

Default curve settings, such as colors, line thickness, and mark symbols, are defined. Also, default text settings, such as font size, color are defined.

If the option **Redraw x-axis when loading new data** is checked, the waveform will be redrawn with the new x-axis range when new data is loaded. If this option is not checked, the x-axis range will be unchanged.

If the option **Engineering Unit** is checked, in the Measure dialog window, curve values will be displayed in engineering unit with suffix such as u, m, k, M (for example, 12.3456u). If this option is not checked, the values will be displayed in scientific unit (for example, 1.23456e-5). The value of **Significant Digits** defines the number of digits after the decimal points.

Also, the right mouse action can be set to either *Show menu*, *Pan*, or *Zoom*.

3.9 Label Menu

The **Label Menu** has the following functions:

Text	Place text on the screen
Line	Draw a line
Dotted Line	Draw a dotted line
Arrow	Draw a line with arrow

To draw a line, first select **Line** from the Label menu. Then click the left mouse at the position where the line begins, and drag the mouse while keeping the left button pressed. Dotted lines and lines with arrows are drawn in the same way.

If one is in the Zoom or Measure mode, and wishes to edit a text or a label, one should first escape from the Zoom/Measure mode by selecting "Escape" in the "View" menu.

3.10 Settings Menu

The **Settings Menu** has the following functions:

Re-Load Settings	Re-load the settings from the .ini file and apply to the current display
Save Settings	Save the current settings to a file with the same file name but with the .ini extension
Save Temporary Settings	Save the current settings temporarily. The temporary settings are not saved to any files, and are discarded when the document is closed.

Load Temporary Settings Load the temporary settings and apply to the current display

Add to Favorites Save the current settings as a favorite. When saving a favorite, one can choose to save the following settings: line color and thickness, text font, Log/dB/FFT display settings, and x and y axis ranges.

Manage Favorites Manage the favorites

When Simview loads a data file (.txt or .smv file), if the corresponding .ini file exists, it will load the settings in the .ini file.

The functions *Load Temporary Settings* and *Save Temporary Settings* are used in situations where one wants to save the settings temporarily and uses it shortly after. For example, when comparing one waveform with another, one can first save the temporary settings when displaying the first waveform. Then display the second waveform and load the temporary settings.

Favorites are a convenient way of storing particular graph settings to be used later. For example, assume that Simview shows two screens, with the top screen displaying V1 in the red color and with certain x-axis and y-axis ranges, and the bottom screen displaying V2 in the blue color with its own y-axis range. If this settings is likely to be used again in the future, the settings can be saved as a favorite and used later.

To apply a favorite to the current display, go to the Settings menu and choose the favorite from the list. Note that when applying the favorite, the number of screens currently on display must be the same as the number of screen in the favorite.

3.11 Exporting Data

FFT results can be saved to a text file. Both simulation results (*.txt) and FFT results (*.fft) are in text format and can be edited using a text editor (such as Microsoft NotePad), or exported to other software (such as Microsoft Excel).

For example, to load a simulate result file “chop-1q.txt” in Microsoft Excel, follow these steps:

- In Microsoft Excel, select **Open** from the File menu. Open the file “chop-1q.txt”.
- In the dialog window “**Text Import Wizard - Step 1 of 3**”, under **Original data type**, choose **Delimited**. Click on **Next**.
- In the dialog window “**Text Import Wizard - Step 2 of 3**”, under **Delimiters**, choose **Space**. Click on **Next**.
- In the dialog window “**Text Import Wizard - Step 3 of 3**”, under **Column data format**, choose **General**. Click on **Finish**.

4
Power Circuit Components

The following chapters provide the detailed information of the PSIM library elements. This chapter focuses on the Power circuit components.

4.1 Resistor-Inductor-Capacitor Branches

PSIM provides several combinations of resistor, inductor, capacitor networks in the element library.

- Individual resistor, inductor, and capacitor. These are multi-level models. Level-1 is basic component. Level-2 includes parasitic and leakage elements as lumped components in equivalent circuits.
- Combined R-L-C series branch
- Symmetrical 3-phase branches of resistor, inductor, capacitor, and their combinations

User can define initial conditions of inductor currents and capacitor voltages for individual component. But the initial inductor currents and capacitor voltages for three-phase branches are all set at zero.

4.1.1 Individual Resistor

Image:

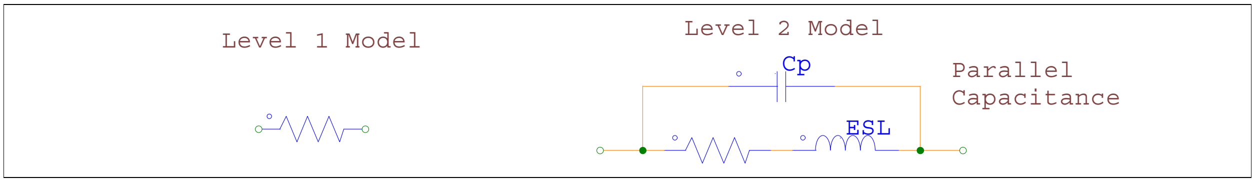
Attributes of Level-1 Model:

Parameters	Description
Resistance	Resistance, in Ohm
Voltage Rating	Voltage rating of the component, in V, for Level 2 model only.
Current Flag	Flag for branch current waveform display. If this flag is set 1, the current flowing through this component will be saved to display in SIMVIEW and Runtime Graphs. The current is positive when it flows into the dotted terminal of the branch.

Attributes of Level-2 Model:

Parameters	Description
Resistance	Resistance, in Ohm
Voltage Rating	Voltage rating of the component, in V.
Power Rating	Power rating of the component, in W.
Inductance ESL	Equivalent series inductance (ESL), in H.
Parallel Capacitance	Equivalent parallel parasitic capacitance, in F.
Current Flag	Flag for branch current waveform display. If this flag is set 1, the current flowing through this component will be saved to display in SIMVIEW and Runtime Graphs. The current is positive when it flows into the dotted terminal of the branch.

Equivalent Circuit:



4.1.2 Individual Inductor

Image:

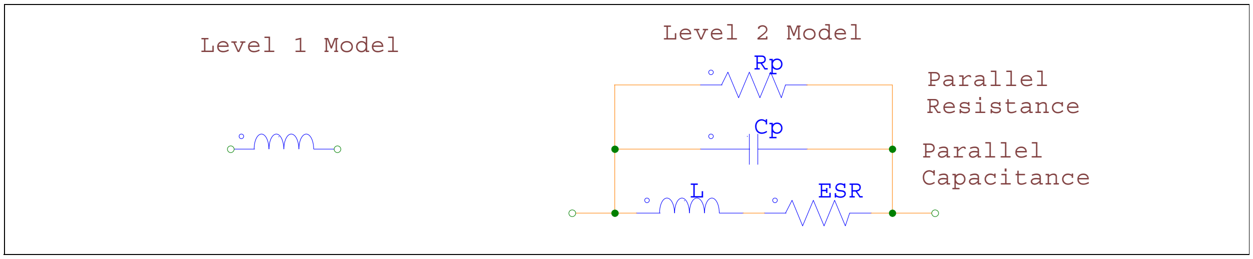
Attributes of Level-1 Model:

Parameters	Description
Inductance	Inductance, in H.
Initial Current	Initial inductor current, in A
Current Flag	Flag for branch current waveform display

Attributes of Level-2 Model

Parameters	Description
Inductance	Inductance, in H.
Peak Current Rating	Peak current rating of the inductor, in A.
Series Resistance	Equivalent series resistance (ESR), in Ohm.
Parallel Resistance	Leakage resistance in parallel with the inductor, in Ohm.
Parallel Capacitance	Parallel parasitic capacitance, in F.
Initial Current	Initial inductor current, in A
Current Flag	Flag for branch current waveform display.

Equivalent Circuit:



4.1.3 Individual and Electrolytic Capacitors

Images:

Capacitor
Capacitor (Electrolytic)

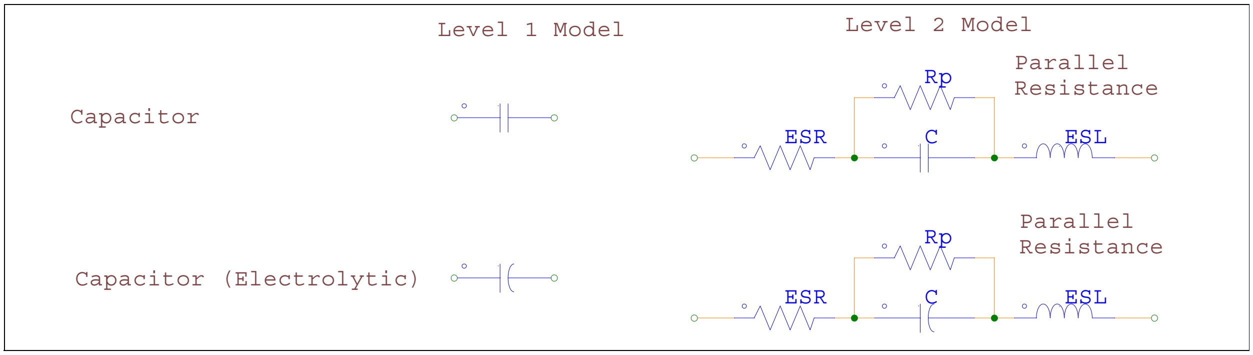
Attributes of Level-1 Model:

Parameters	Description
Capacitance	Capacitance, in F.
Initial Voltage	Initial capacitor voltage, in V
Current Flag	Flag for branch current display

Attributes of Level-2 Model:

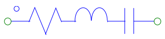
Parameters	Description
Capacitance	Capacitance, in F.
Voltage Rating	Voltage rating of the component, in V.
RMS Current Rating	RMS current rating, in A.
Resistance ESR	Equivalent series resistance (ESR), in Ohm.
Inductance ESL	Equivalent series inductance (ESL), in H.
Leakage Resistance	Leakage resistance, in Ohm.
Initial Voltage	Initial capacitor voltage, in V
Current Flag	Flag for branch current waveform display

Equivalent Circuit:



4.1.4 Combined R-L-C Branch

Image:



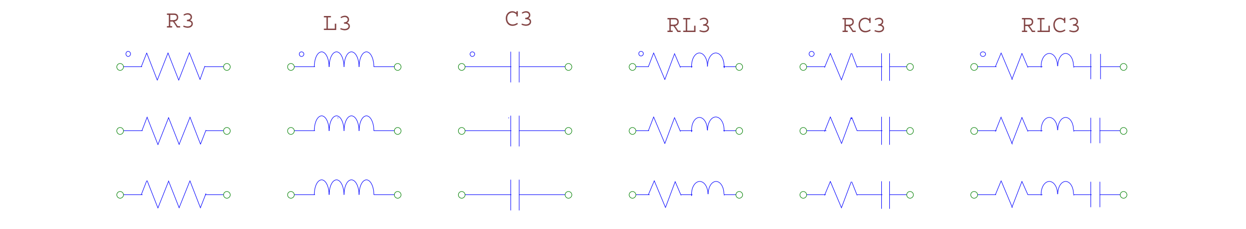
Attributes for R-L-C Branch:

Parameters	Description
Resistance	Resistance, in Ohm
Inductance	Inductance, in H
Capacitance	Capacitance, in F
Initial Current	Initial inductor current, in A
Initial Cap. Voltage	Initial capacitor voltage, in V
Current Flag	Flag for branch current waveform display.

Note: For combined R-L-C brand, at least one of the parameters R, L, or C has to be a non-zero value.

4.1.5 3-Phase R, L, C, and Combination Branches

Images:



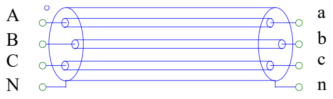
Attributes for 3-Phase R, L, C, and Combination Branches:

Parameters	Description
Resistance	Resistance, in Ohm
Inductance	Inductance, in H
Capacitance	Capacitance, in F
Current Flag_A; Current Flag_B; Current Flag_C	Current flags for Phase A, B, and C of three-phase branches, respectively. If this flag is set 1, the current flowing through this branch will be saved to the file for display in SIMVIEW. It will also be available for display in the runtime graphs (under **Simulate** >> **Runtime Graphs**). The dot in the image indicates phase A. The current is positive when it flows into the dotted terminal of the branch.

4.1.6 3-Phase AC Cable

The 3-phase ac cable model takes into account inductive coupling and capacitances between phases.

Image:



Attributes:

Parameters	Description
Cable Length	Length of the cable, in m
Operating Frequency	Operating frequency of the cable, in Hz
Resistance R_d	Positive sequence resistance R_d of the cable, in Ohm/km
Reactance X_d	Positive sequence reactance X_d of the cable, in Ohm/km
Capacitance C_d	Positive sequence capacitance C_d of the cable, in F/km
Resistance R_o	Zero sequence resistance R_o of the cable, in Ohm/km
Reactance X_o	Zero sequence reactance X_o of the cable, in Ohm/km
Capacitance C_o	Zero sequence capacitance C_o of the cable, in F/km

In the image, the bottom node *N* or *n* on each side of the cable is the cable screen. It is generally connected to the ground on the side of the cable.

The cable parameters should be available from manufacturers' datasheet. When they are not available, assuming that each phase of the cable has a resistance of R, self inductance of L, and mutual inductance of M, the positive sequence and zero sequence parameters can be calculated as follows:

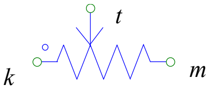
$$R_d = R$$
$$X_d = \omega \cdot (L - M)$$
$$R_o = R$$
$$X_o = \omega \cdot (L + 2M)$$

where $\omega = 2\pi f$ and f is the operating frequency in Hz.

4.1.7 Rheostat

A rheostat is a resistor with a tap.

Image:



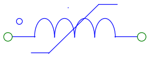
Attributes:

Parameters	Description
Total Resistance	Total resistance of the rheostat R (between Node k and m), in Ohm
Tap Position (0 to 1)	The tap position Tap. The resistance between Node k and t is: R*Tap.
Current Flag	Flag for the current that flows into Node k.

4.1.8 Saturable Inductor

A saturable inductor takes into account the saturation effect of the magnetic core.

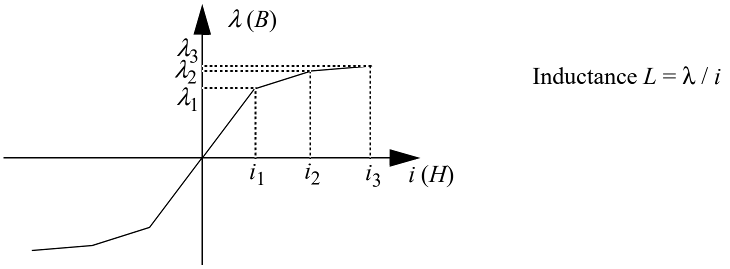
Image:



Attributes:

Parameters	Description
Current vs. Inductance	Characteristics of the current versus the inductance (i_1, L_1), (i_2, L_2), etc.
Current Flag	Flag for the current display

The nonlinear B-H curve is represented by piecewise linear approximation. Since the flux density B is proportional to the flux linkage λ and the magnetizing force H is proportional to the current i, the B-H curve can be represented by the λ-i curve instead, as shown below.



The inductance is defined as: $L = \lambda / i$, the ratio of λ vs. i at each point. The saturation characteristics are defined by a series of data points as: (i_1, L_1), (i_2, L_2), (i_3, L_3), etc.

Note that the defined saturation characteristics must be such that the flux linkage λ is monotonically increasing. That is, $L_1 * i_1 < L_2 * i_2 < L_3 * i_3$, etc.

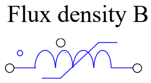
Also, similar to the saturation characteristics in the real world, the slope of each linear segment must be monotonically decreasing as the current increases.

In certain situations, circuits that contain saturable inductors may fail to converge. Connecting a very small capacitor across the saturable inductor may help the convergence.

4.1.9 Saturable Inductor (2)

A second model is provided to model a saturable inductor. As compared to the model in the previous section, this model takes into account the hysteresis loop.

Image:

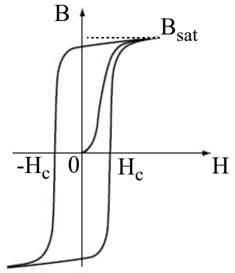


Attributes:

Parameters	Description
Number of Turns	Number of turns of the inductor winding
Winding Resistance	Winding resistance, in Ohm
Core Cross Section Area	Core cross section area, in m^2
Core Length	Effective core length, in m
Relative Permeability	Relative permeability of the core material
Flux Density at Saturation	Flux density B_{sat} at deep saturation, in T
Initial Flux Density	Initial flux density B_{init}, in T
Coercive Force	Coercive force H_c at zero flux density, in A/m
Current Flag	Flag for the current display

The extra node on the side of the inductor image gives the flux density of the core, in T. The node is a control node.

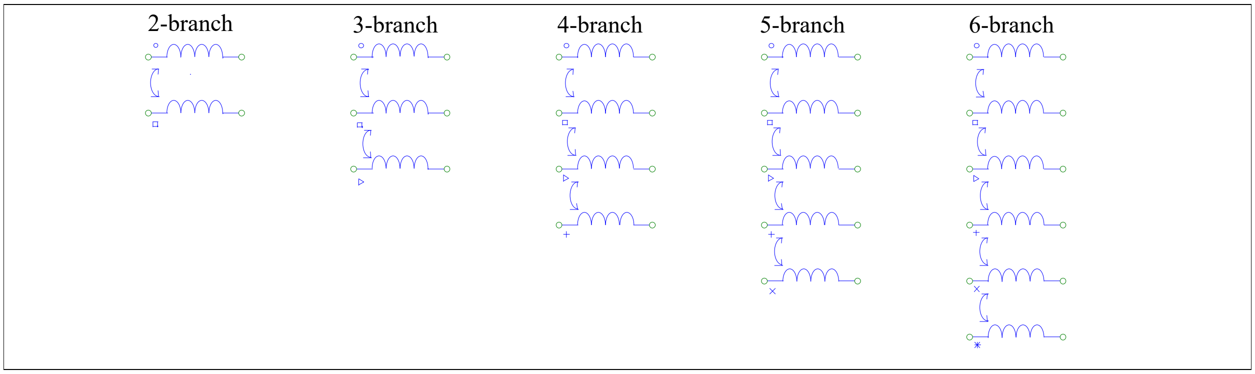
A saturable inductor *B*-*H* curve is shown below.



4.1.10 Coupled Inductors

Coupled inductors with two, three, four, five, and six branches are provided.

Images:

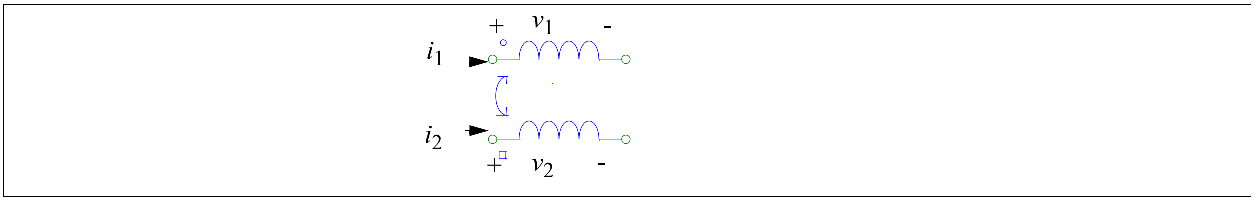


In the images, the small circle indicates the input terminal of inductor 1 while the small square, triangle, + sign, x sign, and * sign indicate the input terminal of inductor 2, 3, 4, 5, and 6.

Attributes:

Parameters	Description
L_{ii} (self)	Self inductance of the inductor *i,* in H
L_{ij} (mutual)	Mutual inductance between Inductor *i* and *j*, in H
Initial Current i_i	Initial current in Inductor *i*
Current Flag_*i*	Flag for the current printout in Inductor *i*

The following shows a coupled inductor with two branches.



Let L_{11} and L_{22} be the self-inductances of Branch 1 and 2, and L_{12} and L_{21} the mutual inductances, the branch voltages and currents have the following relationship:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

The mutual inductances between two windings are assumed to be always equal, i.e., $L_{12} = L_{21}$.

In the SPICE software, coupled inductors are defined by the self inductances and the coupling factors. For the coupled inductor with two branches, the coupling factor K is defined as:

$$K = \frac{L_{12}}{\sqrt{L_{11} \cdot L_{22}}}$$

The mutual inductance is then calculated as:

$$L_{12} = K \cdot \sqrt{L_{11} \cdot L_{22}}$$

The coupling factor has a value between 0 and 1. When the coupling factor is 1, it is perfect coupling. Note that in PSIM, perfect coupling is not allowed. That is, the coupling factor cannot be equal to 1.

Also, when the leakage inductances are ignored (i.e. the coupling factor is close to 1), the self inductances are proportional to the turns ratio squared. That is, if Branch 1 has N_1 turns, and Branch 2 has N_2 turns:

$$\frac{L_1}{L_2} = \frac{N_1^2}{N_2^2}$$

Example:

Two mutually coupled inductors have the self inductances and mutual inductance as: L_{11} = 1 mH, L_{22} = 1.1 mH, and $L_{12} = L_{21}$ = 0.9 mH. The coupling factor is: K = 0.86. The specification of this element will be:

L_{11} (self)	1m
L_{12} (mutual)	0.9m
L_{22} (self)	1.1m

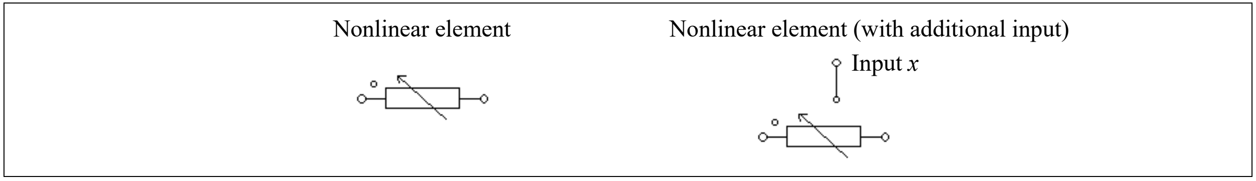
4.1.11 Nonlinear Elements

The following elements with nonlinear voltage-current relationship are provided:

- Resistance-type [$v = f(i)$]
- Resistance-type with additional input x [$v = f(i,x)$]
- Conductance-type [$i = f(v)$]
- Conductance-type with additional input x [$i = f(v,x)$]

The additional input x must be a voltage signal.

Images:


Nonlinear element

Nonlinear element (with additional input)

Attributes:

For resistance-type elements:

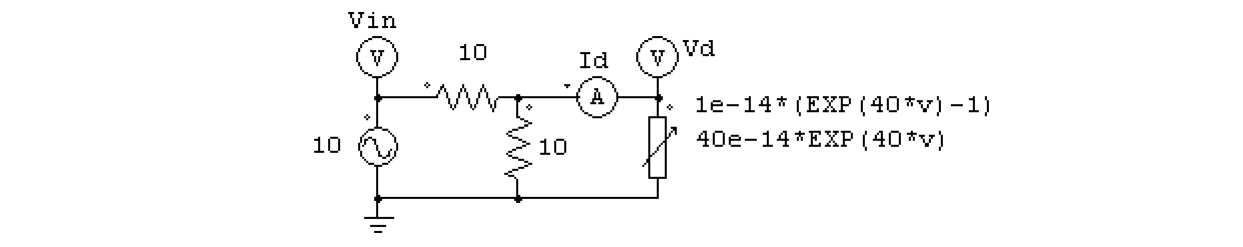
Parameters	Description
Expression $f(i)$ or $f(i,x)$	Expression of v in terms of i and x [$v = f(i)$ or $v = f(i,x)$]
Expression df/di	The derivative of the voltage v versus current i, i.e. $df(i)/di$
Initial Value i_o	The initial value of the current i
Lower Limit of i	The lower limit of the current i
Upper Limit of i	The upper limit of the current i

For conductance-type elements:

Parameters	Description
Expression $f(v)$ or $f(v,x)$	Expression of i in terms of v and x [$i = f(v)$ or $i = f(v,x)$]
Expression df/dv	Derivative of the current i versus voltage v, i.e. $df(v)/dv$
Initial Value v_o	The initial value of the voltage v
Lower Limit of v	The lower limit of the voltage v
Upper Limit of v	The upper limit of the voltage v

A good initial value and lower/upper limits will help the convergence of the solution.

Example: Nonlinear Diode



The nonlinear element (NONI) in the circuit above models a nonlinear diode. The diode current is expressed as a function of the voltage as: $i = 10^{-14} * (e^{40*v}-1)$. In PSIM, the specifications of the nonlinear element will be:

Expression *f*(*v*)	1e-14*(EXP(40*v)-1)
Expression *df*/*dv*	40e-14*EXP(40*v)
Initial Value v_o	0
Lower Limit of *v*	-1e3
Upper Limit of *v*	1

4.2 Switching Devices

There are two basic types of switching devices in PSIM. One is the switchmode type. It operates either in the cut-off region (off state) or saturation region (on state). The other is the linear type. It can operates in either cut-off, linear, or saturation region.

Switches in switchmode include the following:

- Diode and DIAC
- Thyristor and TRIAC
- Self-commutated switches:
  - npn and pnp bipolar junction transistor (BJT)
  - Insulated-Gate Bipolar Transistor (IGBT)
  - n-channel and p-channel Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET)
  - MOSFET with fixed on-resistance $R_{DS(on)}$
  - MOSFET with on-resistance $R_{DS(on)}$ as a function of junction temperature
  - Gate-Turn-Off switch (GTO)
- Bi-directional switch

Switchmode switch models are ideal. Both turn-on and turn-off transients are neglected.

A switch has an on-resistance of 10μΩ. When there is no R-L-C branch connected in parallel with the switch, a 10-MΩ resistor will be connected across the switch internally. This resistance can be viewed as the off-state resistance. In certain cases, this resistance may need to be modified. To change the off-state resistance to, for example, 100 MΩ, connect a 100-MΩ in parallel to the switch. Since PSIM sees that there is already a resistor in parallel to the switch, the 10-MΩ resistor will not be added.

Snubber circuits are not required for switches.

Linear switches include the following:

- 3-State npn and pnp bipolar junction transistor
- 3-State n-channel and p-channel MOSFET

4.2.1 Diode

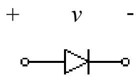
The conduction of a diode or LED is determined by circuit operating conditions. A diode has the following model levels:

- Ideal: A diode is turned on when the positive bias voltage is greater than the threshold voltage, and is turned off when the current drops to zero.
- Level-2: Includes leads parasitic inductance and reverse recovery parameters.
- SPICE Model
- SPICE Subcircuit

Note that SPICE models need the SPICE Module in the license.

The ideal and Level-2 models are described below.

Image:



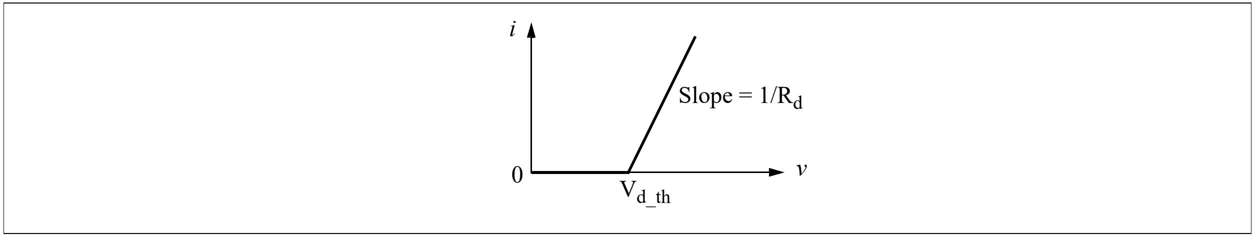
Attributes of the Ideal Model:

Parameters	Description
Forward Voltage	Forward voltage V_{d_th}, in V. A diode starts to conduct when the positive bias voltage is greater than V_{d_th}.
Resistance	On-resistance R_d, in Ohm, after it starts to conduct.
Initial Position	Flag for the initial diode position. If it is 0, the diode is off. If it is 1, the diode is on.
Current Flag	Current flag for diode current display.

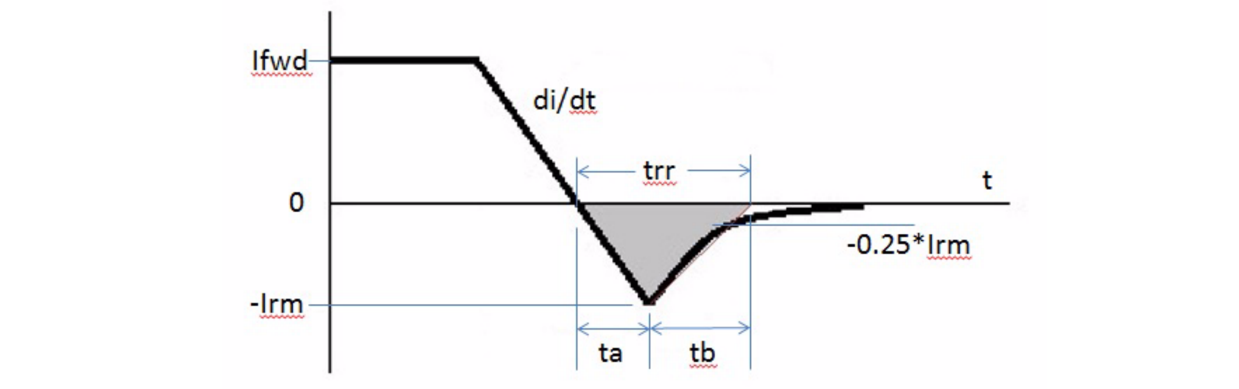
Attributes of the Level-2 Model:

Parameters	Description
Forward Voltage	Forward voltage V_{d_th}, in V. A diode starts to conduct when the positive bias voltage is greater than V_{d_th}.
Resistance	On-resistance R_d, in Ohm, after it starts to conduct.
Parasitic Inductance	Series parasitic inductance, in H
Parallel Capacitance	Parallel capacitance between the diode anode and cathode, in F. If the capacitance value is 0, the capacitor is ignored and is removed from simulation.
Forward Current	Forward current *Ifwd* under test conditions of diode reverse recovery characteristic, in A.
Peak Reverse Current	Peak reverse current *Irm* under test conditions, in A
Current Slope	Current slope *dI/dt* under test conditions, in A/sec.
Reverse Recovery Time	Reverse recovery time *trr* under test conditions, in sec.
Initial Position	Flag for the initial diode position. If it is 0, the diode is off. If it is 1, the diode is on.
Current Flag	Current flag for diode current waveform display.

Ideal Diode v-i Characteristic:



Level-2 Diode Model Reverse Recovery Parameter Definitions:



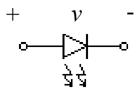
The reverse recovery time *trr* consists of two parts: *ta* and *tb*. During the period *ta*, the current increases from 0 to -*Irm*. Based on JEDEC (Joint Electron Device Engineering Council), *tb* is defined as the interval from the time of -*Irm* to the time when the straight line from -*Irm* through -0.25**Irm* (in red in the diagram) intersects with the time axis.

For further information on how to use the Level-2 model, please refer to the tutorial "Tutorial - Diode model with reverse recovery.pdf".

4.2.2 LED

A light-emitting diode (LED) emits light when it conducts. Its v-i characteristic is the same model as diode's Level-1 model.

Image:



Attributes:

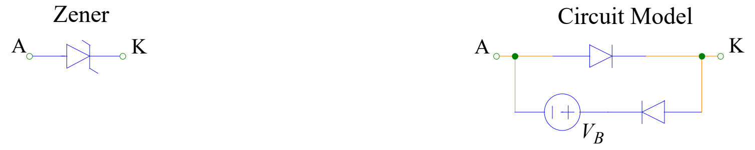
Parameters	Description
Forward Voltage	LED forward threshold voltage V_{d_th}, in V. The LED starts to conduct and emit light when the positive bias voltage is greater than V_{d_th}.
Resistance	LED on-resistance R_d, in Ohm, after it starts to conduct.
Initial Position	Flag for the initial LED position. If the flag is 0, the LED is off. If it is 1, the LED is on.
Current Flag	Current flag for LED current display.

4.2.3 Zener Diode and DIAC

Zener:

A zener diode is modeled by a circuit as shown below.

Images:



Attributes:

Parameters	Description
Breakdown Voltage	Breakdown voltage V_B of the zener diode, in V
Forward Threshold Voltage	Threshold voltage of the forward conduction (from anode to cathode), in V
Forward Resistance	On resistance of the forward conduction, in Ohm
Current Flag	Flag for zener current output (from anode to cathode)

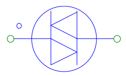
When the zener diode is positively biased, it behaviors as a regular diode. When it is reverse biased, it will block the conduction as long as the cathode-anode voltage V_{KA} is less than the breakdown voltage V_B. When V_{KA} exceeds V_B, the voltage V_{KA} will be clamped to V_B.

Note that when the zener is clamped, since the diode is modelled with an on-resistance of $10\mu\Omega$, the cathode-anode voltage will in fact be equal to: $V_{KA} = V_B + 10\mu\Omega * I_{KA}$. Therefore, depending on the value of I_{KA}, V_{KA} will be slightly higher than V_B. If I_{KA} is very large, V_{KA} can be substantially higher than V_B.

DIAC:

A DIAC is a bi-directional diode. A DIAC does not conduct until the breakover voltage is reached. After that, the DIAC goes into avalanche conduction, and the conduction voltage drop is the breakback voltage.

Image:



Attributes:

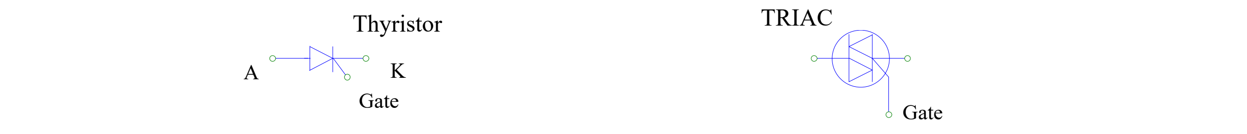
Parameters	Description
Breakover Voltage	Voltage at which breakover occurs and the DIAC begins to conduct, in V
Breakback Voltage	Conduction voltage drop, in V
Current Flag	Current flag

4.2.4 Thyristor and TRIAC

A thyristor is controlled at turn-on. The turn-off is determined by circuit conditions.

A TRIAC is a device that can conduct current in both directions. It behaviors in the same way as two opposite thyristors connected in parallel.

Images:



Attributes:

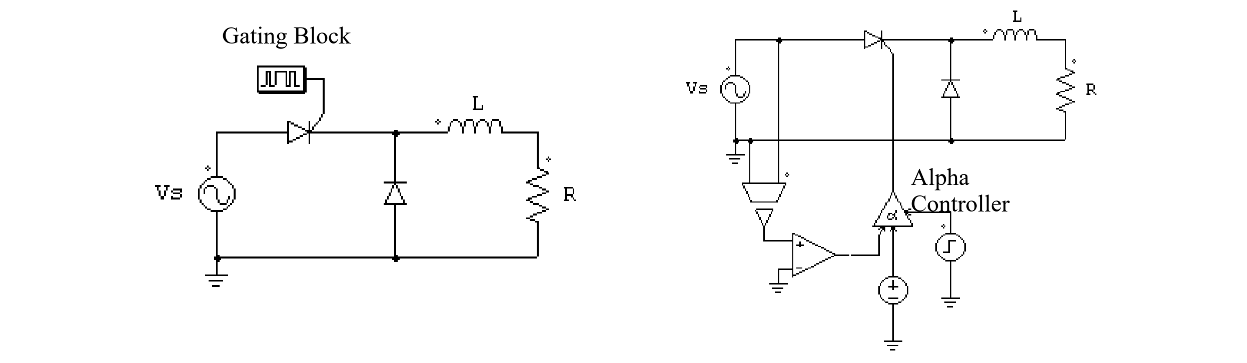
Parameters	Description
Voltage Drop	Thyristor conduction voltage drop, in V
Holding Current	Minimum conduction current below which the device stops conducting and returns to the OFF state (for thyristor only)
Latching Current	Minimum ON state current required to keep the device in the ON state after the triggering pulse is removed (for thyristor only)
Initial Position	Flag for the initial switch position (for thyristor only)
Current Flag	Flag for switch current output

Note that for the TRIAC device, the holding current and latching current are set to zero.

There are two ways to control a thyristor or TRIAC. One is to use a gating block, and the other is to use a switch controller. The gate node of a thyristor or TRIAC must be connected to either a gating block or a switch controller.

The following examples illustrate the control of a thyristor switch.

Examples: Control of a Thyristor Switch



This circuit on the left uses a switching gating block. The switching gating pattern and the frequency are pre-defined, and remain unchanged throughout the simulation. The circuit on the right uses an alpha switch controller. The delay angle alpha, in deg., is specified through the dc source in the circuit.

4.2.5 Bipolar Junction Transistor

A bipolar junction transistor (BJT) model has four levels: Ideal, 3-state, SPICE Model, and SPICE Subcircuit. The ideal and 3-state models are described below. Note that SPICE models need the SPICE Module in the license.

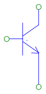
4.2.5.1 Ideal BJT

An ideal bipolar junction transistor model in PSIM is an ideal switching device. Its behavior is a little bit different from that of the device in real life. A BJT switch in PSIM will block reverse voltage (in this sense, it behaves like a GTO). Also, it is controlled by a voltage signal at the gate node, not a current.

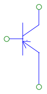
An npn transistor is turned on when the gating signal is a logic high (when a voltage of 1V or higher is applied to the gate node) and the switch is positively biased (collector-emitter voltage is positive). A pnp transistor is turned on when the gating signal is a logic low and the switch is negatively biased (collector-emitter voltage is negative).

Images:

npn
Transistor



pnp
Transistor

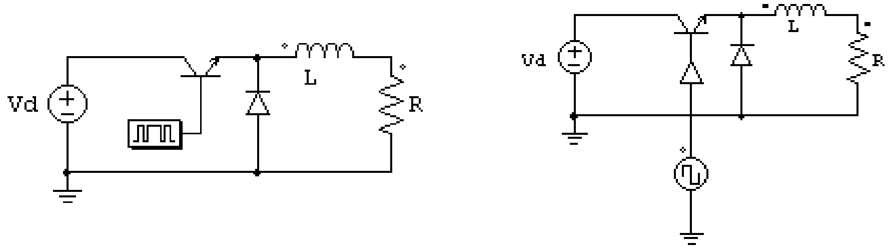


Attributes:

Parameters	Description
Saturation Voltage	The saturation voltage Vce_sat for npn, or Vec_sat for pnp, in V
Initial Position	Initial switch position flag (0: off; 1: on)
Current Flag	Switch current flag (0: no display; 1: display)

Example: Control of a npn Bipolar Junction Transistor

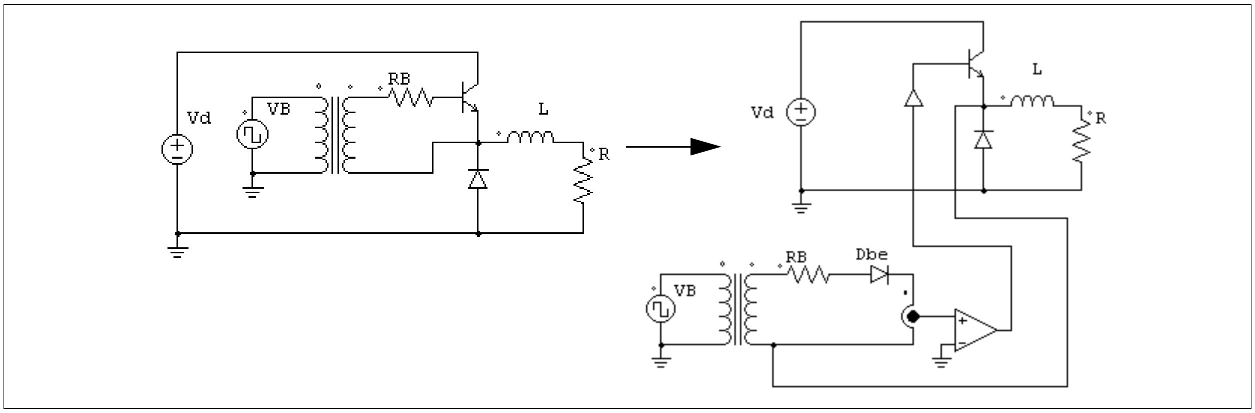
The circuit on the left uses a gating block, and the one on the right uses an on-off switch controller.



If the switch is controlled by a control circuit signal, as shown in the circuit on the right side, an on-off switch controller must be used at the gate node.

The following shows another example of controlling the BJT switch. The circuit on the left shows how a BJT switch is controlled in the real life. In this case, the gating voltage V_B is applied to the transistor base drive circuit through a transformer, and the base current determines the conduction state of the transistor.

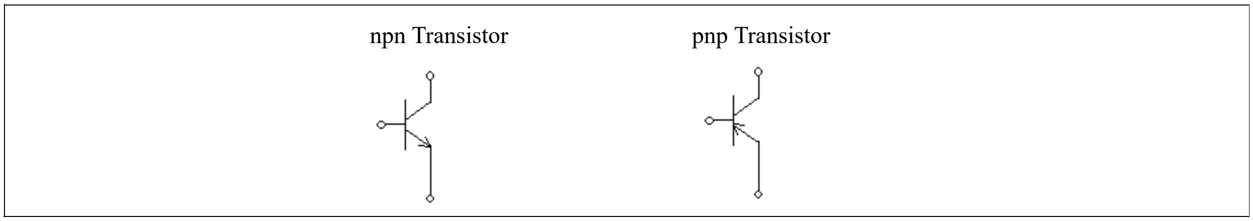
This circuit can be modelled and implemented in PSIM as shown on the right. A diode, D_{be}, with a conduction voltage drop of 0.7V, is used to model the pn junction between the base and the emitter. When the base current exceeds 0 (or a certain threshold value, in which case the base current will be compared to a dc source), the comparator output will be 1, applying the turn-on pulse to the transistor through the on-off switch controller.



4.2.5.2 3-State BJT

A 3-state BJT model is a linear switch. It can operate in either cut-off (off state), linear, or saturation (on state) region.

Images:


npn Transistor

pnp Transistor

Attributes:

Parameters	Description
Current Gain beta	Transistor current gain β, defined as: $\beta = I_c / I_b$
Bias Voltage V_r	Forward bias voltage, in V, between base and emitter for the npn transistor, or between emitter and base for the pnp transistor.
$V_{ce,sat}$ [or $V_{ec,sat}$ for pnp]	Saturation voltage, in V, between collector and emitter for the npn transistor, and between emitter and collector for the pnp transistor.

The properties of an npn transistor in these regions are:

- Cut-off region: $V_{be} < V_r$; $I_b = 0$; $I_c = 0$
- Linear region: $V_{be} = V_r$; $I_c = \beta * I_b$; $V_{ce} > V_{ce,sat}$
- Saturation region: $V_{be} = V_r$; $I_c < \beta * I_b$; $V_{ce} = V_{ce,sat}$

where V_{be} is the base-emitter voltage, V_{ce} is the collector-emitter voltage, and I_c is the collector current.

The properties of a pnp transistor in these regions are:

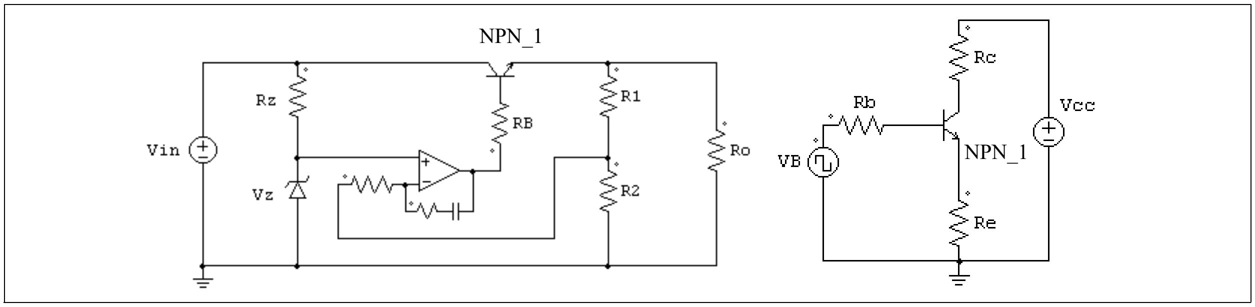
- Cut-off region: $V_{eb} < V_r$; $I_b = 0$; $I_c = 0$
- Linear region: $V_{eb} = V_r$; $I_c = \beta * I_b$; $V_{ec} > V_{ee,sat}$
- Saturation region: $V_{eb} = V_r$; $I_c < \beta * I_b$; $V_{ec} = V_{ec,sat}$

where V_{ee} is the emitter-base voltage, V_{ec} is the emitter-collector voltage, and I_c is the collector current.

Note that for 3-state npn/pnp BJT, the base node (gate node) is a power node, and must be connected to a power circuit component (such as a resistor or a source). It can not be connected to a gating block or a switch controller.

Examples: Circuits Using the Linear BJT

Examples below illustrate the use of linear BJT. The circuit on the left is a linear voltage regulator circuit, and the transistor operates in the linear mode. The circuit on the right is a simple test circuit.



4.2.6 MOSFET

A MOSFET model has 6 levels: Ideal, Level-1, and Level2, SPICE Model, SPICE Subcircuit, SPICE Subcircuit (4-pin), and SPICE Subcircuit (5-pin). The ideal, Level-1, and Level-2 models are described in the sections below. Note that SPICE models need the SPICE Module in the license.

4.2.6.1 Ideal MOSFET

An ideal MOSFET consists of an active switch with an anti-parallel diode. The MOSFET is turned on when the gating signal is a logic high (when a voltage of 1V or higher is applied to the gate node) and the switch is positively biased (drain-source voltage is positive). It is turned off whenever the gating signal is low or the current drops to zero.

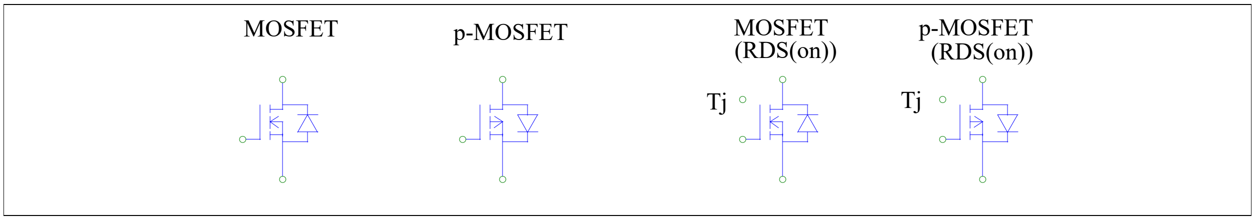
The p-channel MOSFET is turned on when the gating signal is a logic low and it is negatively biased (drain-source voltage is negative).

The switches *MOSFET* (*RDS*(*on*)) and *p-MOSFET* (*RDS*(*on*)) have on-resistance as a function of the junction temperature. In the images of *MOSFET* (*RDS*(*on*)) and *p-MOSFET* (*RDS*(*on*)), the node on top of the gate node is for defining the junction temperature. A voltage signal at this node will define the junction temperature T_j in ^{o}C. For example, a voltage value of 25V would mean a junction temperature of 25^{o}C. The on-resistance of the MOSFET is a function of the junction temperature, as expressed below:

$$R_{DS(on)} = R_{DS(on)\_b} \cdot (1 + K_T \cdot (T_j - T_{j\_b}))$$

A switch can be controlled by either a gating block or a switch controller. They must be connected to the gate (base) node of the switch.

Images:



Attributes for *MOSFET* and *p-MOSFET*:

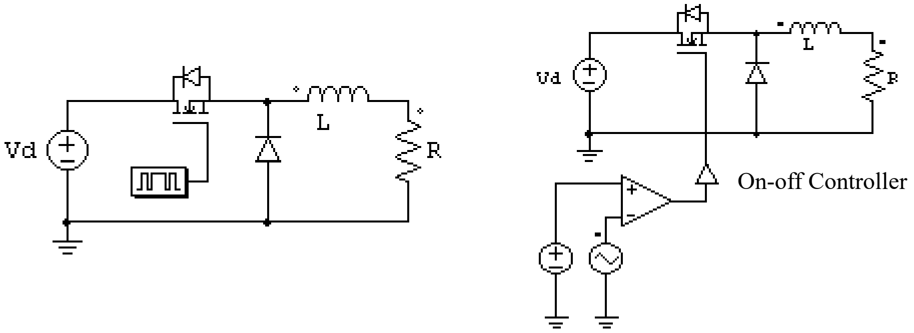
Parameters	Description
On Resistance	The on resistance Rds_on of the MOSFET, in Ohm
Diode Forward Voltage	Anti-parallel diode forward threshold voltage, in V

Diode Resistance	Anti-parallel diode on resistance, in Ohm
Initial Position	Initial switch position flag for the transistor only (0: off; 1: on)
Current Flag	Switch current flag for the whole module (the transistor plus the diode) (0: no display; 1: display)

Attributes for *MOSFET* (*RDS(on)*) and *p-MOSFET* (*RDS(on)*):

Parameters	Description
Tj at Test	Junction temperature T_{j_b} at test, in ^oC
RDS(on) at Test	On-resistance $R_{DS(on)\_b}$ at test, in Ohm
Temperature Coefficient	Temperature coefficient K_T of the on-resistance, in 1/^oC
Diode Forward Voltage	Anti-parallel diode forward threshold voltage, in V
Diode Resistance	Anti-parallel diode on resistance, in Ohm
Initial Position	Initial switch position flag for the transistor only (0: off; 1: on)
Current Flag	Switch current flag for the whole module (the transistor plus the diode)

The following examples illustrate the control of a MOSFET switch.



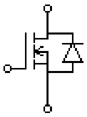
The circuit on the left uses a gating block, and the one on the right uses an on-off switch controller. The gating signal is determined by the comparator output.

4.2.6.2 Level-1 MOSFET

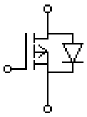
A Level-1 MOSFET is a linear switch that can operate in three states: cut-off, active, or ohmic state.

Images:

MOSFET



p-MOSFET



Attributes:

Parameters	Description
On Resistance	On resistance $R_{DS(on)}$ of the MOSFET, in Ohm
Threshold Voltage $V_{gs(th)}$	Threshold voltage of the gate-to-source voltage, in V, beyond which the MOSFET starts to conduct.
Transconductance g_m	Transconductance g_m of the MOSFET.
Diode Forward Voltage	Forward voltage of the anti-parallel diode, in V
Diode Resistance	On resistance of the anti-parallel diode, in Ohm

A linear MOSFET device is controlled by the gate-to-source voltage V_{gs}. It can operate in one of the three regions: cut-off (off state), active, and ohmic region (on state).

The properties of an n-channel MOSFET device in these regions are:

- Cut-off region: $V_{gs} < V_{gs(th)}$; $I_d = 0$
- Active region: $V_{gs} > V_{gs(th)}$ and $V_{gs} - V_{gs(th)} < V_{ds}$; $I_d = g_m*(V_{gs} - V_{gs(th)})$
- Ohmic region: $V_{gs} > V_{gs(th)}$ and $V_{gs} - V_{gs(th)} > V_{ds}$; $I_d = V_{ds} / R_{DS(on)}$

where V_{gs} is the gate-source voltage, V_{ds} is the drain-source voltage, and I_d is the drain current.

The properties of a p-channel MOSFET device in these regions are:

- Cut-off region: $V_{gs} > V_{gs(th)}$; $I_d = 0$
- Active region: $V_{gs} < V_{gs(th)}$ and $V_{gs} - V_{gs(th)} > V_{ds}$; $I_d = g_m*(V_{gs} - V_{gs(th)})$
- Ohmic region: $V_{gs} < V_{gs(th)}$ and $V_{gs} - V_{gs(th)} < V_{ds}$; $I_d = V_{ds} / R_{DS(on)}$

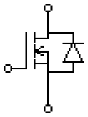
Note that for MOSFET, the gate node is a power node, and must be connected to a power circuit component (such as a resistor or a source). It can not be connected to a gating block or a switch controller.

4.2.6.3 Level-2 MOSFET

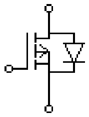
A Level-2 MOSFET takes the intrinsic and parasitic components into consideration.

Images:

MOSFET



p-MOSFET



Attributes:

Parameters	Description
Vbreakdown (drain-source)	Maximum drain-source voltage without causing avalanche breakdown, in V.
On Resistance	On resistance $R_{DS(on)}$ of the MOSFET, in Ohm
Gate Threshold Voltage $V_{gs(th)}$	Threshold voltage of the gate-to-source voltage, in V, above which, the MOSFET starts to conduct.
Internal Gate Resistance	Internal gate resistance, in Ohm.
Transconductance g_m	Transconductance g_m of the MOSFET.
Capacitance Cgs	Gate-to-source intrinsic capacitance, in F.

Capacitance Cgd	Gate-to-drain intrinsic capacitance, in F.
Capacitance Cds	Output capacitance between drain and source, in F.
Diode Forward Voltage	Forward voltage of the anti-parallel diode, in V
Diode Resistance	On resistance of the anti-parallel diode, in Ohm

In the Level-2 MOSFET model, the capacitances C_{gs}, C_{gd}, and C_{ds} can be roughly calculated from the input capacitance C_{iss}, C_{oss}, and C_{rss} from the datasheet as below:

$$C_{gs} = C_{iss} - C_{rss}$$
$$C_{gd} = C_{rss}$$
$$C_{ds} = C_{oss} - C_{rss}$$

These capacitance values may need to be adjusted to obtain the desired transient response.

Note that the gate node is a power node, and must be connected to a power circuit component (such as a resistor or a source). It can not be connected to a gating block or a switch controller.

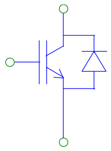
4.2.7 IGBT

An IGBT model has 3 levels: Ideal, Level-2, and SPICE Subcircuit. The ideal and Level-2 models are described in the sections below. Note that the SPICE model needs the SPICE Module in the license.

4.2.7.1 Ideal IGBT

An ideal IGBT switch consist of an active switch with an anti-parallel diode. It is turned on when the gating signal is high (when a voltage of 1V or higher is applied to the gate node) and the switch is positively biased (collector-emitter voltage is positive). It is turned off whenever the gating signal is low or the current drops to zero.

Image:



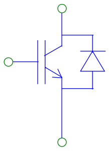
Attributes:

Parameters	Description
Saturation Voltage	Saturation voltage Vce_sat of the IGBT, in V
Transistor Resistance	Transistor on resistance, in Ohm
Diode Forward Voltage	Anti-parallel diode forward threshold voltage, in V
Diode Resistance	Anti-parallel diode on resistance, in Ohm
Initial Position	Initial switch position flag for the transistor only (0: off; 1: on)
Current Flag	Switch current flag for the whole module (the transistor plus the diode)

4.2.7.2 Level-2 IGBT

A Level-2 IGBT model takes into account the turn-on and turn-off transient.

Image:



Attributes:

Parameters	Description
Maximum Vce	Maximum collector-emitter voltage Vce rating, in V
Maximum Vec	Maximum emitter-collector voltage Vec rating, in V. If IGBT has an anti-parallel diode, this voltage will be the diode forward conduction threshold voltage.
Gate Threshold Voltage	Gate threshold voltage Vge_th, in V
Transconductance	Transconductance of the IGBT, in S
Fall Time	Fall time of the current when IGBT is turned off, in sec.
Capacitance Cies	Input capacitance Cies, in F
Capacitance Coes	Output capacitance Coes, in F
Capacitance Cres	Reverse transfer capacitance Cres, in F
Rce_on	Collector-emitter on resistance Rce_on, in Ohm
Vce_threshold	Collector-emitter threshold voltage Vce_th, in V
Internal Gate Resistance	Internal gate resistance Rgate, in Ohm
Current Flag	Switch current flag for the whole module (the transistor plus the diode)

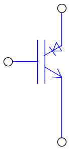
For more information on how to use this model, please refer to the tutorial "Tutorial - IGBT Level-2 model.pdf".

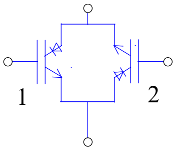
4.2.8 IGBT-RB

An IGBT-RB switch consist of an active switch with reverse blocking capability and without an anti-parallel diode. It is turned on when the gating signal is high (when a voltage of 1V or higher is applied to the gate node) and the switch is positively biased (collector-emitter voltage is positive). It is turned off whenever the gating signal is low or the current drops to zero.

Two versions of IGBT-RB are provided: a single switch, and a dual package with two back-to-back switches.

Images:





Attributes:

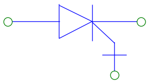
Parameters	Description
Saturation Voltage	Saturation voltage Vce_sat of the IGBT, in V
Transistor Resistance	Transistor on resistance, in Ohm
Initial Position or Initial Position 1 *or* 2	Initial switch position flag for the switch, or in the case of the dual package, for Switch 1 or 2 (0: off; 1: on)
Current Flag	Switch current flag for the whole module (the transistor plus the diode)

For the dual package, when the element is placed in the upright position, Switch 1 is on the left and Switch 2 is on the right.

4.2.9 GTO

A GTO switch is a symmetrical device with both forward-blocking and reverse-blocking capabilities.

Images:



Attributes:

Parameters	Description
Voltage Drop	GTO conduction voltage drop, in V
Initial Position	Initial switch position flag (0: off; 1: on)
Current Flag	Switch current flag (0: no display; 1: display)

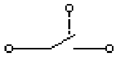
4.2.10 Bi-Directional Switches

A bi-directional switch conducts current in both directions.

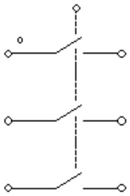
Three types of bi-directional switches are provided: single-phase switch, three-phase switch, and push button switch.

Images:

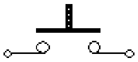
Single switch



3-phase switch



Push button switch



Attributes:

Parameters	Description
Initial Position	Initial switch position flag (for single switch only).
Initial Position (for Phase *A*/*B*/*C*)	Initial switch position for Phase *A*, or *B*, or *C*.

Switch Position	Switch position can be either "On" or "Off" (for push button switch only)
Current Flag	Switch current flag (for single switch only)
Current Flag (for Phase *A*/*B*/*C*)	Switch current flag for Phase *A*, or *B*, or *C*.

In the 3-phase switch image, the phase with a dot is Phase A.

For single and 3-phase bi-directional switches, they are on when the gating signal is high, and are off when the gating signal is low, regardless of the voltage bias conditions. For the push button switch, the switch position is set directly from the parameter input.

4.2.11 Switch Gating Block

A switch gating block defines the gating pattern of a switch or a switch module. The gating pattern can be specified either directly (the element is called *Gating Block* in the library) or in a text file (the element is called *Gating Block* (*file*) in the library).

Note that a switch gating block can be connected to the gate node of a switch ONLY. It can not be connected to any other elements.

Image:

Attributes:

Parameters	Description
Frequency	Operating frequency of the switch or switch module connected to the gating block, in Hz
No. of Points	Number of switching points (for the *Gating Block* element only)
Switching Points	Switching points, in deg. If the frequency is zero, the switching points is in second. (for the *Gating Block* element only)
File for Gating Table	Name of the file that stores the gating table (for the *Gating Block* (*file*) element only)

The number of switching points is defined as the total number of switching actions in one period. Each turn-on or turn-off action is counted as one switching point. For example, if a switch is turned on and off once in one cycle, the number of switching points will be 2.

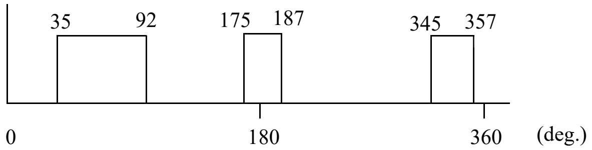
For the *Gating Block* (*file*) element, the file for the gating table must be in the same directory as the schematic file. The gating table file has the following format:

```
n
G1
G2
... ...
Gn
```

where G1, G2, ..., Gn are the switching points.

Example:

Assume that a switch operates at 2000 Hz and has the following gating pattern in one period:



The specification of the *Gating Block* element for this switch will be:

Frequency	2000.
No. of Points	6
Switching Points	35. 92. 175. 187. 345. 357.

The gating pattern has 6 switching points (3 pulses). The corresponding switching angles are 35^o, 92^o, 175^o, 187^o, 345^o, and 357^o, respectively.

If the *Gating Block* (*file*) element is used instead, the specification will be:

Frequency	2000.
File for Gating Table	test.tbl

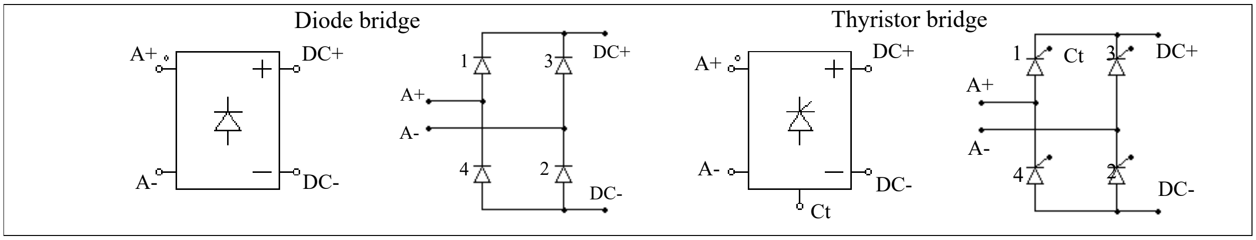
The file "test.tbl" will contain the following:

```
6
35.
92.
175.
187.
345.
357.
```

4.2.12 Single-Phase Switch Modules

Built-in single-phase diode bridge module and thyristor bridge module are provided. The images and internal connections of the modules are shown below.

Images:



Attributes:

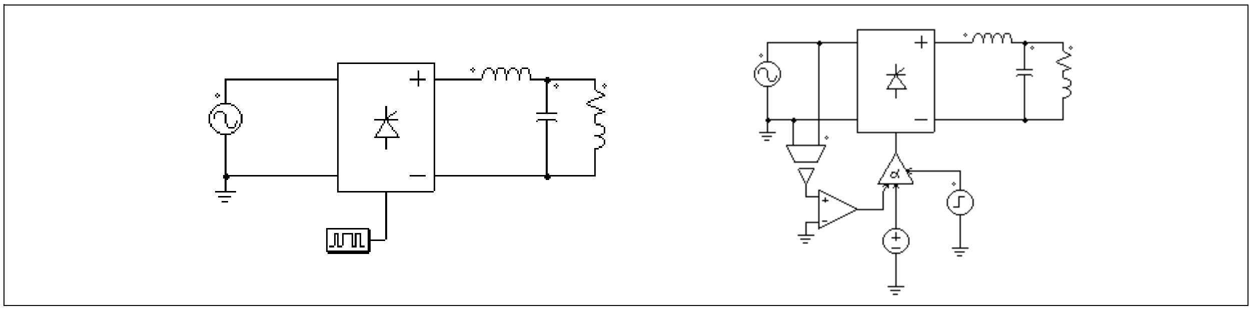
Parameters	Description
Diode Threshold Voltage *or* Voltage Drop	Threshold voltage of the diode, or forward voltage drop of the thyristor, in V
Diode Resistance	On resistance of the diode, in Ohm (for diode bridges only)
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Node C_t at the bottom of the thyristor module is the gating control node for Switch 1. For the thyristor module, only the gating signal for Switch 1 needs to be specified. The gating signals for other switches will be derived internally in the program.

Similar to the single thyristor switch, a thyristor bridge can also be controlled by either a gating block or an alpha controller, as shown in the following examples.

Examples: Control of a Thyristor Bridge

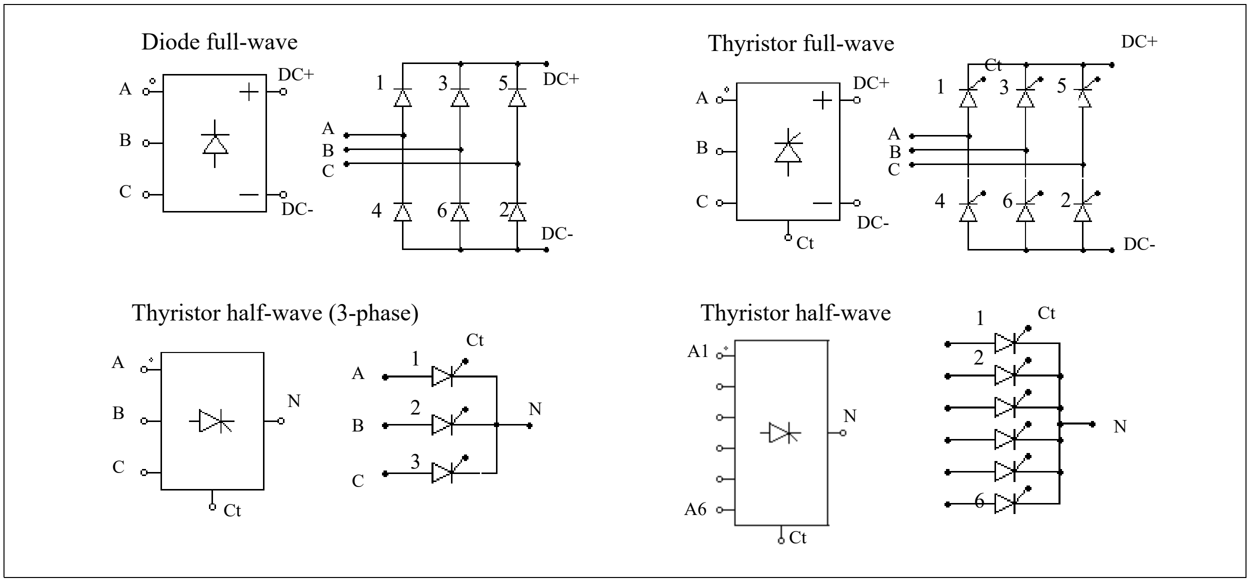
The gating signal for the circuit on the left is specified through a gating block, and the gating signal for the circuit on the right is provided through an alpha controller. A major advantage of the alpha controller is that the delay angle alpha of the thyristor bridge, in deg., can be directly controlled.

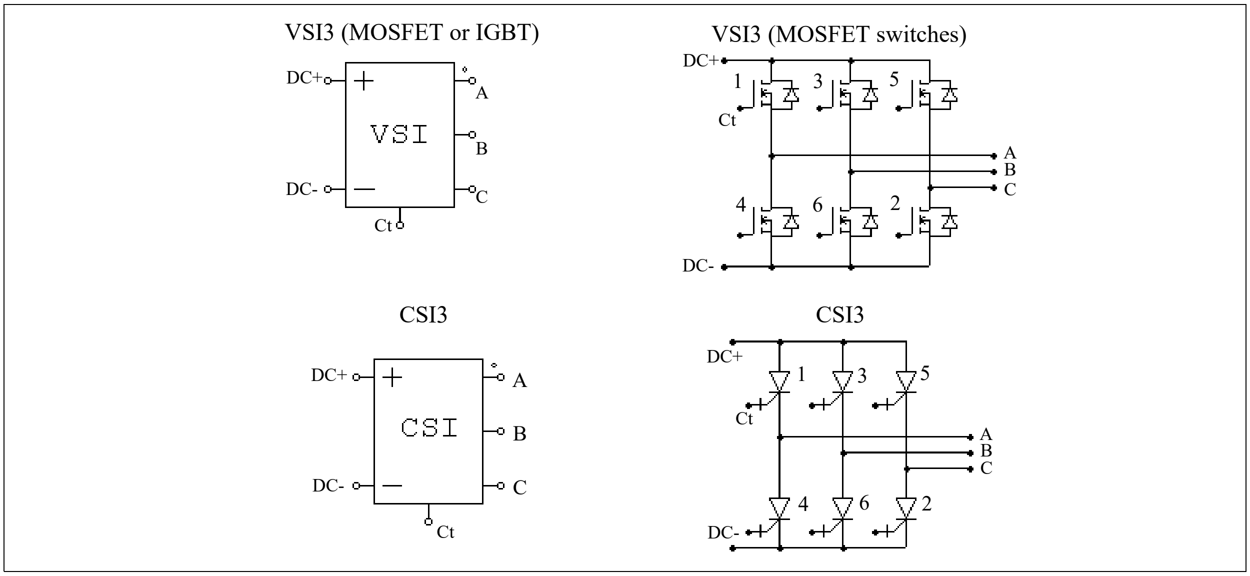


4.2.13 3-Phase Switch Modules

The following figure shows 3-phase switch modules and the internal circuit connections. A 3-phase voltage source inverter module VSI3 consists of either MOSFET-type or IGBT-type switches. A current source inverter module CSI3 consists of GTO-type switches, or equivalently IGBT in series with diodes.

Images:





Attributes for the diode bridge:

Parameters	Description
Diode Threshold Voltage	Threshold voltage drop of the diode, in V
Diode Resistance	On resistance of the diode, in Ohm
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Attributes for thyristor bridges:

Parameters	Description
Voltage Drop	Conduction voltage drop of the thyristor, in V
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Attributes for VSI3 bridge (with MOSFET switches):

Parameters	Description
On Resistance	On resistance of the MOSFET transistor, in Ohm
Diode Threshold Voltage	Threshold voltage drop of the diode, in V
Diode Resistance	On resistance of the diode, in Ohm
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Attributes for VSI3 bridge (with IGBT switches):

Parameters	Description
Saturation Voltage	Saturation voltage Vce_sat of the IGBT transistor, in Ohm
Transistor Resistance	On resistance of the IGBT transistor, in Ohm
Diode Threshold Voltage	Threshold voltage drop of the anti-parallel diode, in V
Diode Resistance	On resistance of the anti-parallel diode, in Ohm
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Attributes for the CSI3 bridge:

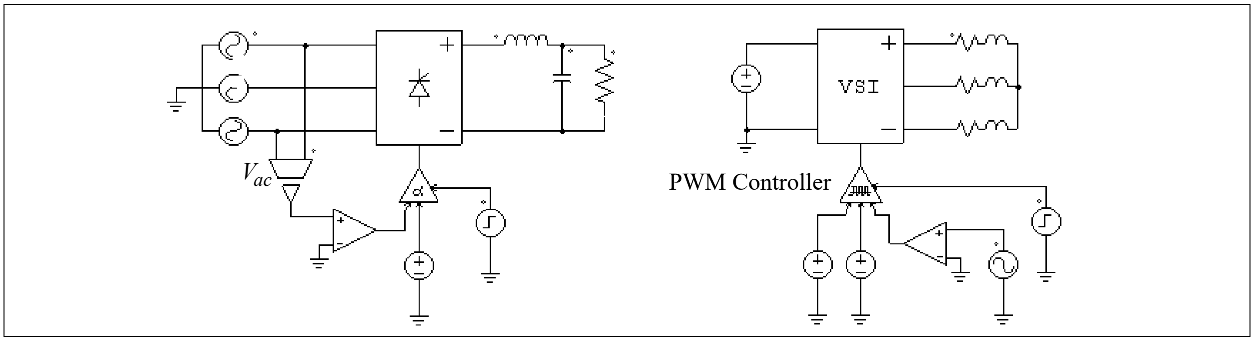
Parameters	Description
Voltage Drop	Forward conduction voltage drop of the switch, in V
Resistance	Forward on resistance of the switch, in Ohm
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Similar to single-phase modules, only the gating signal for Switch 1 need to be specified for three-phase modules. Gating signals for other switches will be automatically derived. For the 3-phase half-wave thyristor bridge, the phase shift between two consecutive switches is 120^o. For all other bridges, the phase shift is 60^o.

Thyristor bridges can be controlled by an alpha controller. Similarly, voltage/current source inverters can be controlled by a PWM lookup table controller.

The following examples illustrate the control of three-phase thyristor and voltage source inverter modules.

Example: Control of 3-Phase Thyristor and VSI Modules



The thyristor circuit on the left uses an alpha controller. For a three-phase circuit, the zero-crossing of the voltage V_{ac} corresponds to the moment when the delay angle alpha is equal to zero. This signal is used to provide synchronization to the controller.

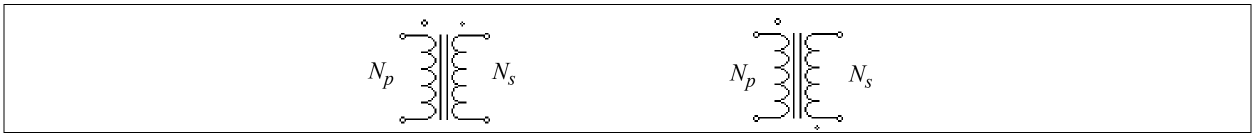
The circuit on the right uses a PWM lookup table controller. The PWM patterns are stored in a lookup table in a text file. The gating pattern is selected based on the modulation index. Other inputs of the PWM lookup table controller include the delay angle, the synchronization, and the enable/disable signal. A detailed description of the PWM lookup table controller is given in the *Switch Controllers* section.

4.3 Transformers

4.3.1 Ideal Transformer

An ideal transformer has no losses and no leakage flux.

Images:



The winding with the larger dot is the primary, and the other winding is the secondary.

Attributes:

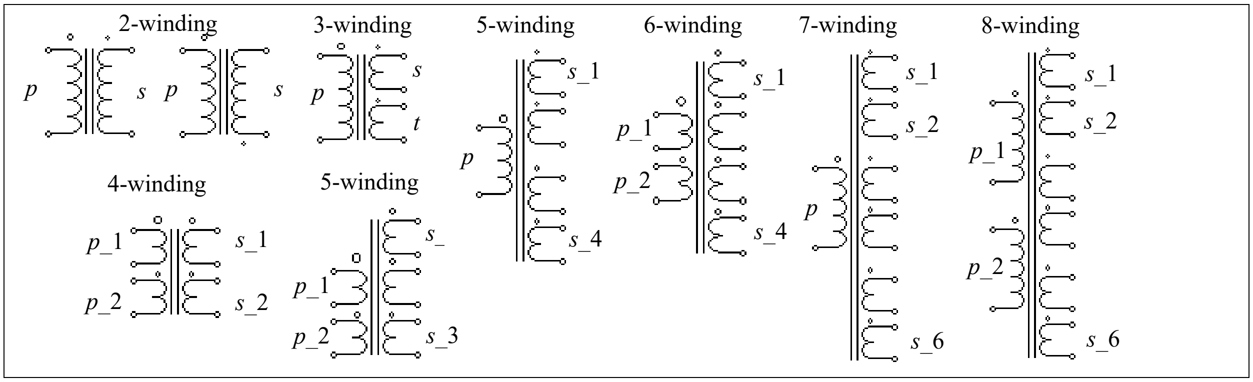
Parameters	Description
Np (primary)	No. of turns of the primary winding
Ns (secondary)	No. of turns of the secondary winding

Since the turns ratio is equal to the ratio of the rated voltages, the number of turns can be replaced by the rated voltage at each side.

4.3.2 Single-Phase Transformers

Single-phase transformers with 1 or 2 primary windings and 1 to 6 secondary windings are provided.

Images:



In the images, *p* refers to primary, *s* refers to secondary, and *t* refers to tertiary. The winding with the largest dot is the primary winding or first primary winding. For the multiple winding transformers, the sequence of the windings is from the top to the bottom. For the transformers with 2 or 3 windings, the attributes are as follows.

Attributes:

Parameters	Description
R_p (primary); R_s (secondary); R_t (tertiary)	Resistance of the primary/secondary/tertiary winding, in Ohm
L_p (pri. leakage); L_s (sec. leakage); L_t (ter. leakage)	Leakage inductance of the primary/secondary/tertiary winding, in H (seen from the primary)
L_m (magnetizing)	Magnetizing inductance, in H
N_p (primary); N_s (secondary); N_t (tertiary)	No. of turns of the primary/secondary/tertiary winding

All the resistances and inductances are referred to the primary winding side. If there are multiple primary windings, they are referred to the first primary winding side.

For a transformer with more than 1 primary or more than 3 secondary windings, the attributes are as follows.

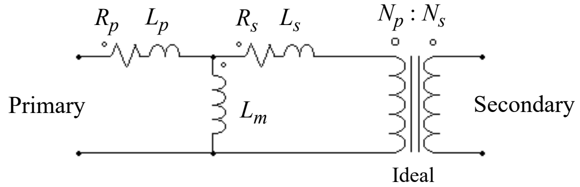
Attributes:

Parameters	Description
R_{p_i} (primary *i*); R_{s_i} (secondary *i*)	Resistance of the i_{th} primary/secondary/tertiary winding, in Ohm
L_{p_i} (pri. *i* leakage); L_{s_i} (sec. *i* leakage)	Leakage inductance of the i_{th} primary/secondary/tertiary winding, in H (referred to the first primary winding)
L_m (magnetizing)	Magnetizing inductance, in H (seen from the first primary winding)
N_{p_i} (primary *i*); N_{s_i} (secondary *i*)	No. of turns of the i_{th} primary/secondary/tertiary winding

All the resistances and inductances are referred to the first primary winding side.

Modeling of a Transformer:

A transformer is modeled as coupled inductors. For example, a single-phase two-winding transformer is modeled as two coupled inductors. The equivalent circuit can be shown as:



In the circuit, R_p and R_s are the primary and secondary winding resistances; L_p and L_s are the primary and secondary winding leakage inductances; and L_m is the magnetizing inductance. All the values are referred to the primary side.

Example:

A single-phase two-winding transformer has a winding resistance of 0.002 Ohm and leakage inductance of 1 mH at both the primary and the secondary side (all the values are referred to the primary). The magnetizing inductance is 100 mH, and the turns ratio is N_p:N_s = 220:440. The transformer will be specified as:

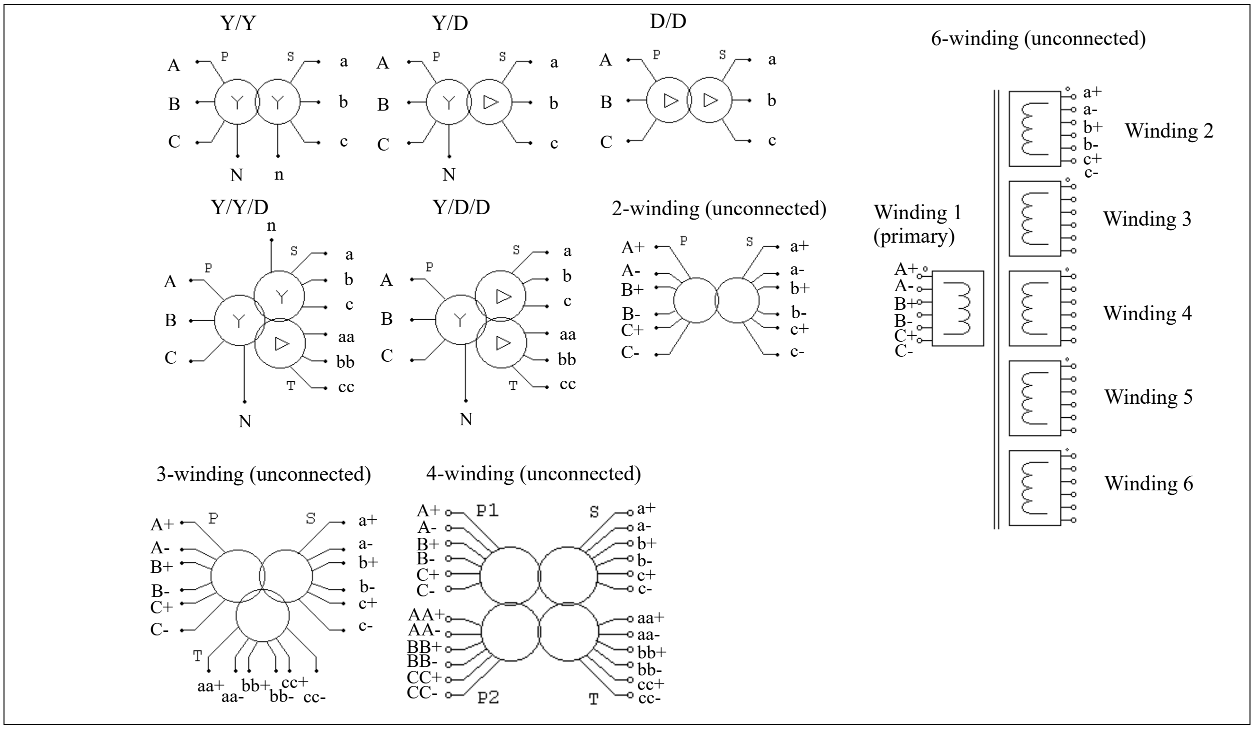
R_p (primary)	2m
R_s (secondary)	2m
L_p (primary)	1m
L_s (secondary)	1m
L_m (magnetizing)	100m
N_p (primary)	220
N_s (secondary)	440

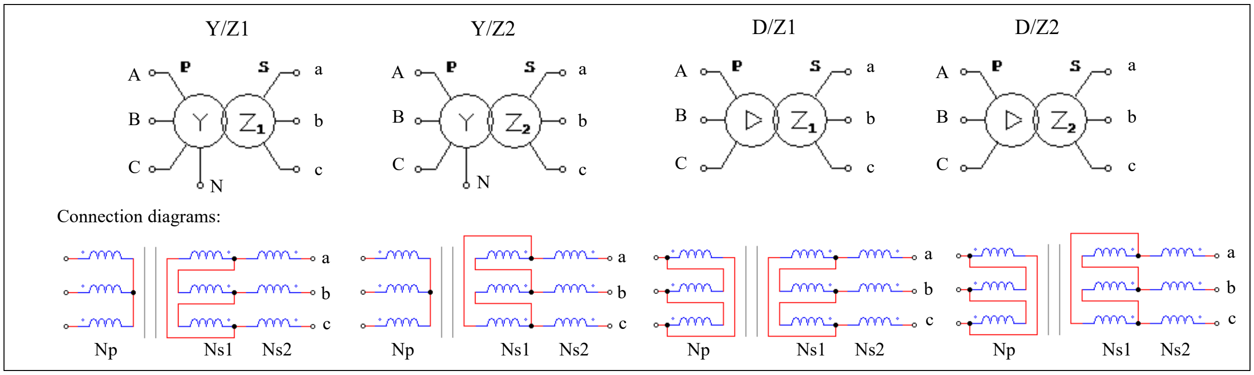
4.3.3 3-Phase Transformers

Two-winding and three-winding transformer modules are provided, as shown below. They all have 3-leg cores.

- 3-phase transformer (windings unconnected)
- 3-phase Y/Y and Y/Δ connected transformers
- 3-phase 3-winding transformer (windings unconnected)
- 3-phase 3-winding Y/Y/Δ and Y/Δ/Δ connected transformers
- 3-phase 3-winding Y/Z1, Y/Z2, Δ/Z1, and Δ/Z2 phase-shifting transformers
- 3-phase 4-winding transformer (windings unconnected)
- 3-phase 6-winding transformer (windings unconnected)

Images:





Attributes for all transformers except phase-shifting transformers:

Parameters	Description
R_p (primary); R_s (secondary); R_t (tertiary)	Resistance of the primary/secondary/tertiary winding, in Ohm
L_p (pri. leakage); L_s (sec. leakage); L_t (ter. leakage)	Leakage inductance of the primary/secondary/tertiary winding, in H
L_m (magnetizing)	Magnetizing inductance, in H (seen from the primary side)
N_p (primary); N_s (secondary); N_t (tertiary)	No. of turns of the primary/secondary/tertiary winding

Attributes for phase-shifting transformers:

Parameters	Description
R_p (primary); R_{s1} (secondary 1); R_{s2} (secondary 2)	Resistance of the primary/1st secondary/2nd secondary winding, in Ohm
L_p (primary leakage); L_{s1} (secondary 1 leakage); L_{s2} (secondary 2 leakage)	Leakage inductance of the primary/1st secondary/2nd secondary winding, in H
L_m (magnetizing)	Magnetizing inductance, in H (seen from the primary side)
N_p (primary); N_{s1} (secondary 1); N_{s2} (secondary 2)	Number of turns of the primary/1st secondary/2nd secondary winding

In the images, *P* refers to primary, *S* refers to secondary, and *T* refers to tertiary. All resistances and inductances are referred to the primary or the first primary winding side.

Three-phase transformers are modeled in the same way as single-phase transformers.

For the 3-phase phase-shifting transformers, let the angle δ be the phase difference between the secondary line voltage V_{ab} and the primary line voltage V_{AB}. The relationship between the angle and the winding turns are:

For the Y/Z1 transformer:

$$\frac{N_{s2}}{N_{s1}+N_{s2}} = \frac{\sin(30^o - \delta)}{\sin(30^o + \delta)}, \qquad 0^o \le \delta \le 30^o$$

$$\frac{N_p}{N_{s1}+N_{s2}} = \frac{1}{2\sin(30^o + \delta)} \cdot \frac{V_{AB}}{V_{ab}}$$

For the Y/Z2 transformer:

$$\frac{N_{s2}}{N_{s1}+N_{s2}} = \frac{\sin(30^o - |\delta|)}{\sin(30^o + |\delta|)}, \qquad -30^o \le \delta \le 0^o$$

$$\frac{N_p}{N_{s1}+N_{s2}} = \frac{1}{2\sin(30^o + |\delta|)} \cdot \frac{V_{AB}}{V_{ab}}$$

For the Delta/Z1 transformer:

$$\frac{N_{s2}}{N_{s1}+N_{s2}} = \frac{\sin(|\delta|)}{\sin(60^o - |\delta|)}, \qquad -30^o \le \delta \le 0^o$$

$$\frac{N_p}{N_{s1}+N_{s2}} = \frac{\sqrt{3}}{2\sin(60^o - |\delta|)} \cdot \frac{V_{AB}}{V_{ab}}$$

For the Delta/Z2 transformer:

$$\frac{N_{s2}}{N_{s1}+N_{s2}} = \frac{\sin(60^o - |\delta|)}{\sin(|\delta|)}, \qquad -60^o \le \delta \le -30^o$$

$$\frac{N_p}{N_{s1}+N_{s2}} = \frac{\sqrt{3}}{2\sin(|\delta|)} \cdot \frac{V_{AB}}{V_{ab}}$$

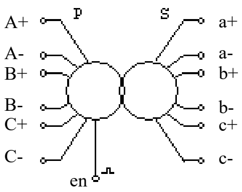
The table below shows typical transformer turns ratios and delay angles.

d	$N_{s2}/(N_{s1}+N_{s2})$				$N_p/(N_{s1}+N_{s2})*V_{ab}/V_{AB}$			
(deg)	Y/Z1	Y/Z2	Δ/Z1	Δ/Z2	Y/Z1	Y/Z2	Δ/Z1	Δ/Z2
30	0				0.577			
20	0.227				0.653			
15	0.366				0.707			
0	1	1	0		1	1	1	
-15		0.366	0.366			0.707	1.225	
-20		0.227	0.532			0.653	1.347	
-30		0	1	1		0.577	1.732	1.732
-40				0.532				1.347
-45				0.366				1.225
-60				0				1

4.3.4 3-Phase Transformer with Saturation

A 3-phase 3-winding transformer that can take into account saturation and residual flux is provided.

Image:



Attributes:

Parameters	Description
Operating Voltage (phase)	Transformer primary side rms voltage per phase under the circuit operating condition, in V
Operating Frequency	Operating frequency of the transformer, in Hz
R_p (primary)	Resistance of the primary winding, in Ohm
L_p (pri. leakage)	Leakage inductance of the primary winding, in H
R_s (secondary)	Resistance of the secondary winding, in Ohm (referred to the primary side)
L_s (sec. leakage)	Leakage inductance of the secondary winding, in H (referred to the primary side)
R_m (core losses)	Resistance that represents transformer core losses, in Ohm
I_m vs. L_m	Values of the magnetizing current I_m (in A) vs. the magnetizing inductance L_m (in H) that defines the magnetizing characteristics seen from the primary winding
Phase A Residual Flux	Phase A residual flux, in per unit value
Phase B Residual Flux	Phase B residual flux, in per unit value
Phase C Residual Flux	Phase C residual flux, in per unit value
N_p (primary)	Number of turns of the primary winding
N_s (secondary)	Number of turns of the secondary winding

All the resistances and inductances are referred to the primary side.

In the images, *p* refers to the primary side and *s* refers to the secondary side. The node *en* is a control node, to be connected to a control logic signal. The moment when the signal changes from 0 to 1 indicates the moment that the transformer is energized.

Note that the operating voltage refers to the voltage per phase in the actual operating condition, not necessarily the rated voltage. For example, if a transformer rated at 63kV (primary side line-to-line rms) operates under the rated condition, and if the primary winding is connected in Y, the operating voltage per phase will be 63,000/ $\sqrt{3}$ V. If the primary winding is connected in delta, the operating voltage per phase will be 63,000V. If the same transformer is connected in delta but the actual operating voltage is only 60,000V, for example, the operating voltage per phase will be 60,000V instead.

4.4 Magnetic Elements

A set of magnetic elements, including winding, leakage flux path, air gap, linear core, and saturable core, is provided to model magnetic devices. These elements are the basic building blocks of magnetic equivalent circuits, and they provide a very powerful and convenient way of modeling any types of magnetic devices.

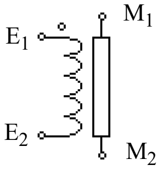
Different unit systems have been used in the magnetic circuit analysis. The table below shows magnetic quantities in the SI (System International) and the mixed CGS (Centimeter Gram Second) unit systems, and their conversion.

Quantity	SI Unit	CGS Unit	CGS-SI Conversion
Flux ϕ	Weber	Maxwell	1 Weber = 10^8 Maxwell
Flux Density *B*	Tesla	Gauss	1 Tesla = 10^4 Gauss
Field Intensity *H*	A*Turns/m	Oersted	1 A*Turns/m = $\frac{4\pi}{1000}$ Oersted
Magnetomotive Force (mmf) *F*	A*Turns	Gilbert	1 A*Turns = $\frac{4\pi}{10}$ Gilbert
Permeability μ_o (in space)	$4\pi*10^{-7}$	1	$4\pi*10^{-7}$

4.4.1 Winding

A winding element provides the interface between the electric circuit and the magnetic equivalent circuit.

Image:



Attributes:

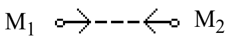
Parameters	Description
Number of Turns	No. of turns of the winding
Winding Resistance	Winding resistance, in Ohm

This element represents a winding on a magnetic core. The two electric nodes (E_1 and E_2) are connected to an electric circuit, while the two magnetic nodes (M_1 and M_2) are connected to other magnetic elements (such as leakage flux path, air gap, and magnetic core).

4.4.2 Leakage Flux Path

This element models the flow path of the leakage flux.

Image:

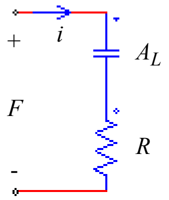


Attributes:

Parameters	Description
Inductance Factor A_L	Inductance factor A_L, in H, defined as the inductance per turn squared
Resistance for Losses	Resistance R, in Ohm, that represents the losses due to the leakage flux.
Current Flag	Display flag of the current that flows through the resistor R

The resistance R represents the losses due to the leakage flux.

Assuming that the mmf (magnetomotive force) applied across the leakage flux path is F, the electric equivalent circuit of the leakage flux path is as follows:



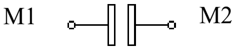
The mmf, in the form of a voltage source, applies across the capacitor (the capacitance is A_L) and the resistor R. Let the current flowing through this branch be i, and the rms value be I_{rms}, the relationship between the losses due to the leakage flux and the resistance R is:

$$P_{loss} = I_{rms}^2 \cdot R$$

4.4.3 Air Gap

The image and attributes of an air gap element are as follows.

Image:



The input parameters of the air gap can be defined in two ways. One is to define the air gap length and the cross section area, and the other is to define the inductance factor A_L. They are as follows:

Attributes:

For the element *Air Gap*:

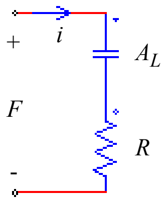
Parameters	Description
Air Gap Length	The length of the air gap, l_g, in m

Cross Section Area	Cross section of the air gap, A_c, in m^2
Resistance for Losses	Resistance R, in ohm, that represents the losses due to the air gap fringing effect
Current Flag	Display flag of the current that flows through the resistor R

For the element *Air Gap (AL)*:

Parameters	Description
Inductance Factor A_L	Inductance factor A_L, in H, defined as the inductance per turn squared
Resistance for Losses	Resistance R, in ohm, that represents the losses due to the air gap fringing effect
Current Flag	Display flag of the current that flows through the resistor R

The resistance R represents the losses due to the air gap fringing effect. Assuming that the mmf (magnetomotive force) applied across the air gap is F, the electric equivalent circuit of the air gap is as follows:



The mmf, in the form of a voltage source, applies across the capacitor (the capacitance has the value of the inductance factor A_L) and the resistor R. For the element *Air Gap*, the inductance factor can be calculated from the air gap length and the cross section area as:

$$A_L = \frac{\mu_o \cdot A_c}{l_g}$$

where $\mu_o = 4\pi * 10^{-7}$.

The losses on the resistor represents the losses due to the fringing effect, which can be expressed as:

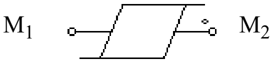
$$P_{loss} = I_{rms}^2 \cdot R$$

where I_{rms} is the rms value of the current i flowing through the resistor.

4.4.4 Linear Core

This element represents a linear loss less core.

Image:



Attributes:

Parameters	Description
Inductance Factor A_L	Inductance factor A_L of the core, in H, defined as the inductance per turn squared

If the length of the core is L_{length} and the cross section area is A_c, the inductance factor A_L is expressed as:

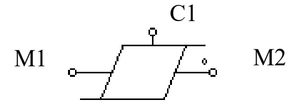
$$A_L = \frac{\mu_o \cdot \mu_r \cdot A_c}{L_{length}}$$

where μ_r is the relative permeability of the core material.

4.4.5 Saturable Core

This element models a magnetic core with saturation and hysteresis.

Image:



Attributes:

Parameters	Description
Inductance Factor A_L	Inductance factor A_L of the core, in H, defined as the inductance per turn squared
Resistance for Losses	Resistance R, in Ohm, that represents the core losses
Coefficient phi_sat	Coefficient Φ_{sat} for the core B-H curve, in Weber
Coefficient K1	Coefficient K_1 for the core B-H curve
Coefficient Kexp1	Coefficient K_{exp1} for the core B-H curve
Coefficient K2	Coefficient K_2 for the core B-H curve
Coefficient Kexp2	Coefficient K_{exp2} for the core B-H curve
Initial Flux phi_o	Initial flux Φ_o of the core, in Weber
Current Flag	Display flag of the electric current that flows through the resistor R. If the rms value of the current is I_{rms}, the core losses can be calculated as: $P_{core\_loss} = I_{rms}^2 * R$.

In the element image, the nodes M_1 and M_2 are the two nodes that connect the core to other magnetic elements (such as winding, flux leakage path, air gap, etc.). The node marked with a dot is Node M_2. Node C_1 is a control output node, which shows the flux (in Weber) flowing through the core, from Node M_2 to M_1.

The initial flux of the core, Φ_o, is the flux flowing from Node M_2 to M_1 at the beginning, corresponding to the field intensity H = 0. The initial flux density B_o can be calculated as: $B_o = \Phi_o/A_c$ (where A_c is the core cross section area), and the initial mmf F_o is: $F_o = \Phi_o/A_L$.

A differential voltage probe connected between Node M_2 to M_1 will measure the mmf (in ampere*turn) applied to the core.

The coefficients Φ_{sat}, K_1, K_{exp1}, K_2, and K_{exp2} are used to fit the B-H curve of an actual magnetic material. A good initial guess of Φ_{sat} is the maximum flux of the B-H curve in deep saturation. To calculate this flux, multiply the corresponding flux density B by the cross section area of the core. Coefficient K_1 usually varies between 0.7 and 1, depending on the core material. Coefficient K_{exp1} mainly affects the rate of the core saturation, and is in the range between 10 and 200 (10 for low permeability ferrite, and 200 for metglas).

The coefficients K_2 and K_{exp2} are used in very rare occasions, such as for ferroresonant regulators. They are normally set as follows to keep them from affecting the B-H curve:

$K_2 > 2$
$K_{exp2} > 20$

A tutorial on how to define the core coefficients is provided in the "doc" sub-folder in the PSIM directory. Also, a program is provided to plot the B-H curve of the core given a set of coefficients. To launch this program, in PSIM, go to **Utilities** >> **B-H Curve**.

4.5 Other Elements

4.5.1 Operational Amplifier

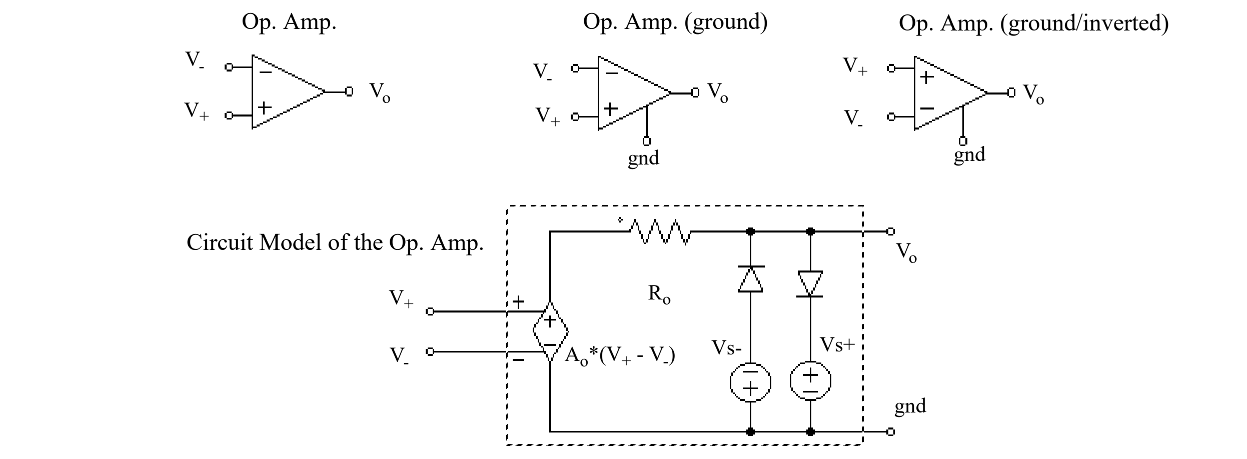
Two types of operational amplifier (op. amp.) models are provided: ideal op. amp. model, and non-ideal op. amp. model (Level-1 model).

Note that the ideal op. amp. model works in the negative feedback mode but does not work in the positive feedback mode. The non-ideal op. amp. model does not have such a restriction.

4.5.1.1 Ideal Operational Amplifier

Three ideal op. amp. elements are provided: *Op. Amp.*, *Op. Amp. (ground)*, and *Op. Amp. (ground/inverted)*. An ideal op. amp. is modelled using power circuit elements, as shown below.

Images:



where

V+; V-	- noninverting and inverting input voltages
V_o	- output voltage
A_o	- op. amp. dc gain (A_o is set to 100,000.)
R_o	- output resistance (R_o is set to 80 Ohms)

Attributes:

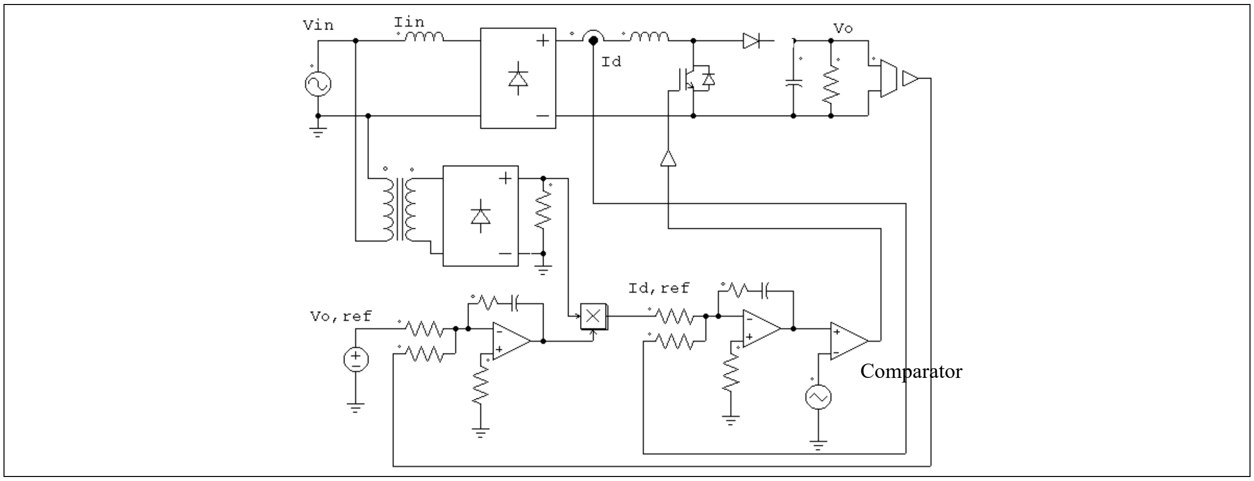
Parameters	Description
Voltage Vs+	Upper voltage source level of the op. amp.
Voltage Vs-	Lower voltage source levels of the op. amp.

The difference between the element *Op. Amp.* and *Op. Amp.* (*ground*) or *Op. Amp.* (*ground/inverted*) is that, for the *Op. Amp.* element, the reference ground of the op. amp. model is connected to the power ground, whereas for *Op. Amp.* (*ground*) or *Op. Amp.* (*ground/inverted*), the reference ground node of the model is accessible and can be floating.

Note that the image of an op. amp. is similar to that of a comparator. For the op. amp., the inverting input is at the upper left and the noninverting input is at the lower left. For the comparator, it is the opposite.

Example: A Boost Power Factor Correction Circuit

The figure below shows a boost power factor correction circuit. The PI regulators of both the inner current loop and the outer voltage loop are implemented using op. amp.



4.5.1.2 Non-Ideal Operational Amplifier

As compared to the ideal op. amp. model, the non-ideal op. amp. model, also referred to as Level-1 op. amp. model, includes the following characteristics in the model:

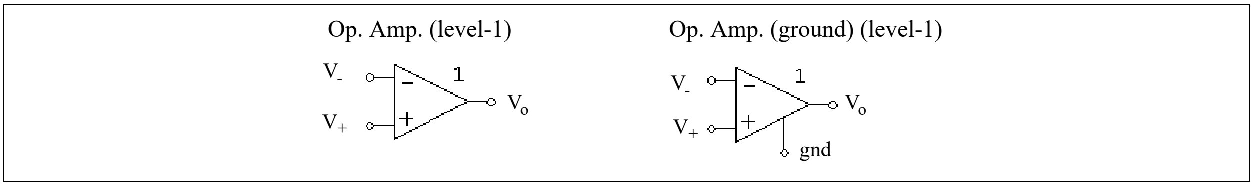
- Op. amp. bandwidth
- Output current limit

The non-ideal op. amp. is modelled using power circuit elements, and similar to the ideal op. amp., is treated as a power element in PSIM.

Two non-ideal op. amp. elements are provided: *Op. Amp. (level-1)* and *Op. Amp. (ground) (level-1)*. The difference between these two elements is that, for *Op. Amp. (level-1)*, the reference ground of the op. amp. model is connected to the power ground, whereas for *Op. Amp. (ground) (level-1)*, the reference ground node of the model is accessible and can be floating.

The non-ideal op. amp. images have the number "1" at the upper right corner, as shown below.

Images:

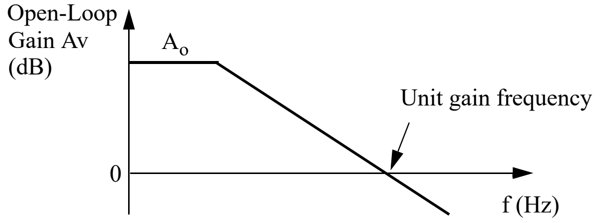

Op. Amp. (level-1)

Op. Amp. (ground) (level-1)

Attributes:

Parameters	Description
Input Resistance Rin	Input resistance of the op. amp., in Ohm
DC Gain A_o	DC gain of the op. amp.
Unit Gain Frequency	Frequency at which the op. amp. gain is 1, in Hz.
Output Resistance Ro	Output resistance of the op. amp., in Ohm
Maximum Output Current	Maximum current that the op. amp. output can provide, in A
Voltage Vs+	Upper voltage source level of the op. amp.
Voltage Vs-	Lower voltage source levels of the op. amp.

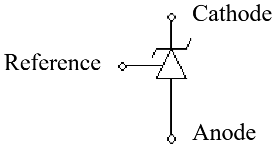
The figure below shows the definition of the dc gain A_o and the unit gain frequency.



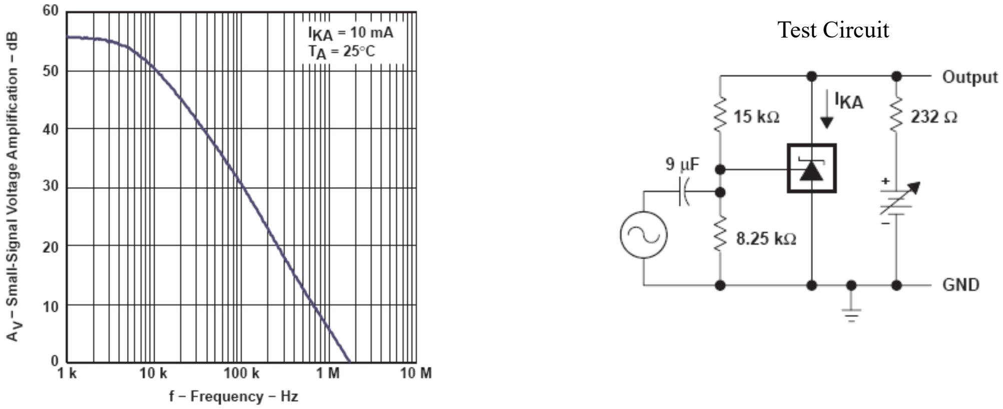
4.5.2 TL431 Shunt Regulator

The image of the TL431 shunt regulator is shown as below.

Image:



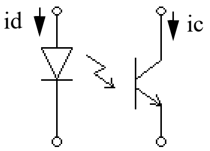
The TL431 regulator maintains the reference node voltage to be around 2.495V. In this model, the limited bandwidth of the TL431 is modelled. That is, the ac frequency response of the TL431 model resembles the following figure from the datasheet.



4.5.3 Opto-Coupler

The image and parameters of a opto-coupler are shown below.

Image:



Attributes:

Parameters	Description
Current Transfer Ratio	The current transfer ratio *CTR* between the transistor current i_c and the diode current i_d, that is, $CTR = i_c/i_d$.
Diode Resistance	Diode resistance R_d, in Ohm. Note that the resistance must be greater than 0.
Diode Threshold Voltage	The forward diode threshold voltage V_{d_th}, in V
Transistor Vce_sat	The transistor saturation voltage V_{ce_sat}, in V
Transistor-side Capacitance	Capacitance C_p across the collector and emitter of the transistor, in F

These parameters can either be read directly from the manufacturer datasheet, or can be calculated from the datasheet information. Take the Motorola optocoupler 4N25 as an example. From the datasheet, we can read the current transfer ratio and the transistor saturation voltage as:

$CTR = 70\% = 0.7$; $V_{ce\_sat} = 0.15$ V

From the "LED Forward Voltage versus Forward Current" characteristics on the datasheet, one can calculate the diode resistance R_d and the threshold voltage V_{d_th}. We assume that the diode forward voltage is approximated by a straight line. That is,

$v_d = V_{d\_th} + R_d * i_d$

With $T_A = 25^oC$, we can obtain two readings from the curve: $i_d = 10$ mA and $v_d = 1.15$ V; $i_d = 40$ mA and $v_d = 1.3$ V. Based on these two points, we can calculate R_d and V_{d_th} as: $R_d = 5$ Ohm; $V_{d\_th} = 1.1$ V.

The optocoupler model does not take into account the delay, but it does model the turn-on/turn-off transient through the capacitor C_p across the collector and emitter of the transistor. The capacitance value can be obtained from the fall time t_{fall} and the switching time test condition as:

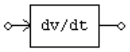
$$C_p = \frac{t_{fall}}{2.2 \cdot R_L}$$

where R_L is the load resistor in the switching time test circuit. For 4N25, for example, from the datasheet, we can obtain: $t_{fall} = 1.3$ µs, and $R_L = 100$ Ohm. The capacitance is calculated as: $C_p = 6$ nF.

4.5.4 dv/dt Block

A dv/dt block has the same function as the differentiator in the control circuit, except that it is for the power circuit.

Image:



The output of the dv/dt block is equal to the derivative of the input voltage versus time. It is calculated as:

$$V_o = \frac{V_{in}(t) - V_{in}(t - \Delta t)}{\Delta t}$$

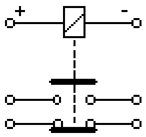
where $V_{in}(t)$ and $V_{in}(t\text{-}\Delta t)$ are the input values at the current and previous time step, and Δt is the simulation time step.

4.5.5 Relays

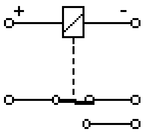
Two relay blocks with one normally open (NO) switch and one normally closed (NC) switch are provided.

Images:

Relay (1 NO 1 NC)



Relay (1 NO 1 NC changeover)



Attributes:

Parameters	Description
Rated Coil Voltage	Rated voltage of the relay coil, in V
Coil Resistance	Resistance of the coil, in Ohm
Operate Voltage	Voltage at which the relay will operate, in V
Release Voltage	Voltage at which the relay will return to the default position, in V
Operate Time	Time from the moment the operate voltage is reached to the moment the switches operate, in sec.
Release Time	Time from the moment the release voltage is reached to the moment the switches return to the default positions, in sec.

The relays have one normally open switch and one normally closed switch. When a dc voltage is applied to the relay coil in the polarity as shown in the image, if the voltage reaches the Operate Voltage, after a time delay defined by the Operate Time, the NO switch will be closed, and the NC switch will be opened.

When the coil voltage reduces to the Release Voltage, after a time delay defined by the Release Time, the two switches will return to their default positions.

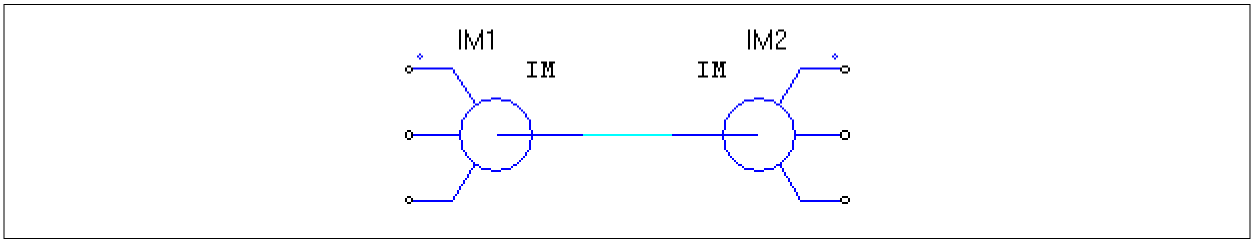
4.6 Motor Drive Module

The Motor Drive Module is an add-on module to the basic PSIM program. It provides machine models and mechanical load models for motor drive system studies.

The Motor Drive Module includes electric machines as described in this section, and mechanical elements and speed/torque/position sensors as described in Section 4.9.

4.6.1 Reference Direction of Mechanical Systems

In a motor drive system, in order to formulate equations for the mechanical system, a position notation needs to be defined. Take the following motor drive system as an example:



The system consists of two induction machines, IM1 and IM2, connected back-to-back. One operates as a motor, and the other as a generator. From the point of view of the first machine IM1, the mechanical equation can be written as:

$$(J_1 + J_2) \cdot \frac{d\omega_m}{dt} = T_{em1} - T_{em2}$$

where J_1 and J_2 are the moment of inertia, and T_{em1} and T_{em2} are the developed torques of the machine IM1 and IM2, respectively.

From the point of view of the second machine IM2, however, the mechanical equation can be written as:

$$(J_1 + J_2) \cdot \frac{d\omega_m}{dt} = T_{em2} - T_{em1}$$

These two equations are equally valid, but will produce opposite mechanical speed.

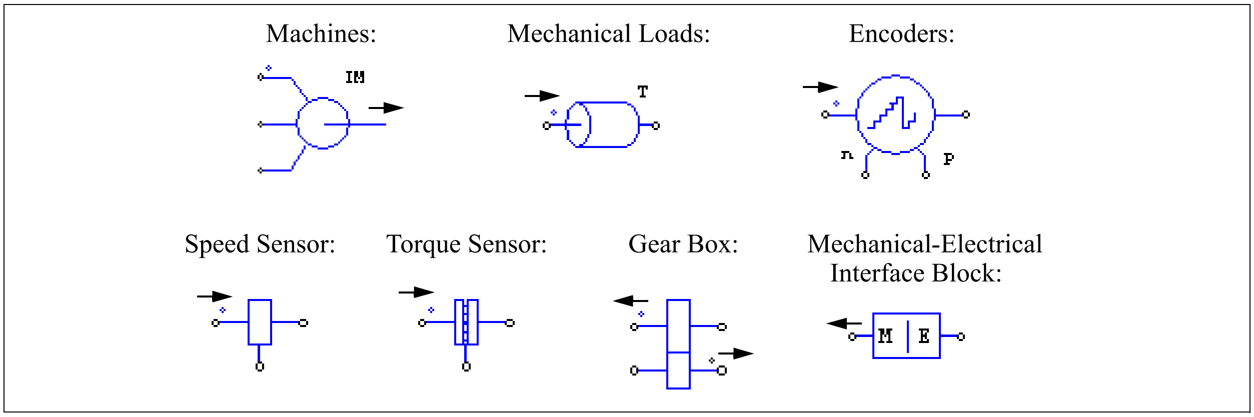
In order to avoid this ambiguity, in PSIM, the concept "reference direction" is used in the mechanical system so that the mechanical equation can be uniquely defined.

In a mechanical system, one element is designated as the master unit (this element is considered to operate in the master mode), and the rest of the elements are in the slave mode. Elements that can be master units are: Electric machines, mechanical-to-electrical interface blocks, and gear boxes.

The master unit defines the reference direction of the mechanical system. The direction is define as the direction from the shaft node of the master unit, along the shaft, to the rest of the mechanical system.

Once the reference direction of the mechanical system is defined, the speed and torque reference of the mechanical system can be defined. For example, if we use the right-hand method, with the thumb pointing in the reference direction of the mechanical system, by rotating the right hand, the fingers will point to the positive direction of the speed and the torque.

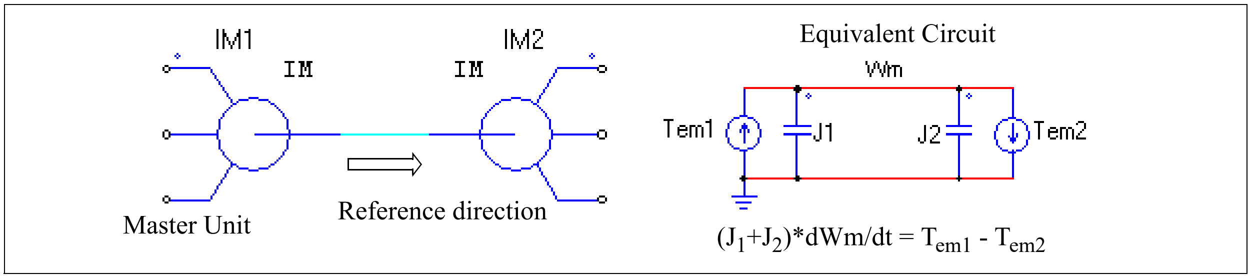
Moreover, each mechanical element has its own reference direction. The following diagram shows the reference direction of each mechanical element, as indicated by the arrow:



The reference direction of each element and the reference direction of the overall mechanical system determine how the element interacts with the mechanical system.

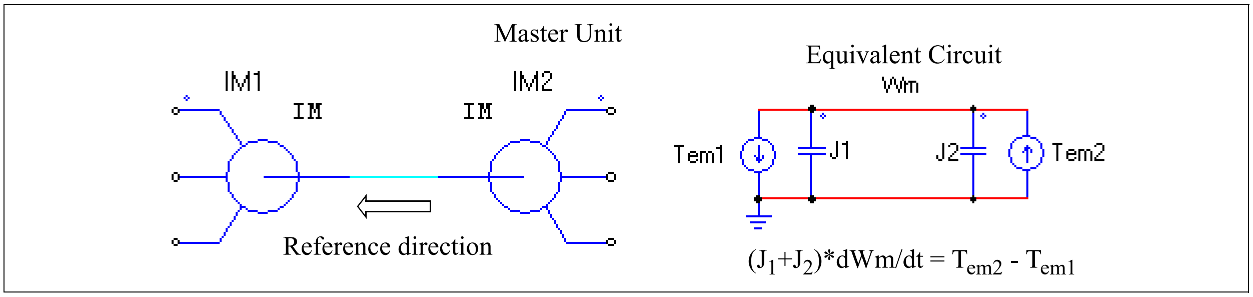
For example, if the reference direction of a machine is along the same direction as the reference direction of the mechanical system, the developed torque of the machine will contribute to the shaft rotation in the positive direction. However, if the reference direction of the machine is opposite to that of the mechanical system, the developed torque will contribute to the shaft rotation in the negative direction.

In the two-machine example above, using the notation of the "reference direction", if we define the machine IM1 as the master unit, the reference direction of the overall mechanical system will be from left to right, as shown below. Based on this direction, the machine IM1 will be along the reference direction, and the machine IM2 will be opposite to the reference direction. This leads to the equivalent circuit of the mechanical system as shown on the right.



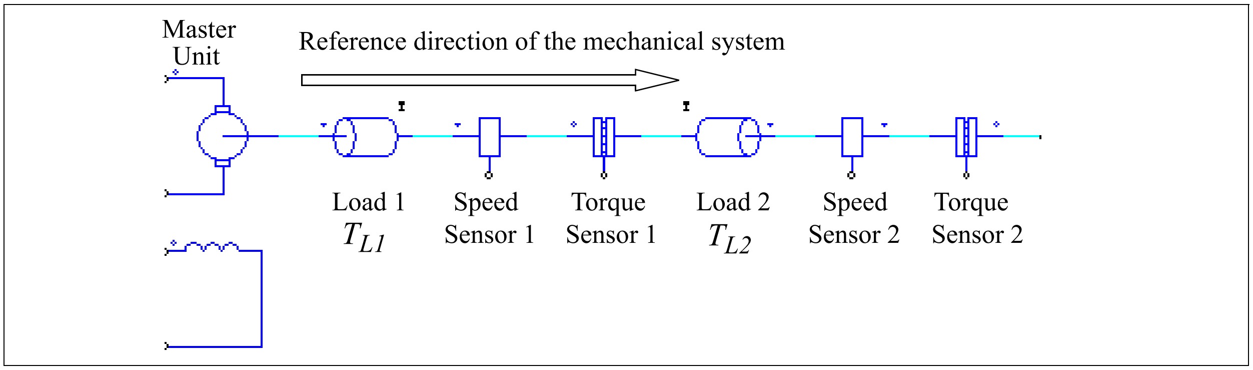
$(J_1+J_2)*dWm/dt = T_{em1} - T_{em2}$

Similarly, if we define the machine IM2 as the master unit, the reference direction of the overall mechanical system will be from right to left, as shown below. Based on this direction, the machine IM1 will be opposite to the reference direction, and the machine IM2 will be along the reference direction. This leads to the equivalent circuit of the mechanical system as shown on the right.



$(J_1+J_2)*dWm/dt = T_{em2} - T_{em1}$

The following shows another mechanical system with sensors and loads connected in different ways.



In this mechanical system, the machine on the left is the master unit. The reference direction of the mechanical system is from left to the right along the mechanical shaft. Comparing this direction with the reference direction of each element, Load 1, Speed Sensor 1, and Torque Sensor 1, will be along the reference direction, and Load 2, Speed Sensor 2, and Torque Sensor 2 will be opposite to the reference direction of the mechanical system.

Therefore, if the speed of the machine is positive, Speed Sensor 1 reading will be positive, and Speed Sensor 2 reading will be negative.

Similarly, the two constant-torque mechanical loads, with the amplitudes of T_{L1} and T_{L2}, interact with the machine in different ways. Load 1 is along the reference direction, and the loading torque of Load 1 to the master machine will be T_{L1}. On the other hand, Load 2 is opposite to the reference direction, and the loading torque of Load 2 to the machine will be $-T_{L2}$.

4.6.2 Induction Machines

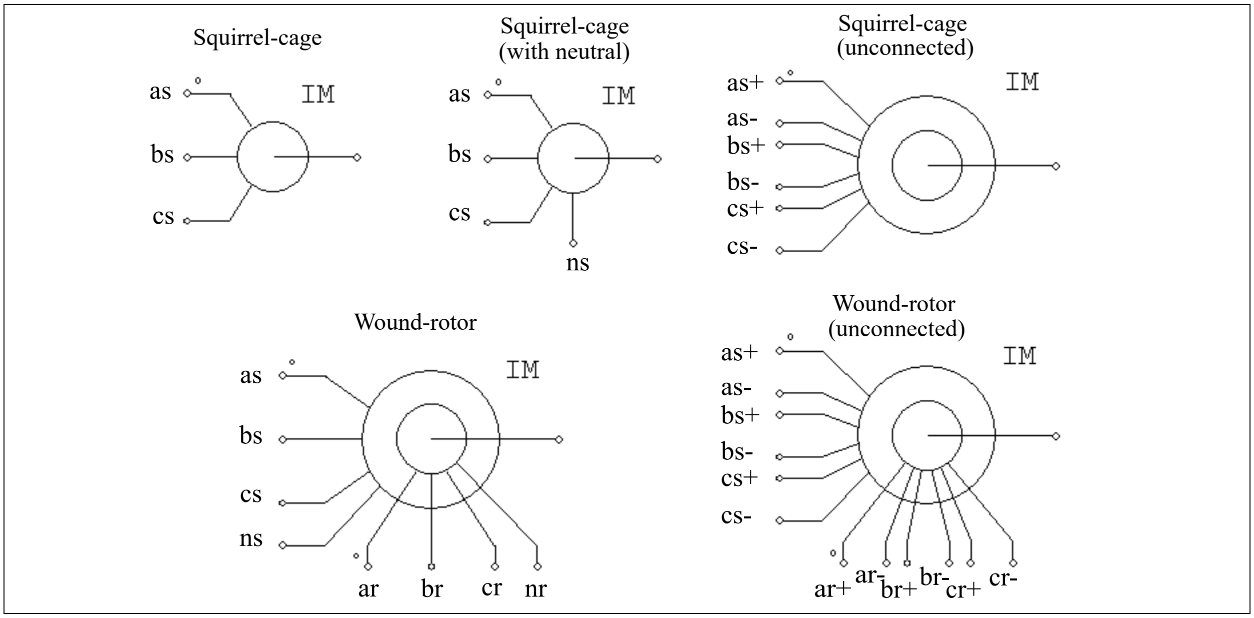
Linear and nonlinear models are provided for squirrel-cage and wound-rotor induction machines. The linear model is further divided into general type and symmetrical type. This section describes the linear models.

Four linear models are provided:

- Symmetrical 3-phase squirrel-cage induction machine
- General 3-phase squirrel-cage induction machine
- Symmetrical 3-phase wound-rotor induction machine
- General 3-phase wound-rotor induction machine

The images and parameters are shown as follows.

Images:



Attributes:

Parameters	Description
R_s (stator)	Stator winding resistance, in Ohm
L_s (stator)	Stator winding leakage inductance, in H
R_r (rotor)	Rotor winding resistance referred to the stator side, in Ohm
L_r (rotor)	Rotor winding leakage inductance referred to the stator side, in H
L_m (magnetizing)	Magnetizing inductance, in H
Ns/Nr Turns Ratio	Stator and rotor winding turns ratio (for wound-rotor machine only)
No. of Poles	Number of poles *P* of the machine (an even integer)
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Torque Flag	Output flag for internal torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

All the parameters are referred to the stator side.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The models of the squirrel-cage induction machine with and without the neutral are the same internally.

The operation of a 3-phase induction machine is described by the following equations:

$$[v_{abc,s}] = [R_s] \cdot [i_{abc,s}] + [L_s] \cdot \frac{d}{dt}[i_{abc,s}] + [M_{sr}] \cdot \frac{d}{dt}[i_{abc,r}]$$

$$[v_{abc,r}] = [R_r] \cdot [i_{abc,r}] + [L_r] \cdot \frac{d}{dt}[i_{abc,r}] + [M_{sr}]^T \cdot \frac{d}{dt}[i_{abc,s}]$$

where

$$[v_{abc,s}] = \begin{bmatrix} v_{a,s} \\ v_{b,s} \\ v_{c,s} \end{bmatrix} \qquad [v_{abc,r}] = \begin{bmatrix} v_{a,r} \\ v_{b,r} \\ v_{c,r} \end{bmatrix} \qquad [i_{abc,s}] = \begin{bmatrix} i_{a,s} \\ i_{b,s} \\ i_{c,s} \end{bmatrix} \qquad [i_{abc,r}] = \begin{bmatrix} i_{a,r} \\ i_{b,r} \\ i_{c,r} \end{bmatrix}$$

For squirrel-cage machines, $v_{a,r} = v_{b,r} = v_{c,r} = 0$. The parameter matrices are defined as:

$$[R_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \qquad [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}$$

$$[L_s] = \begin{bmatrix} L_s + M_{sr} & -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & L_s + M_{sr} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} & L_s + M_{sr} \end{bmatrix} \qquad [L_r] = \begin{bmatrix} L_r + M_{sr} & -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & L_r + M_{sr} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} & L_r + M_{sr} \end{bmatrix}$$

$$\left[M_{sr}\right] = M_{sr} \cdot \begin{bmatrix} \cos\theta & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\theta & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\theta \end{bmatrix}$$

where M_{sr} is the mutual inductance between the stator and rotor windings, and θ is the mechanical angle. The mutual inductance is related to the magnetizing inductance as:

$$L_m = \frac{3}{2}M_{sr}$$

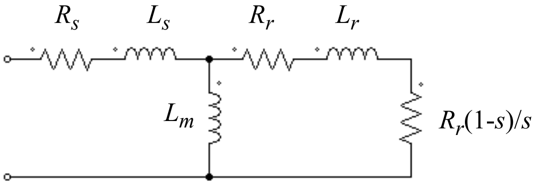
The mechanical equation is expressed as:

$$J \cdot \frac{d\omega_m}{dt} = T_{em} - T_L$$

where the developed torque T_{em} is defined as:

$$T_{em} = \frac{P}{2} \cdot \left[i_{abc,s}\right]^T \cdot \frac{d}{d\theta}\left[M_{sr}\right] \cdot \left[i_{abc,r}\right]$$

For a symmetrical squirrel-cage induction machine, the steady state equivalent circuit is shown below. In the figure, s is the slip.

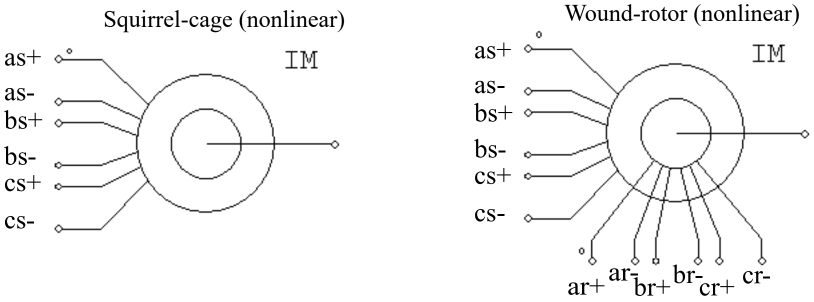


4.6.3 Induction Machine with Saturation

Two models of induction machines with saturation are provided:

- 3-phase squirrel-cage induction machine
- 3-phase wound-rotor induction machine

Images:



Attributes:

Parameters	Description
R_s (stator)	Stator winding resistance, in Ohm
L_s (stator)	Stator winding leakage inductance, in H

R_r (rotor)	Rotor winding resistance referred to the stator side, in Ohm
L_r (rotor)	Rotor winding leakage inductance referred to the stator side, in H
Ns/Nr Turns Ratio	Stator and rotor winding turns ratio (for wound-rotor machine only)
No. of Poles	Number of poles P of the machine (an even integer)
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Torque Flag	Output flag for internal torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)
I_m vs. L_m (I_{m1},L_{m1}) ...	Characteristics of the magnetizing current I_m versus the magnetizing inductance $[(I_{m1},L_{m1})\ (I_{m2},L_{m2})\ ...]$ where I_m is in A and L_m is in H.

All the parameters are referred to the stator side.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The operation of a 3-phase induction machine with saturation is described by the following equations:

$$[v_{abc,s}] = [R_s]\cdot[i_{abc,s}] + L_s\cdot\frac{d}{dt}[i_{abc,s}] + \frac{d}{dt}[\lambda_{abc,s}]$$

$$[v_{abc,r}] = [R_r]\cdot[i_{abc,r}] + L_r\cdot\frac{d}{dt}[i_{abc,r}] + \frac{d}{dt}[\lambda_{abc,r}]$$

where

$$[\lambda_{abc,s}] = M_{sr}\cdot\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}\cdot[i_{abc,s}] + M_{sr}\cdot\begin{bmatrix} \cos\theta & \cos\left(\theta+\frac{2\pi}{3}\right) & \cos\left(\theta-\frac{2\pi}{3}\right) \\ \cos\left(\theta-\frac{2\pi}{3}\right) & \cos\theta & \cos\left(\theta+\frac{2\pi}{3}\right) \\ \cos\left(\theta+\frac{2\pi}{3}\right) & \cos\left(\theta-\frac{2\pi}{3}\right) & \cos\theta \end{bmatrix}[i_{abc,r}]$$

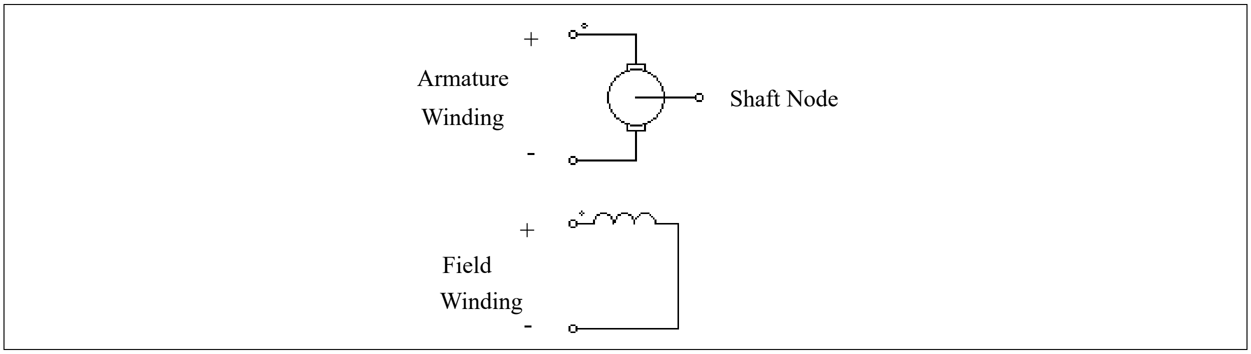
$$[\lambda_{abc,s}] = M_{sr}\cdot\begin{bmatrix} \cos\theta & \cos\left(\theta-\frac{2\pi}{3}\right) & \cos\left(\theta+\frac{2\pi}{3}\right) \\ \cos\left(\theta+\frac{2\pi}{3}\right) & \cos\theta & \cos\left(\theta-\frac{2\pi}{3}\right) \\ \cos\left(\theta-\frac{2\pi}{3}\right) & \cos\left(\theta+\frac{2\pi}{3}\right) & \cos\theta \end{bmatrix}\cdot[i_{abc,s}] + M_{sr}\cdot\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}[i_{abc,r}]$$

In this case, the inductance M_{sr} is no longer constant, but a function of the magnetizing current I_m. The magnetizing current I_m and inductance L_m are defined by a series of data points in pair, for example, "(0, 0.041) (5, 0.035) (20, 0.03)". Between two data points, the inductance L_m is interpolated linearly. When the magnetizing current is less than the minimum or greater than the maximum, the inductance will be clamped to the value at the first point or the last point.

4.6.4 DC Machine

The image and parameters of a dc machine are as follows:

Image:



Attributes:

Parameters	Description
R_a (armature)	Armature winding resistance, in Ohm
L_a (armature)	Armature winding inductance, in H
R_f (field)	Field winding resistance, in Ohm
L_f (field)	Field winding inductance, in H
Moment of Inertia	Moment of inertia of the machine, in kg*m^2
V_t (rated)	Rated armature terminal voltage, in V
I_a (rated)	Rated armature current, in A
n (rated)	Rated mechanical speed, in rpm
I_f (rated)	Rated field current, in A
Torque Flag	Output flag for internal torque T_{em}
Master/Slave Flag	The master/slave flag of the machine (1: master; 0: slave)

When the torque flag is set to 1, the internal torque generated by the machine will be saved to the output file for display.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The operation of a dc machine is described by the following equations:

$$v_t = E_a + i_a \cdot R_a + L_a \frac{di_a}{dt}$$

$$v_f = i_f \cdot R_f + L_f \frac{di_f}{dt}$$

$$E_a = k \cdot \phi \cdot \omega_m$$

$$T_{em} = k \cdot \phi \cdot i_a$$

$$J \cdot \frac{d\omega_m}{dt} = T_{em} - T_L$$

where v_t, v_f, i_a, and i_f are the armature and field winding voltage and current, respectively; E_a is the back emf,

ω_m is the mechanical speed in rad./sec., T_{em} is the internal developed torque, and T_L is the load torque. The back emf and the internal torque can also be expressed as:

$$E_a = L_{af} \cdot i_f \cdot \omega_m$$

$$T_{em} = L_{af} \cdot i_f \cdot i_a$$

where L_{af} is the mutual inductance between the armature and the field windings. It can be calculated based on the rated operating conditions as:

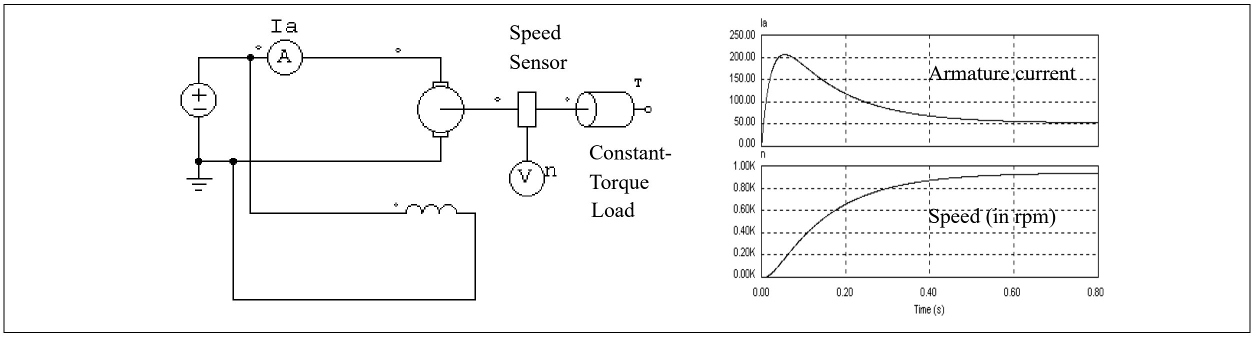
$$L_{af} = \frac{(V_t - I_a \cdot R_a)}{I_f \cdot \omega_m}$$

Note that the dc machine model assumes magnetic linearity. Saturation is not considered.

Example: A DC Motor with a Constant-Torque Load

The circuit below shows a shunt-excited dc motor with a constant-torque load T_L. Since the load is along the reference direction of the mechanical system, the loading torque to the machine is T_L. Also, the speed sensor is along the reference direction. It will give a positive output for a positive speed.

The simulation waveforms of the armature current and the speed are shown on the right.

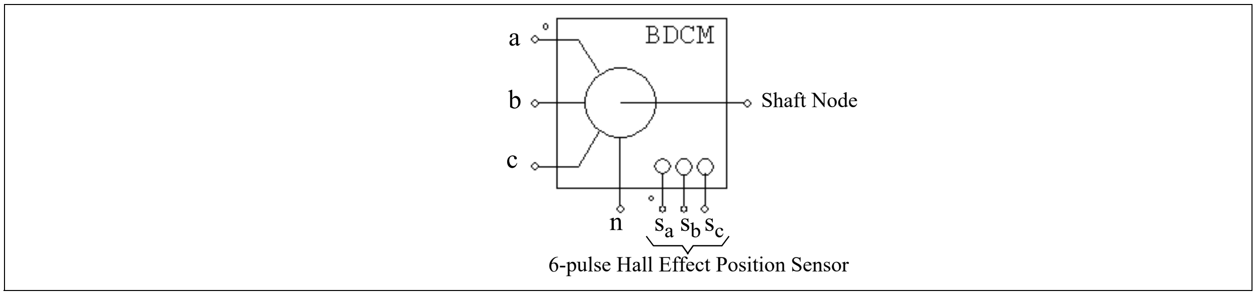


4.6.5 Brushless DC Machine

A 3-phase brushless dc machine is a type of permanent magnet synchronous machine with trapezoidal waveform back emf. It has 3-phase windings on the stator, and permanent magnet on the rotor.

The image of the 3-phase brushless dc machine is shown below. Two types of parameter inputs are provided: One based on machine model parameters, and the other based on manufacturer datasheet information.

Image:



Attributes based on machine model parameters:

Parameters	Description
R (stator resistance)	Stator phase resistance *R*, in Ohm
L (stator self ind.)	Stator phase self inductance *L*, in H

M (stator mutual ind.)	Stator mutual inductance *M*, in H The mutual inductance *M* is a negative value. Depending on the winding structure, the ratio between *M* and the stator self inductance *L* is normally between -1/3 and -1/2. If *M* is unknown, a reasonable value of *M* equal to -0.4**L* can be used as the default value.
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed)
Vrms / krpm	RMS line-to-line back emf constant, in V/krpm (mechanical speed). The values of Vpk/krpm and Vrms/krpm should be available from the machine data sheet. If these values are not available, they can be obtained through experiments by operating the machine as a generator at 1000 rpm and measuring the peak and rms values of the line-to-line voltage.
No. of Poles *P*	Number of poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Shaft Time Constant	Shaft time constant τ_{shaft}
theta_0 (deg.)	Initial rotor angle θ_r, in electrical deg. The initial rotor angle is the rotor angle at t=0. The zero rotor angle position is defined as the position where Phase A back emf crosses zero (from negative to positive) under a positive rotation speed.
theta_advance (deg.)	Position sensor advance angle $\theta_{advance}$, in electrical deg. The advance angle is defined as such that, for a brushless dc machine with a 120^o trapezoidal back emf waveform, if the advance angle is 0, the leading edge of the Phase A hall effect sensor signal will align with the intersection of the rising ramp and the flat-top of the back emf trapezoidal waveform.
Conduction Pulse Width	Position sensor conduction pulse width, in electrical deg. Positive conduction pulse can turn on the upper switch and negative pulse can turn on the lower switch in a full bridge inverter. The conduction pulse width is 120 electrical deg. for 120^o conduction mode.
Torque Flag	Output flag for internal developed torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

Attributes based on manufacturer datasheet information:

Parameters	Description
Resistance (phase-phase)	Phase-to-phase (or line-to-line) resistance, in Ohm
Inductance (phase-phase)	Phase-to-phase (or line-to-line) inductance, in H
Speed Constant	Speed constant K_v, defined as the ratio between the speed and the applied voltage, in rpm/V
Torque Constant	Torque constant K_t, defined as the ratio between the generated torque and the applied current, in N*m/A
No. of Poles P	Number of poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
No Load Speed	The motor speed at no load with the nominal voltage applied, in rpm

No Load Current	The current under no load operation, in A
Torque Flag	Output flag for internal developed torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

The node assignments of the image are: Nodes a, b, and c are the stator winding terminals for Phase *A*, *B*, and *C*, respectively. The stator windings are Y connected, and Node n is the neutral point. The shaft node is the connecting terminal for the mechanical shaft. They are all power nodes and should be connected to the power circuit.

Node s_a, s_b, and s_c are the outputs of the built-in 6-pulse hall effect position sensors for Phase *A*, *B*, and *C*, respectively. The sensor output is a bipolar commutation pulse (1, 0, and -1). The sensor output nodes are all control nodes and should be connected to the control circuit.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The equations of the 3-phase brushless dc machine are:

$$v_a = R \cdot i_a + (L - M) \cdot \frac{di_a}{dt} + E_a$$

$$v_b = R \cdot i_b + (L - M) \cdot \frac{di_b}{dt} + E_b$$

$$v_c = R \cdot i_c + (L - M) \cdot \frac{di_c}{dt} + E_c$$

where v_a, $v_{b,}$ and v_c are the phase voltages, i_a, $i_{b,}$ and i_c are the phase currents, *R*, *L*, and *M* are the stator phase resistance, self inductance, and mutual inductance, and E_a, $E_{b,}$ and E_c are the back emf of Phase *A*, *B*, and *C*, respectively.

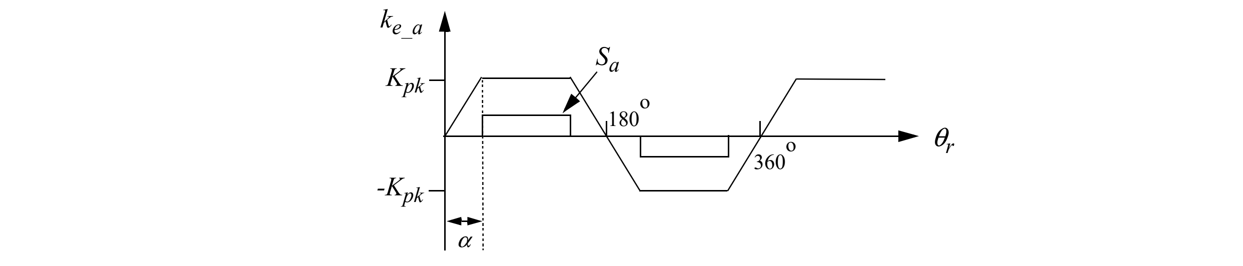
The back emf voltages are a function of the rotor mechanical speed ω_m and the rotor electrical angle θ_{r}, that is:

$$E_a = k_{e\_a} \cdot \omega_m$$

$$E_b = k_{e\_b} \cdot \omega_m$$

$$E_c = k_{e\_c} \cdot \omega_m$$

The coefficients k_{e_a}, $k_{e\_b,}$ and k_{e_c} are dependent on the rotor angle $\theta_{r.}$ In this model, an ideal trapezoidal waveform profile is assumed, as shown below for Phase *A*. Also shown is the Phase *A* hall effect sensor signal S_a.



where K_{pk} is the peak trapezoidal value, in V/(rad./sec.), which is defined as: $K_{pk} = \frac{V_{pk}/krpm}{2} \cdot \frac{1}{1000 \cdot 2\pi/60}$. Given the values of Vpk/krpm and Vrms/krpm, the angle α is determined automatically in the program.

The developed torque of the machine is:

$$T_{em} = (E_a \cdot i_a + E_b \cdot i_b + E_c \cdot i_c)/\omega_m$$

The mechanical equations are:

$$J \cdot \frac{d\omega_m}{dt} = T_{em} - B \cdot \omega_m - T_{load}$$

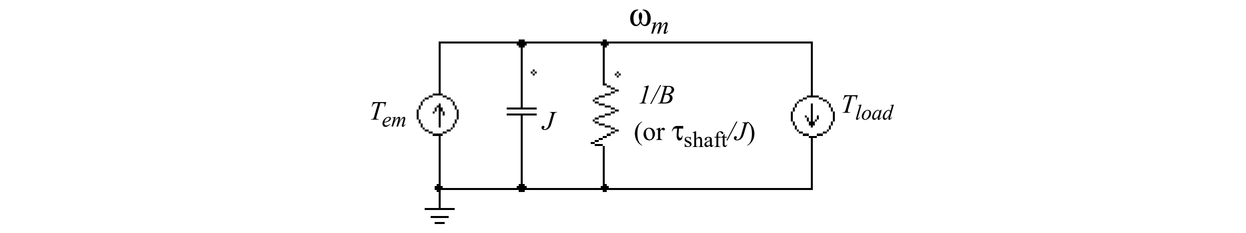
$$\frac{d\theta_r}{dt} = \frac{P}{2} \cdot \omega_m$$

where B is the friction coefficient, T_{load} is the load torque, and P is the number of poles. The coefficient B is calculated from the moment of inertia J and the shaft time constant τ_{shaft} as below:

$$B = \frac{J}{\tau_{shaft}}$$

The shaft time constant τ_{shaft}, therefore, reflects the effect of the friction and windage of the machine. Note that when the shaft time constant is set to 0, the friction term is $B^*\omega_m$ is ignored.

To better understand the definition of the shaft time constant, we can represent the mechanical equation with the following equivalent circuit:



This circuit shows that the shaft time constant τ_{shaft} is equal to the RC time constant of the resistor $1/B$ and the capacitor J. Therefore, the shaft time constant can be measured by the following test:

- Connect the machine to an external mechanical source. With the stator side in open circuit, drive the machine to a certain speed.
- Remove the mechanical source. The shaft time constant will be equal to the time that it takes the machine to decelerate to 36.8% of its initial speed.

More Explanation on the Hall Effect Sensor:

A hall effect position sensor consists of a set of hall switches and a set of trigger magnets.

The hall switch is a semiconductor switch (e.g. MOSFET or BJT) that opens or closes when the magnetic field is higher or lower than a certain threshold value. It is based on the hall effect, which generates an emf proportional to the flux-density when the switch is carrying a current supplied by an external source. It is common to detect the emf using a signal conditioning circuit integrated with the hall switch or mounted very closely to it. This provides a TTL-compatible pulse with sharp edges and high noise immunity for connection to the controller via a screened cable. For a three-phase brushless dc motor, three hall switches are spaced 120 electrical deg. apart and are mounted on the stator frame.

The set of trigger magnets can be a separate set of magnets, or it can use the rotor magnets of the brushless motor. If the trigger magnets are separate, they should have the matched pole spacing (with respect to the rotor magnets), and should be mounted on the shaft in close proximity to the hall switches. If the trigger magnets use the rotor magnets of the machine, the hall switches must be mounted close enough to the rotor magnets, where they can be energized by the leakage flux at the appropriate rotor positions.

Example: Defining Brushless DC Motor Parameters from Manufacturer Datasheet

This example illustrates how to define brushless dc motor parameters from manufacturer datasheet. Below is the information provided on the datasheet of the brushless dc motor *Maxon EC-22-16730* (32V, 50W) from Maxon Motor.

Values at nominal voltage	
Nominal voltage (V)	32
No load speed (rpm)	38700
No load current (mA)	327

Characteristics

Terminal resistance phase to phase (Ohm)	0.363
Terminal inductance phase to phase (mH)	0.049
Torque constant (mNm/A)	7.85
Speed constant (rpm/V)	1220
Rotor inertia (gcm^2)	4.2

Other specifications:

Number of pole pairs	1
Number of phases	3

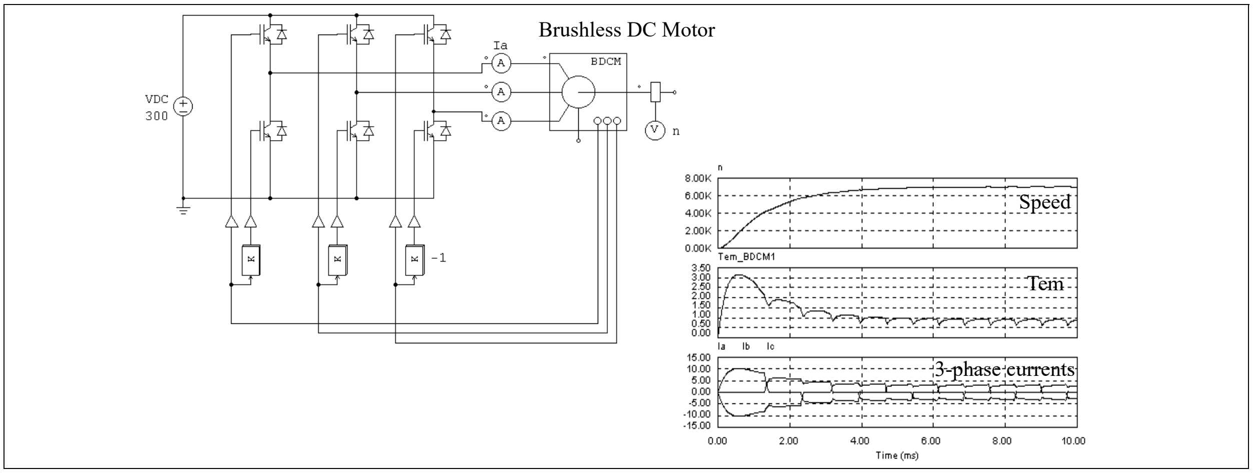
Using the element based on manufacturer datasheet information, and after converting all the quantities to the SI units, the motor parameters in PSIM are defined as follows:

Resistance (phase-phase)	0.363
Inductance (phase-phase)	0.049m
Speed Constant	1220
Torque Constant	7.85m
No. of Poles P	2
Moment of Inertia	4.2e-7
No Load Speed	38700
No Load Current	327m
Torque Flag	1
Master/Slave Flag	1

Example: Start-Up of an Open-Loop Brushless DC Motor

The figure below shows an open-loop brushless dc motor drive system. The motor is fed by a 3-phase voltage source inverter. The outputs of the motor hall effect position sensors are used as the gatings signals for the inverter, resulting a 6-pulse operation.

The simulation waveforms show the start-up transient of the mechanical speed (in rpm), developed torque T_{em}, and 3-phase input currents.



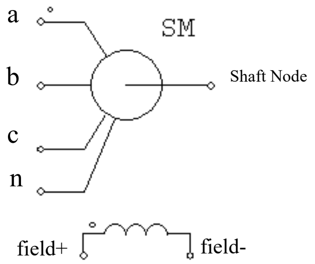
4.6.6 Synchronous Machine with External Excitation

The structure of a conventional synchronous machine consists of three stator windings, one field winding on either a salient or cylindrical rotor, and an optional damping winding on the rotor.

Depending on the way the internal model interfaces with the external stator circuitry, there are two types of interface: one is the voltage-type interface, and the other is the current-type interface. The model for the voltage-type interface consists of controlled voltage sources on the stator side, and this model is suitable in situations where the machine operates as a generator and/or the stator external circuit is in series with inductive branches. On the other hand, The model for the current-type interface consists of controlled current sources on the stator side, and this model is suitable in situations where the machine operates as a motor and/or the stator external circuit is in parallel with capacitive branches.

The image and parameters of the machine are shown as follows.

Image:



Attributes:

Parameters	Description
R_s (stator)	Stator winding resistance, in Ohm
L_s (stator)	Stator leakage inductance, in H
L_{dm} (d-axis mag. ind.)	d-axis magnetizing inductance, in H
L_{qm} (q-axis mag. ind.)	q-axis magnetizing inductance, in H.
R_f (field)	Field winding resistance referred to the stator side, in Ohm
L_{fl} (field leakage ind.)	Field winding leakage inductance referred to the stator side, in H
R_{dr} (damping cage)	Rotor damping cage d-axis resistance referred to the stator side, in Ohm
L_{drl} (damping cage)	Rotor damping cage d-axis leakage inductance referred to the stator side, in H
R_{qr} (damping cage)	Rotor damping cage q-axis resistance referred to the stator side, in Ohm
L_{qrl} (damping cage)	Rotor damping cage q-axis leakage inductance referred to the stator side, in H
Ns/Nf (effective)	Stator-field winding effective turns ratio
Number of Poles P	Number of Poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Torque Flag	Output flag for internal developed torque T_{em}
Master/slave Flag	Master/slave flag of the machine (1: master; 0: slave)

All the parameters are referred to the stator side.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The equations of the synchronous machine can be expressed as follows:

$$[V] = [R] \cdot [I] + \frac{d}{dt}[\lambda]$$

where

$$[V] = \begin{bmatrix} v_a & v_b & v_c & v_f & 0 & 0 \end{bmatrix}^T \qquad [I] = \begin{bmatrix} i_a & i_b & i_c & i_f & i_{dr} & i_{qr} \end{bmatrix}^T$$

$$[R] = diag\begin{bmatrix} R_s & R_s & R_s & R_f & R_{dr} & R_{qr} \end{bmatrix} \quad [\lambda] = \begin{bmatrix} \lambda_a & \lambda_b & \lambda_c & \lambda_f & \lambda_{dr} & \lambda_{qr} \end{bmatrix}^T$$

and [λ] = [L]*[I]. The inductance matrix is defined as follows:

$$[L] = \begin{bmatrix} [L_{11}] & [L_{12}] \\ [L_{12}]^T & [L_{22}] \end{bmatrix}$$

and

$$[L_{11}] = \begin{bmatrix} L_s + L_o + L_2\cos(2\theta_r) & -\frac{L_o}{2} + L_2\cos\left(2\theta_r - \frac{2\pi}{3}\right) & -\frac{L_o}{2} + L_2\cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ -\frac{L_o}{2} + L_2\cos\left(2\theta_r - \frac{2\pi}{3}\right) & L_s + L_o + L_2\cos\left(2\theta_r + \frac{2\pi}{3}\right) & -\frac{L_o}{2} + L_2\cos(2\theta_r) \\ -\frac{L_o}{2} + L_2\cos\left(2\theta_r + \frac{2\pi}{3}\right) & -\frac{L_o}{2} + L_2\cos(2\theta_r) & L_s + L_o + L_2\cos\left(2\theta_r - \frac{2\pi}{3}\right) \end{bmatrix}$$

$$[L_{12}] = \begin{bmatrix} L_{sf}\cos(2\theta_r) & L_{sd}\cos(2\theta_r) & -L_{sq}\sin(2\theta_r) \\ L_{sf}\cos\left(2\theta_r - \frac{2\pi}{3}\right) & L_{sd}\cos\left(2\theta_r - \frac{2\pi}{3}\right) & -L_{sq}\sin\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{sf}\cos\left(2\theta_r + \frac{2\pi}{3}\right) & L_{sd}\cos\left(2\theta_r + \frac{2\pi}{3}\right) & -L_{sq}\sin\left(2\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}$$

$$[L_{22}] = \begin{bmatrix} L_f & L_{fdr} & 0 \\ L_{fdr} & L_{dr} & 0 \\ 0 & 0 & L_{qr} \end{bmatrix}$$

where θ_r is the rotor angle.

The developed torque can be expressed as:

$$T = \frac{P}{2} \cdot [I] \cdot \frac{d}{d\theta_r}[L] \cdot [I]$$

The mechanical equations are:

$$J \cdot \frac{d\omega_m}{dt} = T_{em} - T_{load}$$

$$\frac{d\theta_r}{dt} = \frac{P}{2} \cdot \omega_m$$

The input parameters and the parameters in the equations are related in the following way:

$$L_{dm} = 1.5 \cdot (L_o + L_2)$$

$$L_{qm} = 1.5 \cdot (L_o - L_2)$$

$$L_{sf} = \left(\frac{N_f}{N_s}\right)^2 \cdot L_{dm}$$

$$L_{sd} = L_{dm}$$

$$L_{sq} = L_{qm}$$

$$L_{fl} = L_f - L_{sf}$$

$$L_{fdr} = L_{dm}$$

$$L_{drl} = L_{dr} - L_{dm}$$

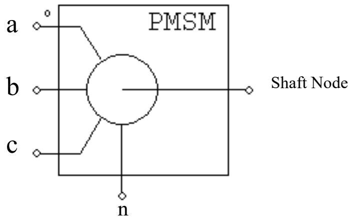
$$L_{qrl} = L_{qr} - L_{qm}$$

4.6.7 Permanent Magnet Synchronous Machine

A 3-phase permanent magnet synchronous machine has 3-phase windings on the stator, and permanent magnet on the rotor. The difference between this machine and the brushless dc machine is that the machine back emf is sinusoidal.

The image and parameters of the machine are shown as follows.

Image:



Attributes:

Parameters	Description
R_s (stator resistance)	Stator winding resistance, in Ohm
L_d (d-axis ind.)	Stator d-axis inductance, in H
L_q (q-axis ind.)	Stator q-axis inductance, in H. The d-q coordinate is defined such that the d-axis passes through the center of the magnet, and the q-axis is in the middle between two magnets. The q-axis is leading the d-axis.
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed). The value of Vpk/krpm should be available from the machine data sheet. If this data is not available, it can be obtained through an experiment by operating the machine as a generator at 1000 rpm and measuring the peak line-to-line voltage.
No. of Poles *P*	Number of poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Shaft Time Constant	Shaft time constant τ_{shaft}
Initial Rotor Angle	Initial rotor electrical angle, in deg.
Torque Flag	Output flag for internal developed torque T_{em}
Master/slave Flag	Master/slave flag of the machine (1: master; 0: slave)

The node assignments of the image are: Nodes *a*, *b*, and *c* are the stator winding terminals for Phase *a*, *b*, and *c*, respectively. The stator windings are Y connected, and Node *n* is the neutral point. The shaft node is the connecting terminal for the mechanical shaft. They are all power nodes and should be connected to the power circuit.

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The equations of the permanent-magnet synchronous machine are:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}$$

where v_a, $v_{b,}$ v_c, and i_a, $i_{b,}$ and i_c, and λ_a, λ_b, λ_c are the stator phase voltages, currents, and flux linkages, respectively, and R_s is the stator phase resistance. The flux linkages are further defined as:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_{pm} \cdot \begin{bmatrix} \cos(\theta_r) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}$$

where θ_r is the rotor electrical angle, and λ_{pm} is the peak stator phase flux linkage defined as:

$$\lambda_{pm} = \frac{60 \cdot V_{pk}/krpm}{\sqrt{3} \cdot \pi \cdot P \cdot 1000}$$

where P is the number of poles.

The stator self and mutual inductances are rotor position dependent, and are defined as:

$$L_{aa} = L_s + L_o + L_2 \cdot \cos(2\theta_r)$$

$$L_{bb} = L_s + L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right)$$

$$L_{cc} = L_s + L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right)$$

$$L_{ab} = L_{ba} = -\frac{L_o}{2} + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right)$$

$$L_{ac} = L_{ca} = -\frac{L_o}{2} + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right)$$

$$L_{bc} = L_{cb} = -\frac{L_o}{2} + L_2 \cdot \cos(2\theta_r)$$

where L_s is the stator leakage inductance. The variables in the abc frame can be converted to the dq0 frame using the following transformation:

$$\begin{bmatrix} x_d \\ x_q \\ x_0 \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}$$

The inverse transformation is as follows:

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) & 1 \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \\ x_0 \end{bmatrix}$$

The d-axis and q-axis inductances are associated with the inductances in the abc frame as follows:

$$L_d = L_s + \frac{3}{2}L_o + \frac{3}{2}L_2$$

$$L_q = L_s + \frac{3}{2}L_o - \frac{3}{2}L_2$$

The developed torque can be expressed as:

$$T_{em} = -\frac{P}{2} \cdot L_2 \cdot \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \cdot \begin{bmatrix} \sin(2\theta_r) & \sin\left(2\theta_r - \frac{2\pi}{3}\right) & \sin\left(2\theta_r + \frac{2\pi}{3}\right) \\ \sin\left(2\theta_r - \frac{2\pi}{3}\right) & \sin\left(2\theta_r + \frac{2\pi}{3}\right) & \sin(2\theta_r) \\ \sin\left(2\theta_r + \frac{2\pi}{3}\right) & \sin(2\theta_r) & \sin\left(2\theta_r - \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \frac{P}{2} \cdot \lambda_{pm} \cdot \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \cdot \begin{bmatrix} \sin(\theta_r) \\ \sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}$$

In the dq0 frame, the developed torque can be expressed as:

$$T_{em} = \frac{3}{2} \cdot \frac{P}{2} \cdot (\lambda_{pm} i_q + (L_d - L_q) i_d i_q)$$

The mechanical equations are:

$$J \cdot \frac{d\omega_m}{dt} = T_{em} - B \cdot \omega_m - T_{load}$$

$$\frac{d\theta_r}{dt} = \frac{P}{2} \cdot \omega_m$$

where B is a coefficient, T_{load} is the load torque, and P is the number of poles. The coefficient B is calculated from the moment of inertia J and the shaft time constant τ_{shaft} as below:

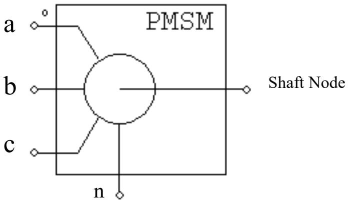
$$B = \frac{J}{\tau_{shaft}}$$

4.6.8 Permanent Magnet Synchronous Machine with Saturation

A 3-phase PMSM machine with saturation differs from that of a linear 3-phase PMSM machine in that the d-axis and q-axis magnetizing inductances L_{dm} and L_{qm} can be expressed as a nonlinear function of the d-axis and q-axis currents in the lookup table form.

The image and parameters of the machine are shown as follows.

Image:



Attributes:

Parameters	Description
R_s (stator resistance)	Stator winding resistance, in Ohm
L_s (stator leakage ind.)	Stator d-axis inductance, in H
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed). This value should be available from the machine data sheet. If not, it can be obtained through experiment by operating the machine as a generator at 1000 rpm and measuring the peak line-to-line voltage.
No. of Poles P	Number of poles P
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Shaft Time Constant	Shaft time constant τ_{shaft}, in sec. It is associated with the friction coefficient B as: B = J / $\tau_{shaft.}$
Initial Rotor Angle	Initial rotor electrical angle, in deg.
L_d Lookup Table File	File name of the lookup table for L_{dm}
L_q Lookup Table File	File name of the lookup table for L_{qm}
dq Flag	Flag for the lookup table. When the flag is 0, L_{dm} and L_{qm} are function of I_d and I_q. When the flag is 1, L_{dm} and L_{qm} are function of the current magnitude I_m and the angle.
Transformation Flag	Flag for the transformation convention (see details below)
Torque Flag	Output flag for internal developed torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The relationship between the d-axis and q-axis inductances L_d and L_q and the magnetizing inductances L_{dm} and L_{qm} is as follows:

$$L_d = L_s + L_{dm}$$

$$L_q = L_s + L_{qm}$$

where L_s is the stator leakage inductance. Since L_s is normally very small, L_d can be considered equivalent to

L_{dm}, and L_q can be considered equivalent to L_{qm}.

The *Transformation Flag* defines the transformation convention between the abc frame and the dq frame. When the *Transformation Flag* is 0:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

$$I_m = \sqrt{I_d^2 + I_q^2}$$

$$\theta_m = \operatorname{atan2}(I_q, I_d)$$

The current vector angle is in deg., and is from -180^o to 180^o.

When the *Transformation Flag* is 1:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

$$I_m = \sqrt{\frac{2}{3}} \cdot \sqrt{I_d^2 + I_q^2}$$

$$\theta_m = \operatorname{atan2}(-I_d, I_q)$$

The current vector angle is in deg., and is from 0^o to 360^o.

The L_{dm} and L_{qm} lookup tables have the following format:

M, N
$V_{r1}, V_{r2}, ..., V_{rM}$
$V_{c1}, V_{c2}, ..., V_{cN}$
$L_{11}, L_{12}, ..., L_{1N}$
$L_{21}, L_{22}, ..., L_{2N}$
... ...
$L_{M1}, L_{M2}, ..., L_{MN}$

where M is the number of rows and N is the number of columns; V_r is the row vector and V_c is the column vector; and $L_{i,j}$ is the L_{dm} or L_{qm} inductance value, in H, at Row i and Column j. Note that Vectors V_r and V_c must be monotonically increasing.

When the *dq flag* is 0, the row vector is I_q and the column vector is I_d. When the flag is 1, the row vector is the angle and the column vector is the amplitude I_m.

If the input is between two points, interpolation is used to calculate the value. If the input is less than the minimum or greater than the maximum value, the input will be set to the minimum or maximum value.

This PMSM model with saturation can also be used as the linear PMSM model if the lookup tables are defined such that L_{dm} and L_{qm} are linear function of I_d and I_q.

The following shows an example of the lookup table:

4,15

-5.7155 -4.8990 -4.0825 -3.2660

-5.7155 -4.8990 -4.0825 -3.2660 -2.4495 -1.6330 -0.8165 0 0.8165 1.6330 2.4495 3.2660 4.0825 4.8990

5.7155

0.0109 0.0109 0.0107 0.0104 0.0102 0.0100 0.0098 0.0098 0.0098 0.0100 0.0102 0.0104 0.0107 0.0109 0.0109

0.0109 0.0109 0.0109 0.0106 0.0109 0.0106 0.0105 0.0105 0.0105 0.0106 0.0109 0.0106 0.0109 0.0109 0.0109

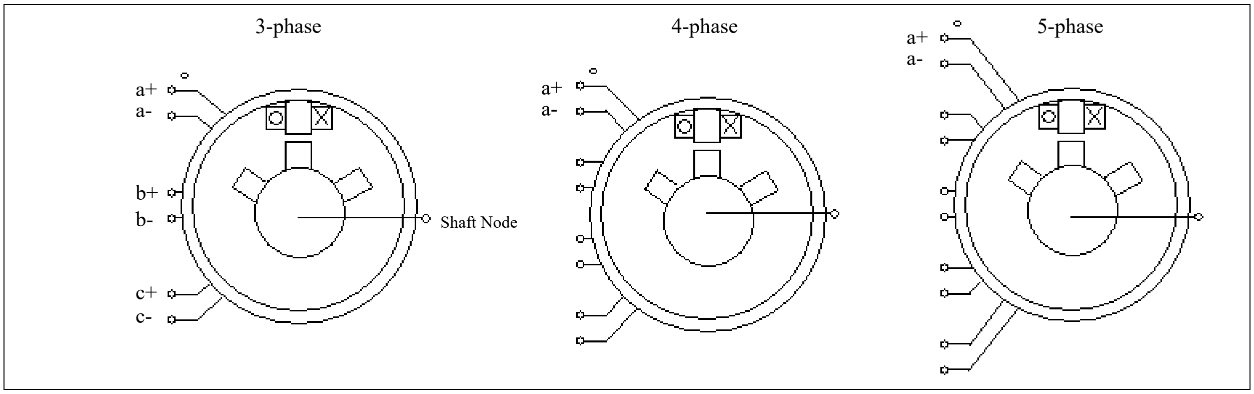
0.0109 0.0109 0.0109 0.0109 0.0111 0.0108 0.0106 0.0106 0.0106 0.0108 0.0111 0.0109 0.0109 0.0109 0.0109

0.0110 0.0110 0.0111 0.0110 0.0110 0.0109 0.0108 0.0107 0.0108 0.0109 0.0110 0.0110 0.0111 0.0110 0.0110

4.6.9 Switched Reluctance Machine

Three types of switched reluctance machine models are provided: 3-phase, 4-phase, and 5-phase.

Images:



Attributes:

Parameters	Description
Resistance	Stator phase resistance R, in Ohm
Inductance L_{min}	Minimum phase inductance L_{min}, in H
Inductance L_{max}	Maximum phase inductance L_{max}, in H
Theta_min (deg.)	Duration of the interval θ_{min} where the inductance is at the minimum, in deg.
Theta_max (deg.)	Duration of the interval θ_{max} where the inductance is at the maximum, in deg.
Stator Pole Number	Number of stator poles (teeth)
Rotor Pole Number	Number of rotor poles (teeth)
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Torque Flag	Output flag for internal torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

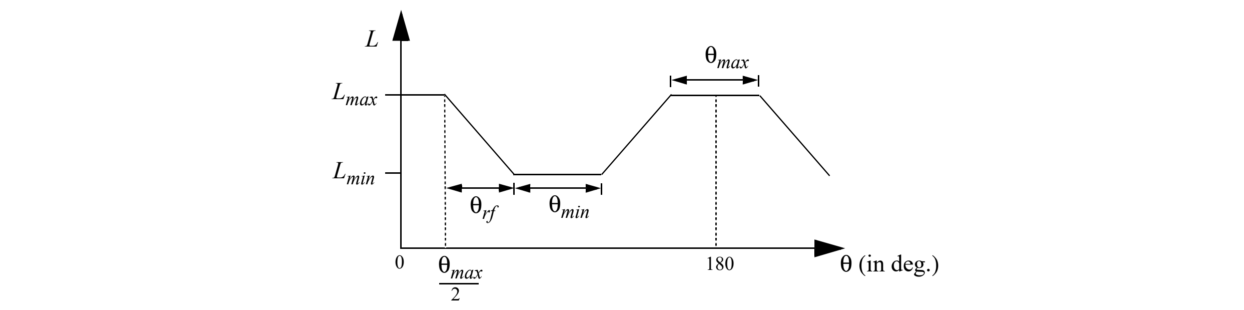
For more details on the definition and use of the master/slave flag, refer to Section 4.6.1.

The node assignments are: Nodes a+, a-, b+, b-, c+, c-, etc. are the stator winding terminals for Phase *a*, *b*, c, *d*, and *e*, respectively. The shaft node is the connecting terminal for the mechanical shaft. They are all power nodes and should be connected to the power circuit.

The equation of the switched reluctance machine for one phase is:

$$v = i \cdot R + \frac{d(L \cdot i)}{dt}$$

where v is the phase voltage, i is the phase current, R is the phase resistance, and L is the phase inductance. The phase inductance L is a function of the rotor angle θ (in electrical deg.), as shown in the following figure. The inductance profile is half-wave symmetrical. That is, it repeats itself after 180^o.



The rotor angle is defined such that, when the stator and the rotor teeth are completely aligned, θ = 0. The value of the inductance can be in either rising stage, flat-top stage, falling stage, or flat-bottom stage.

If we define the constant k as the rising slope of the inductance from L_{min} to L_{max} (which is the same as the falling slope from L_{max} to L_{min}), and the interval of the rising/falling slope is θ_{rf}, we can express the inductance L as a function of the rotor angle θ (from 0 to 180^o) as follows:

$$L = L_{max} \quad \text{for} \quad 0 \le \theta \le \frac{\theta_{max}}{2}$$

$$L = L_{max} - k \cdot \left(\theta - \frac{\theta_{max}}{2}\right) \quad \text{for} \quad \frac{\theta_{max}}{2} \le \theta \le \frac{\theta_{max}}{2} + \theta_{rf}$$

$$L = L_{min} \quad \text{for} \quad \frac{\theta_{max}}{2} + \theta_{rf} \le \theta \le \frac{\theta_{max}}{2} + \theta_{rf} + \theta_{min}$$

$$L = L_{min} + k \cdot \left(\theta - \frac{\theta_{max}}{2} - \theta_{rf} - \theta_{min}\right) \quad \text{for} \quad \frac{\theta_{max}}{2} + \theta_{rf} + \theta_{min} \le \theta \le \frac{\theta_{max}}{2} + 2\theta_{rf} + \theta_{min}$$

$$L = L_{max} \quad \text{for} \quad \frac{\theta_{max}}{2} + 2\theta_{rf} + \theta_{min} \le \theta \le 180^o$$

The developed torque of the machine per phase is:

$$T_{em} = \frac{1}{2} \cdot i^2 \cdot \frac{dL}{d\theta}$$

Based on the inductance expression, we have the developed torque in each stage as:

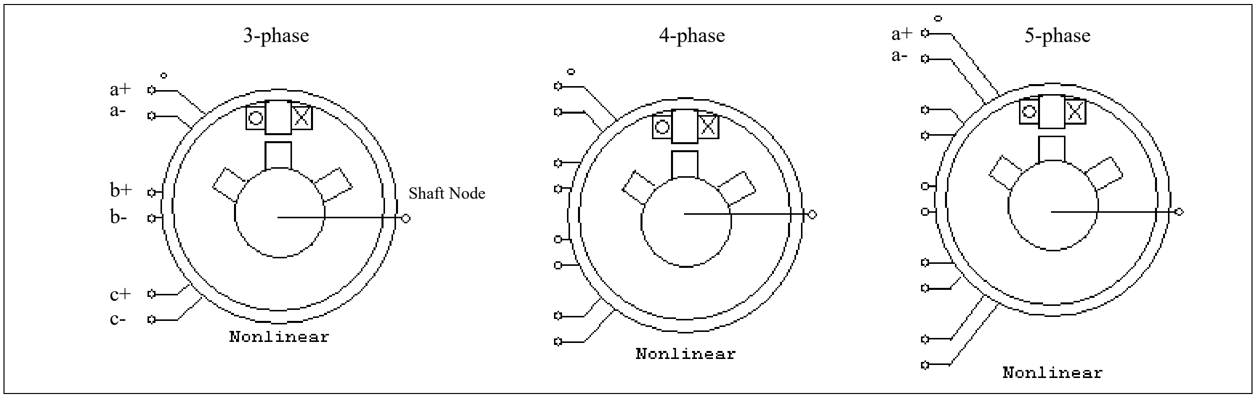
$T_{em} = i^2 * k / 2$	[rising stage]
$T_{em} = 0$	[flat-top stage]
$T_{em} = - i^2 * k / 2$	[falling stage]
$T_{em} = 0$	[flat-bottom stage]

4.6.10 Nonlinear Switched Reluctance Machine

In the previous switched reluctance models, the inductance is only a function of the rotor position, and it remains constant during the flat-top and flat-bottom states. In the nonlinear models, the inductance can be a nonlinear function of not only the rotor position, but also the current. The relationship between the inductance and the rotor position and current is defined through a 2-dimensional lookup table.

The images and parameters are shown as follows.

Images:



Attributes:

Parameters	Description
Resistance	Stator phase resistance R, in Ohm
Stator Pole Number	Number of stator poles (teeth)
Rotor Pole Number	Number of rotor poles (teeth)
Inductance Table File	File that stores the 2-dimensional table of the inductance versus the rotor position and current.
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Torque Flag	Output flag for internal torque T_{em}
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)

The inductance table file has the following format:

m, n
$\theta_1, \theta_2, ..., \theta_m$
$I_1, I_2, ..., I_n$
$L_{1,1}, L_{1,2}, ..., L_{1,n}$
$L_{2,1}, L_{2,2}, ..., L_{2,n}$
... ...
$L_{m,1}, L_{m,2}, ..., L_{m,n}$

where m is the number of rows and n is the number of columns; θ_1 to θ_m is the row vector for the rotor position (in deg.), and I_1 and I_n is the column vector for the phase current (in A); and $L_{i,j}$ is the inductance value (in H) at Row i and Column j. For example, when the rotor position is θ_2 and the phase current is I_1, the inductance is $L_{2,1}$.

Because of the half-wave symmetry, only inductance values from 0 to 180^o need to be provided. Note that the row vector and column vector must be monotonically increasing.

The equation of a nonlinear switched reluctance machine for one phase is:

$$v = i \cdot R + \frac{d(L \cdot i)}{dt}$$

where v is the phase voltage, i is the phase current, R is the phase resistance, and L is the phase inductance which depends on both the rotor angle and the current as defined by the lookup table.

The developed torque per phase is:

$$T_{em} = \frac{dW_{co}}{d\theta}$$

where W_{co} is the co-energy defined as:

$$W_{co} = \int_0^{i_o} (L \cdot i) \cdot di$$

The integration and derivative calculations are done using the inductance lookup table.

4.6.11 Motor Control Blocks

The following control blocks for different machine types are provided:

- Maximum-Torque-Per-Ampere Control
- Field Weakening Control

4.6.11.1 Maximum-Torque-Per-Ampere Control

There are two Maximum-Torque-Per-Ampere (MTPA) control blocks in PSIM's Motor Drive Module. Their images are shown below. On the left side is the MTPA for linear internal permanent magnet synchronous machine (linear IPM), and on the right side is for nonlinear IPM

Images:

MTPA (IPM)
Id
Is
Iq
MTPA (NL)
Is
Id
Ld
Lq
Lambda
Iq

Attributes of MTPA for Linear IPM:

Parameters	Description
Ld (d-axis inductance)	d-axis inductance of the PMSM machine, in H
Lq (q-axis inductance)	q-axis inductance of the PMSM machine, in H
Vpk / krpm	Peak line-to-line back emf constant of the machine, in V/krpm (mechanical speed)
Number of Poles	Number of poles of the machine
Maximum Inverter Current	Maximum inverter output current amplitude (peak), in A
Base Voltage Value	System base voltage value, in V
Base Current Value	System base current value, in A
Base Mechanical Speed	System base mechanical speed, in rad/sec

The Maximum-Torque-Per-Ampere control block has the following inputs and outputs, all in per-unit values:

Is: Inverter current amplitude reference, input.
Id: d-axis current reference, output.

Iq:	q-axis current reference, output

This block is for the control of linear PMSM machines only. Given the current amplitude reference, this block will calculate the d-axis and q-axis current references Id and Iq such that the maximum amount of torque is generated by the machine.

The block requires the parameters of the PMSM controlled and the inverter current rating.

Attributes MPTA for Nonlinear IPM:

Parameters	Description
Maximum Inverter Current	Maximum inverter output current amplitude (peak), in A
Base Current Value	System base current value, in A

This block is for the control of nonlinear internal permanent magnet synchronous machines (IPM) only. It has the following inputs and outputs, all in per unit value except the inductances Ld and Lq and the flux linkage Lambda. If the base value Ib is set to 1, all input and output quantities are in real values.

Input Signals:

Is:	Inverter current amplitude reference
Ld:	d-axis inductance Ld, in H
Lq:	q-axis inductance Lq, in H
Lambda:	Peak stator phase flux linkage, in Weber

Output Signals:

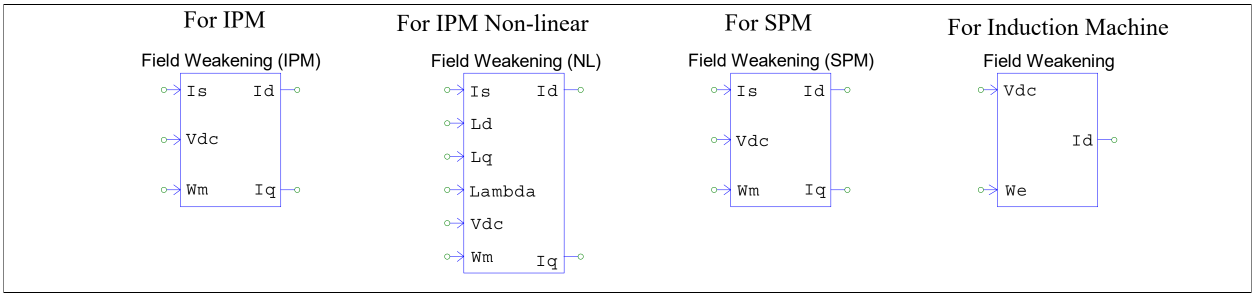
Id:	d-axis current reference
Iq:	q-axis current reference

Given the current amplitude reference, this block will calculate the d-axis and q-axis current references Id and Iq such that the maximum amount of torque is generated by the machine.

4.6.11.2 Field Weakening Control

The images of field weakening control blocks are shown below.

Images:



Attributes for Field Weakening (IPM):

Parameters	Description
Rs (stator resistance)	Stator winding resistance of the machine, in Ohm
Ld (d-axis inductance)	d-axis inductance of the PMSM machine, in H
Lq (q-axis inductance)	q-axis inductance of the PMSM machine, in H
Vpk / krpm	Peak line-to-line back emf constant of the machine, in V/krpm (mechanical speed)
Number of Poles	Number of poles of the machine

Maximum Inverter Current	Maximum inverter output current amplitude (peak), in A
Maximum Inverter Voltage	Maximum inverter output voltage amplitude (phase peak), in V
Base Voltage Value	System base voltage value, in V
Base Current Value	System base current value, in A
Base Mechanical Speed	System base mechanical speed, in rad/sec

The field weakening (IPM) control block accepts the following input signals, all in per-unit value.

Is: Inverter current amplitude reference
Vdc: DC bus voltage feedback
Wm: Motor mechanical speed, in rad/sec.

It has the following output signals, all in per-unit value:

Id: d-axis current reference
Iq: q-axis current reference

This block is for the control of linear PMSM machines only. When the machine speed is higher than a certain value, the machine will not be able to provide the maximum amount of torque. Instead, it will be limited by the machine power rating. Given the current amplitude reference, dc bus voltage, and the motor speed, the field weakening block will calculate the d-axis and q-axis current references Id and Iq to achieve a maximum power operation.

This block requires the parameters of the PMSM controlled and the inverter voltage and current ratings.

Attributes for Field Weakening (Non-Linear IPM):

Parameters	Description
Number of Poles	Number of poles of the machine
Maximum Inverter Current	Maximum inverter output current amplitude (peak), in A
Maximum Inverter Voltage	Maximum inverter output voltage amplitude (phase peak), in V
Base Voltage Value	System base voltage value, in V
Base Current Value	System base current value, in A
Base Mechanical Speed	System base mechanical speed, in rad/sec

The block has the following inputs and outputs, all in per unit value except Ld, Lq, and Lambda. If base voltage value Vb, Ib, and Wmb are all set to 1, all input and output quantities are in true values.

Input Signals:

Is: Inverter current amplitude reference.
Ld: d-axis inductance at the rated operating conditions, in real value in H
Lq: q-axis inductance at the rated operating conditions, in real value in H
Lambda: Peak stator phase flux linkage at the rated operating conditions, in real value in Weber
Vdc: DC bus voltage feedback
Wm: Motor mechanical speed in

Output Signals:

Id: d-axis current reference.
Iq: q-axis current reference.

This block is for the control of nonlinear interior PMSM (IPM) machines only. It is not for surface mounted PMSM (SPM) machines.

When the machine speed is higher than a certain value, the machine will not be able to generate the maximum torque. Instead, it will be limited by the machine power rating. Given the current amplitude reference, dc bus voltage, and the motor speed, the field weakening control block will calculate the d-axis and q-axis current references Id and Iq to operate in the constant power region.

Attributes for Field Weakening (SPM):

Parameters	Description
Ld (d-axis inductance)	The inductance on d-axis, in H
Vpk/krpm	The ratio of peak voltage versus speed. in V/krpm
Number of Poles	Number of poles of the machine
Maximum Inverter Current	Maximum inverter output current amplitude (peak), in A
Maximum Inverter Voltage	Maximum inverter output voltage amplitude (phase peak), in V
Base Voltage Value	System base voltage value, in V
Base Current Value	System base current value, in A
Base Mechanical Speed	System base mechanical speed, in rad/sec

The block has the following inputs and outputs, all in per unit value. If base values Vb, Ib, and Wmb are all set at 1, input and output quantities are in real values.

Input Signals:

Is:	Inverter current amplitude reference
Vdc:	DC bus voltage feedback
Wm:	Motor mechanical speed

Output Signal:

Id:	d-axis current reference
Iq:	q-axis current reference

This block is for the control of linear surface-mounted PMSM (SPM) machines only. It is not for internal PMSM (IPM) machines. In a SPM, the d-axis and q-axis inductance values are equal.

When the machine speed is higher than a certain value, the machine will not be able to generate the maximum torque. Instead, it will be limited by the machine power rating. Given the current amplitude, dc bus voltage, and the motor speed, the field weakening control block will calculate the d-axis and q-axis current references Id and Iq to operate in the constant power region.

Attributes for Field Weakening (IPM):

Parameters	Description
Rs (stator)	Stator winding resistance of the machine, in Ohm
Lls (stator leakage)	Stator leakage inductance of the PMSM machine, in H
Rr (rotor)	Rotor winding resistance of the machine, in Ohm
Llr (rotor leakage)	Rotor leakage inductance of the machine, in H
Lm (magnetizing)	Magnetizing inductance, in H
Number of Poles	Number of poles of the machine
Rated Speed (rpm)	Rated speed of the machine, in rpm
AC Source Frequency	AC source frequency, in Hz
AC Voltage (line-line rms)	AC source voltage, line-to-line rms, in V
Base Voltage Value	System base voltage value, in V
Base Current Value	System base current value, in A

The block has the following inputs and outputs, all in per unit value. If base values Vb, Ib, and Wmb are all set at 1.0, all input and output quantities are in real values.

Input Signals:
Vdc: DC bus voltage
We: Stator electrical speed (in per unit, with the base value of Wmb)

Output Signal:
Id: d-axis current reference

This block is for the control of linear induction machines only.

When the machine speed is higher than a certain value, the machine will not be able to generate the maximum torque. Instead, it will be limited by the machine power rating. Given the dc bus voltage and the stator speed, the field weakening control block will calculate the d-axis current reference Id to operate in the constant power region.

4.7 MagCoupler Module

The MagCoupler Module provides interface for co-simulation between PSIM and the software JMAG. JMAG is an electromagnetic field analysis software for the development and design of electric machines, actuators, and other electrical and electronic devices and components. With the MagCoupler Module, one can perform power electronics and control in PSIM, and electromagnetic field analysis in JMAG.

The MagCoupler Module includes the MagCoupler-DL block and the MagCoupler block, as well as the mechanical elements and speed/torque/position sensors as described in Section 4.9.

The differences between the MagCoupler-DL block and the MagCoupler block and are:

- With the MagCoupler block, PSIM and JMAG run at the same time step, whereas with the MagCoupler-DL block, PSIM and JMAG can run at different time steps. The flexibility to set different time steps would make it possible to run JMAG at a larger time step and speed up the overall simulation.
- The interface ports of the MagCoupler block are signal ports. That is, electric currents do not flow into or out of the interface ports, and an electric current must first be converted into a voltage signal before it can be connected to the block. On the other hand, the MagCoupler-DL block is a native power element, and the interface ports behavior in the same way as other power circuit nodes (i.e. voltages can be applied and currents can flow in and out of the nodes). Because of the difference, the MagCoupler-DL block provides more efficient and robust interface between PSIM and JMAG.

To run the MagCoupler Module, the JMAG installation directory and the directories that include the JMAG DLL files "jbdll.dll" and "RT_Dll.dll" must be added to the PSIM's search path. To add to PSIM's search path, in PSIM, go to **Options** >> **Set Path**, and click on **Add Folder**.

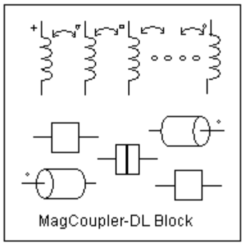
Also, the MagCoupler Module requires Microsoft Internet Explorer Version 6 or higher. It will not work with Internet Explorer Version 5.

The description of the MagCoupler-DL block and the MagCoupler block is given below.

4.7.1 MagCoupler-DL Block

The MagCoupler-DL block has the following image and properties.

Image:



Attributes:

Parameter	Description
Netlist XML File	The file defines the interface between PSIM and JMAG. The file extension is .xml.
JMAG Input File	The JMAG input data file. It has the .jcf extension. Note that the .xml and .jcf files must be in the same directory.
JMAG Case Text	Comments for the JMAG circuit
Terminal Names	Terminal names of the block

The power circuit nodes and mechanical shaft nodes, as well as the rest of the interface between PSIM and the JMAG data files, are defined in the *Netlist XML File*. This file is in XML format, and is generated by JMAG. To specify this file, click on the browse button at the right of the edit field.

When a specific XML is selected, the .jcf data file will be read, and the terminals (nodes) of the block will be displayed. In the block image, the power circuit nodes will be placed at the top of the block, arranged from the left to the right. The mechanical shaft nodes will be placed on the left and right of the block, with the first shaft node on the right, and the second shaft node on the left.

The *JMAG Input File* is the JMAG data file for the device modeled. The file has the .jcf extension and is defined in the netlist XML file. Note that the .jcf file and the .xml file must be in the same directory.

The *JMAG Case Text* is a text identifying the specific JMAG study case. It can be any text.

The parameters of a sample permanent-magnet synchronous machine is shown below:

Parameter	Description
dt_of_JMAG	Time step used in JMAG.
Ir	Rated current. It is used in JMAG
dI_threshold	Current threshold. It is used in JMAG
dr_threshold	Displacement threshold. It is used in JMAG
Moment of Inertia	Moment of inertia of the machine, in kg*m^2.
Shaft Time Constant	Shaft time constant of the machine, in sec.
Back emf Flag	Display flag for all FEM coil currents (1: display; 0: no display)
Rotor Angle Flag	Display flag for all shaft angles
Speed Flag	Display flag for all shaft speeds
Torque Flag	Display flag for all developed torques
shaft1 Master Flag	Master/slave flag of the machine (1: master; 0: slave)

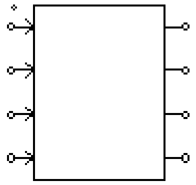
Please consult JMAG manuals and documents on how to set up the co-simulation on the JMAG side.

4.7.2 MagCoupler Block

The MagCoupler block has the following image and properties.

Image:

Block with 4 inputs and 4 outputs



Attributes:

Parameter	Description
Link Table File	The XML file that defines the interface between PSIM and JMAG. It has the .xml extension.
JMAG Input File	The JCF input data file for JMAG. It has the .jcf extension. Note that the .xml file and the .jcf file must be in the same directory.

JMAG Case Text	Comments for the JMAG circuit
IN Nodes	Nodes that pass the values from PSIM to JMAG
OUT Nodes	Nodes that pass the values from JMAG to PSIM

The number of input and output nodes may vary, depending on the actual number of input/output nodes in a particular circuit.

The MagCoupler block accepts voltages, currents, and positions as inputs, and it provides voltages, currents, positions, torques, and force as the outputs. In PSIM, the MagCoupler block is a power circuit element. The way it interfaces with the rest of the circuit is that both the inputs and outputs are voltage signals (no electric current flows into the input node). To convert a branch current into a voltage signal, or vice versa, one can use a current-controlled voltage source, or voltage-controlled current source.

The *Link Table File*, in XML format, defines the input/output interface and corresponding functions in JMAG. This file is generated automatically by JMAG. To locate this file, click on the browse button at the right of the edit field.

The *JMAG Input File* is the JCF input data file that is read by the JMAG solver. The name is defined in the Link Table File.

Note that JCF input file .jcf must be in the same directory as the input link table file .xml. If any material database is used in JMAG, it should also be placed in the directory of the .xml file. Also, the .xml file does not have to be in the same directory as the schematic file. However, if a .xml file with the same name is present in the schematic directory, PSIM will read the one in the schematic directory first.

The *JMAG Case Text* is a text identifying the specific JMAG circuit. It can be any text describing the JMAG circuit.

The *IN Nodes* are the nodes through which PSIM passes the values to JMAG. In the MagCoupler block image, the order of the input nodes is from the top to the bottom. The order can be changed by highlighting the node and click on the upper or down arrow.

The *OUT Nodes* are the nodes through which JMAG passes the values back to PSIM. In the MagCoupler block image, the order of the output nodes is from the top to the bottom. The order can be changed by highlighting the node and click on the upper or down arrow.

By clicking on the **Edit Image** button, one can edit and customize the image of the MagCoupler block. Clicking on the **Display File** button will display the Link Table File in the Microsoft Internet Explorer environment, and clicking on the **Read File** button will read or re-load the Link Table File.

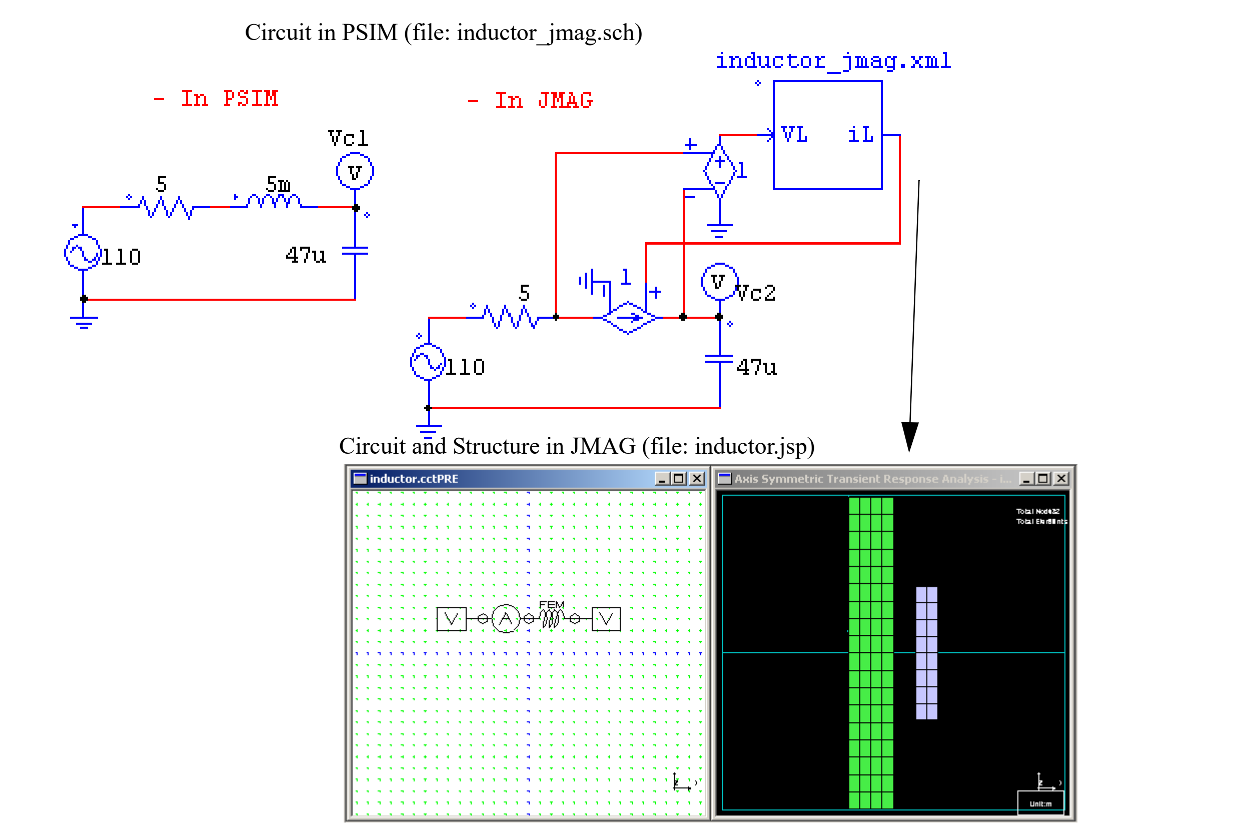
Set-up in JMAG and PSIM:

Using the MagCoupler block, it is easy to set up the link between JMAG and PSIM for co-simulation. It involves two main steps: setting up the circuit in JMAG and generating the link table file, and loading the link table file into PSIM.

An inductor example below is used to illustrate this process.

In the PSIM circuit of this example, the circuit on the left uses the built-in inductor element from the PSIM library, and the circuit on the right has the inductor implemented in JMAG. In this case, the inductor is modelled as a controlled current source in PSIM. The voltage across the inductor is first converted to a node-to-ground voltage through a voltage-controlled voltage source, and the value is passed to the input node VL of the MagCoupler block. At each time step, PSIM calls JMAG functions which calculate the inductor current based on the voltage input. This current is then sent back to PSIM in the voltage form, and is used to control the current source that represents the inductor.

In the JMAG circuit of this example, the voltage function (on the left side) receives the voltage from PSIM, and through the current probe in series with the FEM coil, the current is calculated and sent back to PSIM. The inductor structure in the JMAG environment is shown on the lower right.



Circuit in PSIM (file: inductor_jmag.sch)

Circuit and Structure in JMAG (file: inductor.jsp)

The setup process of calling JMAG in PSIM through the MagCoupler block is as follows.

In JMAG:

- In the JMAG circuit, connect a voltage function to the right of the FEM coil. Under **Electrical Potential** in the property window, choose **Constant Value**, and set **Constant Value [V]** to 0.
- Connect a current probe to the left of the FEM coil.
- Connect another voltage function to the left of the current probe (the circuit will look like what is shown above). In the property window, choose **Cooperates with an external circuit simulator**.
- Highlight the inductor structure window. Go to the menu **Conditions** >> **Create Conditions**. From the **Conditions List**, highlight **Coupled External Circuit Simulator**, and click **Modify**.
- On the **Coupled External Circuit Simulator** dialog window, there are two lists. The list on the right, called **JMAG**, contains all the functions that can be used to interface with PSIM. The list on the left, called **External Circuit Simulator**, contains the functions that are selected to interface with PSIM. In this case, there are two items in the JMAG list, one is the Voltage Function, and the other is the Current Probe.
- Highlight the Voltage Function, and click on the <- button to move the item from the list on the right to the list on the left. Repeat the same step to the Current Probe. Now both items should appear in the list on the left.
- Highlight the Voltage Function, and change the terminal name to VL. Also, change the Current Probe terminal name to iL. Close the dialog window.
- Go to the menu **File** >> **Export** and select **JCF...**. With the JCF file name defined as "inductor", the JCF file "inductor.jcf" and the link table file "inductor_csl.xml" will be generated.
- Copy the JCF file "inductor.jcf" and the link table file "inductor_csl.xml" to the folder containing the PSIM schematic file "inductor_jmag.sch". Rename the link table file to "inductor_jmag.xml". Note that the XML file does not have to be renamed, and both the JCF and XML files do not have to be moved to the folder of the schematic file. They are done here for the simplicity of file management

In PSIM:

- After the rest of the power circuit is created, go to **Elements** >> **Power** >> **MagCoupler Module**, and select **MagCoupler Block**. Place the block on the schematic.

- Double click on the MagCoupler block to bring out the property window. click on the browser button ... next to the **Link Table File** edit field to locate and select the file "inductor_jmag.xml". After the file is read, the property window will display the IN node VL and the OUT node iL.
- Connect the MagCouple block to the rest of the circuit in the schematic.

The setup is now complete and the simulation is ready to run.

4.8 MagCoupler-RT Module

The MagCoupler-RT Module provides interface between PSIM and JMAG-RT data files. JMAG-RT is another way of modeling electromagnetic devices. The JMAG-RT data files are obtained by running the JMAG simulation in advance, and the data files are stored in a lookup table form. During the PSIM simulation, JMAG is not needed, and PSIM interfaces directly with the JMAG-RT data.

The biggest advantage of JMAG-RT is that, since the JMAG-RT data files are obtained from the JMAG dynamic simulation, the accuracy of the JMAG-RT model is comparable to that of a JMAG dynamic model. However, since JMAG is not involved in the PSIM simulation, the PSIM simulation is significantly faster.

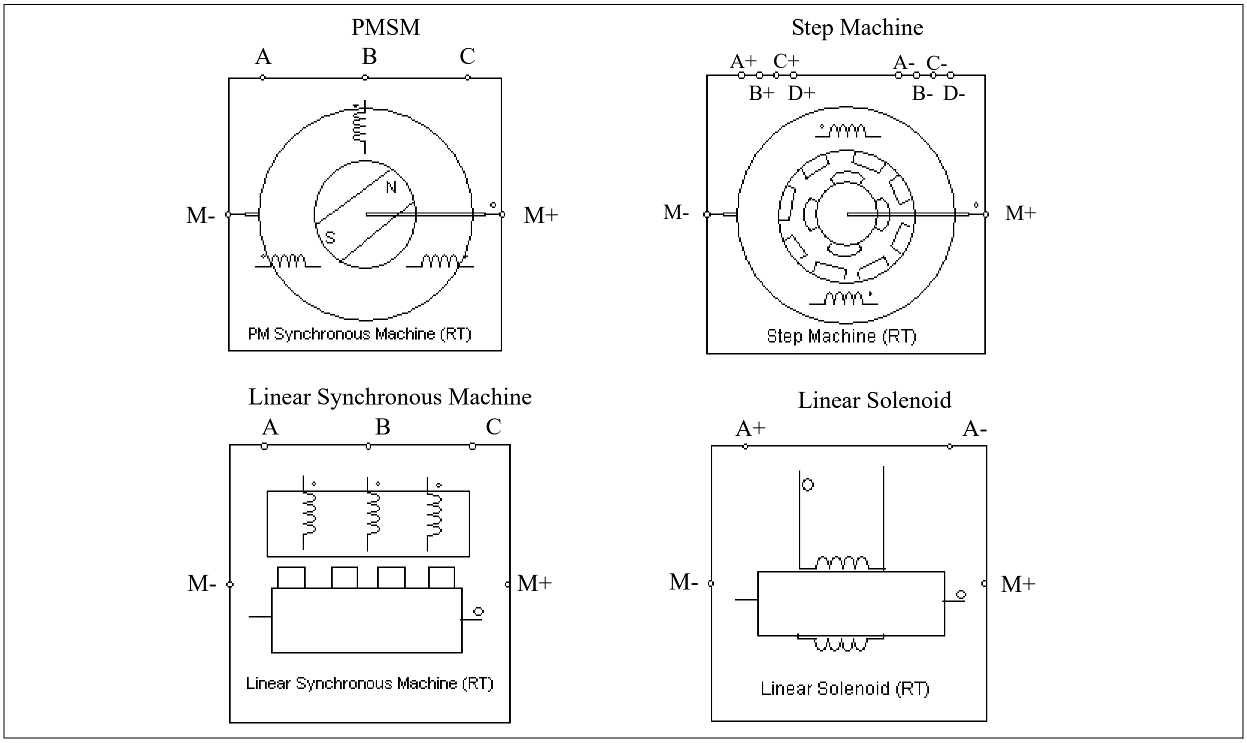
The MagCoupler-RT Module includes the blocks described in this section, and mechanical elements and speed/torque/position sensors as described in Section 4.9.

Four MagCoupler-RT blocks are provided:

- 3-phase permanent-magnet synchronous machine (PMSM)
- 2-phase step machine
- Linear synchronous machine
- Linear solenoid

Each block has the dedicated image, as shown below.

Images:



Attributes:

Parameter	Description
Netlist XML File	The XML file that defines the interface between PSIM and JMAG-RT. It has the .xml extension.
JMAG-RT Input File	The JMAG-RT data file. It has the .rtt extension. Note that the .xml file and the .rtt file must be in the same directory.
JMAG Case Text	Comments for the JMAG-RT circuit
Terminal Names	Terminal names of the block

In the MagCoupler-RT block images, the power circuit nodes (such as *A*, *B*, *C*, *A*+, *A*-, *B*+, *B*-, *C*+, *C*-, *D*+, and *D*- as shown above) are placed at the top of the block, arranged from the left to the right. The mechanical shaft nodes are placed on the left and right of the block, with the first shaft node (such as *M*+ as shown above) on the right, and the second shaft node (such as *M*-) on the left.

The electric nodes and rotor shaft nodes, as well as the rest of the interface between PSIM and the JMAG-RT data files (.rtt file), are defined in the *Netlist XML File*. This file is in XML format, and is generated by the JMAG-RT Manager. To specify this file, click on the browse button at the right of the edit field.

Note that for each type of MagCoupler-RT block, the "NetlistElement" must be the same as the predefined name. The predefined names are:

- "PM Synchronous Machine" for 3-phase PMSM
- "Step Machine" for 2-phase step machine
- "Linear Synchronous Machine" for linear synchronous machine
- "Linear Solenoid" for linear solenoid

For example, the first several lines of a netlist XML file for a PMSM may look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<CircuitElement file="imp.rtt" name="MagCoupler-RT">
<NetlistElement type="JMAGRT" element="PM Synchronous Machine" pole_number="4">
… … …
```

The *JMAG-RT Input File* is the JMAG-RT data file for the device modeled. The file has the .rtt extension and is defined in the netlist XML file. Note that the .rtt file and the .xml file must be in the same directory.

The *JMAG Case Text* is a text identifying the specific JMAG-RT study case. It can be any text.

The *Terminal Names* are the names of the interface nodes.

The parameters of a 3-phase PMSM is shown below:

Parameter	Description
shaft1_MomentofInertia	Moment of inertia of the machine, in kg*m^2
shaft1_ShaftTimeConstant	Shaft time constant of the machine, in sec.
RU_resistance	Resistance of Phase U, in Ohm
RV_resistance	Resistance of Phase V, in Ohm
RW_resistance	Resistance of Phase W, in Ohm
OffsetAngle	Initial rotor angle, in mechanical deg.
coef_inductance	Inductance coefficient (used in JMAG-RT)
coef_flux	Flux coefficient (used in JMAG-RT)
coef_torque	Torque coefficient (used in JMAG-RT)
coef_magnet	Magnet coefficient (used in JMAG-RT)

coef_material	Material coefficient (used in JMAG-RT)
turns_coil1	Coil 1 turns (used in JMAG-RT)
turns_coil2	Coil 2 turns (used in JMAG-RT)
Current Flag	Display flag for 3-phase FEM coil currents (1: display; 0: no display)
Back emf Flag	Display flag for 3-phase FEM coil back emf
Rotor Angle Flag	Display flag for the rotor angle in mechanical deg.
Speed Flag	Display flag for the shaft mechanical speed in rad./sec.
Torque Flag	Display flag for the developed torque in N*m
shaft1 Master Flag	Master/slave flag of the machine (1: master; 0: slave)

The parameters of a 2-phase step machine is shown below:

Parameter	Description
RA+	Resistance of Phase A+, in Ohm
RA-	Resistance of Phase A-, in Ohm
RB+	Resistance of Phase B+, in Ohm
RB-	Resistance of Phase B-, in Ohm
MomentofInertia	Moment of inertia of the machine, in kg*m^2
ShaftTimeConstant	Shaft time constant of the machine, in sec.
OffsetAngle	Initial rotor angle, in mechanical deg.
turns_coil2	Coil 2 turns (used in JMAG-RT)
coef_inertia1	Inertia 1 coefficient (used in JMAG-RT)
coef_inertia2	Inertia 2 coefficient (used in JMAG-RT)
Current Flag	Display flag for 3-phase FEM coil currents (1: display; 0: no display)
Back emf Flag	Display flag for 3-phase FEM coil back emf
Rotor Angle Flag	Display flag for the rotor angle in mechanical deg.
Speed Flag	Display flag for the shaft mechanical speed in rad./sec.
Torque Flag	Display flag for the developed torque in N*m
shaft1 Master Flag	Master/slave flag of the machine (1: master; 0: slave)

The parameters of a linear solenoid is shown below:

Parameter	Description
RA	Resistance of the solenoid, in Ohm
Mass	Mass of the solenoid, in kg
ShaftTimeConstant	Shaft time constant of the solenoid, in sec.
SpringConstant	Spring constant of the solenoid (used in JMAG-RT)
DispLimitMax	Maximum limit of the displacement of the solenoid, in m
DispLimitMin	Minimum limit of the displacement of the solenoid, in m
OffsetDisp	Initial displacement, in m

turns_coil2	Coil 2 turns (used in JMAG-RT)
coef_mass1	Mass 1 coefficient (used in JMAG-RT)
coef_mass2	Mass 2 coefficient (used in JMAG-RT)
Current Flag	Display flag for 3-phase FEM coil currents (1: display; 0: no display)
Back emf Flag	Display flag for 3-phase FEM coil back emf
Position Flag	Display flag for the solenoid position in m
Velocity Flag	Display flag for the solenoid velocity in m/sec.
Force Flag	Display flag for the developed force in N
mass1 Master Flag	Master/slave flag of the solenoid (1: master; 0: slave)

The parameters of a linear synchronous machine is shown below:

Parameter	Description
RU	Resistance of Phase U, in Ohm
RV	Resistance of Phase V, in Ohm
RW	Resistance of Phase W, in Ohm
Mass	Mass of the machine, in kg
ShaftTimeConstant	Shaft time constant of the solenoid, in sec.
OffsetDisp	Initial displacement, in m
coef_inductance	Inductance coefficient (used in JMAG-RT)
coef_flux	Flux coefficient (used in JMAG-RT)
coef_force	Torque coefficient (used in JMAG-RT)
coef_magnet	Magnet coefficient (used in JMAG-RT)
coef_material	Material coefficient (used in JMAG-RT)
turns_coil1	Coil 1 turns (used in JMAG-RT)
turns_coil2	Coil 2 turns (used in JMAG-RT)
coef_mass1	Mass 1 coefficient (used in JMAG-RT)
coef_mass2	Mass 2 coefficient (used in JMAG-RT)
Current Flag	Display flag for 3-phase FEM coil currents (1: display; 0: no display)
Back emf Flag	Display flag for 3-phase FEM coil back emf
Position Flag	Display flag for the solenoid position in m
Velocity Flag	Display flag for the solenoid velocity in m/sec.
Force Flag	Display flag for the developed force in N
mass1 Master Flag	Master/slave flag of the solenoid (1: master; 0: slave)

In all the parameter definitions above, except the flags (current flag, back emf flag, rotor angle/position flag, speed/velocity flag, torque/force flag, and master flag), all other parameters are read from the .rtt file defined in the XML file. These values can be changed, and for these parameters noted with "used in JMAG-RT", the changed values will be sent back to JMAG-RT for calculation.

Several MagCoupler-RT examples are provided under the "examples\MagCoupler-RT" folder.

4.9 Mechanical Elements and Sensors

This section describes elements that are shared by Motor Drive Module, MagCoupler Module, and MagCoupler-RT Module. The elements include mechanical loads, gear boxes, mechanical coupling blocks, mechanical-electrical interface blocks, and various speed/torque/position sensors.

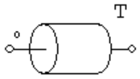
4.9.1 Mechanical Loads

Several mechanical load models are provided: constant-torque, constant-power, constant-speed, general-type, and externally controlled loads.

4.9.1.1 Constant-Torque Load

The image of a constant-torque load is:

Image:



Attributes:

Parameters	Description
Constant Torque	Torque constant T_{const}, in N*m
Moment of Inertia	Moment of inertia of the load, in kg*m^2

If the reference direction of a mechanical system enters the dotted terminal, the load is along the reference direction, and the loading torque to the master machine is T_{const}. Otherwise the loading torque will be -T_{const}. See Section 2.6.1 for more detailed explanation on the reference direction.

A constant-torque load is expressed as:

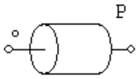
$$T_L = T_{\text{const}}$$

The torque does not depend on the speed direction.

4.9.1.2 Constant-Power Load

The image of a constant-power load is:

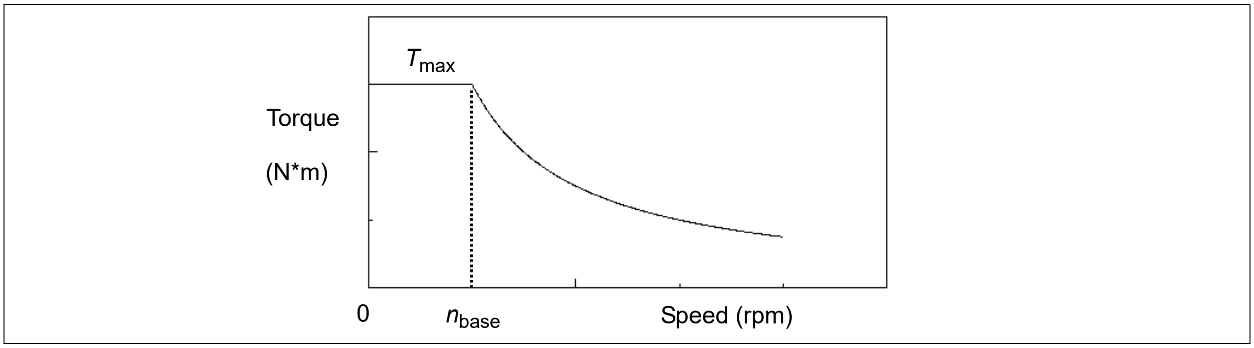
Image:



Attributes:

Parameters	Description
Maximum Torque	Maximum torque T_{max} of the load, in N*m
Base Speed	Base speed n_{base} of the load, in rpm
Moment of Inertia	Moment of inertia of the load, in kg*m^2

The torque-speed curve of a constant-power load is shown below:



When the mechanical speed is less than the base speed n_{base}, the load torque is:

$$T_L = T_{\max}$$

When the mechanical speed is above the base speed, the load torque is:

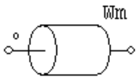
$$T_L = \frac{P}{|\omega_m|}$$

where $P = T_{\max} * \omega_{\text{base}}$ and $\omega_{\text{base}} = 2\pi * n_{\text{base}}/60$. The mechanical speed ω_m is in rad./sec.

4.9.1.3 Constant-Speed Load

The image of a constant-speed load is:

Image:



Attributes:

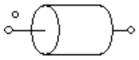
Parameters	Description
Constant Speed (rpm)	Speed constant, in rpm
Moment of Inertia	Moment of inertia of the load, in kg*m^2

A constant-speed mechanical load defines the speed of a mechanical system, and the speed will remain constant, as defined by the speed constant.

4.9.1.4 General-Type Load

The image of a general-type mechanical load is as follows.

Image:



Attributes:

Parameters	Description
Tc	Constant torque term
k_1 (coefficient)	Coefficient for the linear term

k_2 (coefficient)	Coefficient for the quadratic term
k_3 (coefficient)	Coefficient for the cubic term
Moment of Inertia	Moment of inertia of the load, in kg*m^2

A general-type load is expressed as:

$$T_L = sign(\omega_m) \cdot (T_c + k_1 \cdot |\omega_m| + k_2 \cdot \omega_m^2 + k_3 \cdot |\omega_m|^3)$$

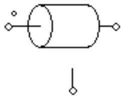
where ω_m is the mechanical speed in rad./sec.

Note that the torque of the general-type load is dependent on the speed direction.

4.9.1.5 Externally-Controlled Load

An externally-controlled mechanical load is used to define a load of an arbitrary load profile.

Image:



Attributes:

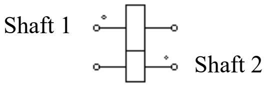
Parameters	Description
Speed Flag	Flag for speed dependency Flag = 0: The load is frictional and is always against the rotational direction; Flag = 1: The load is independent of the rotational direction.
Moment of Inertia	Moment of inertia of the load, in kg*m^2

The value of the mechanical load is defined by the voltage value at the control node (1V corresponds to 1 N*m). This node is a control circuit node.

4.9.2 Gear Box

The image is a gear box is shown below.

Image:



Attribute:

Parameter	Description
Gear Ratio	The gear ratio *a*
Shaft 1 Master/Slave Flag	Master/slave flag for Shaft 1
Shaft 2 Master/Slave Flag	Master/slave flag for Shaft 2

The shaft with the bigger dot is Shaft 1.

If the numbers of teeth of the first gear and the second gear are n_1 and n_2, respectively, the gear ratio a is defined as: $a = n_1 / n_2$. Let the radius, torque, and speed of these two gears be: r_1, r_2, T_1, T_2, ω_1, and ω_2, we have: T_1 /

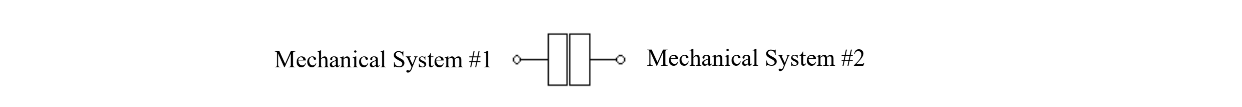
$T_2 = r_1 / r_2 = \omega_2 / \omega_1 = a$.

The two shafts of the gear box can be in either master mode or slave mode. For more information on the definition and use of the master/slave flag, refer to Section 4.6.1.

4.9.3 Mechanical Coupling Block

The mechanical coupler block is used to couple two mechanical systems.

Image:

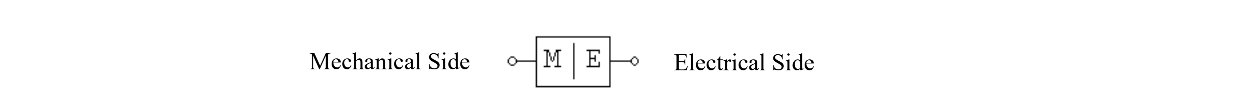


This block is used in situations where both mechanical systems have a device in the master mode, and they must be connected together to form one system. Based on the connection convention in PSIM, a mechanical system can have only one master device. In this case, the mechanical coupling block can be inserted in between, and the mechanical system on each side of the coupling block can have its own device in the master mode.

4.9.4 Mechanical-Electrical Interface Block

This block allows users to access the internal equivalent circuit of the mechanical system of a machine.

Image:



Attribute:

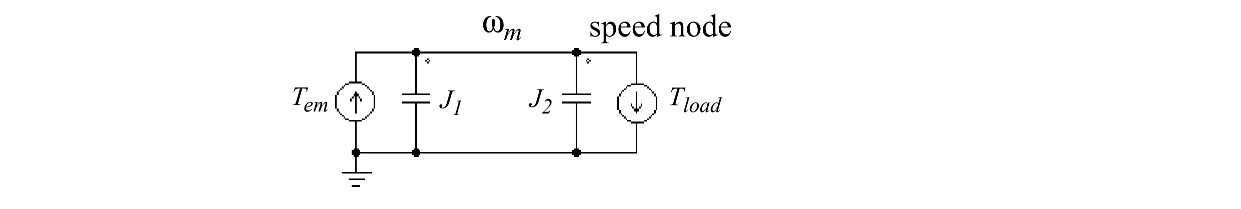
Parameter	Description
Master/Slave Flag	Flag for the master/slave mode (1: master, 0: slave)

Similar to electric machines, the mechanical-electrical interface block can be used to define the reference direction of a mechanical system through the master/slave flag. When the interface block is set to the master mode, the reference direction is along the mechanical shaft, away from the mechanical node, and towards the rest of the mechanical elements.

Let's assume that a drive system consists of a motor (with a developed torque of T_{em} and a moment of inertia of J_1) and a mechanical load (with a load torque of T_{load} and a moment of inertia of J_2). The equation that describes the mechanical system is:

$$(J_1 + J_2) \cdot \frac{d\omega_m}{dt} = T_{em} - T_{load}$$

where ω_m is the shaft mechanical speed. In PSIM, this equation is modelled by an equivalent circuit as shown below.



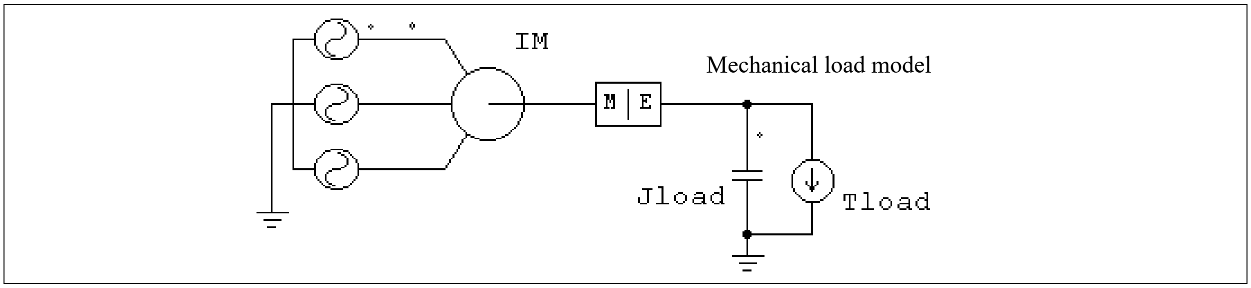
In this circuit, the two current sources have the values of T_{em} and T_{load}, and the capacitors have the values of J_1 and J_2. The node-to-ground voltage (speed node voltage) represents the mechanical speed ω_m. This is analogous to C*dV/dt = i for a capacitor where $C = J_1$+J_2, $V = \omega_m$, and $i = T_{em}$-T_{load}.

In PSIM, mechanical equivalent circuits for motors and mechanical loads all use the capacitor-based circuit model. The mechanical-electrical interface block provides the access to the internal mechanical equivalent circuit. If the mechanical side of an interface block (with the letters “M”) is connected to a mechanical shaft, the electrical side (with the letters “E”) will be the speed node of the mechanical equivalent circuit. One can thus connect any electrical circuits to this node.

With this element, users can connect built-in motors or mechanical loads with user-defined load or motor models.

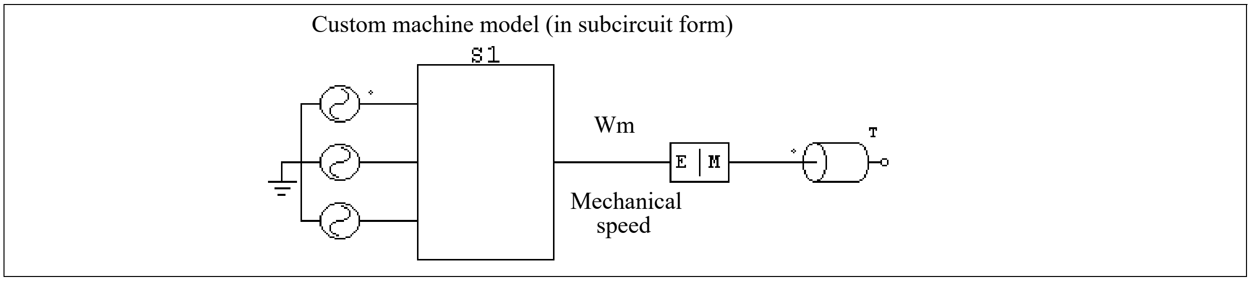
Example: An induction machine with a custom mechanical load model

The figure below shows an induction machine connected to a user defined mechanical load model through the mechanical-electrical interface block. As explained above, the voltage at the electrical side represents the shaft mechanical speed. A current source flowing out of this node represents a mechanical load, and a capacitor connected to this node represents the load moment of inertia.



Example: A custom machine model with a constant-torque load

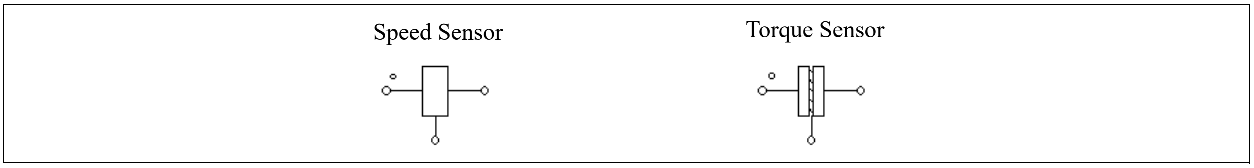
Similarly, one can build a custom machine model and connect it to the mechanical load in PSIM. The figure below shows such a circuit. The custom machine model must use the capacitor analogy to model the mechanical equation. The node representing the mechanical speed is then made available and is connected to the electrical side of the mechanical-electrical interface block.



4.9.5 Speed/Torque Sensors

A speed sensor or torque sensor is used to measure the mechanical speed or torque.

Images:



Attribute:

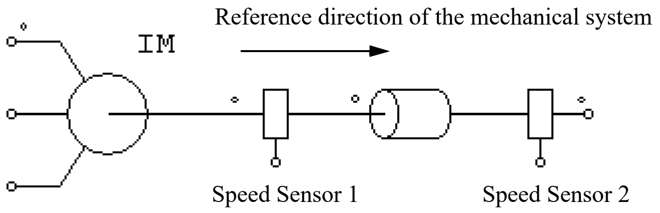
Parameter	Description
Gain	Gain of the sensor

The output of the speed sensor is in rpm.

The output of the speed/torque sensor depends on how the sensor is connected in a mechanical system.

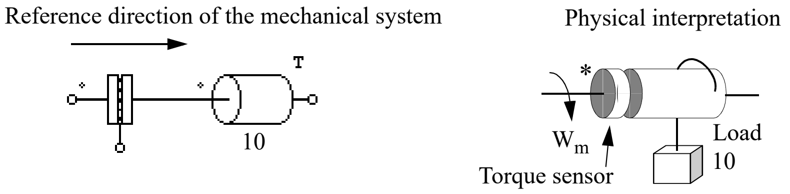
For the speed sensor, if the sensor is along the reference direction of the mechanical system (refer to Section 4.6.1 for more details on the definition and use of the reference direction), a positive mechanical speed would give a positive sensor output. Otherwise, if the sensor is opposite to the reference direction of the mechanical system, a positive mechanical speed would give a negative sensor output.

For example, in the mechanical system below, Speed Sensor 1 is along the reference direction, and Speed Sensor 2 is opposite to the reference direction of the mechanical system. If the actual mechanical speed is positive, the output of Speed Sensor 1 will be positive, and the output of Speed Sensor 2 will be negative.



The torque sensor measures the torque difference between the dotted side of the sensor and the undotted side. To understand the physical meaning of the torque sensor measurement, we use the diagram below as an illustration.

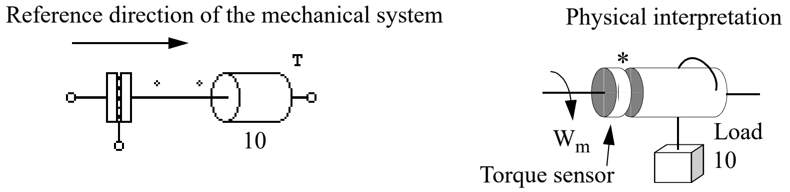
The figure on the left shows a torque sensor connected with a 10-N*m mechanical load, and the reference direction of the mechanical system is from left to right. Based on the reference direction, if we use the right-hand method, by pointing the thumb in the reference direction and rotating the right hand, the direction of the fingers will show the direction of the positive speed and torque. The physical interpretation of the system is shown on the right.



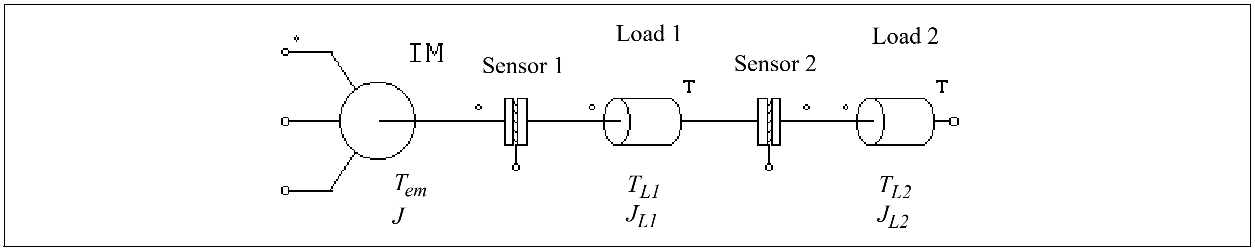
In this case, the direction of the positive speed and torque is in the clockwise direction. The dotted side of the sensor is on the left, and the load is in such a way that it tries to slow down the shaft (the load torque is in the counter-clockwise direction).

The physical meaning of the torque sensor is that, if the dotted side of the sensor is fixed, the sensor will measure the torque tension on the undotted side of the sensor, and a positive sensor output would mean that the torque is opposite to the direction of the speed reference. Therefore, for the example above, the positive speed reference is in the clockwise direction, and the load torque is in the counter-clockwise direction. This will give a torque sensor reading of 10 N*m.

Similarly, if the undotted side of the sensor is fixed, the sensor will measure the torque tension on the dotted side of the sensor, in the positive direction of the speed reference. For example, in the system below, the torque sensor is flipped with the dotted side on the right. If the undotted side is fixed, the load torque is applied to the dotted side of the sensor, in the opposite direction of the speed reference. The torque sensor output will be -10 N*m instead.



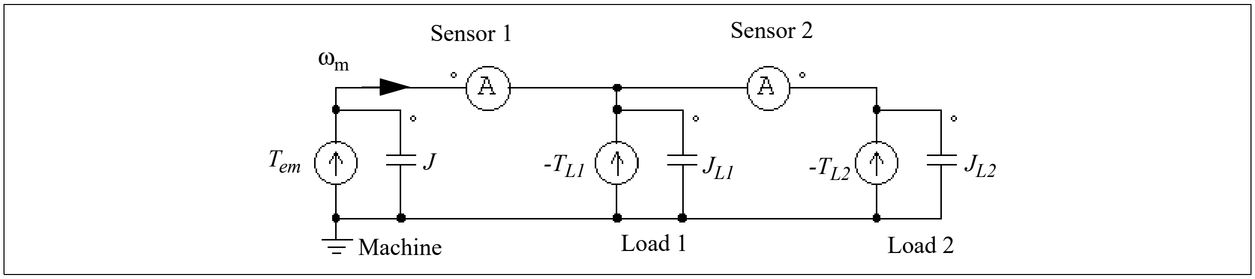
To understand how the torque sensor is modeled in the equivalent circuit of the mechanical system, we use the following system as an example.



The system consists of one machine, 2 torque sensors, and 2 mechanical loads. The torques and moment of inertia for the machine and the loads are as labelled in the diagram. The reference direction of this mechanical system is from left to right. The equation for this system can be written as:

$$(J + J_{L1} + J_{L2}) \cdot \frac{d\omega_m}{dt} = T_{em} - T_{L1} - T_{L2}$$

The equivalent electrical circuit of the equation is shown below:



The node voltage in the circuit represents the mechanical speed ω_m. The current probe on the left represents the reading of the Torque Sensor 1. Similarly, the current probe on the right represents the reading of the Torque Sensor 2. Note that the second current probe is from right to left since Sensor 2 is opposite to the reference direction of the mechanical system.

The equivalent circuit also illustrates how mechanical power is transferred. The multiplication of the current to the voltage, which is the same as the torque times the mechanical speed, represents the mechanical power. If the power is positive, it is transferred in the direction of the speed ω_m.

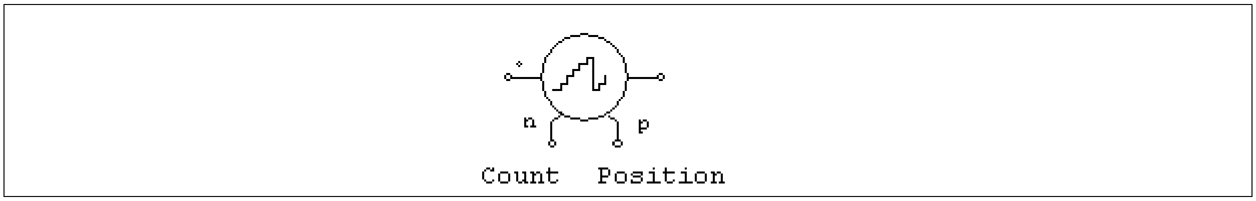
4.9.6 Position Sensors

Four types of position sensors are provided: absolute encoder, incremental encoder, resolver, and hall effect position sensor. They are connected to the mechanical shaft, similar to the speed sensor and torque sensor, and the output signals are control signals.

4.9.6.1 Absolute Encoder

An absolute encoder is a position sensor that provides the shaft position within a 360^o range (mechanical degree).

Image:



Attribute:

Parameter	Description
Initial Position (deg.)	Initial shaft position, in deg.
No. of Bits of Resolution	Number of Bits of resolution N

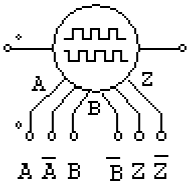
The encoder output resolution is determined by the number of bits N. The encoder has two outputs: one is the number of counts (the range is from 0 to 2^N-1), and the other is the position, in mechanical deg. (the range is from 0 to 360^o).

An example of a PMSM drive system using the absolute encoder is given in the sample file "Absolute Encoder PMSM Drive.sch".

4.9.6.2 Incremental Encoder

An incremental encoder is a position sensor that produces quadrature outputs which indicate the speed, angle, and direction of the shaft.

Image:



Attribute:

Parameter	Description
Initial Position (deg.)	Initial shaft position, in deg.
No. of Lines	Number of lines that are in the code pattern of the code disk of the encoder

The two quadrature outputs are A and $\overline{A}$ ($\overline{A}$ is the inverse of A), and B and $\overline{B}$. They are offset by 90^o. In addition, the encoder provides separate index signal output Z and $\overline{Z}$ that provide one count per revolution.

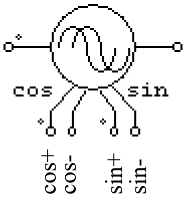
An example of an induction motor drive system using the incremental encoder is given in the sample file "Incremental Encoder INDM Drive.sch".

4.9.6.3 Resolver

A resolver is essentially a rotary transformer with one rotor winding and two stator windings. These two stator windings, referring to as the COS winding and SIN winding, are located 90^o apart.

As the shaft rotates, the output voltages of the COS and SIN windings vary as the cosine and sine functions of the shaft angle.

Image:



Attribute:

Parameter	Description
Initial Position (deg.)	Initial shaft position, in deg.
No. of Poles	Number of poles of the resolver

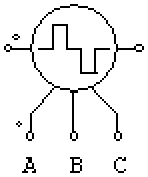
The resolver has four outputs: cos+, cos- (the inverse of cos+), sin+, and sin- (the inverse of sin+). The peak amplitude of all the outputs is 1.

An example of a PMSM drive system using the resolver is given in the sample file "Resolver PMSM Drive.sch".

4.9.6.4 Hall Effect Sensor

A hall effect sensor is a type of position sensors that provides three pulses depending on the shaft position. The sensor consists of a set of semiconductor switches and trigger magnets. The switches open or close when the magnetic field is higher or lower than a certain threshold value.

Image:



Attribute:

Parameter	Description
Initial Position (deg.)	Initial shaft position, in deg.
No. of Poles	Number of poles of the sensor

The hall effect sensor provides three logic signal outputs A, B, and C, which are spaced 120 electrical deg. apart.

The hall effect sensor is the same as the built-in hall effect sensor in the brushless dc machine.

Examples of BDCM motor drive systems using the hall effect sensor are given in the "examples" sub-folder of the PSIM directory.

4.10 Thermal Module

The Thermal Module is an add-on module to the PSIM program. It provides a quick way of estimating the losses of semiconductor devices (diodes, IGBT, and MOSFET), as well as the core and winding losses of inductors.

The core of the Thermal Module is the device database. A device database editor is provided to allow users to add new devices to the database and to manage the database easily. The devices in the database can then be used in the simulation for the loss calculation.

A database device simulation contains two parts of calculation:

- The behavior model of the device is used in the simulation. The behavior model of a switch takes into account the static characteristics (such as conduction voltage drop, on-state resistance, etc.), but not the dynamic characteristics (such as turn-on and turn-off transients). The behavior model of an inductor is an ideal inductor.
- Based on the voltage, current, and temperature values from the simulation, for switches, PSIM accesses the device database and calculates the conduction losses or switching losses. The static characteristics of the device are updated for the next simulation. For inductors, PSIM accesses the device database and calculates the core and winding losses.

  In switch loss calculation, voltage and current values of a device right before and after the switching instant are used. For a device operating in hard switching, if a snubber circuit is present, it will change the voltage and current waveforms and will result in incorrect loss calculation. It is recommended that snubber circuits be temporarily disabled when using Thermal Module devices for loss calculation.

Loss calculation of inductors takes into account the material characteristics of the device (such as core material, shape, winding type, size, and air gap, etc.).

Note that the loss calculation is only an approximation, and the accuracy of the results depends on the accuracy of the device data as well as the proper scaling of the results from the device test condition to the actual circuit operating conditions. Users should always verify the results with the measurement from experimental setup.

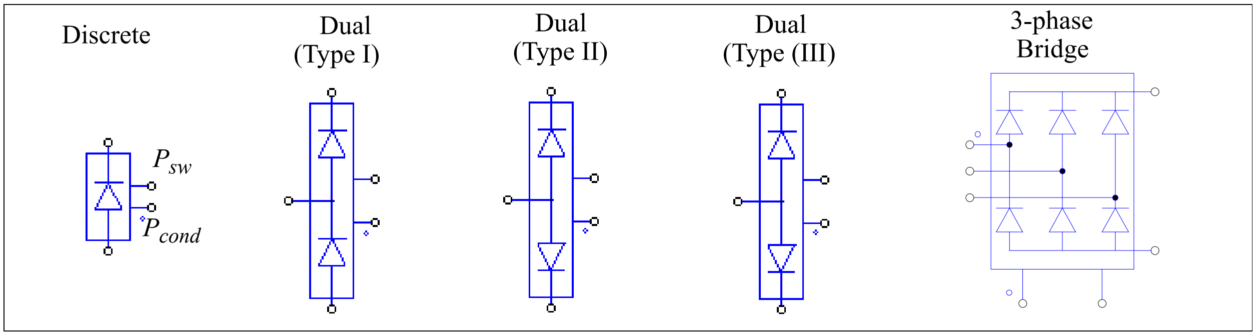
4.10.1 Diode Thermal Model

4.10.1.1 Diode Device in Database

The following information is defined for a diode device in the database:

General Information:

- Manufacturer:Device manufacture
- Part Number:Manufacturer's part number
- Package:It can be discrete, dual, or 3-phase bridge package, as shown in the figure below:



In the images, beside the diode anode and cathode terminals, there are two extra nodes. The node with a dot is for the diode conduction losses P_{cond}, and the node without a dot is for the diode switching losses P_{sw}.

The style of the package (for example, TO-220, TO-247, etc.) can be specified in the *Style* field.

Absolute Maximum Ratings:

$V_{rrm,max}$ (V):	Peak reverse blocking voltage
$I_{F,max}$ (A):	Maximum dc current
$T_{j,max}$ (°C):	Maximum junction temperature

Electrical Characteristics:

V_d vs. I_F:	Forward conduction voltage drop V_d vs. forward current I_F
t_{rr} vs. I_F:	Reverse recovery time t_{rr} vs. current I_F
I_{rr} vs. I_F:	Peak reverse recovery current I_{rr} vs. current I_F
Q_{rr} vs. I_F:	Reverse recovery charge Q_{rr} vs. current I_F
E_{rr} vs. I_F:	Reverse recovery energy losses E_{rr} vs. current I_F

Thermal Characteristics:

$R_{th(j\text{-}c)}$:	Junction-to-case thermal resistance, in °C/W
$R_{th(c\text{-}s)}$:	Case-to-sink thermal resistance, in °C/W

Dimensions and Weight:

Length (mm):	Length of the device, in mm
Height (mm):	Height of the device, in mm
Weight (g):	Weight of the device, in g

Note that the condition "Reverse blocking voltage VR (V)" in characteristics E_{rr} vs. I_F is used to scale the loss calculation, and must be entered correctly.

Also, parameters under *Absolute Maximum Ratings*, *Thermal Characteristics*, and *Dimensions and Weight* are not used in the loss calculation, and are optional.

In addition, the diode forward conduction voltage drop V_d changes depending on the forward current I_F. The new voltage drop is used in the subsequent simulation.

4.10.1.2 Diode Loss Calculation

A diode device in the database can be selected and used in the simulation for loss calculation. A diode in the Thermal Module library has the following parameters:

Attributes:

Parameters	Description
Device	The specific device selected from the device database
Frequency	Frequency, in Hz, under which the losses are calculated
P_{cond} Calibration Factor	The calibration factor K_{cond} of the conduction losses P_{cond}
P_{sw} Calibration Factor	The calibration factor K_{sw} of the switching losses P_{sw}
Number of Parallel Devices	Number of identical diode devices in parallel

The parameter *Frequency* refers to the frequency under which the losses are calculated. For example, if the device operates at the switching frequency of 10 kHz, and the parameter *Frequency* is also set to 10 kHz, the losses will be the values for one switching period. However, if the parameter *Frequency* is set to 60 Hz, then the losses will be the value for a period of 60 Hz.

The parameter P_{cond} *Calibration Factor* is the correction factor for the conduction losses. For the example, if the calculated conduction losses before the correction is P_{cond_cal}, then

$$P_{cond} = K_{cond} * P_{cond\_cal}$$

Similarly, the parameter P_{sw} *Calibration Factor* is the correction factor for the switching losses. For the example, if the calculated switching losses before the correction is P_{sw_cal}, then

$$P_{sw} = K_{sw} * P_{sw\_cal}$$

When several identical diodes are in parallel, one should have just one device in the schematic, and set the correct number of devices in the parameter input. This is because when several identical devices are in parallel in the schematic, the device currents may not be exactly equal due to small differences in the simulation. When the number of parallel devices is greater than one, the total current through the group of the parallel devices will be equally divided among the devices. The total losses are then obtained by multiplying the losses in each device by the number of parallel devices. The currents flowing out of Nodes P_{cond} and P_{sw} are the total losses of all the parallel devices combined.

The voltage at the conduction losses node P_{cond} or the switching losses node P_{sw} represents the calculated junction temperature T_j of the diode, and this junction temperature is used when the database curves are used for loss calculation. If the calculated T_j is between the junction temperatures of two curves, interpolation will be used. If the calculated T_j is lower than the lowest T_j or higher than the highest T_j in the curves, the curve corresponding to the lowest or highest T_j will be used. If there is only one curve, that curve is used regardless of the calculated T_j.

Conduction Losses:

The diode conduction losses are calculated as:

Conduction Losses = $V_d * I_F$

where V_d is the diode voltage drop, and I_F is the diode forward current. When the diode is conducting periodically with an on duty cycle of D, the conduction losses are calculated as:

Conduction Losses = $V_d * I_F * D$

Switching Losses:

In calculating the switching losses, the diode turn-on losses are neglected and are not considered.

The diode turn-off losses due to the reverse recovery is calculated as:

$$P_{sw\_off} = E_{rr} * f * V_R / V_{R\_datasheet}$$

or

$$P_{sw\_off} = 1/4 * Q_{rr} * V_R * f$$

where E_{rr} is the reverse recovery energy losses, Q_{rr} is the reverse recovery charge, f is the frequency as defined in the input parameter *Frequency*, V_R is the actual reverse blocking voltage, and $V_{R\_datasheet}$ is the reverse blocking voltage in the E_{rr} characteristics of the datasheet, defined as "Reverse blocking voltage VR (V)" in the test conditions.

If only one point is given for Q_{rr}, the loss calculation will be scaled based on the actual current I_F as follows:

$$P_{sw\_off} = 1/4 * Q_{rr} * V_R * f * I_F / I_{F\_datasheet}$$

where $I_{F\_datasheet}$ is the test condition current under which Q_{rr} and t_{rr} are obtained.

The reverse recovery charge Q_{rr} is defined as:

$$Q_{rr} = 1/2 * t_{rr} * I_{rr}$$

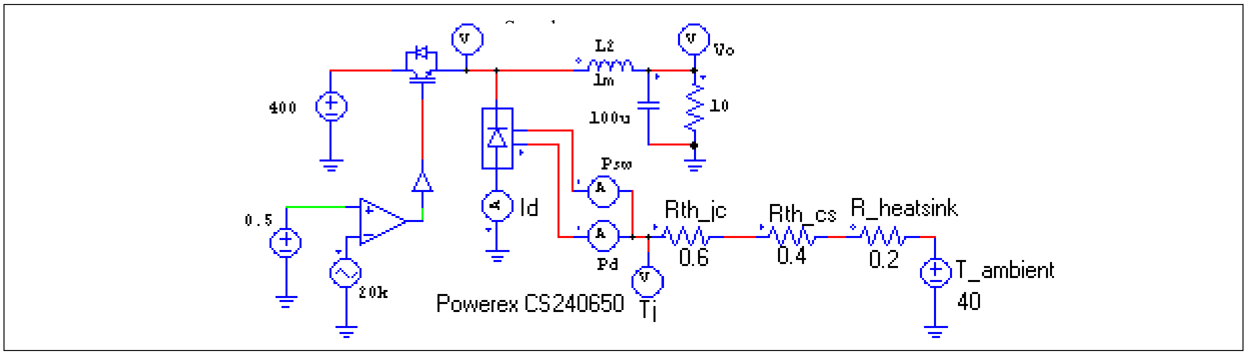
Whenever E_{rr} is given in the device database, the losses will be calculated based on E_{rr}. If E_{rr} is not given, the losses will be calculated based on Q_{rr}. If Q_{rr} is not given, the losses will be calculated based on t_{rr} and I_{rr}. If both are not given, the losses will be treated as 0.

The losses P_{cond} and P_{sw}, in watts, are represented in the form of currents which flow out of these nodes. Therefore, to measure and display the losses, an ammeter should be connected between the P_{cond} or P_{sw} node and the ground. When they are not used, these two nodes cannot be floating, and must be connected to ground.

Example: Diode Loss Calculation

The circuit below shows a sample circuit that uses the Powerex's discrete diode CS240650 (600V, 50A). The conduction losses and the switching losses are measured through two ammeters.

Once the information of the losses is available, by building the thermal equivalent circuit, the device junction temperature can be calculated. This junction temperature is in turn fed back and used in the loss calculation. The circuit shows a thermal circuit without considering the thermal transient.



4.10.2 IGBT Thermal Model

4.10.2.1 IGBT Device in Database

An IGBT device has three types of packages: discrete, dual, or 6-pack.

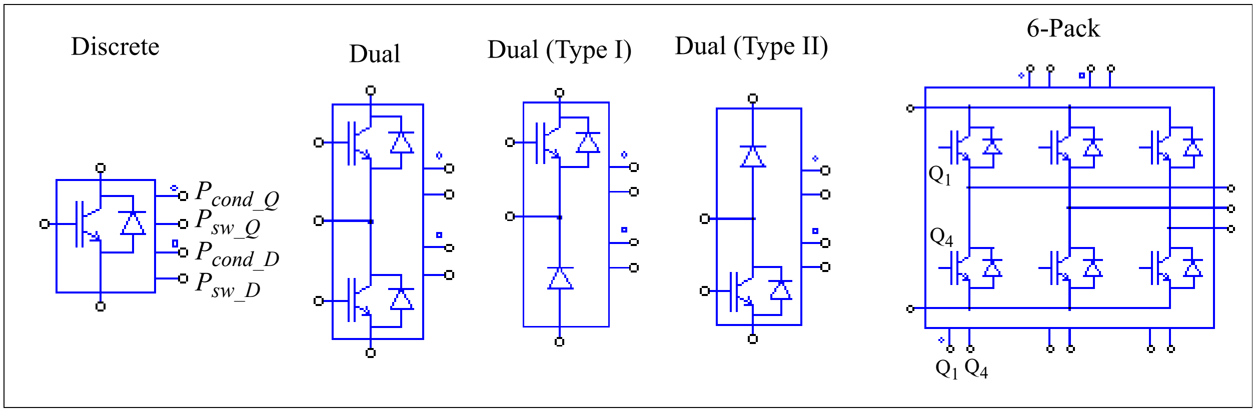
For the dual package, both the top and the bottom switches can be IGBT's (full-bridge configuration), or one of the switches is IGBT and the other is a free-wheeling diode (half-bridge configuration). For the half-bridge dual IGBT device, since the free-wheeling diode parameters can be different from these of the anti-parallel diode, this type of device is referred to as the IGBT-Diode device, and is treated as a different type in the simulation. But for the convenience of discussion, both devices are referred to as the IGBT devices here.

The following information is defined for an IGBT device in the database:

General Information:

Manufacturer:	Device manufacture
Part Number:	Manufacturer's part number
Package:	It can be discrete, dual, or 6-pack, as shown in the figure below

Images:



In the images, beside the IGBT and diode terminal nodes, there are four extra nodes from the top to the bottom (or from the left to the right on the top for the 6-pack package). They are the nodes for transistor conductor losses P_{cond_Q} (the node with a circle), for transistor switching losses P_{sw_Q}, for diode conductor losses P_{cond_D} (the node with a square), and for diode switching losses P_{sw_D}, respectively.

The style of the package (for example, TO-247, TO-268, etc.) can be specified in the *Style* field.

Absolute Maximum Ratings:

$V_{ce,max}$ (V): Maximum collector-emitter voltage
$I_{c,max}$ (A): Maximum collector current
$T_{j,max}$ (^oC): Maximum junction temperature

Electrical Characteristics - Transistor:

$V_{ce(sat)}$ vs. I_c: Collector-emitter saturation voltage $V_{ce(sat)}$ vs. collector current I_c
E_{on} vs. I_c: Turn-on energy losses E_{on} vs. collector current I_c
E_{off} vs. I_c: Turn-off energy losses E_{off} vs. collector current I_c

Electrical Characteristics - Diode (or Anti-Parallel Diode):

V_d vs. I_F: Forward conduction voltage drop V_d vs. forward current I_F
t_{rr} vs. I_F: Reverse recovery time t_{rr} vs. current I_F
I_{rr} vs. I_F: Peak reverse recovery current I_{rr} vs. current I_F
Q_{rr} vs. I_F: Reverse recovery charge Q_{rr} vs. current I_F
E_{rr} vs. I_F: Reverse recovery energy losses E_{rr} vs. current I_F

Electrical Characteristics - Free-Wheeling Diode (for IGBT-Diode device only):

V_d vs. I_F: Forward conduction voltage drop vs. forward current I_F
t_{rr} vs. I_F: Reverse recovery time t_{rr} vs. current I_F
I_{rr} vs. I_F: Peak reverse recovery current I_{rr} vs. current I_F
Q_{rr} vs. I_F: Reverse recovery charge Q_{rr} vs. current I_F
E_{rr} vs. I_F: Reverse recovery charge E_{rr} vs. current I_F

Thermal Characteristics:

$R_{th(j-c)}$ (transistor): Transistor junction-to-case thermal resistance, in $^oC/W$
$R_{th(j-c)}$ (diode): Diode junction-to-case thermal resistance, in $^oC/W$
$R_{th(c-s)}$: Case-to-sink thermal resistance, in $^oC/W$

Dimensions and Weight:

Length (mm): Length of the device, in mm
Width (mm): Width of the device, in mm
Height (mm): Height of the device, in mm
Weight (g): Weight of the device, in g

Note that the condition "DC bus voltage (V)" in the characteristics E_{on} vs. I_c and E_{off} vs. I_c for transistors, and the condition "Reverse blocking voltage VR (V)" in the characteristics E_{rr} vs. I_F for anti-parallel and free-wheeling diodes, are used to scale the loss calculation, and must be entered correctly.

Also, parameters under *Absolute Maximum Ratings*, *Thermal Characteristics*, and *Dimensions and Weight* are not used in the loss calculation, and are optional.

In addition, the collector-emitter saturation voltage $V_{ce(sat)}$ of the transistor and the forward conduction voltage drop V_d of the diode change depending on the currents. The new values are used in the subsequent simulation.

4.10.2.2 IGBT Loss Calculation

An IGBT device in the database can be selected and used in the simulation for loss calculation. An IGBT device in the Thermal Module library has the following parameters:

Attributes:

Parameters	Description
Device	The specific device selected from the device database
Frequency	Frequency, in Hz, under which the losses are calculated

$\mathrm{P_{cond\_Q}}$ Calibration Factor	Calibration factor K_{cond_Q} of the transistor conduction losses P_{cond_Q}
$\mathrm{P_{sw\_Q}}$ Calibration Factor	Calibration factor K_{sw_Q} of the transistor switching losses P_{sw_Q}
$\mathrm{P_{cond\_D}}$ Calibration Factor	Calibration factor K_{cond_D} of the diode conduction losses P_{cond_D}
$\mathrm{P_{sw\_D}}$ Calibration Factor	Calibration factor K_{sw_D} of the diode switching losses P_{sw_D}
Number of Parallel Devices	Number of identical IGBT devices in parallel

The parameter *Frequency* refers to the frequency under which the losses are calculated. For example, if the device operates at the switching frequency of 10 kHz, and the parameter *Frequency* is also set to 10 kHz, the losses will be the values for one switching period. However, if the parameter *Frequency* is set to 60 Hz, then the losses will be the value for a period of 60 Hz.

The parameter P_{cond_Q} *Calibration Factor* is the correction factor for the transistor conduction losses. For the example, if the calculated conduction losses before the correction is $P_{cond\_Q\_cal}$, then

$$P_{cond\_Q} = K_{cond\_Q} * P_{cond\_Q\_cal}$$

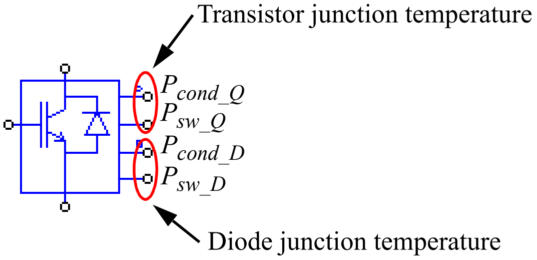
Similarly, the parameter P_{sw_Q} *Calibration Factor* is the correction factor for the transistor switching losses. For the example, if the calculated switching losses before the correction is $P_{sw\_Q\_cal}$, then

$$P_{sw\_Q} = K_{sw\_Q} * P_{sw\_Q\_cal}$$

Parameters P_{cond_D} *Calibration Factor* and P_{sw_D} *Calibration Factor* work in the same way, except that they are for the diode losses.

When several identical IGBT devices are in parallel, one should have just one device in the schematic, and set the correct number of devices in the parameter input. This is because when several identical devices are in parallel in the schematic, the device currents may not be exactly equal due to small differences in the simulation. When the number of parallel devices is greater than one, the total current through the group of the parallel devices will be equally divided among the devices. The total losses are then obtained by multiplying the losses in each device by the number of parallel devices. The currents flowing out of Nodes P_{cond_D}, P_{sw_D}, P_{cond_Q}, and P_{sw_Q} are the total losses of all the parallel devices combined.

The voltage at the conduction losses node P_{cond_Q} or the switching losses node P_{sw_Q} represents the calculated junction temperature T_{j_Q} of the transistor, and the voltage at the conduction losses node P_{cond_D} or the switching losses node P_{sw_D} represents the calculated junction temperature T_{j_D} of the anti-parallel diode, as shown below.



The calculated junction temperatures T_{j_Q} and T_{j_D} are used when the database curves are used for loss calculation. If the calculated junction temperature is between the junction temperatures of two curves, interpolation will be used. If the calculated junction temperature is lower than the lowest junction temperature or higher than the highest junction temperature in the curves, the curve corresponding to the lowest or highest junction temperature will be used. If there is only one curve, that curve is used regardless of the calculated junction temperature.

Conduction Losses:

The transistor conduction losses is calculated as:

$$P_{cond\_Q} = V_{ce(sat)} * I_c$$

where $V_{ce(sat)}$ is the transistor collector-emitter saturation voltage, and I_c is the collector current. When the transistor is conducting periodically with an on duty cycle of D, the conduction losses are calculated as:

$$P_{cond\_Q} = V_{ce(sat)} * I_c * D$$

Switching Losses:

The transistor turn-on losses is calculated as:

$$P_{sw\_Q\_on} = E_{on} * f * V_{cc} / V_{cc\_datasheet}$$

where E_{on} is the transistor turn-on energy losses, f is the frequency as defined in the input parameter *Frequency,* V_{cc} is the actual dc bus voltage, and $V_{cc\_datasheet}$ is the dc bus voltage in the E_{on} and E_{off} characteristics of the datasheet, defined as "DC bus voltage (V)" in the test conditions.

The transistor turn-off losses is calculated as:

$$P_{sw\_Q\_off} = E_{off} * f * V_{cc} / V_{cc\_datasheet}$$

where E_{off} is the transistor turn-off energy losses.

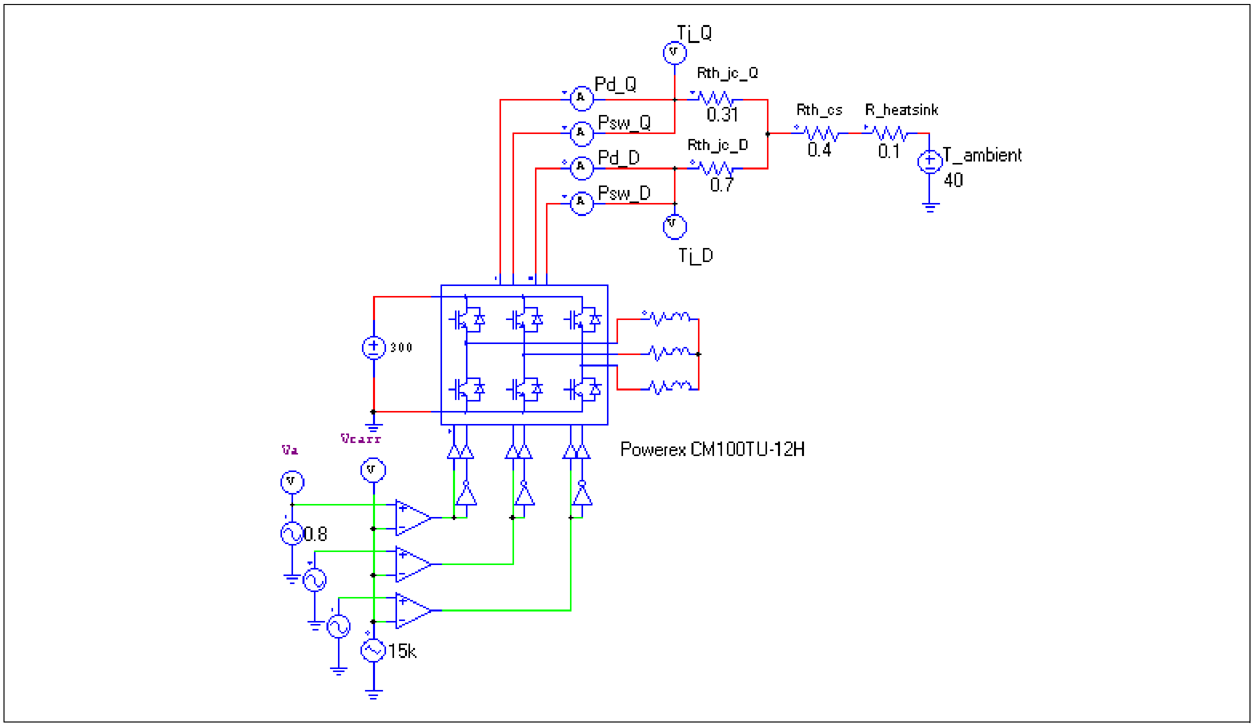
The loss calculation for the anti-parallel diode or free-wheeling diode is the same as described in the section for the diode device.

The losses P_{cond_Q}, P_{sw_Q}, P_{cond_D}, and P_{sw_D}, in watts, are represented in the form of currents which flow out of these nodes. Therefore, to measure and display the losses, an ammeter should be connected between the nodes and the ground. When they are not used, these nodes cannot be floating and must be connected to ground.

Example: IGBT Loss Calculation

The circuit below shows a sample circuit that uses Powerex's 6-pack IGBT module CM100TU-12H (600V, 100A). The conduction losses and the switching losses of the transistors and the diodes are added separately, and a thermal equivalent circuit is provided to calculate the temperature raise.

With the Thermal Module, users can quickly check the thermal performance of a device under different operating conditions, and compare the devices of different manufactures.



4.10.3 IGBT-RB Thermal Model

4.10.3.1 IGBT-RB Device in Database

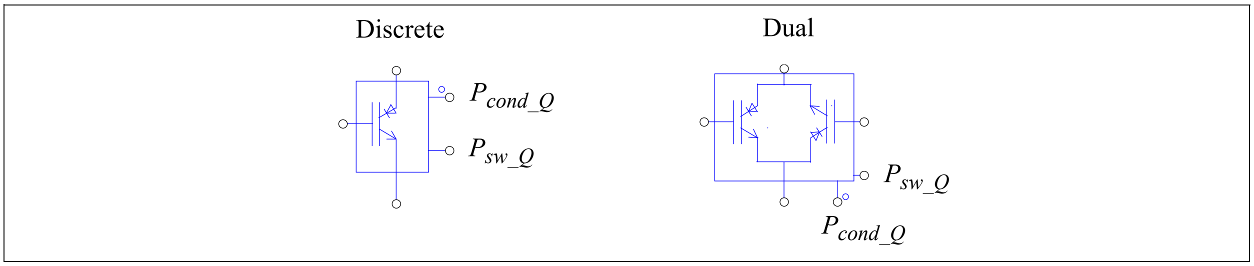
An IGBT-RB device has two types of packages: discrete or dual.

The following information is defined for an IGBT device in the database:

General Information:

Manufacturer: Device manufacture
Part Number: Manufacturer’s part number
Package: It can be discrete or dual, as shown in the figure below

Images:



In the images, beside the IGBT terminal nodes, there are two extra nodes. They are the nodes for transistor conductor losses P_{cond_Q} (the node with a circle), and transistor switching losses P_{sw_Q}.

The style of the package can be specified in the *Style* field.

Absolute Maximum Ratings:

$V_{ce,max}$ (V): Maximum collector-emitter voltage
$I_{c,max}$ (A): Maximum collector current
$T_{j,max}$ (^oC): Maximum junction temperature

Electrical Characteristics - Transistor:

$V_{ce(sat)}$ vs. I_c: Collector-emitter saturation voltage $V_{ce(sat)}$ vs. collector current I_c
E_{on} vs. I_c: Turn-on energy losses E_{on} vs. collector current I_c
E_{off} vs. I_c: Turn-off energy losses E_{off} vs. collector current I_c
E_{rr} vs. I_c: Reverse recovery energy losses E_{rr} vs. collector current I_c

Thermal Characteristics:

$R_{th(j-c)}$ (transistor): Transistor junction-to-case thermal resistance, in $^oC/W$
$R_{th(c-s)}$: Case-to-sink thermal resistance, in $^oC/W$

Dimensions and Weight:

Length (mm): Length of the device, in mm
Width (mm): Width of the device, in mm
Height (mm): Height of the device, in mm
Weight (g): Weight of the device, in g

Note that the condition "DC bus voltage (V)" in the characteristics E_{on} vs. I_c and E_{off} vs. I_c for transistors, and the condition "Reverse blocking voltage VR (V)" in the characteristics E_{rr} vs. I_c are used to scale the loss calculation, and must be entered correctly.

Also, parameters under *Absolute Maximum Ratings*, *Thermal Characteristics*, and *Dimensions and Weight* are not used in the loss calculation, and are optional.

In addition, the collector-emitter saturation voltage $V_{ce(sat)}$ of the transistor depending on the currents. The new values are used in the subsequent simulation.

4.10.3.2 IGBT-RB Loss Calculation

An IGBT-RB device in the database can be selected and used in the simulation for loss calculation. An IGBT-RB device in the Thermal Module library has the following parameters:

Attributes:

Parameters	Description
Device	The specific device selected from the device database
Frequency	Frequency, in Hz, under which the losses are calculated
$\mathrm{P}_{\mathrm{cond\_Q}}$ Calibration Factor	Calibration factor K_{cond_Q} of the transistor conduction losses P_{cond_Q}
$\mathrm{P}_{\mathrm{sw\_Q}}$ Calibration Factor	Calibration factor K_{sw_Q} of the transistor switching losses P_{sw_Q}
Number of Parallel Devices	Number of identical IGBT-RB devices in parallel

The parameter *Frequency* refers to the frequency under which the losses are calculated. For example, if the device operates at the switching frequency of 10 kHz, and the parameter *Frequency* is also set to 10 kHz, the losses will be the values for one switching period. However, if the parameter *Frequency* is set to 60 Hz, then the losses will be the value for a period of 60 Hz.

The parameter P_{cond_Q} *Calibration Factor* is the correction factor for the transistor conduction losses. For the example, if the calculated conduction losses before the correction is $P_{cond\_Q\_cal}$, then

$$P_{cond\_Q} = K_{cond\_Q} * P_{cond\_Q\_cal}$$

Similarly, the parameter P_{sw_Q} *Calibration Factor* is the correction factor for the transistor switching losses. For the example, if the calculated switching losses before the correction is $P_{sw\_Q\_cal}$, then

$$P_{sw\_Q} = K_{sw\_Q} * P_{sw\_Q\_cal}$$

When several identical IGBT-RB devices are in parallel, one should have just one device in the schematic, and set the correct number of devices in the parameter input. This is because when several identical devices are in parallel in the schematic, the device currents may not be exactly equal due to small differences in the simulation. When the number of parallel devices is greater than one, the total current through the group of the parallel devices will be equally divided among the devices. The total losses are then obtained by multiplying the losses in each device by the number of parallel devices. The currents flowing out of Nodes P_{cond_Q} and P_{sw_Q} are the total losses of all the parallel devices combined.

The voltage at the conduction losses node P_{cond_Q} or the switching losses node P_{sw_Q} represents the calculated junction temperature T_{j_Q} of the transistor.

The calculated junction temperature T_{j_Q} is used when the database curves are used for loss calculation. If the calculated junction temperature is between the junction temperatures of two curves, interpolation will be used. If the calculated junction temperature is lower than the lowest junction temperature or higher than the highest junction temperature in the curves, the curve corresponding to the lowest or highest junction temperature will be used. If there is only one curve, that curve is used regardless of the calculated junction temperature.

Conduction Losses:

The transistor conduction losses is calculated as:

$$P_{cond\_Q} = V_{ce(sat)} * I_c$$

where $V_{ce(sat)}$ is the transistor collector-emitter saturation voltage, and I_c is the collector current. When the transistor is conducting periodically with an on duty cycle of D, the conduction losses are calculated as:

$$P_{cond\_Q} = V_{ce(sat)} * I_c * D$$

Switching Losses:

The transistor turn-on losses is calculated as:

$$P_{sw\_Q\_on} = E_{on} * f * V_{cc} / V_{cc\_datasheet}$$

where E_{on} is the device turn-on energy losses, f is the frequency as defined in the input parameter *Frequency,* V_{cc} is the actual dc bus voltage, and $V_{cc\_datasheet}$ is the dc bus voltage in the E_{on} and E_{off} characteristics of the datasheet, defined as "DC bus voltage (V)" in the test conditions.

The transistor turn-off losses is calculated as:

$$P_{sw\_Q\_off} = (E_{off} + E_{rr}) * f * V_{cc} / V_{cc\_datasheet}$$

where E_{off} is the transistor turn-off energy losses and E_{rr} is the reverse recovery energy losses.

The losses P_{cond_Q} and P_{sw_Q}, in watts, are represented in the form of currents which flow out of these nodes. Therefore, to measure and display the losses, an ammeter should be connected between the nodes and the ground. When they are not used, these nodes cannot be floating and must be connected to ground.

4.10.4 MOSFET Thermal Model

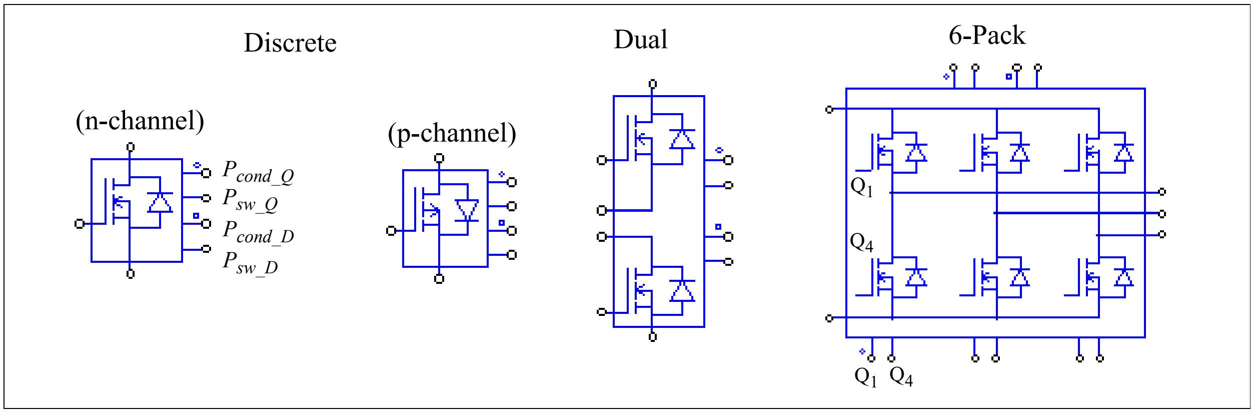
4.10.4.1 MOSFET Device in Database

The following information is defined for a MOSFET device in the database:

General Information:

Manufacturer:	Device manufacture
Part Number:	Manufacturer’s part number
Package:	Discrete, dual, or 6-pack, as shown in the figure below:

Images:



In the images, beside the MOSFET and diode terminal nodes, there are four extra nodes from the top to the bottom (or from the left to the right on the top for the 6-pack package). They are the node for transistor conductor losses P_{cond_Q} (the node with a circle), for transistor switching losses P_{sw_Q}, for diode conductor losses P_{cond_D} (the node with a square), and for diode switching losses P_{sw_D}, respectively.

The style of the package (for example, TO-220, TO-262, TO-247, etc.) can be specified in the *Style* field.

Absolute Maximum Ratings:

$V_{DS,max}$ (V):	Maximum drain-to-source voltage
$I_{D,max}$ (A):	Maximum continuous drain current
$T_{j,max}$ (^oC):	Maximum junction temperature

Electrical Characteristics - Transistor:

$R_{DS(on)}$ (ohm): Drain-to-source on-resistance $R_{DS(on)\_b}$ at the test conditions (test conditions: junction temperature T_{j_b} in ^{o}C (usually 25 ^{o}C), gate-to-source voltage V_{GS} in V, and drain current I_D in A)

Temperature Coefficient: Temperature coefficient K_T of the on-resistance, in 1/^{o}C

$V_{GS(th)}$ (V): Gate threshold voltage $V_{GS(th)}$ (test condition: drain current I_D in A)

g_{fs} (S): Forward transconductance g_{fs}. In the linear region of the device, we have

$g_{fs} = \frac{I_D}{V_{GS} - V_{GS(th)}}$ (test conditions: drain-to-source voltage V_{DS} in V, and drain current I_D in A)

t_r (ns) and t_f (ns): Rise time t_r and fall time t_f (test conditions: drain-to-source voltage V_{DS} in V; drain current I_D in A; and gate resistance R_g in ohm)

Q_g, Q_{gs}, and Q_{gd}: Total gate charge Q_g, gate-to-source charge Q_{gs}, and gate-to-drain ("Miller") charge Q_{gd}, respectively, all in nC (test conditions: drain-to-source voltage V_{DS} in V; gate-to-source voltage V_{DS} in V; and drain current I_D in A)

C_{iss}, C_{oss}, and C_{rss}: Input capacitance C_{iss}, output capacitance C_{oss}, and reverse transfer capacitance C_{rss}, respectively, all in pF (test conditions: drain-to-source voltage V_{DS} in V; gate-to-source voltage V_{DS} in V; and test frequency in MHz)

Electrical Characteristics - Diode:

V_d vs. I_F: Forward conduction voltage drop V_d vs. forward current I_F

t_{rr} and Q_{rr}: Reverse recovery time t_{rr} in ns, and reverse recovery charge Q_{rr} in uC (test conditions: forward current I_F in A; rate of change of the current di/dt in A/us, and junction temperature T_j in ^{o}C)

Thermal Characteristics:

$R_{th(j-c)}$: Junction-to-case thermal resistance, in ^{o}C/W

$R_{th(c-s)}$: Case-to-sink thermal resistance, in ^{o}C/W

Dimensions and Weight:

Length (mm):	Length of the device, in mm
Width (mm):	Width of the device, in mm
Height (mm):	Height of the device, in mm
Weight (g):	Weight of the device, in g

Note that the parameters under *Absolute Maximum Ratings*, *Thermal Characteristics*, and *Dimensions and Weight*, and the rise time t_r and fall time t_f parameters, are not used in the loss calculation, and are optional.

The MOSFET on-resistance is a function of the transistor junction temperature. The voltage at the node P_{cond_Q} or P_{sw_Q} of the MOSFET image represents the calculated transistor junction temperature T_j in ^{o}C. For example, a voltage value of 25V would mean a junction temperature of 25^{o}C. The on-resistance of the MOSFET is expressed as:

$$R_{(DS)on} = R_{(DS)on\_b} \cdot (1 + K_T \cdot (T_j - T_{j\_b}))$$

The temperature coefficient is calculated as:

$$K_T = \frac{\frac{R_{(DS)on}}{R_{(DS)on\_b}} - 1}{T_j - T_{j\_b}}$$

The base values are normally obtained at the test conditions of 25^{o}C. Using the normalized value of $R_{DS(on)}$ based on the base value, we have:

$$K_T = \frac{R_{(DS)on\_\text{normalized}} - 1}{T_j - 25}$$

The on-resistance $R_{DS(on)}$ is calculated at every time step and is used in the simulation.

Also, the forward conduction voltage drop V_d of the diode changes depending on the current. The new conduction voltage value is used in the simulation.

4.10.4.2 MOSFET Loss Calculation

A MOSFET device in the database can be selected and used in the simulation for loss calculation. A MOSFET in the Thermal Module library has the following parameters:

Attributes:

Parameters	Description
Device	The specific device selected from the device database
Frequency	Frequency, in Hz, under which the losses are calculated
V_{GG+} (upper level)	Upper level of the gate source voltage, in V
V_{GG-} (lower level)	Lower level of the gate source voltage, in V
R_{g_on} (turn-on)	Gate resistance during turn-on
R_{g_off} (turn-off)	Gate resistance during turn-off. In most cases, the turn-on gate resistance R_{g_on} and the turn-off gate resistance R_{g_off} are identical.
$R_{DS(on)}$ Calibration Factor	The calibration factor of the on-state resistance $R_{DS(on)}$
g_{fs} Calibration Factor	The calibration factor of the forward transconductance g_{fs}
P_{cond_Q} Calibration Factor	The calibration factor K_{cond_Q} of the transistor conduction losses P_{cond_Q}
P_{sw_Q} Calibration Factor	The calibration factor K_{sw_Q} of the transistor switching losses P_{sw_Q}
P_{cond_D} Calibration Factor	The calibration factor K_{cond_D} of the diode conduction losses P_{cond_D}
P_{sw_D} Calibration Factor	The calibration factor K_{sw_D} of the diode switching losses P_{sw_D}
Number of Parallel Devices	Number of identical devices in parallel

The parameter *Frequency* refers to the frequency under which the losses are calculated. For example, if the device operates at the switching frequency of 10 kHz, and the parameter *Frequency* is also set to 10 kHz, the losses will be the values for one switching period. However, if the parameter *Frequency* is set to 60 Hz, then the losses will be the value for a period of 60 Hz.

The parameter P_{cond_Q} *Calibration Factor* is the correction factor for the transistor conduction losses. For the example, if the calculated conduction losses before the correction is $P_{cond\_Q\_cal}$, then

$$P_{cond\_Q} = K_{cond\_Q} * P_{cond\_Q\_cal}$$

Similarly, the parameter P_{sw_Q} *Calibration Factor* is the correction factor for the transistor switching losses. For the example, if the calculated switching losses before the correction is $P_{sw\_Q\_cal}$, then

$$P_{sw\_Q} = K_{sw\_Q} * P_{sw\_Q\_cal}$$

Parameters P_{cond_D} *Calibration Factor* and P_{sw_D} *Calibration Factor* work in the same way. except that they are for the diode losses.

When several identical MOSFET devices are in parallel, one should have just one device in the schematic, and set the correct number of devices in the parameter input. This is because when several identical devices are in parallel in the schematic, the device currents may not be exactly equal due to small differences in the simulation. When the number of parallel devices is greater than one, the total current through the group of the parallel devices will be equally divided among the devices. The total losses are then obtained by multiplying the losses in each device by the number of parallel devices. The currents flowing out of Nodes P_{cond_D}, P_{sw_D}, P_{cond_Q}, and P_{sw_Q} are the total losses of all the parallel devices combined.

The voltage at the diode loss node P_{cond_D} or P_{sw_D} of the MOSFET image represents the calculated junction temperature T_j of the anti-parallel diode, and this junction temperature is used when the database curves are used for determining the forward conduction diode voltage drop. If the calculated T_j is between the junction temperatures of two curves, interpolation will be used. If the calculated T_j is lower than the lowest T_j or higher than the highest T_j in the curves, the curve corresponding to the lowest or highest T_j will be used. If there is only one curve, that curve is used regardless of the calculated T_j.

Conduction Losses:

The transistor conduction losses is calculated as:

$$P_{cond\_Q} = I_D^2 * R_{DS(on)}$$

where I_D is the drain current, and $R_{DS(on)}$ is the static on-resistance. When the transistor is conducting periodically with an on duty cycle of D, the conduction losses are calculated as:

$$P_{cond\_Q} = I_D^2 * R_{DS(on)} * D$$

Switching Losses:

The transistor turn-on losses is calculated as:

$$P_{sw\_Q\_on} = E_{on} * f$$

where E_{on} is the transistor turn-on energy losses, and f is the frequency as defined in the input parameter *Frequency.*

The transistor turn-off losses is calculated as:

$$P_{sw\_Q\_off} = E_{off} * f$$

where E_{off} is the transistor turn-off energy losses.

The energy losses E_{on} and E_{off} are calculated using the rise times and the fall times of the voltage and current waveforms based on the information of the MOSFET gate current, input/output/reverse transfer capacitances, and gate charges.

Please note that the gate charge losses are not included in the switching losses above. The gate charge losses are usually quite small compared to the turn-on/turn-off switching losses, and can be neglected in the full load conditions. However, they can become substantial in the light load conditions. To calculate the gate charge losses P_{loss_Qg}:

$$P_{loss\_Qg} = Q_g * V_{GG} * f_{sw}$$

where Q_g is the total gate charge, V_{GG} is the gate source voltage, and f_{sw} is the switching frequency.

Also, the MOSFET device must be properly selected to ensure that it is sufficiently rated to conduct the current. If the MOSFET current rating is too low, PSIM will not be able to calculate the switching rise/fall times and will give an error message. In simulation, the maximum drain-to-source current that a MOSFET device is allowed to conduct is:

$$I_{o,max} = g_{fs} * (V_{GG} - V_{GS(th)})$$

If the current exceeds $I_{o,max}$, one should either increase the gate source voltage level or select another MOSFET device with a larger forward transconductance value.

The loss calculation for the anti-parallel diode or free-wheeling diode is the same as described in the diode device section.

The losses P_{cond_Q}, P_{sw_Q}, P_{cond_D}, and P_{sw_D}, in watts, are represented in the form of currents which flow out of these nodes. Therefore, to measure and display the losses, an ammeter should be connected between the nodes and the ground. When they are not used, these nodes cannot be floating and must be connected to ground.

4.10.5 Inductor Thermal Model

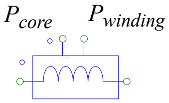
4.10.5.1 Inductor in Database

The following information is defined for an inductor in the database:

General Information:

Manufacturer:	Inductor manufacture
Part Number:	Inductor manufacturer’s part number
Package:	Only the basic package is available.

Image:



Electrical Characteristics:

L (uH):	Inductance, in uH

Maximum Ratings:

Imax,rms (A):	Maximum rms current rating, in A.

Core Information:

Core Type and Size:	Specify core shape and size
Core Material:	Specify core material and magnetic characteristics

Winding Information:

Winding Type and Size:	Specify conductor shape, size, and insulation type
Winding Distribution:	Specify winding number of turns, layers, distance between layers, distance between wires, and distance between core and winding
Distribution of Parallel Wires:	Specify parallel wire distribution in radial and height directions

Gap Information:

Gap Data:	Specify the number of gaps and the length of each gap.

A new inductor can be added to the database by selecting **Device** >> **New Inductor**. Databases of core materials, cores, and winding can be added and edited by selecting **Device** >> **Edit Inductor Core Database**, **Edit Winding Database**, and **Edit Core Material Database**.

4.10.5.2 Inductor Loss Calculation

A database inductor can be selected and used in the simulation for loss calculation. An inductor in the Thermal Module library has the following parameters:

Attributes:

Parameters	Description
Device	The specific inductor selected from the device database.
Frequency	Fundamental frequency of the inductor, in Hz.
Temperature Flag	The flag that defines how core temperature is defined. It can be either *Calculated* or *Fixed*. If it is *Calculated*, the core temperature will be calculated based on circuit operating conditions and this temperature will be used for loss calculation. If it is *Fixed*, the core temperature will be specified by users. This will give users a clear idea the losses at a specific temperature.
Ambient Temperature	The ambient temperature of the inductor, in ^{o}C.

Convection	Type of cooling. It can be *Forced* for forced convection or *Natural* for natural convection.
Air Speed	The air speed, in m/sec., if forced convection is selected.
Estimated Core Temp.	Estimated core temperature, in ^oC, if the Temperature Flag is set to Fixed.
Estimated Winding Temp.	Estimated winding temperature, in ^oC, if the Temperature Flag is set to Fixed.
Loss Calculation Flag	If the flag is 0, the loss calculation will be performed from the beginning of the simulation. If the flag is 1, the loss calculation will be performed only in the last fundamental cycle of the simulation. Since loss calculation will slow down simulation, calculating the losses only in the last cycle will speed up the simulation.
Current Flag	Display flag of the inductor current.

The losses P_{core} and $P_{winding}$, in watts, are represented in the form of currents which flow out of the nodes. To measure and display the losses, an ammeter should be connected between the nodes and the ground. When they are not used, these nodes cannot be floating and must be connected to ground.

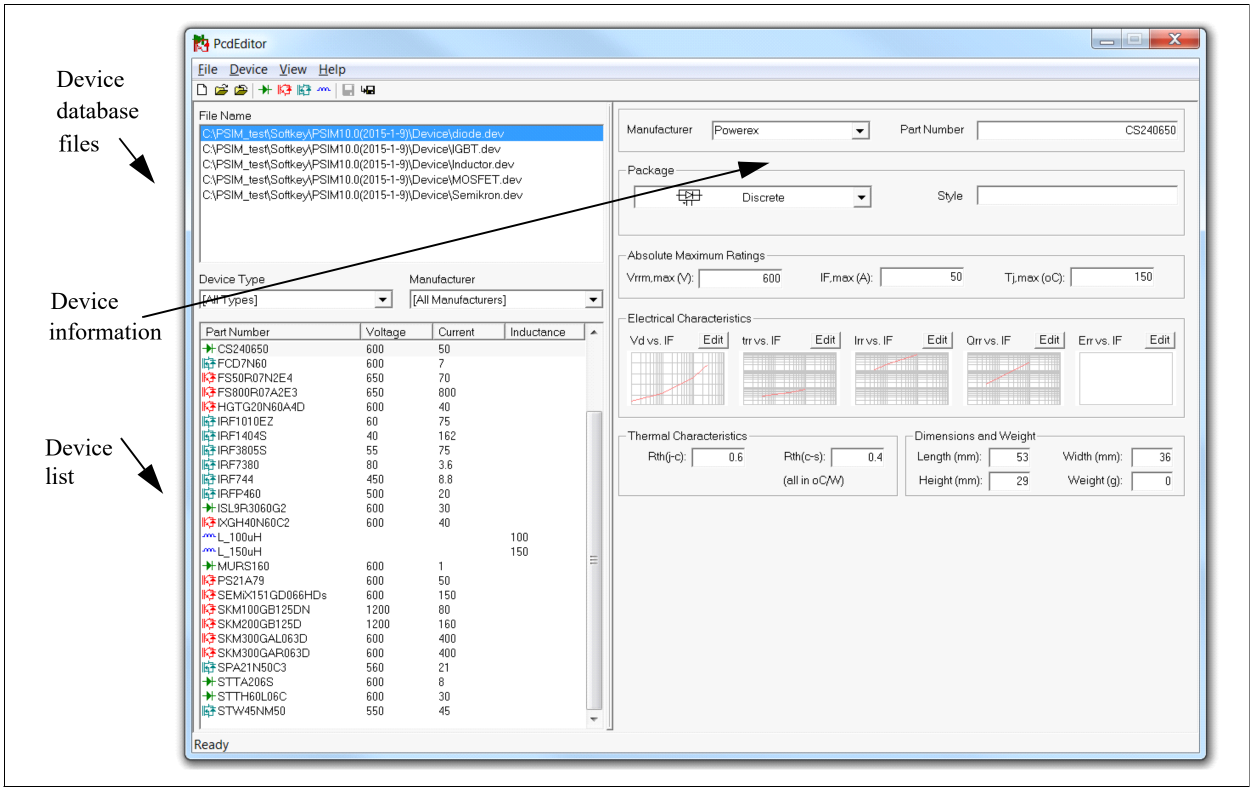
In the parameters, **Frequency** refers to the main frequency under which the inductor is excited. For example, if an inductor conducts a 60-Hz current in an ac circuit which also has 20kHz switching harmonic components, the frequency should be set to 60 Hz. If an inductor conducts dc current in a dc circuit which has 20kHz switching harmonic components, the frequency should be set to 20kHz.

For more information on how to use the Thermal Module inductor, please refer to the tutorial "Tutorial - Inductor loss calculation in Thermal Module.pdf".

4.10.6 Device Database Editor

The device database editor, PcdEditor.exe, provides an easy and convenient way of adding, editing, and managing devices.

Pull down PSIM’s **Utilities** menu, an image of the database editor is shown below.



File Name

C:\PSIM_test\Softkey\PSIM10.0(2015-1-9)\Device\diode.dev
C:\PSIM_test\Softkey\PSIM10.0(2015-1-9)\Device\IGBT.dev
C:\PSIM_test\Softkey\PSIM10.0(2015-1-9)\Device\Inductor.dev
C:\PSIM_test\Softkey\PSIM10.0(2015-1-9)\Device\MOSFET.dev
C:\PSIM_test\Softkey\PSIM10.0(2015-1-9)\Device\Semikron.dev

Device Type: [All Types]
Manufacturer: [All Manufacturers]

Part Number	Voltage	Current	Inductance
CS240650	600	50	
FCD7N60	600	7	
FS50R07N2E4	650	70	
FS800R07A2E3	650	800	
HGTG20N60A4D	600	40	
IRF1010EZ	60	75	
IRF1404S	40	162	
IRF3805S	55	75	
IRF7380	80	3.6	
IRF744	450	8.8	
IRFP460	500	20	
ISL9R3060G2	600	30	
IXGH40N60C2	600	40	
L_100uH			100
L_150uH			150
MURS160	600	1	
PS21A79	600	50	
SEMiX151GD066HDs	600	150	
SKM100GB125DN	1200	80	
SKM200GB125D	1200	160	
SKM300GAL063D	600	400	
SKM300GAR063D	600	400	
SPA21N50C3	560	21	
STTA206S	600	8	
STTH60L06C	600	30	
STW45NM50	550	45	

Manufacturer: Powerex
Part Number: CS240650

Package: Discrete
Style:

Absolute Maximum Ratings
Vrrm,max (V): 600
IF,max (A): 50
Tj,max (oC): 150

Electrical Characteristics
Vd vs. IF
trr vs. IF
Irr vs. IF
Qrr vs. IF
Err vs. IF

Thermal Characteristics
Rth(j-c): 0.6
Rth(c-s): 0.4
(all in oC/W)

Dimensions and Weight
Length (mm): 53
Width (mm): 36
Height (mm): 29
Weight (g): 0

On the left are the device database files that are loaded into the database editor, and the list of the devices. The devices can be displayed based on either **Device Type** or **Manufacturer**. Also, the device list can be sorted by **Part Number**, **Voltage** rating, or **Current** rating, by clicking on the title bars of the list.

To create a new device file, choose **File** >> **New Device File**.

To load a device files into the editor, choose **File** >> **Open Device File**.

To unload a device file from the editor, choose **File** >> **Close Device File**.

On the right is the information of each device. In general, the following information is required to define a switching device for thermal simulation:

- Manufacturer and Part Number
- Package type and style
- Absolute maximum ratings
- Electrical characteristics
- Thermal characteristics
- Dimension and weight

The following information are required to define a database inductor for thermal simulation.

- Manufacturer and Part Number
- Package type and style
- Maximum ratings
- Electrical characteristics
- Core, winding, and gap information

Five types of elements can be added to a database device file: inductor and four types of switching devices: diode, IGBT, dual IGBT-diode modules, and MOSFET.

To create a new device, go to the **Device** menu, and choose either **New Diode**, **New IGBT**, **New IGBT-Diode**, **New MOSFET,** or **New Inductor**.

To make a copy of an existing device in the same database file, highlight the device in the list, and choose **Device** >> **Save Device As**.

To make a copy of an existing device and save it in a different database file, first highlight the device in the list, then highlight the file name in the File Name list, and choose **Device** >> **Save Device As**.

4.10.7 Adding a Switching Device to Database

The step-by-step procedure is illustrated with an example: add Powerex discrete diode CS240650 (600V, 50A) into a new device database file "diode_new.dev".

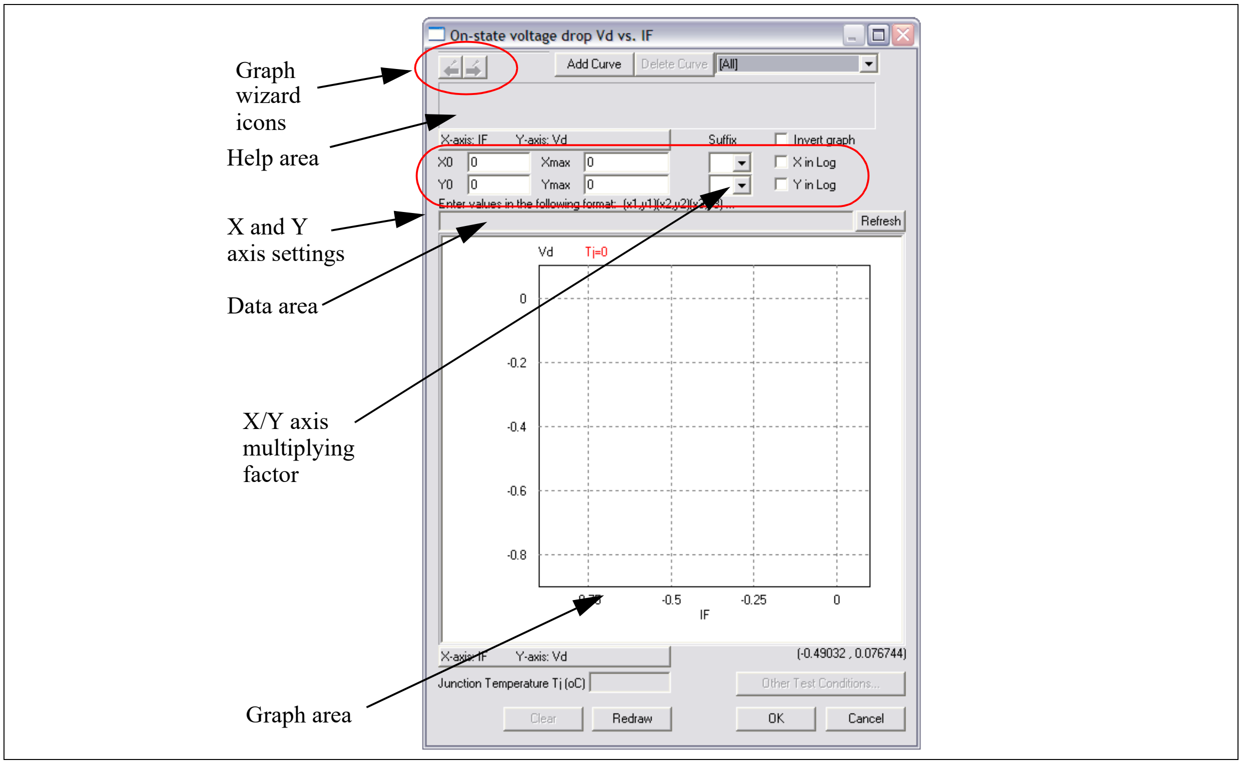
- **Create new device file:** Launch PcdEditor.exe. Go to **File** >> **New Device File**, and create a file called "diode_new.dev". This file will be placed in the *device* sub-folder under the PSIM program folder by default.
- **Create new device:** With the file name "diode_new" highlighted in the "File Name" list, Choose **Device** >> **New Diode**. A diode will be added to the database file with *Manufacturer* as "New" and *Part Number* as "New".
- **Enter basic information:** Obtain the datasheet of Powerex diode CS240650 from the web site www.pwrx.com. By referring to the information from the datasheet, in the database editor, enter the following information for this device:

  Manufacturer:Powerex
  Part Number:CS240650
  Package:Discrete

  and under *Absolute Maximum Ratings*:

  Vrrm,max (V):600
  IF,max (A):50
  Tj,max (^{o}C):150
- **Define forward voltage characteristics V_d vs. I_F:**

Under *Electrical Characteristics* by clicking on the **Edit** button on top of the V_d vs. I_F graph area. The following dialog window will appear.



X-axis: IF Y-axis: Vd

□ Invert graph

X0 0 Xmax 0 □ X in Log

Y0 0 Ymax 0 □ Y in Log

Enter values in the following format: (x1,y1)(x2,y2)(x3,y3) ...

X-axis: IF Y-axis: Vd

Junction Temperature Tj (oC)

The dialog window has two pages: *Graph* and *Conditions.*

The *Graph* page contains thee x and y axis settings as well as the data points and the graph. In this case, the y axis is the conduction voltage drop V_d, and the x axis is the forward current I_F. The x and y axis can have multiplying factors (such as m for 10^{-3}, u for 10^{-6}, etc.).

The *Conditions* page contains the conditions under which the graph is obtained.

There are two ways to define the graph:

- Enter the graph data points manually if there are only a few data points

- Use the Graph Wizard to capture the graph directly from the datasheet image if the graph image is available from datasheet.

To Define the Graph Manually:

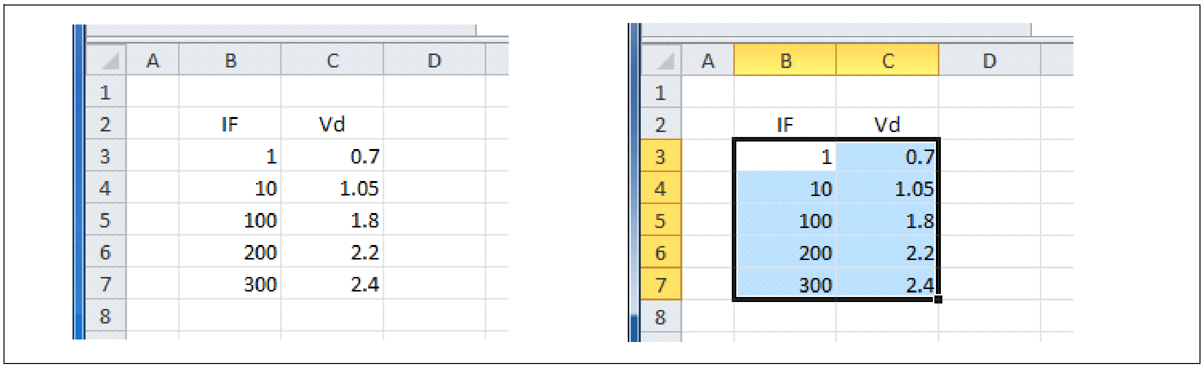
- On the datasheet, refer to the "Maximum On-State Characteristics", and enter the values for the x/y axis settings as follows:

X0:	1
Xmax:	1000
Y0:	0.6
Ymax:	2.6
X in log:	checked

- Visually inspect the graph, and select a few data points. Enter the data points in the data area as follows, then click on the **Refresh** button to display the graph.:

  (1,0.7) (10,1.05) (100,1.8) (200,2.2) (300,2.4)

  Or if the data is in an Excel file, it can be copied directly from Excel into the data area. The figure on the left shows the data in Excel. To copy the data, select the two data columns as shown in the figure on the right, press Ctrl-C to copy. Then move the cursor into the data area, and press Ctrl-V to paste.



	A	B	C	D
1				
2		IF	Vd	
3		1	0.7	
4		10	1.05	
5		100	1.8	
6		200	2.2	
7		300	2.4	
8				

	A	B	C	D
1				
2		IF	Vd	
3		1	0.7	
4		10	1.05	
5		100	1.8	
6		200	2.2	
7		300	2.4	
8				

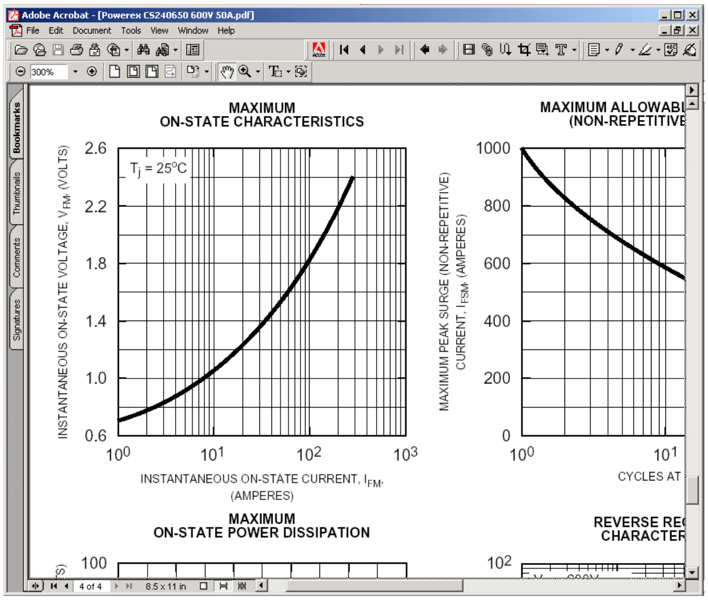
- Click on the *Conditions* tab and enter the *Junction Temperature* as 25 °C.

Alternatively, the graph can be defined in this case using the Graph Wizard.

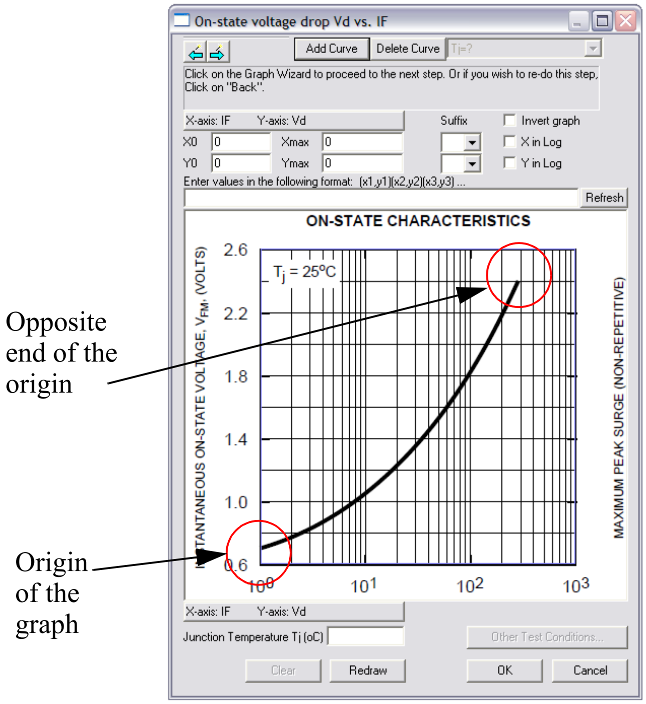
To Define the Graph Using the Graph Wizard:

- Click on the **Add Curve** button. Then click on the forward wizard icon to start the Graph Wizard.

- Display the graph of the datasheet on the screen as follows, then press the Print Screen key (the key is labeled as "Prnt Scr" on some keyboards) to copy the screen image to the clipboard.


Adobe Acrobat - [Powerex CS240650 600V 50A.pdf]
File Edit Document Tools View Window Help
300%
Bookmarks
Thumbnails
Comments
Signatures
MAXIMUM
ON-STATE CHARACTERISTICS
T_j = 25°C
INSTANTANEOUS ON-STATE VOLTAGE, V_{FM}, (VOLTS)
2.6
2.2
1.8
1.4
1.0
0.6
10^0
10^1
10^2
10^3
INSTANTANEOUS ON-STATE CURRENT, I_{FM}, (AMPERES)
MAXIMUM ALLOWABL
(NON-REPETITIVE
MAXIMUM PEAK SURGE (NON-REPETITIVE) CURRENT, I_{FSM}, (AMPERES)
1000
800
600
400
200
0
10^0
10^1
CYCLES AT
MAXIMUM
ON-STATE POWER DISSIPATION
100
REVERSE REC
CHARACTER
10^2
4 of 4
8.5 x 11 in

- Click on the forward wizard icon to paste the screen image into the graph window in the database editor. Position the graph image properly in the graph window by dragging the left mouse so that the complete graph is displayed within the window.
- If the graph image is either too large or too small, go back to the previous step by clicking on the backward wizard icon. Then resize the image of the graph in the Adobe Acrobat, and copy the screen image to the clipboard again.
- The graph dialog window should look something like follows. Click on the forward wizard icon to move on to the next step.


On-state voltage drop Vd vs. IF

Click on the Graph Wizard to proceed to the next step. Or if you wish to re-do this step, Click on "Back".

X-axis: IF Y-axis: Vd

X0 0 Xmax 0

Y0 0 Ymax 0

Suffix

□ Invert graph

□ X in Log

□ Y in Log

Enter values in the following format: (x1,y1)(x2,y2)(x3,y3) ...

X-axis: IF Y-axis: Vd

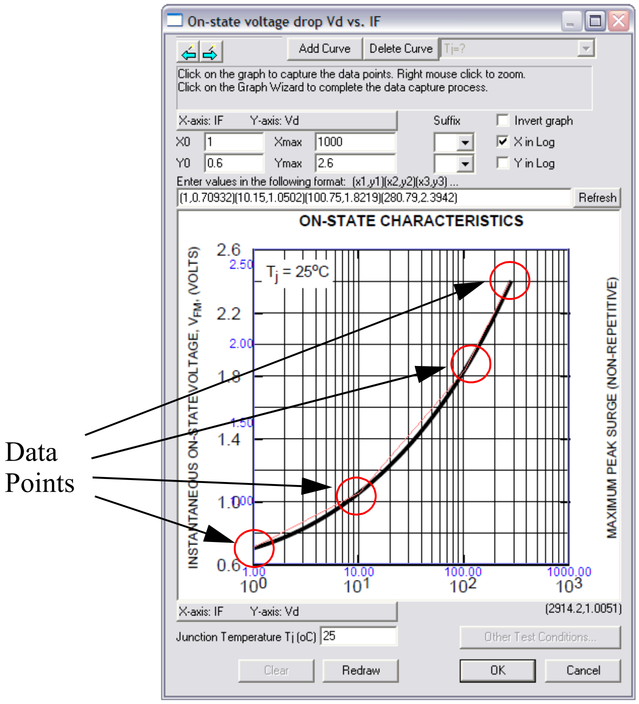
Junction Temperature Tj (oC)

Opposite end of the origin

Origin of the graph

- In this step, the border of the graph area is defined by first left clicking at the origin of the graph (usually the lower left corner), then left clicking again at the opposite corner of the origin (usually the upper right corner). Note that the graph origin does not have to be the lower left corner, and it can be any one of the four corners.

- To locate the origin of the corner more accurately, right mouse click to zoom in, and press the Esc (escape) key to exit the zoom. After this, a blue rectangle will appear around the border of the graph, and the dialog window will appear as follows.



On-state voltage drop Vd vs. IF

Click on the graph to capture the data points. Right mouse click to zoom.
Click on the Graph Wizard to complete the data capture process.

X-axis: IF Y-axis: Vd

X0: 1 Xmax: 1000

Y0: 0.6 Ymax: 2.6

□ Invert graph

☑ X in Log

□ Y in Log

Enter values in the following format: (x1,y1)(x2,y2)(x3,y3) ...

(1,0.70932)(10.15,1.0502)(100.75,1.8219)(280.79,2.3942)

(2914.2,1.0051)

Junction Temperature Tj (oC): 25

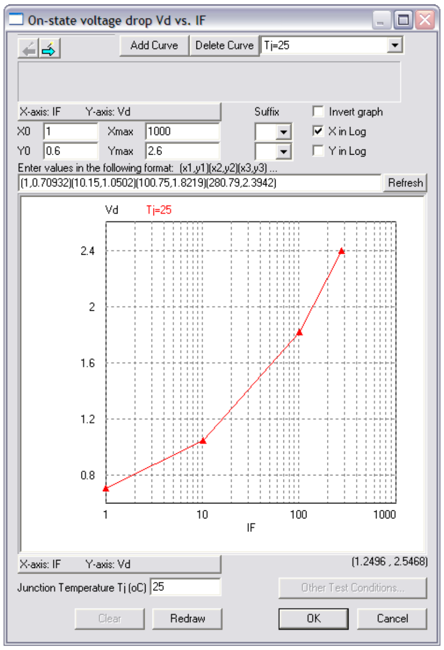
Then click on the forward wizard icon to move on to the next step.

- In this step, the x and y axis settings will be defined. Enter the settings as follows:

X0:	1
Xmax:	1000
Y0:	0.6
Ymax:	2.6
X in log:	checked

- Leave the checkbox **Invert graph** unchecked since the diode forward current I_F is in the x-axis in this case. But in other situations, if the current is in y-axis, check the **Invert graph** checkbox.
- Enter the junction temperature as 25 ^{o}C. Click on the forward wizard icon to move on to the next step.
- Left click on top of the graph to capture the data points. In this case, for example, four data points at the current values of around 1A, 10A, 100A, and 280A are captured. Again, right click to zoom in. You can capture as many data points as desired.
- As data points are captured, red lines will appear that will connect the data points.

- Then click on the forward wizard icon to complete the data capture process. The final graph dialog window should appear as follows.



X-axis: IF Y-axis: Vd

X0 1 Xmax 1000

Y0 0.6 Ymax 2.6

☐ Invert graph

☑ X in Log

☐ Y in Log

Enter values in the following format: (x1,y1)(x2,y2)(x3,y3) ...

(1,0.70932)(10.15,1.0502)(100.75,1.8219)(280.79,2.3942)

X-axis: IF Y-axis: Vd

(1.2496 , 2.5468)

Junction Temperature Tj (oC) 25

- To see the x and y axis values of a particular data point on the graph, place the cursor inside the graph area. The cursor image will change to a cross image, and the x and y coordinates of the cursor will be displayed at the upper right corner of the dialog window. Place the cursor on top of the curve to read the x and y axis readings.
- If there are multiple curves for different junction temperatures, repeat the same process and enter the junction temperature value for each curve accordingly. Use the drop-down menu at the upper right corner of the dialog window to show one particular curve, or to show all the curves at the same time.
- With the same process, define the reverse recovery characteristics t_{rr} vs. I_F, I_{rr} vs. I_F, and Q_{rr} vs. I_F.
- Enter the *Thermal Characteristics* as:

  $R_{th(j-c)}$: 0.6
  $R_{th(c-s)}$: 0.4
- Enter the *Dimension and Weight* as:

  Length (mm):53
  Width (mm):36
  Height (mm):29
- Choose **Device** >> **Save Device** to save the device information. This completes the process of adding the diode into the database.

4.11 Renewable Energy Module

Renewable Energy Module library contains the following elements:

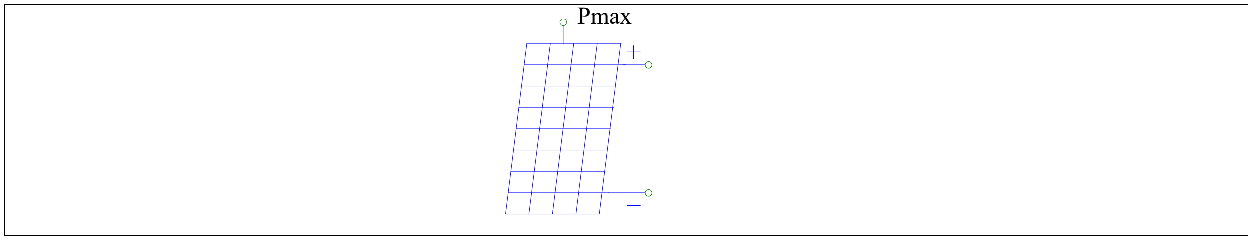
- Solar modules: physical model, functional model, cSi model, and thin-film model;
- Wind turbine
- Lithium-ion and other battery models
- Ultracapacitor

4.11.1 Solar Modules

4.11.1.1 Solar Module - Functional Model

Functional model represents the solar module based on *i-v* characteristics. Functional model of solar module is easy to use because it requires only four parameters which can be obtained on any manufacturer’s datasheet.

Image:



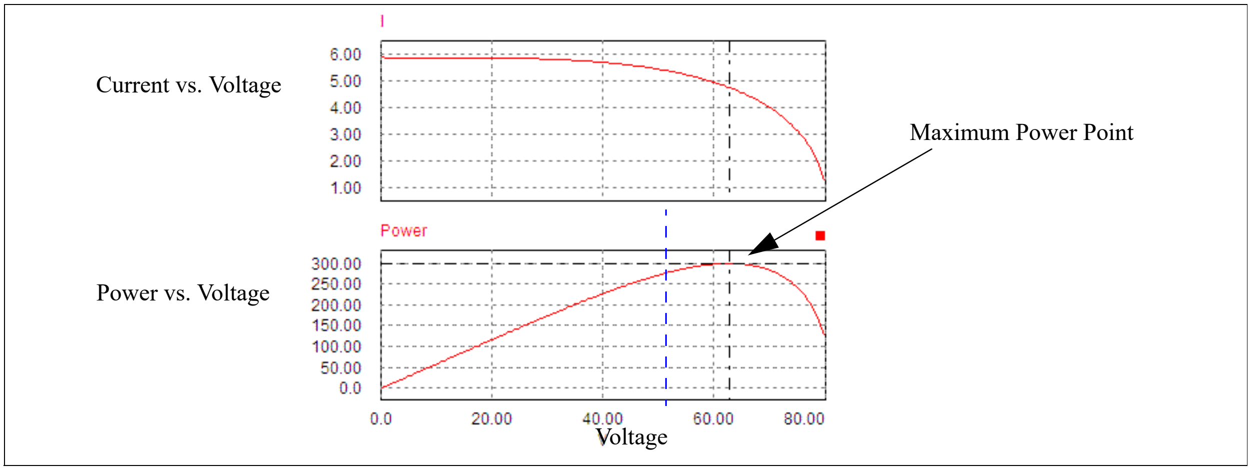
In the image, The nodes marked with the "+" and "-" signs are the positive and negative terminals. The node on the top is theoretical maximum power (in W) given the operating conditions.

While the positive and negative terminal nodes are power circuit nodes, the other nodes are all control circuit nodes.

Attributes:

Parameter	Description
Open Circuit Voltage Voc	Voltage measured when solar cell terminals are open circuit, in V.
Short Circuit Current Isc	Current measured when the solar cell terminals are short circuited, in A.
Maximum Power Voltage Vm	Solar cell terminal voltage when the output power is at the maximum, in V
Maximum Power Current Im	Solar cell terminal current when the output power is at the maximum, in A

Using the four input parameters, the functional model will create the i-v curve of the solar cell. The figure below shows the i-v and p-v curves of a typical solar cell.

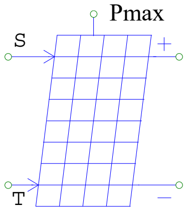


The curves show that the solar cell output power reaches the maximum at a specific voltage level. Many control schemes have been proposed in the literature to track this maximum power point (so called Maximum Power Point Tracking, or MPPT) so that the solar cell output power is at the maximum.

4.11.1.2 Solar Module - Physical Model

The physical model of solar module simulates the behavior of the solar module with more accuracy because it takes into account the light intensity and temperature variation.

Image:



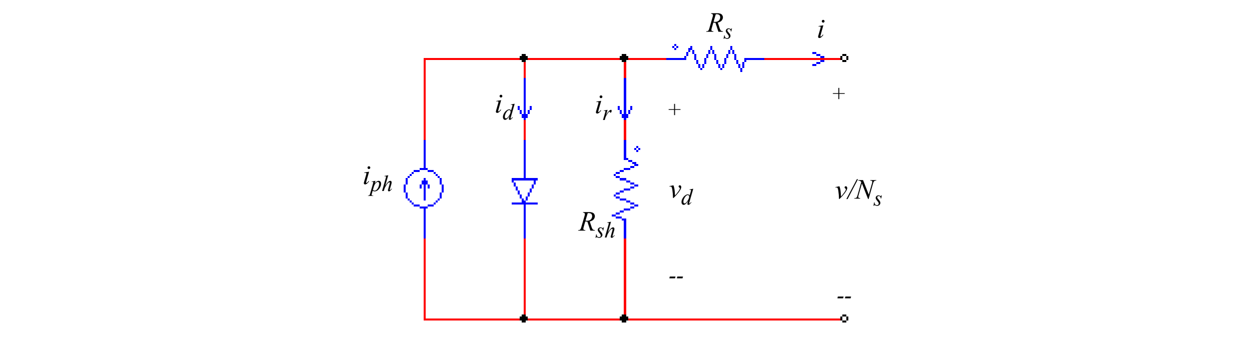
In the image, The nodes with the "+" and "-" signs are the positive and negative terminals. The node with the letter "S" refers to the light intensity input (in W/m^2), and The node with the letter "T" refers to the ambient temperature input (in ^{o}C). The node on the top is theoretical power (in W) given the operating conditions.

While the positive and negative terminal nodes are power circuit nodes, the other nodes are all control circuit nodes.

Attributes

Parameter	Description
Number of Cells Ns	Number of cells N_s of the solar module. A solar module consists of N_s solar cells in series.
Standard Light Intensity S0	Light intensity S_0 under the standard test conditions, in W/m^2. The value is normally 1000 W/m^2 in manufacturer datasheet.
Ref. Temperature Tref	Temperature T_{ref} under the standard test conditions, in ^{o}C
Series Resistance Rs	Series resistance R_s of each solar cell, in Ohm
Shunt Resistance Rsh	Shunt resistance R_{sh} of each solar cell, in Ohm
Short Circuit Current Isc0	Short circuit current I_{sc0} of each solar cell at the reference temperature T_{ref}, in A
Saturation Current Is0	Diode saturation current I_{s0} of each solar cell at the reference temperature T_{ref}, in A
Band Energy Eg	Band energy of each solar cell, in eV. It is around 1.12 for crystalline silicon, and around 1.75 for amorphous silicon.
Ideality Factor A	Ideality factor A of each solar cell, also called emission coefficient. It is around 2 for crystalline silicon, and is less than 2 for amorphous silicon.
Temperature Coefficient Ct	Temperature coefficient C_t, in A/^{o}C or A/^{o}K
Coefficient Ks	Coefficient k_s that defines how light intensity affects the solar cell temperature.

A solar module consists of N_s solar cells in series, and the equivalent circuit of the physical model of one solar cell is shown below:



The equations that describe a solar cell are:

$$i = i_{ph} - i_d - i_r$$

and

$$i_{ph} = I_{sc0} \cdot \frac{S}{S_0} + C_t \cdot (T - T_{ref})$$

$$i_d = I_0 \cdot \left(e^{\frac{q v_d}{AkT}} - 1 \right)$$

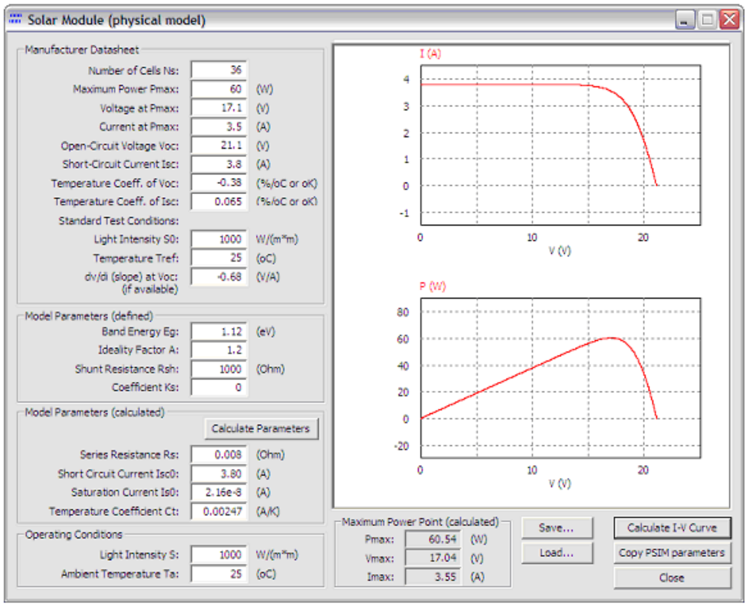
$$I_0 = I_{s0} \cdot \left(\frac{T}{T_{ref}} \right)^3 \cdot e^{\frac{qE_g}{Ak}\left(\frac{1}{T_{ref}} - \frac{1}{T} \right)}$$

$$i_r = \frac{v_d}{R_{sh}}$$

$$v_d = \frac{v}{N_s} + i \cdot R_s$$

$$T = T_a + k_s \cdot S$$

where q is the electron charge (q = 1.6 x 10^{-19} C); k is the Boltzmann constant (k = 1.3806505 x 10^{-23}); S is the light intensity input; T_a is the ambient temperature input; v is the voltage across the entire solar module; and i is the current flowing out of the positive terminal of the solar module.



Manufacturer Datasheet

Number of Cells Ns: 36
Maximum Power Pmax: 60 (W)
Voltage at Pmax: 17.1 (V)
Current at Pmax: 3.5 (A)
Open-Circuit Voltage Voc: 21.1 (V)
Short-Circuit Current Isc: 3.8 (A)
Temperature Coeff. of Voc: -0.38 (%/oC or oK)
Temperature Coeff. of Isc: 0.065 (%/oC or oK)
Standard Test Conditions:
Light Intensity S0: 1000 W/(m*m)
Temperature Tref: 25 (oC)
dv/di (slope) at Voc: (if available) -0.68 (V/A)

Model Parameters (defined)

Band Energy Eg: 1.12 (eV)
Ideality Factor A: 1.2
Shunt Resistance Rsh: 1000 (Ohm)
Coefficient Ks: 0

Model Parameters (calculated)

Series Resistance Rs: 0.008 (Ohm)
Short Circuit Current Isc0: 3.80 (A)
Saturation Current Is0: 2.16e-8 (A)
Temperature Coefficient Ct: 0.00247 (A/K)

Operating Conditions

Light Intensity S: 1000 W/(m*m)
Ambient Temperature Ta: 25 (oC)

Maximum Power Point (calculated)

Pmax: 60.54 (W)
Vmax: 17.04 (V)
Imax: 3.55 (A)

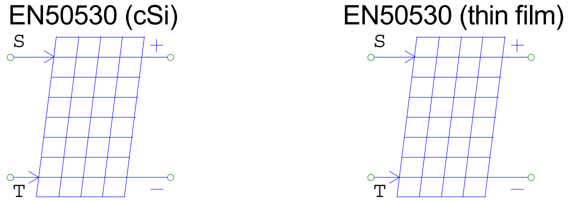
Some of the parameters of the physical model can be obtained from manufacturer datasheet, and the rest of the parameters can be obtained by trial-and-error. A utility tool *Solar Module* (*physical model*) under the **Utilities** menu is provided to help obtaining the parameters from manufacturer datasheet. The interface of the tool is shown on the right.

For more details on how to use this tool to obtain the model parameters, please refer to the tutorial "Tutorial - Solar Module (physical module).pdf" in the "doc" sub-folder in the PSIM directory.

4.11.1.3 Solar Module - cSi and Thin-File Models

The solar module’s cSi (Crystalline Silicon) and Thin-Film models are developed according to EN50530 Standard. Only three parameters are needed. This is because material related data are already included in the v-i characteristic equations in these models.

Images:



In the image, the nodes marked with the "+" and "-" signs are the positive and negative terminals. The node with the letter "S" refers to the light intensity input (in W/m^2), and The node with the letter "T" refers to the ambient temperature input (in ^{o}C).

While the positive and negative terminal nodes are power circuit nodes, the other nodes are all control circuit nodes.

Attributes:

Parameter	Description
Maximum Power	Solar cell maximum output power, in W.
Maximum Power Voltage	Solar cell terminal voltage when the output power is at the maximum, in V
Test Condition Temperature	Test condition temperature, in degree C.

The current-voltage characteristic of cSi and thin-film models equations are:

$$I_{pv} = I_{ph} - I_0(e^{(Vpv + IpvRs)/(mVr)} - 1) - (V_{pv} + I_{pv}R_s)/R_p$$

Where,

$$I_0 = C_0 T_{mod}^3 e^{-\frac{Vpv}{Vr}}$$

and

$$V_T = \frac{kT_{mod}}{e_0}$$

Linear temperature model for the module temperature:

$$T_{mod} = T + \frac{c}{1000\frac{W}{m^2}}G$$

In the above equations:

I_{pv}	Module current
I_0	Diode saturation current
I_{ph}	Photo current (source current)

V_{pv}	Module voltage
V_t	Temperature voltage
V_{gap}	bandgap
R_s	serial resistance
R_p	Parallel resistance
T	Absolute ambient temperature (K)
T_{mod}	Module temperature (K)
G	(Irradiance (W/m2)
c	Constant for the linear temperature model
C_0	Coefficient of diode saturation current
m	diode factor
e_0	Elementary charge
k	Boltzmann constant.

The technology dependable parameters providing the v-i curve of this model are listed in the table below:

	cSi	Thin Film	Tolerance
$\frac{V_{mpp}\|_{G = 200(W/m^2)}}{V_{mpp}\|_{G = 1000(W/m^2)}}$	0.95	0.98	+/- 1%
$\frac{V_{mppSTC}}{V_{ocSTC}}$	0.8	0.72	< 1%
$\frac{I_{mppSTC}}{I_{scSTC}}$	0.9	0.8	< 1%

PV generator model for MPPT performance tests:

MPP to open circuit voltage ratio:

$$FF_V = \frac{V_{mppSTC}}{V_{ocSTC}}$$

MPP to short circuit current ratio:

$$FF_I = \frac{I_{mppSTC}}{I_{scSTC}}$$

Formula for the PV current as a function of PV voltage:

$$I_{pv} = I_{sc}\left(e^{\frac{V_{pv}}{V_{oc}C_{AG}}} - 1\right)$$

Irradiance G and temperature T dependent short circuit current:

$$I_{sc} = I_{scSTC}\frac{G}{G_{STC}} \bullet [1 + \alpha \cdot (T_{pv} - T_{STC})]$$

Irradiance and temperature dependent open circuit voltage:

$$V_{OC} = V_{ocSTC} \cdot (1 + \beta \cdot (T_{pv} - T_{STC}))\left(\ln\left(\frac{G}{C_G} + 1\right) \cdot C_v - (C_R \cdot G)\right)$$

Where the temperature of the PV generator should follow the ambient conditions as follows:

$$T_{PV} = T_{amb} + T_0 + \frac{k}{1 + \tau \cdot s} \cdot G$$

where

T_{PV}	Computed PV generator temperature;

T_{amb} Ambient temperature;

T_0 Correction temperature (T0 = -30 ^{o}C);

k Irradiance gain (k = 0.03 km^2/W);

τ Time constant (τ = 5 minutes);

α Temperature coefficient of the current;

β Temperature coefficient of the voltage;

C_R, C_V, C_G Technology depending correction factor

Irradiance dependent current I0 is given as:

$$I_0 = I_{scSTC}\left((1-FF_I)^{[1/(1-FF_V)]} \cdot \frac{G}{G_{STC}}\right)$$

Constant C_{AQ} is given as:

$$C_{AQ} = \frac{FF_V - 1}{\ln(1-FF_I)}$$

Voltage ratio from V_{MPP} at an irradiance of 200W/m2 to V_{MPP} at an irradiance of 1000W/m^2 is given as:

$$V_{L2H} = \frac{V_{MPP}\big|_{G = 200(W/m^2)}}{V_{MPP}\big|_{G = 1000(W/m^2)}}$$

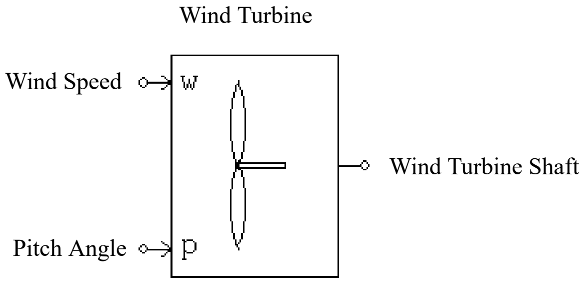
The parameters of the PV generator model must be set as listed in the table below:

	cSi	Thin-Film	Tolerance
FF_V	0.8	0.72	< 1%
FF_I	0.9	0.8	< 1%
C_G [W/m^2]	2.514E-03	1.252E-03	-
C_V	8.593E-02	8.419E-02	-
C_R [m^2/W]	1.088E-04	1.476E-04	-
V_{L2H}	0.95	0.98	±1 %
α	0.04	0.02	
β	-0.4	-0.2	

4.11.2 Wind Turbine

The image and parameter of the wind turbine are described below.

Image:



Attributes:

Parameter	Description
Nominal Output Power	The maximum output power of the wind turbine, in W, at 0^o pitch angle. This power is considered as the maximum power operating point of the turbine, and it should not exceed the rated power of the generator.
Base Wind Speed	The base wind speed that would produce the nominal output power, in m/s
Base Rotational Speed	The base rotational speed of the turbine that would produce the nominal output power, in rpm
Initial Rotational Speed	The initial rotational speed of the turbine, in rpm
Moment of Inertia	Moment of inertia of the wind turbine blade, in kg*m^2
Torque Flag	Flag to display the internal torque of the wind turbine (0: no display; 1: display)
Master/Slave Flag	Master/slave flag for the connected mechanical system (0: slave; 1: master)

In the image, the node with the letter "w" is for the wind speed input in m/s, and the node with the letter "p" is for the blade pitch angle input in deg. Both nodes are control circuit nodes.

The power generated by a wind turbine can be expressed as:

$$P = \frac{1}{2} \cdot A \cdot v_{wind}^3 \cdot \rho \cdot C_p$$

where A is the area of the rotor blade (in m^2), v_{wind} is the wind speed (in m/sec.), ρ is the air density (it is approximately 1.225 kg/m^3), and C_p is the power coefficient. The power coefficient C_p is a function of the tip speed ratio λ and the blade pitch angle β. It can be expressed as [Heier[1]]:

$$C_p = c_1 \cdot (c_2 - c_3\beta - c_4\beta^x - c_5) \cdot e^{-c_6} + c_7$$

where $c_1 = 0.5$, $c_2 = 116*\lambda'$, $c_3 = 0.4$, $c_4 = 0$, $c_5 = 5$, $c_6 = 21*\lambda'$, $c_7 = 0.01*\lambda$, and

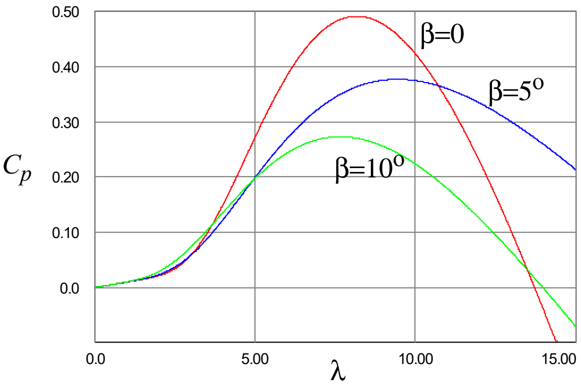
$$\lambda = \frac{\omega_m \cdot R_{blade}}{v_{wind}}$$

$$\lambda' = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}$$

where ω_m is the rotor rotational speed (in rad/sec.) and R_{blade} is the radius of the rotor blade (in m).

The relationship between the power coefficient C_p and the tip speed ratio λ and the blade pitch angle β can be plotted in the figure below.

[1] S. Heier and R. Waddington, *Grid Integration of Wind Energy Conversion Systems*, Wiley, 2006



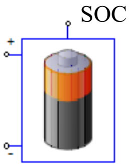
The figure shows that the power coefficient C_p reaches the maximum of 0.49 when the tip speed ratio λ is 8.18. We choose the values of C_p and λ at the maximum as the nominal values, i.e., $C_{p\_nom} = 0.49$, $\lambda_{nom} = 8.18$. One way to control the wind turbine, therefore, is to maintain the tip speed ratio at or close to the nominal value so that the power that the wind turbine generates is at the maximum.

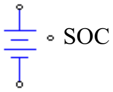
Several elements related to renewable energy applications are included here.

4.11.3 Lithium-Ion Battery Model

A lithium-ion battery model is provided. It comes with two images: one with the battery image, and the other with the battery cell symbol image.

Images:





Attributes:

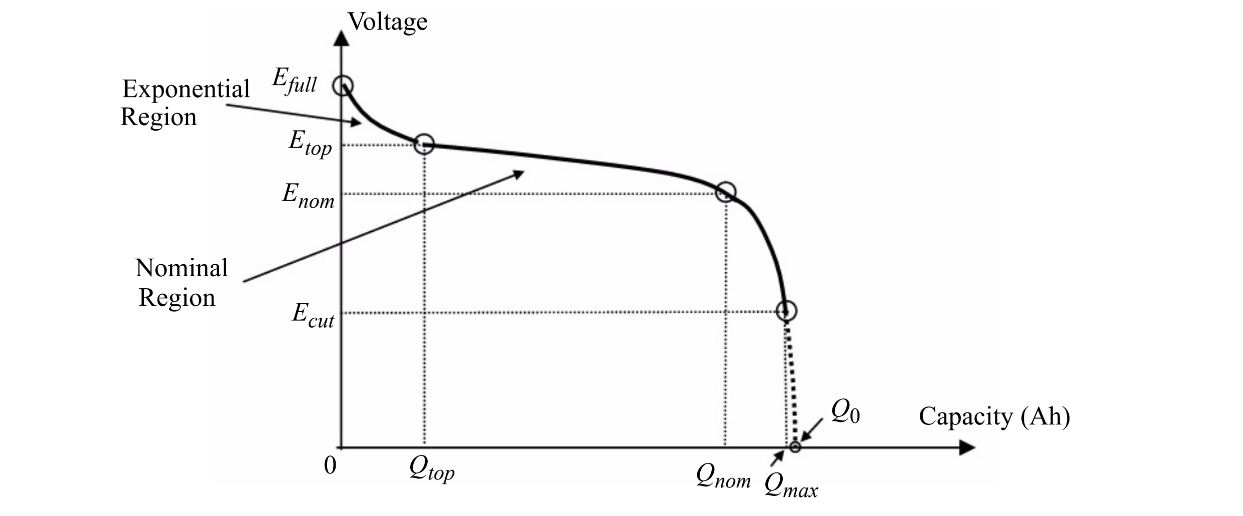
Parameter	Description
No. of Cells in Series	Number of cells N_s in series of a battery pack
No. of Cells in Parallel	Number of cells N_p in parallel of a battery pack
Voltage Derating Factor	Voltage derating factor K_s, from 0 (100% derating) to 1 (no derating)
Capacity Derating Factor	Capacity derating factor K_p, from 0 (100% derating) to 1 (no derating)
Rated Voltage	Rated voltage E_{rated} of the battery cell, in V
Discharge Cut-off Voltage	Battery voltage E_{cut} corresponding to the maximum capacity, in V
Rated Capacity	Rated capacity Q_{rated} of the battery cell, in Ah
Internal Resistance	Internal resistance $R_{battery}$ of the battery cell, in Ohm
Discharge Current	Discharge current of the curve under which model parameters are obtained, in A
Capacity Factor	Capacity factor
Full Voltage	Full (or maximum) voltage E_{full} of the battery cell, in V

Exponential Point Voltage	Battery voltage E_{top} at the end of the exponential region in the discharge curve, in V
Nominal Voltage	Battery voltage E_{nom} at the end of the nominal region in the discharge curve, in V
Maximum Capacity	Maximum capacity Q_{max} of the battery cell, in Ah
Exponential Point Capacity	Battery capacity Q_{top} at the end of the exponential region in the discharge curve, in Ah
Nominal Capacity	Battery capacity Q_{nom} at the end of the nominal region in the discharge curve, in Ah
Initial State of Charge	Initial state of charge (SOC) (from 0 to 1)

In the images, the extra node at the top of the battery image or at the side of the battery cell symbol image is for the SOC output. It is a control circuit node.

Parameter E_{rated}, E_{cut}, and Q_{rated} can be directly read from manufacturer datasheet. Some other parameters can be obtained from the battery discharge curve.

A typical discharge curve is shown in the figure below.



The capacity factor is roughly the ratio between Q_0 (the capacity at 0V) and Q_{max}. It is a value close to 1, and it needs to be adjusted such that the battery voltage is equal to E_{cut} when SOC = 0.

Note that the battery parameters are all for one battery cell, while the model can be used to define a battery pack where the number of cells in series or in parallel is not 1. For a battery pack, all voltages need to be multiplied by N_s*K_s, all capacities by N_p*K_p, and the resistance by N_s/N$_p$. For example, for the entire battery pack:

$$\begin{aligned}
E_{rated\_total} &= N_s \cdot K_s \cdot E_{rated} \\
E_{cut\_total} &= N_s \cdot K_s \cdot E_{cut} \\
E_{full\_total} &= N_s \cdot K_s \cdot E_{full} \\
E_{top\_total} &= N_s \cdot K_s \cdot E_{top} \\
E_{nom\_total} &= N_s \cdot K_s \cdot E_{nom} \\
Q_{rated\_total} &= N_p \cdot K_p \cdot Q_{rated} \\
Q_{max\_total} &= N_p \cdot K_p \cdot Q_{max} \\
Q_{top\_total} &= N_p \cdot K_p \cdot Q_{top} \\
Q_{nom\_total} &= N_p \cdot K_p \cdot Q_{nom}
\end{aligned}$$

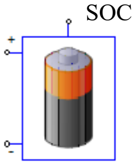
$$R_{battery\_total} = \frac{N_s}{N_p} \cdot R_{battery}$$

For a detailed description on how to define and use the lithium-ion battery model, please refer to the document "Tutorial - How to use Lithium-Ion battery model.pdf".

4.11.4 Battery (Lookup Table) Model

A battery model based on lookup tables is provided.

Image:



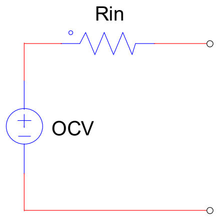
Attributes:

Parameter	Description
No. of Cells in Series	Number of cells N_s in series of a battery pack
No. of Cells in Parallel	Number of cells N_p in parallel of a battery pack
Voltage Derating Factor	Voltage derating factor K_s, from 0 (100% derating) to 1 (no derating)
Capacity Derating Factor	Capacity derating factor K_p, from 0 (100% derating) to 1 (no derating)
Maximum Capacity	Maximum capacity Q_{max} of the battery cell, in Ah
Initial State of Charge	Initial state of charge (SOC) (from 0 to 1)
OCV-SOC Table	Table of the open circuit voltage (OCV) vs. the state of charge (SOC)
Rin-SOC Table (discharge)	Table of the internal resistance Rin vs. SOC during the discharge process
Rin-SOC Table (charge)	Table of the internal resistance Rin vs. SOC during the charge process

In the image, the extra node at the top of the battery image is for the SOC output. It is a control circuit node.

Note that the battery parameters are all for one battery cell, while the model can be used to define a battery pack where the number of cells in series or in parallel is not 1. For a battery pack, all voltages need to be multiplied by N_s*K_s, all capacities by N_p*K_p, and the resistance by N_s/N_p.

The equivalent circuit of the battery is shown below.



For the internal resistance Rin, two tables can be defined: one during the discharge process, and another during the charge process.

Below is the procedure to obtain the lookup tables from manufacture datasheet:

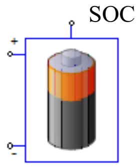
- From the **Utilities** menu, select **Curve Capture Tool**, and capture two discharge curves of the battery voltage versus the discharge capacity from the datasheet at different discharge currents.

- Load the file "Extracting OCV and Rin_discharge.psimsch" (in "examples\batteries\lookup table"). Set the discharge currents and use the tables from the previous step. Run simulation, and display OCV and Rin_discharge separately with SOC as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The OCV and Rin_discharge tables are ready to use by the model.
- From the **Utilities** menu, select **Curve Capture Tool**, and capture the charge curves of the battery voltage, charge current, and charge capacity versus time.
- Load the file "Extracting Rin_charge.psimsch" (in "examples\batteries\lookup table"). Use the tables from the previous step. Run simulation, and display Rin_charge with SOC as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The Rin_charge table is ready to use by the model.

4.11.5 Battery (Lookup Table SOC) Model

A battery model based on lookup tables is provided. The difference between this model and the battery (lookup table) model is that in this model, the discharge capacity points corresponding to SOC=0 and SOC=1 can be defined.

Image:



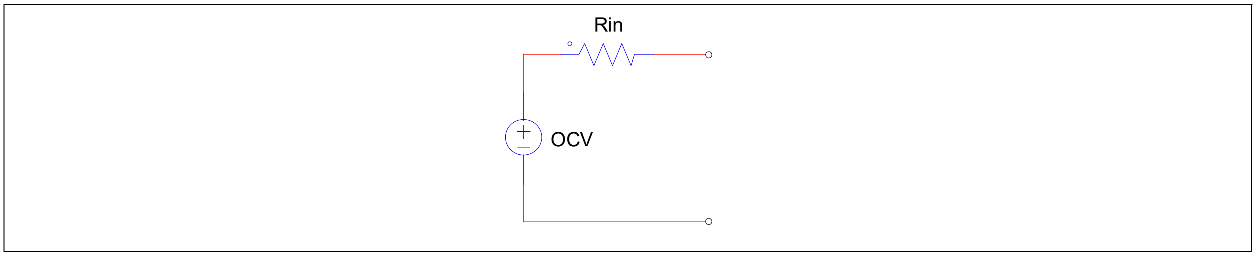
Attributes:

Parameter	Description
No. of Cells in Series	Number of cells N_s in series of a battery pack
No. of Cells in Parallel	Number of cells N_p in parallel of a battery pack
Voltage Derating Factor	Voltage derating factor K_s, from 0 (100% derating) to 1 (no derating)
Capacity Derating Factor	Capacity derating factor K_p, from 0 (100% derating) to 1 (no derating)
Maximum Capacity	Maximum capacity Q_{max} of the battery cell, in Ah
Discharge Capacity SOC=1	Discharge capacity of the battery cell Qext1 corresponding to the state of charge SOC=1.
Discharge Capacity SOC=0	Discharge capacity of the battery cell Qext0 corresponding to SOC=0
Initial State of Charge	Initial state of charge (SOC) (from 0 to 1)
OCV-SOC Table	Table of the open circuit voltage (OCV) vs. the state of charge (SOC)
Rin-SOC Table (discharge)	Table of the internal resistance Rin vs. SOC during the discharge process
Rin-SOC Table (charge)	Table of the internal resistance Rin vs. SOC during the charge process

In the image, the extra node at the top of the battery image is for the SOC output. It is a control circuit node.

Note that the battery parameters are all for one battery cell, while the model can be used to define a battery pack where the number of cells in series or in parallel is not 1. For a battery pack, all voltages need to be multiplied by N_s*K_s, all capacities by N_p*K_p, and the resistance by N_s/N$_p$.

The equivalent circuit of the battery is shown below.

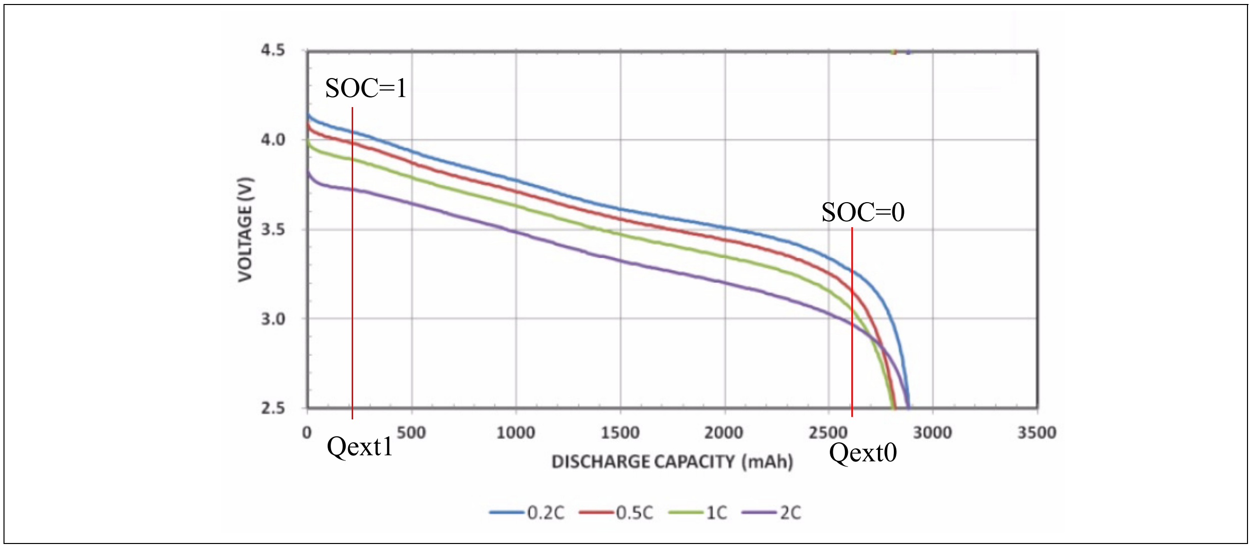


For the internal resistance Rin, two tables can be defined: one during the discharge process, and another during the charge process.

For a discharge curve, typically SOC is defined such that when the discharge capacity is 0 (at the beginning of the discharge), SOC=1; and when the discharge capacity is Qmax, SOC=0. However, in this model, one has the option to define where the points for SOC=0 and SOC=1 are, as shown in the figure below.

In this model, SOC can be less than 0 or greater than 1.

Note that the three lookup tables OCV, Rin (discharge), and Rin (charge) are all obtained with the traditional SOC definition. That is, SOC=1 when the discharge capacity is 0, and SOC=0 when the discharge capacity is Qmax.



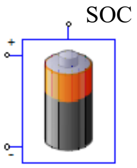
Below is the procedure to obtain the lookup tables from manufacture datasheet:

- From the **Utilities** menu, select **Curve Capture Tool**, and capture two discharge curves of the battery voltage versus the discharge capacity from the datasheet at different discharge currents.
- Load the file "Extracting OCV and Rin_discharge.psimsch" (in "examples\batteries\lookup table (SOC)"). Set the discharge currents and use the tables from the previous step. Run simulation, and display OCV and Rin_discharge separately with SOC as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The OCV and Rin_discharge tables are ready to use by the model.
- From the **Utilities** menu, select **Curve Capture Tool**, and capture the charge curves of the battery voltage, charge current, and charge capacity versus time.
- Load the file "Extracting Rin_charge.psimsch" (in "examples\batteries\lookup table (SOC)"). Use the tables from the previous step. Run simulation, and display Rin_charge with SOC as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The Rin_charge table is ready to use by the model.

4.11.6 Battery (Lookup Table Q) Model

A battery model based on lookup tables is provided. This model is the same as the battery (lookup table SOC) model except that the lookup tables are a function of the charge/discharge capacity instead of SOC. Similarly, in this model, the discharge capacity points corresponding to SOC=0 and SOC=1 can be defined.

Image:



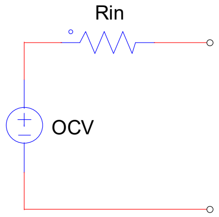
Attributes:

Parameter	Description
No. of Cells in Series	Number of cells N_s in series of a battery pack
No. of Cells in Parallel	Number of cells N_p in parallel of a battery pack
Voltage Derating Factor	Voltage derating factor K_s, from 0 (100% derating) to 1 (no derating)
Capacity Derating Factor	Capacity derating factor K_p, from 0 (100% derating) to 1 (no derating)
Maximum Capacity	Maximum capacity Q_{max} of the battery cell, in Ah
Discharge Capacity SOC=1	Discharge capacity of the battery cell Qext1 corresponding to the state of charge SOC=1.
Discharge Capacity SOC=0	Discharge capacity of the battery cell Qext0 corresponding to SOC=0
Initial State of Charge	Initial state of charge (SOC) (from 0 to 1)
OCV-Qdischarge Table	Table of the open circuit voltage (OCV) vs. the discharge capacity Qdischarge
Rin-Qdischarge Table	Table of the internal resistance Rin vs. Qdischarge during the discharge process
Rin-Qcharge Table	Table of the internal resistance Rin vs. the charge capacity Qcharge during the charge process

In the image, the extra node at the top of the battery image is for the SOC output. It is a control circuit node.

Note that the battery parameters are all for one battery cell, while the model can be used to define a battery pack where the number of cells in series or in parallel is not 1. For a battery pack, all voltages need to be multiplied by N_s*K_s, all capacities by N_p*K_p, and the resistance by N_s/N$_p$.

The equivalent circuit of the battery is shown below.



For the internal resistance Rin, two tables can be defined: one during the discharge process, and another during the charge process.

For a discharge curve, typically SOC is defined such that when the discharge capacity is 0 (at the beginning of the discharge), SOC=1; and when the discharge capacity is Qmax, SOC=0. However, in this model, one has the option to define where the points for SOC=0 and SOC=1 are, as shown in the figure below.

In this model, SOC can be less than 0 or greater than 1.

Note that the three lookup tables OCV, Rin (discharge), and Rin (charge) are all obtained with the traditional SOC definition. That is, SOC=1 when the discharge capacity is 0, and SOC=0 when the discharge capacity is Qmax.

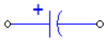
Below is the procedure to obtain the lookup tables from manufacture datasheet:

- From the **Utilities** menu, select **Curve Capture Tool**, and capture two discharge curves of the battery voltage versus the discharge capacity from the datasheet at different discharge currents.
- Load the file "Extracting OCV and Rin_discharge.psimsch" (in "examples\batteries\lookup table (Q)"). Set the discharge currents and use the tables from the previous step. Run simulation, and display OCV and Rin_discharge separately with the discharge capacity as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The OCV and Rin_discharge tables are ready to use by the model.
- From the **Utilities** menu, select **Curve Capture Tool**, and capture the charge curves of the battery voltage, charge current, and charge capacity versus time.
- Load the file "Extracting Rin_charge.psimsch" (in "examples\batteries\lookup table (Q)"). Use the tables from the previous step. Run simulation, and display Rin_charge with the charge capacity Qcharge as the x-axis variable. In Simview, select **File** >> **Save Display As** and save the curve to a text file. Open the file with a text editor and remove the first line. The Rin_charge table is ready to use by the model.

4.11.7 Ultracapacitor Model

Ultracapacitors are electrical energy storage devices that offer high power density and a high number of charge and discharge cycles.

Images:



Attributes:

Parameter	Description
Number of Cells in Series	Number of cells in series
Number of Cells in Parallel	Number of cells in parallel
Capacitance per Cell	Nominal capacitance per cell, in F.
Coefficient Kv	Voltage coefficient
Resistance R1	Resistance coefficient R1, in Ohm
Capacitance C1	Capacitance coefficient C1, in F
Resistance R2	Resistance coefficient R2, in Ohm
Capacitance C2	Capacitance coefficient C2, in F
Resistance R3	Resistance coefficient R3, in Ohm
Capacitance C3	Capacitance coefficient C3, in F
Resistance R4	Resistance coefficient R4, in Ohm.
Maximum Voltage	Maximum voltage rating per cell, in V
Initial Voltage	Initial capacitor voltage per cell, in V

Parameters Kv, R1, and C1 are associated with the capacitor response in the short term (in seconds.). Parameters R2 and C2 are associated with the response in the medium term (in minutes. Parameters R3 and C3 are associated with the response in the long term (in hundreds of minutes). The parameter R4 is associated with capacitor losses due to self-discharge.

For further information on how to use the model, please refer to the tutorial "Tutorial - Ultracapacitor model.pdf".

5

Control Circuit Components

This chapter provides descriptions for the components in PSIM's element library.

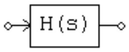
5.1 Transfer Function Blocks

A transfer function block is expressed in polynomial form as:

$$G(s) = k \cdot \frac{B_n \cdot s^n + ... + B_2 \cdot s^2 + B_1 \cdot s + B_0}{A_n \cdot s^n + ... + A_2 \cdot s^2 + A_1 \cdot s + A_0}$$

Two types of transfer function blocks are provided: one with zero initial values (the element is called *s-domain Transfer Function* in the PSIM library) and the other with initial values as input parameters (called *s-domain Transfer Function* (*initial value*) in the PSIM library).

Image:



Attributes:

Parameters	Description
Order n	Order *n* of the transfer function
Gain	Gain *k* of the transfer function
Coeff. B_n...B_o	Coefficients of the numerator (from B_n to B_o)
Coeff. A_n...A_o	Coefficients of the denominator (from A_n to A_o)
Initial Values x_n..x_1	Initial values of the state variables x_n to x_1 (for the element *s-domain Transfer Function* (*initial value*) only)

Let $Y(s) = G(s)*U(s)$ where $Y(s)$ is the output and $U(s)$ is the input, we can convert the s-domain expression into the differential equation form as follows:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & ... & 0 & -A_0/A_n \\ 1 & 0 & 0 & ... & 0 & -A_1/A_n \\ 0 & 1 & 0 & ... & 0 & -A_2/A_n \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 & -A_{n-1}/A_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ ... \\ x_n \end{bmatrix} + \frac{k}{A_n} \cdot \begin{bmatrix} B_0 - A_0 \cdot B_n/A_n \\ B_1 - A_1 \cdot B_n/A_n \\ B_2 - A_2 \cdot B_n/A_n \\ ... \\ B_{n-1} - A_{n-1} \cdot B_n/A_n \end{bmatrix} \cdot u$$

The output equation in the time domain can be expressed as:

$$y = x_n + k \cdot \frac{B_n}{A_n} \cdot u$$

The initial values of the state variables x_n to x_1 can be specified as the inputs in the element *s-domain Transfer Function* (*initial value*).

Example:

The following is a second-order transfer function:

$$G(s) = 1.5 \cdot \frac{400.e^3}{s^2 + 1200 \cdot s + 400.e^3}$$

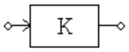
In PSIM, the specification will be:

Order n	2
Gain	1.5
Coeff. B_n...B_o	0. 0. 400.e3
Coeff. A_n...A_o	1. 1200. 400.e3

5.1.1 Proportional Controller

The output of a proportional (P) controller is equal to the input multiplied by a gain.

Image:



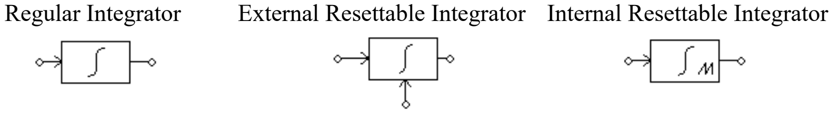
Attribute:

Parameter	Description
Gain	Gain *k* of the transfer function

5.1.2 Integrator

There are three types of integrators: regular integrator, external resettable integrator, and internal resettable integrator.

Images:


Regular Integrator
External Resettable Integrator
Internal Resettable Integrator
M

Attributes:

For *Regular Integrator*:

Parameters	Description
Time Constant	Time constant *T* of the integrator, in sec.
Initial Output Value	Initial value of the output

For *External Resettable Integrator*:

Parameters	Description
Time Constant	Time constant *T* of the integrator, in sec.
Initial Output Value	Initial value of the output
Reset Flag	Reset flag (0: edge reset; 1: level reset)

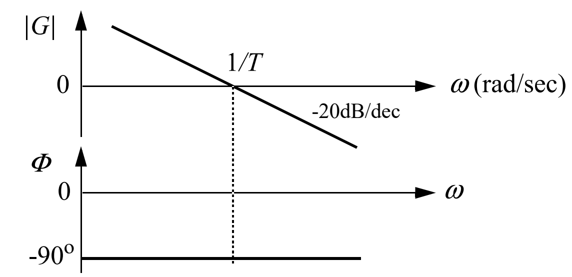
For *Internal Resettable Integrator*:

Parameters	Description
Time Constant	Time constant *T* of the integrator, in sec.
Initial Output Value	Initial value of the output
Lower Output Limit	Lower limit of the output
Upper Output Limit	Upper limit of the output

The transfer function of an integrator is:

$$G(s) = \frac{1}{sT}$$

The Bode plot of the amplitude $|G|$ and the phase angle Φ of the integrator is shown as below:



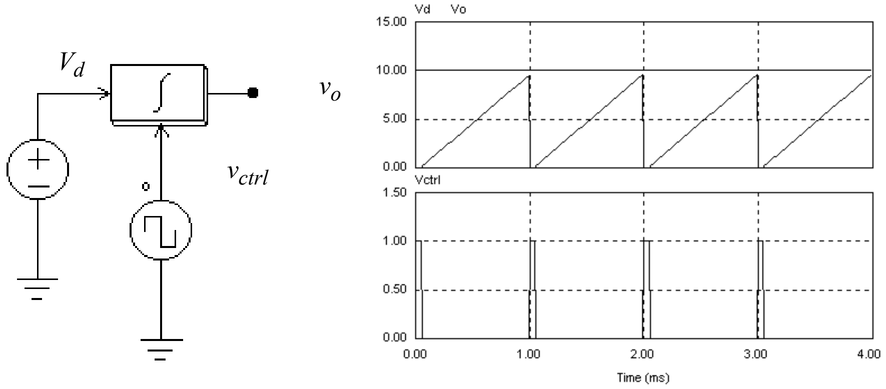
The output of the external resettable integrator can be reset by an external control signal (at the bottom of the block). For the edge reset (reset flag = 0), the integrator output is reset to zero at the rising edge of the control signal. For the level reset (reset flag = 1), the integrator output is reset to zero as long as the control signal is high (1).

The output of the internal resettable integrator is reset to 0 when the output reaches either the lower limit or the upper limit. It works in the same way as the external resettable integrator with the edge reset, except that in this case users do not need to set up the external reset circuit.

When a limiter is connected to the integrator output, anti-windup is implemented automatically.

Example:

The following circuit illustrates the use of the resettable integrator. The input of the integrator is a dc quantity. The control input of the integrator is a pulse waveform which resets the integrator output at the end of each cycle. The reset flag is set to 0.



5.1.3 Differentiator

The transfer function of a differentiator is:

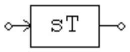
$$G(s) = sT$$

A differentiator is calculated as follows:

$$v_o(t) = T \cdot \frac{v_{in}(t) - v_{in}(t - \Delta t)}{\Delta t}$$

where Δt is the simulation time step, $v_{in}(t)$ and $v_{in}(t\text{-}\Delta t)$ are the input values at the present and the previous time step.

Image:



Attribute:

Parameter	Description
Time Constant	Time constant *T* of the differentiator, in sec.

Since sudden changes of the input will generate spikes at the output, it is recommended that a low-pass filter be placed at the input of the differentiator.

5.1.4 Proportional-Integral Controller

A proportional-integral (PI) controller is defined as below.

Image:

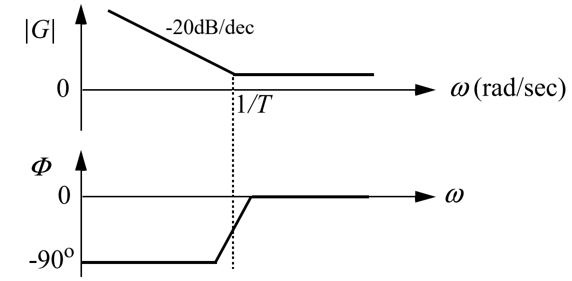
Attributes:

Parameters	Description
Gain	Gain *k* of the PI controller
Time Constant	Time constant *T* of the PI controller

The transfer function of a PI controller is defined as:

$$G(s) = k \cdot \frac{1 + sT}{sT}$$

The Bode plot of the amplitude $|G|$ and the phase angle Φ of the PI controller is shown as below:

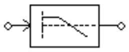


When a limiter is connected to the PI output, anti-windup is implemented automatically.

5.1.5 Single-Pole Controller

A single-pole controller is a proportional controller with a pole. It is defined as below.

Image:



Attributes:

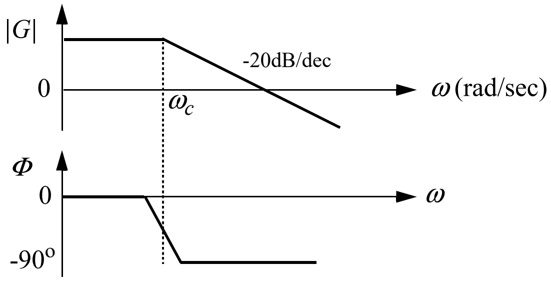
Parameters	Description
Gain	Gain *k* of the controller
Pole Frequency	Frequency of the pole f_c, in Hz

The transfer function of the single-pole is defined as:

$$G(s) = k \cdot \frac{\omega_c}{s + \omega_c}$$

where $\omega_c = 2\pi f_c$.

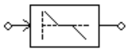
The Bode plot of the amplitude $|G|$ and the phase angle Φ of the single-pole controller is shown as below:



5.1.6 Modified PI Controller

A modified PI controller (Type-2) consists of a PI with a pole.

Image:



Attributes:

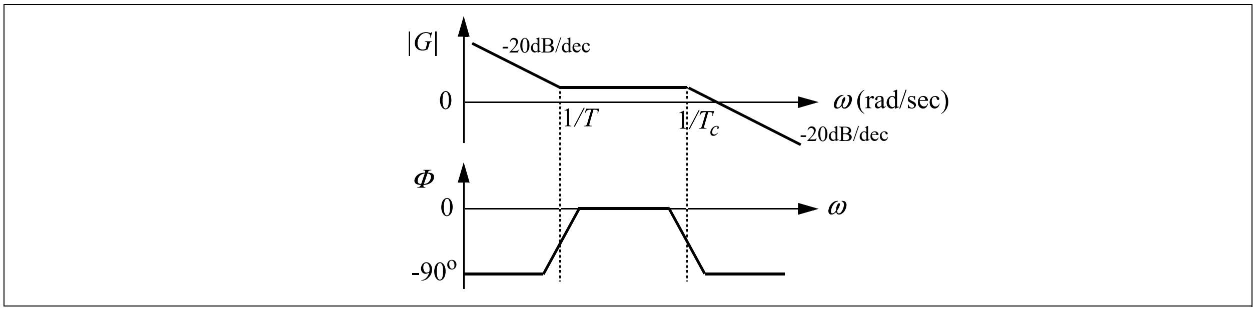
Parameters	Description
Gain	Gain *k* of the PI controller
Time Constant	Time constant *T* of the PI portion of the controller
Pole Frequency	Frequency of the pole f_c, in Hz

The transfer function of a modified PI controller is defined as:

$$G(s) = k \cdot \frac{1 + sT}{sT} \cdot \frac{1}{1 + sT_c}$$

where $T_c = \frac{1}{\omega_c}$ and $\omega_c = 2\pi f_c$.

The Bode plot of the amplitude $|G|$ and the phase angle Φ of the PI controller is shown as below:

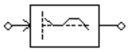


When a limiter is connected to the PI output, anti-windup is implemented automatically.

5.1.7 Type-3 Controller

A Type-3 controller consists of two zeros and two poles.

Image:



Attributes:

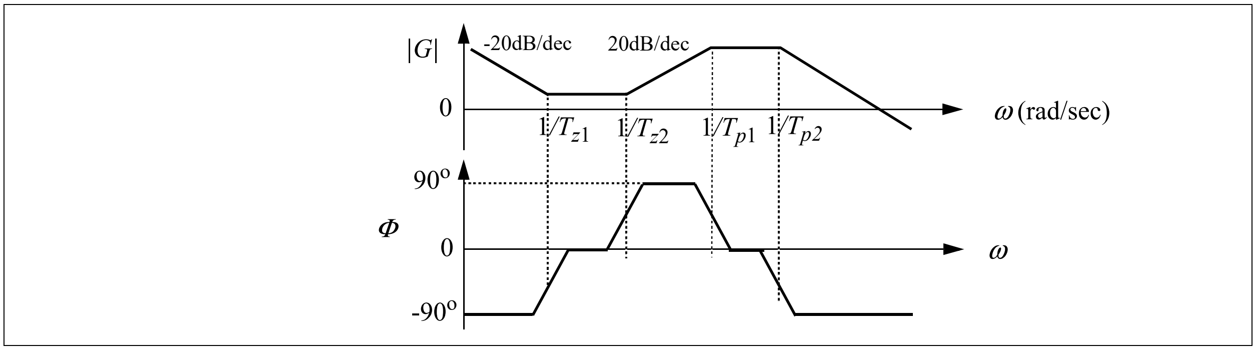
Parameters	Description
Gain	Gain *k* of the PI controller
Frequency fz1	Frequency f_{z1} of the first zero, in Hz
Frequency fz2	Frequency f_{z2} of the second zero, in Hz
Frequency fp1	Frequency f_{p1} of the first pole, in Hz
Frequency fp2	Frequency f_{p2} of the second pole, in Hz

The transfer function of a Type-3 controller is defined as:

$$G(s) = k \cdot \frac{1 + sT_{z1}}{sT_{z1}} \cdot \frac{1 + sT_{z2}}{(1 + sT_{p1}) \cdot (1 + sT_{p2})}$$

where $T_{z1} = \frac{1}{2\pi f_{z1}}$, $T_{z2} = \frac{1}{2\pi f_{z2}}$, $T_{p1} = \frac{1}{2\pi f_{p1}}$, and $T_{p2} = \frac{1}{2\pi f_{p2}}$.

The Bode plot of the amplitude |*G*| and the phase angle *Φ* of the Type-3 controller is shown as below:

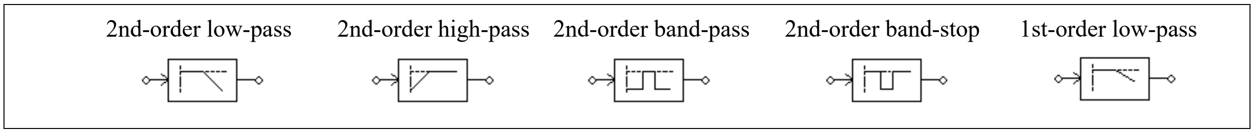


When a limiter is connected to the controller output, anti-windup is implemented automatically.

5.1.8 Built-in Filter Blocks

Four second-order filters and one first-order filter are provided as built-in blocks in PSIM.

Images:



Attributes for second-order filters:

Parameters	Description
Gain	Gain k
Damping Ratio	Damping ratio ξ
Cut-off Frequency	Cut-off frequency f_c ($f_c = \frac{\omega_c}{2\pi}$) for low-pass and high-pass filters, in Hz
Center Frequency	Center frequency f_o ($f_o = \frac{\omega_o}{2\pi}$) for band-pass and band-stop filter, in Hz
Passing Band; Stopping Band	Frequency width f_b ($f_b = \frac{B}{2\pi}$) of the passing/stopping band for band-pass/band-stop filters, in Hz

Attributes for first-order low-pass filter:

Parameters	Description
Gain	Gain k
Cut-off Frequency	Cut-off frequency f_c ($f_c = \frac{\omega_c}{2\pi}$) of the low-pass filter, in Hz

The transfer function of these filters are listed below.

For second-order low-pass filter:

$$G(s) = k \cdot \frac{\omega_c^2}{s^2 + 2\xi\omega_c s + \omega_c^2}$$

For second-order high-pass filter:

$$G(s) = k \cdot \frac{s^2}{s^2 + 2\xi\omega_c s + \omega_c^2}$$

For second-order band-pass filter:

$$G(s) = k \cdot \frac{B \cdot s}{s^2 + B \cdot s + \omega_o^2}$$

For second-order band-stop filter:

$$G(s) = k \cdot \frac{s^2 + \omega_o^2}{s^2 + B \cdot s + \omega_o^2}$$

For first-order low-pass filter:

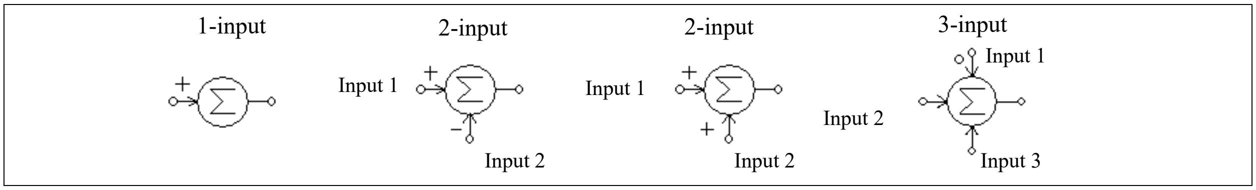
$$G(s) = k \cdot \frac{\omega_c}{s + \omega_c}$$

5.2 Computational Function Blocks

5.2.1 Summer

The input of a one-input summer or two-input summer can be either a scalar or a vector. The input of a three-input summer can only be a scalar.

Images:



Attribute:

Parameter	Description
Gain_*i*	Gain k_i for the i_{th} input

For the three-input summer, the input with a dot is the first input.

If the inputs are scalar, the output of a summer with n inputs is defined as:

$$V_o = k_1 V_1 + k_2 V_2 + \ldots + k_n V_n$$

If the input is a vector, the output of a two-input summer will also be a vector, which is defined as:

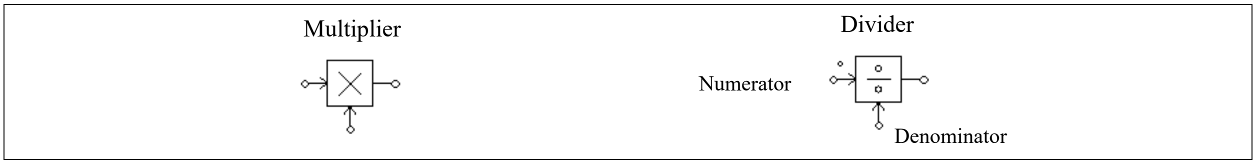
$V_1 = [a_1\ a_2\ \ldots\ a_n]$
$V_2 = [b_1\ b_2\ \ldots\ b_n]$
$V_o = V_1 + V_2 = [a_1+b_1\ a_2+b_2\ \ldots\ a_n+b_n]$

The output of a one-input summer, however, will still be a scalar which is equal to the summation of the input vector elements, that is, $V_o = a_1 + a_2 + \ldots a_n$.

5.2.2 Multiplier and Divider

The output of a multipliers or divider is equal to the multiplication or division of two inputs.

Images:



For the divider, the dotted node is for the numerator input.

The input of a multiplier can be either a vector or a scalar. If the two inputs are vectors, their dimensions must be equal. Let the two inputs be:

$$V_1 = [a_1\ a_2\ \ldots\ a_n]$$
$$V_2 = [b_1\ b_2\ \ldots\ b_n]$$

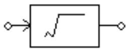
The output, which is a scalar, will be:

$$V_o = V_1 * V_2^T = a_1 * b_1 + a_2 * b_2 + a_n * b_n$$

5.2.3 Square-Root

A square-root function block calculates the square root of the input.

Image:



5.2.4 Exponential/Power/Logarithmic Functions

The images and attributes of these function blocks are shown below.

Images:

Exponential
Power
LOG
LOG10
a x
x a
log
log 10

Attributes (for exponential and power blocks):

Parameters	Description
Coefficient k_1	Coefficient k_1
Coefficient k_2	Coefficient k_2

The output of an exponential function block is defined as:

$$V_o = k_1 \cdot k_2^{V_{in}}$$

For example, if $k_1 = 1$, $k_2 = 2.718281828$, and $V_{in} = 2.5$, then $V_o = e^{2.5}$ where e is the base of the natural logarithm.

The output of a power function block is defined as:

$$V_o = k_1 \cdot V_{in}^{k_2}$$

The function block LOG gives the natural logarithm (base e) of the input, and the block LOG10 gives the common logarithm (base 10) of the input.

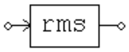
5.2.5 Root-Mean-Square

A root-mean-square function block calculates the RMS value of the input over a period specified by the base frequency f_b. The output is defined as:

$$V_{rms} = \sqrt{\frac{1}{T}\int_0^T v_{in}^2(t)dt}$$

where $T = 1/f_b$. The output is only updated at the beginning of each period.

Image:



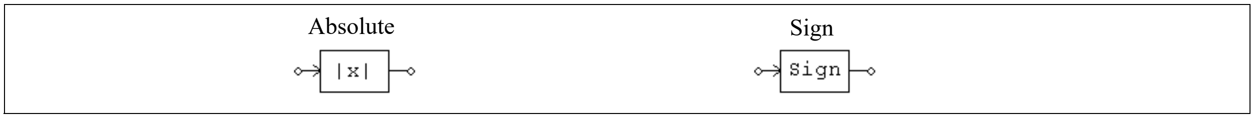
Attribute:

Parameter	Description
Base frequency	Base frequency f_b, in Hz

5.2.6 Absolute and Sign Functions

An absolute value function block gives the absolute value of the input. A sign function block gives the sign of the input, i.e., the output is 1 if the input is positive, 0 if the input is 0, and -1 if the input is negative.

Images:

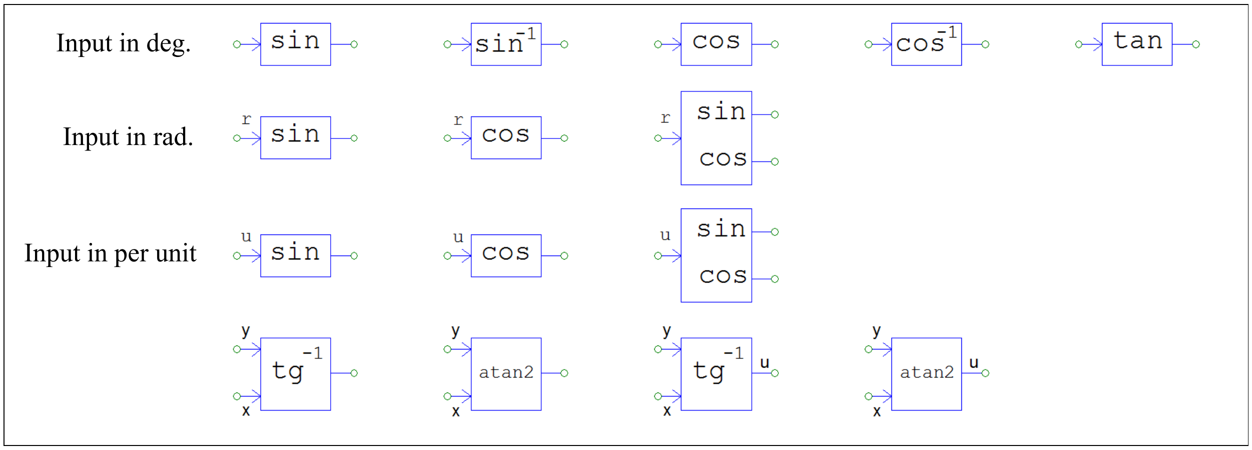


5.2.7 Trigonometric Functions

The following trigonometric functions are provided: sine (sin), cosine (cos), arc-sine ($\sin^{-1}$), arc-cosine ($\cos^{-1}$), tangent (tan), arc-tangent (tg^{-1}), and arc-tangent 2 (atan2).

The output is equal to the corresponding trigonometric function of the input. For the sin, cos, and tan blocks, the input can be in degree, radian, or per unit. and for the arcsin, arccos, and arctangent blocks, the output can be in degree, radian, or per unit.

Images:



If the input of a block has the letter "r", it is in radian. If the input has the letter "u", it is in per unit. Otherwise it is in deg.

In the arctangent and arctangent 2 blocks, the input node marked with *x* is the real part, and the other input node marked with *y* is the imaginary part. If the output has the letter "u", the output is in per unit.

The output of the arctangent and arctangent 2 blocks is the inverse tangent of the ratio between the imaginary part *y* and the real part *x*, i.e. $\theta = tg^{-1}\left(\frac{y}{x}\right)$. The output of the arctangent block is in degree, and the range is from 0 to 360°. On the other hand, the output of the arctangent 2 block is in radian, and the range is from -π to +π. The arctangent 2 block behaves in the same way as the function atan2(y,x) in the C language.

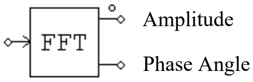
5.2.8 Fast Fourier Transform

A Fast Fourier Transform block calculates the fundamental component of the input signal. The FFT algorithm is based on the radix-2/decimation-in-frequency method. The number of sampling points within one fundamental period should be 2^N (where *N* is an integer). The maximum number of sampling points allowed is 1024.

The output gives the peak amplitude and the phase angle of the input fundamental component. The output voltage (in complex form) is defined as:

$$v_o = \frac{2}{N} \cdot \sum_{n=0}^{n=\frac{N}{2}-1} \left(\left[v_{in}(n) - v_{in}\left(n + \frac{N}{2}\right) \right] \cdot e^{-j\frac{2\pi n}{N}} \right)$$

Image:



Attributes:

Parameters	Description
No. of Sampling Points	No. of sampling points N
Fundamental Frequency	Fundamental frequency f_b, in Hz.

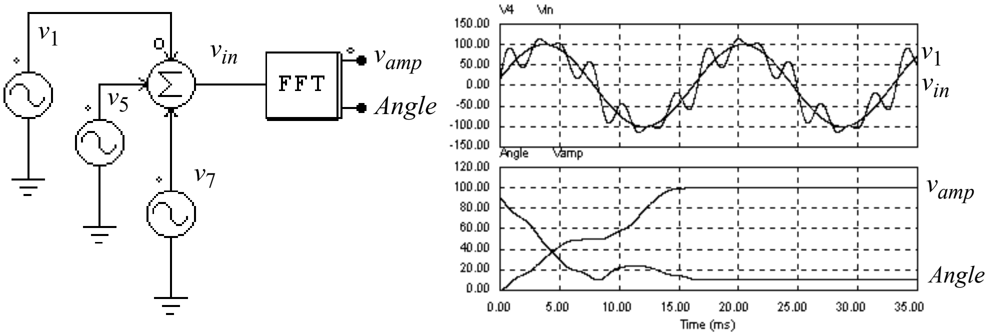
The dotted node of the block refers to the output of the amplitude. Note that the phase angle output has been internally adjusted such that a sine function V_m*sin(ωt) will give a phase angle output of 0.

Note that the FFT block only works if the input signal contains dc, the fundamental component, and harmonics of the fundamental components (with the frequencies at multiple integer times of the fundamental frequency). If the input signal contains another component that is not multiple integer of the fundamental frequency, the FFT block will not give the correct result.

Also, the FFT block only measures the fundamental component, not any other harmonics. For example, if the input signal contains 60 Hz (fundamental) and 180 Hz, if the parameter "Fundamental Frequency" is set to 180, the FFT block output will be incorrect.

Example:

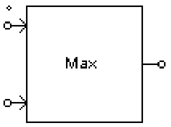
In the circuit below, the voltage v_{in} contains a fundamental component v_1 (100 V at 60 Hz), a 5th harmonic voltage v_5 (25 V at 300 Hz), and a 7th harmonic v_7 (25 V at 420 Hz). After one cycle, the FFT block output reaches the steady state with the amplitude of 100 V and the phase angle of 0^o.



5.2.9 Maximum/Minimum Function

A maximum/minimum function block detects the maximum or minimum of the inputs.

Image:



Attributes:

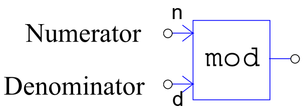
Parameters	Description
Number of Inputs	Number of inputs
Block Function Type	Function type of the block. It can be either *Maximum* or *Minimum*.

The number of inputs can be changed through the drop-down menu.

5.2.10 MOD Function

A MOD function block performs the modulo operation.

Image:



The block has two inputs: the numerator input V_{in_n} on the top left (with the letter "n"), and the denominator input V_{in_d} on the bottom left (with the letter "d").

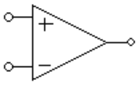
The block gives the remainder of the division V_{in_n} / V_{in_d}. For example, if V_{in_n} = 10.1 and V_{in_d} = 2, the output will be 0.1.

5.3 Other Function Blocks

5.3.1 Comparator

The output of a comparator is high when the positive input is higher than the negative input. When the positive input is lower, the output is zero. If the two input are equal, the output is undefined and it will keep the previous value.

Image:

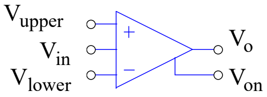


Note that the comparator image is similar to that of the op. amp. For the comparator, the noninverting input is at the upper left and the inverting input is at the lower left. For the op. amp., however, it is the opposite.

5.3.2 Hysteresis Comparator

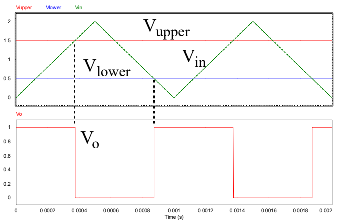
A hysteresis comparator provides a hysteresis band when comparing the input.

Image:



The node definitions are shows above. The output V_{on} is complementary to the output V_o.

When the input V_{in} is greater than the lower voltage threshold V_{lower}, the output V_o will be 1. When V_{in} is greater than the upper voltage threshold V_{high}, the output V_o will be 0. The waveforms are shown below.

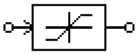


5.3.3 Limiters

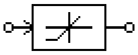
Four types of limiters are provided to limit the amplitude of a signal: lower/upper limiter, lower limiter, upper limiter, and range limiter.

Images:

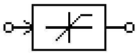
Lower/upper Limiter



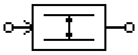
Lower Limiter



Upper Limiter



Range Limiter



Attributes:

Parameters	Description
Lower Limit	Lower limit V_{lower} of the limiter
Upper Limit	Upper limit V_{upper} of the limiter

For a lower/upper limiter, or lower limiter, or upper limiter, the output of the limiter is clamped to the upper or lower limit whenever the input exceeds the limiter range. If the input is within the limit, the output is equal to the input.

A range limiter limits the output v_o between the lower limit and the upper limit, that is:

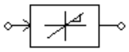
$$V_{lower} \leq v_o < V_{upper}$$

Let the range of the upper limit and the lower limit be V_{range}. When the output is equal to or greater than the upper limit, the output is subtracted by the range V_{range} until it is within the range. When the output is below the lower limit, it is added by range V_{range} until it is within the range. When the input is within the limit, the output is equal to the input.

5.3.4 Gradient (dv/dt) Limiter

A gradient (dv/dt) limiter limits the rate of change of the input. If the rate of change is within the limit, the output is equal to the input.

Image:



Attribute:

Parameter	Description
dv/dt Limit	Limit of the rate of change (dv/dt) of the input

5.3.5 Trapezoidal and Square Blocks

Trapezoidal waveform blocks and square waveform blocks are specific types of lookup tables: the output and the input relationship is either a trapezoidal or a square waveform.

Images:

Trapezoidal Waveform
Square Waveform

For the trapezoidal waveform block:

Attributes:

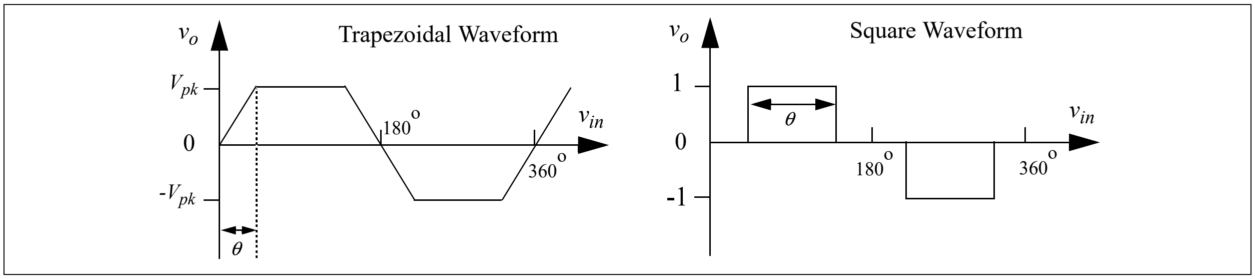
Parameters	Description
Rising Angle theta	Rising angle θ, in deg.
Peak Value	Peak value V_{pk} of the waveform

For the square waveform block:

Attribute:

Parameter	Description
Pulse Width (deg.)	Pulse width θ in half cycle, in deg.

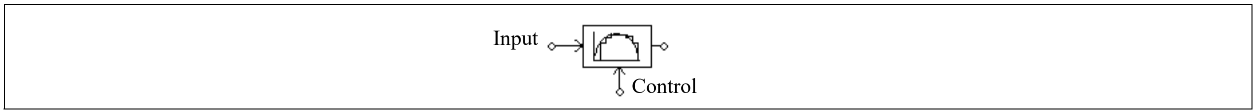
The waveforms of these two blocks are shown below. Note that the input v_{in} is in deg., and can be in the range of -360° to 360°. Both waveforms are half-wave and quarter-wave symmetrical.



5.3.6 Sampling/Hold Block

A sampling/hold block samples the input when the control signal changes from low to high (from 0 to 1), and holds this value until the next point is sampled.

Image:

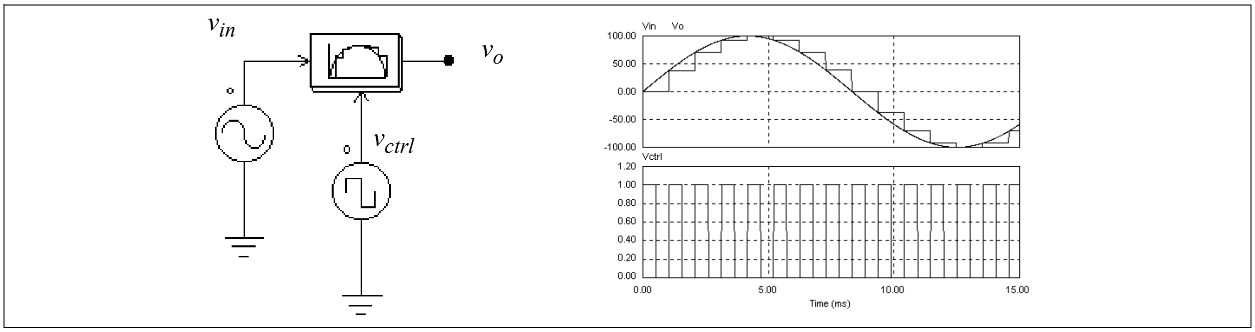


The difference between this block and the zero-order hold block (ZOH) in Digital Control Module is that this block is treated as a continuous element and sampling moments can be controlled externally; whereas the zero-order hold block is a discrete element and the sampling moments are fixed and of equal distance.

For a discrete system, the zero-order hold block should be used.

Example:

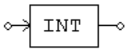
In this example, a sinusoidal input is sampled. The control signal is a square wave voltage source with an amplitude of 1.



5.3.7 Round-Off Block

The image of a round-off block is shown below:

Image:



Attributes:

Parameters	Description
No. of Digits	No. of digits N after the decimal point
Truncation Flag	Truncation flag (1: truncation; 0: round-off)

Let the input of the round-off block be V_{in}. The input is first scaled based on the following expression:

$$V_{in,\,new} = V_{in} \cdot 10^N$$

If the truncation flag is 1, the output will be equal to $V_{in,new}$ truncated, and then divided by 10^N. Otherwise, the output will be equal to $V_{in,new}$ rounded off to the nearest integer, and then divided by 10^N.

Examples:

If V_{in} = 34.5678, N = 0, truncation flag = 0, then we have the output V_{out} = 35. Similarly, if V_{in} = 34.5678, N = 0, truncation flag = 1, then V_{out} = 34.

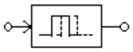
If V_{in} = 34.5678, N = 1, truncation flag = 1, then V_{out} = 34.5. If V_{in} = 34.5678, N = -1, truncation flag = 1, then V_{out} = 30.

5.3.8 Time Delay Blocks

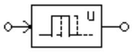
Two types of time delay blocks are provided: one with arbitrary delay time, and the other with only one simulation time step.

Images:

Time Delay



Unit Time Delay



Attribute:

Parameter	Description
Delay Time	Delay time, in sec. (for the Time Delay block only)

A time delay block delays the input signal by a specified amount of time interval. The input signal can be an analog signal or logic signal. For example, it can be used to model the propagation delay of a logic element.

A unit time delay block delays the input signal by just one simulation time step. If the simulation time step is changed, the delay time will be changed accordingly.

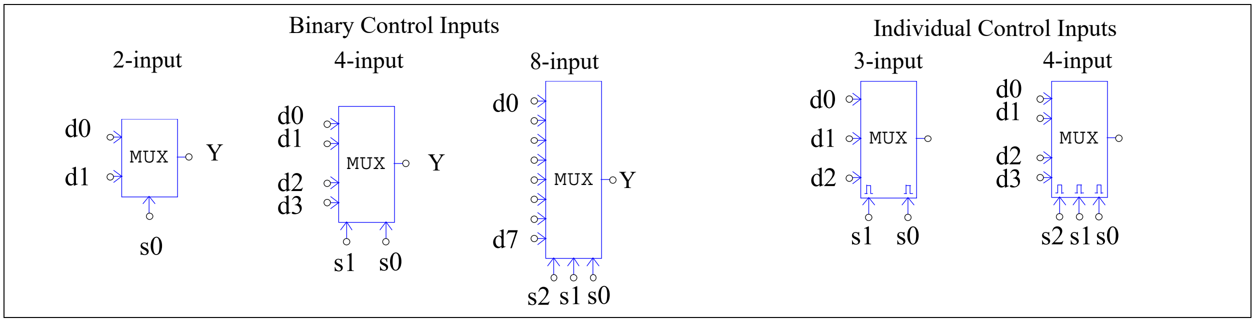
Note that the difference between the unit time delay block and the unit delay block in Digital Control Module is that this block is a continuous element and the delay time is one simulation time step; whereas the unit delay block is a discrete element and the delay time is equal to the sampling period.

For a discrete system, the unit delay block should be used.

5.3.9 Multiplexer

The output of a multiplexer is equal to a selected input depending on the control signal. Two types of multiplexers are provided: One with binary control inputs, and the other with individual control inputs.

Images:



In the images, d0...d7 are the data inputs; and s0..s2 are the control inputs. The truth tables of the multiplexers are as follows.

For multiplexers with binary control inputs:

2-Input	
s0	Y
0	d0
1	d1

4-Input		
s1	s0	Y
0	0	d0
0	1	d1
1	0	d2
1	1	d3

8-Input			
s2	s1	s0	Y
0	0	0	d0
0	0	1	d1
0	1	0	d2
0	1	1	d3
1	0	0	d4
1	0	1	d5
1	1	0	d6
1	1	1	d7

For multiplexers with individual control inputs:

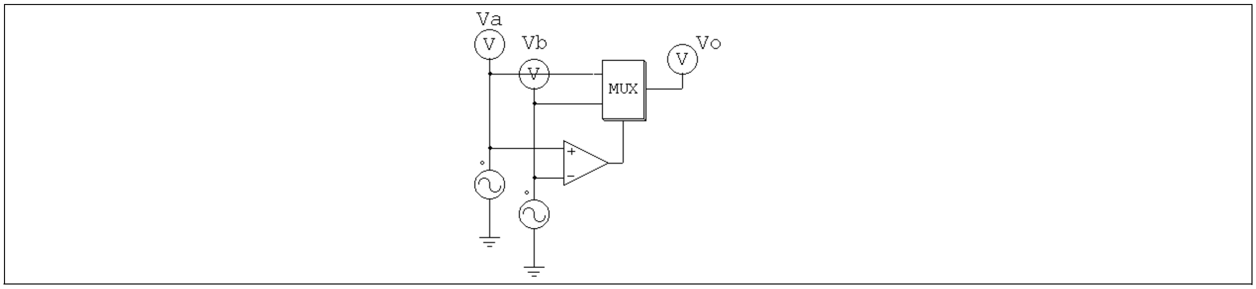
3-Input		
s1	s0	Y
x	1	d0
1	0	d1
otherwise		d2

4-Input			
s2	s1	s0	Y
x	x	1	d0
x	1	0	d1
1	0	0	d2
otherwise			d3

Note that the data input could be either an analog or digital signal.

Example:

The following circuit selects the maximum value out of two inputs. When V_a is greater than V_b, the comparator output will be 1, and $V_o = V_a$. Otherwise $V_o = V_b$.



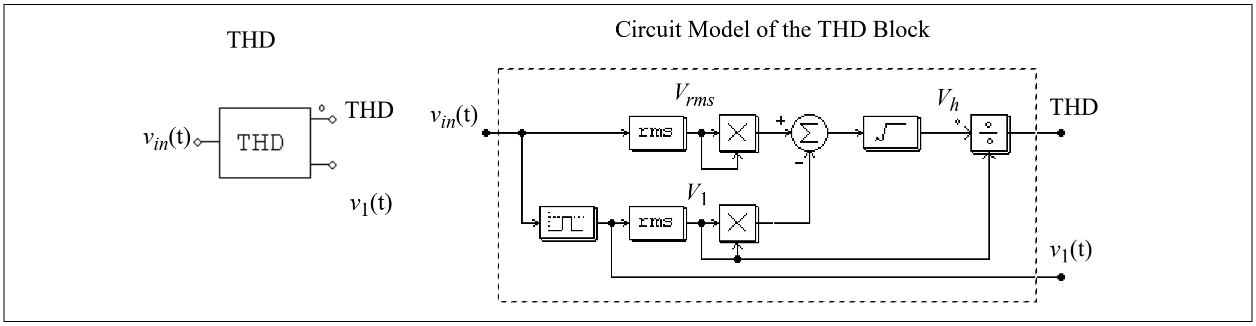
5.3.10 THD Block

The total harmonic distortion (THD) of an ac waveform that contains both the fundamental and harmonic components is defined as:

$$THD = \frac{V_h}{V_1} = \frac{\sqrt{V_{rms}^2 - V_1^2}}{V_1}$$

where V_1 is the fundamental component (rms), V_h is the harmonic rms value, and V_{rms} is the overall rms value of the waveform. The THD block is modelled as shown below.

Image:



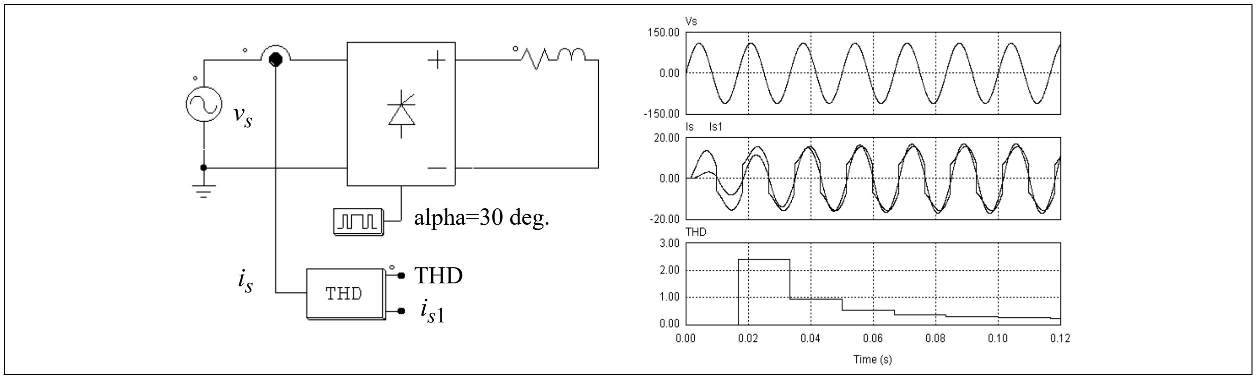
A second-order band-pass filter is used to extract the fundamental component. The center frequency and the passing band of the band-pass filter need to be specified.

Attributes:

Parameters	Description
Fundamental Frequency	Fundamental frequency of the input, in Hz
Passing Band	Passing band of the band-pass filter, in Hz

Example:

In the single-phase thyristor circuit below, a THD block is used to measure the THD of the input current. The delay angle of the thyristor bridge is chosen as 30^o. For the THD block, the fundamental frequency is set at 60 Hz and the passing band of the filter is set at 20 Hz. The simulation results are shown on the right.

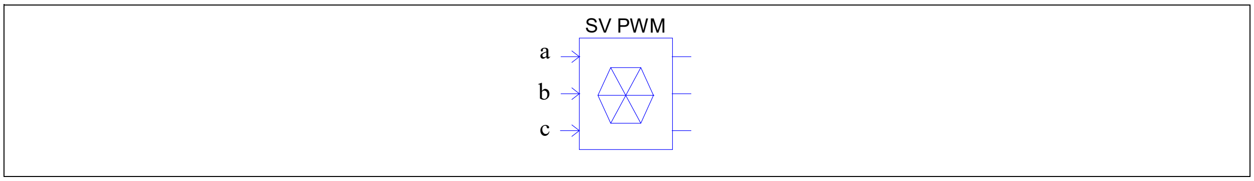


One of the THD block output is the input current fundamental component i_{s1}. By comparing the phase difference between the input voltage v_s and the current i_{s1}, one can calculate the input displacement power factor. This, together with the THD value, can be used to calculate the input power factor.

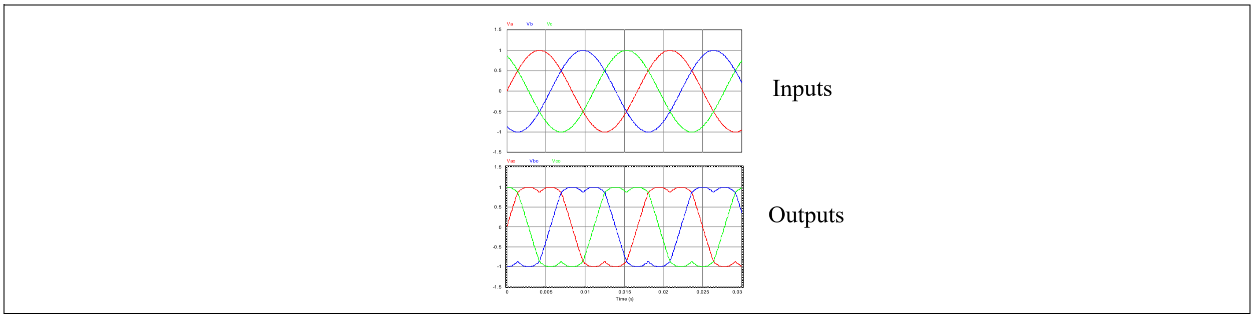
5.3.11 Space Vector PWM

A space vector PWM block is used in a carrier-wave based PWM scheme. It transforms 3-phase input signals to achieve the effect of space vector PWM.

Image:



When a 3-phase sinusoidal signal is applied to the inputs of the block, the output waveforms are shown below.

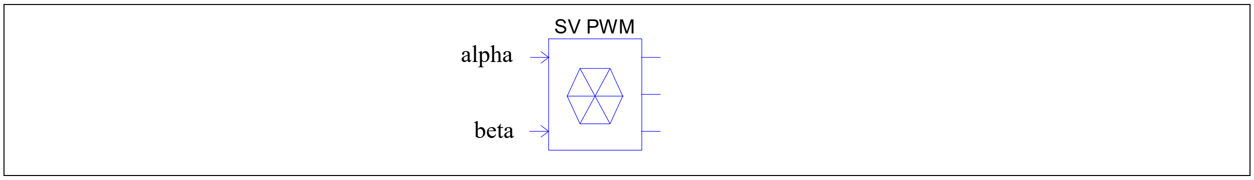


The fundamental component of the output is 1.155 times the value of the input.

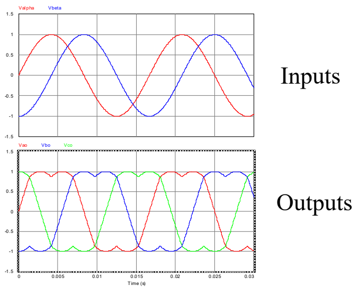
5.3.12 Space Vector PWM (alpha/beta)

The space vector PWM (alpha/beta) block is used in a carrier-based PWM scheme. It transforms two-phase orthogonal alpha/beta input signals to achieve the effect of space vector PWM.

Image:



When two sinusoidal signals, with the beta input lagging the alpha input by 90 deg., are applied to the inputs of the block, the output waveforms are shown below.



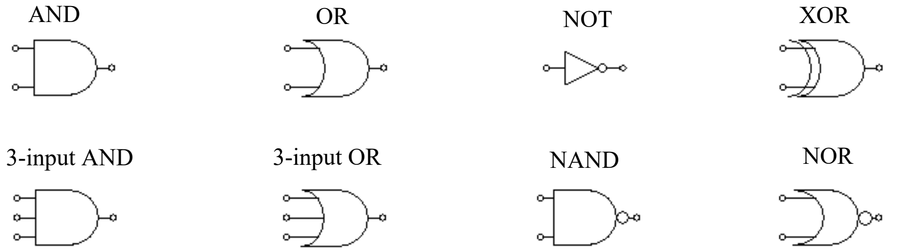
The fundamental component of the output is 1.155 times the value of the input.

5.4 Logic Components

5.4.1 Logic Gates

Basic logic gates are AND, OR, XORGATE (exclusive-OR), NOT, NAND, and NOR gates.

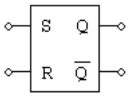
Images:



5.4.2 Set-Reset Flip-Flop

There are two types of set-reset flip-flops. One is edge-triggered and the other is level-triggered.

Image:



Attribute:

Parameter	Description
Trigger Flag	Trigger flag (0: edge-triggered; 1: level-triggered)

An edge-triggered flip-flop only changes the states at the rising edge of the set/reset input. The truth table of an edge-triggered flip-flop is:

S	R	Q	Q
0	0	no change	
0	↑	0	1
↑	0	1	0
↑	↑	not used	

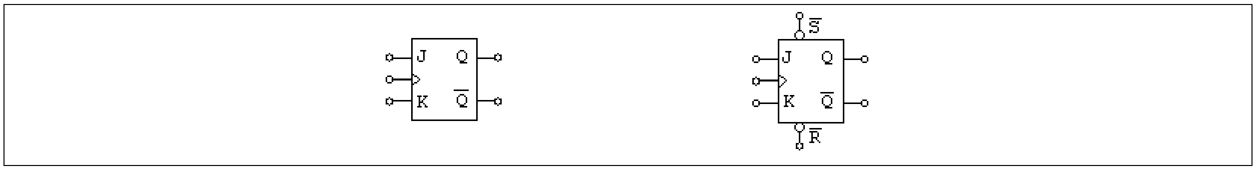
A level-triggered flip-flop, on the other hand, changes the states based on the input level. The truth table of a level-triggered set-reset flip-flop is:

S	R	Q	Q
0	0	no change	
0	1	0	1
1	0	1	0
1	1	not used	

5.4.3 J-K Flip-Flops

Two types of J-K flip-flops are provided: One without the set/reset inputs, and the other with the set input ($\overline{S}$) and reset input ($\overline{R}$). For the element without the set and reset inputs, it is assumed that both set/reset inputs are high (1).

Images:



A J-K flip-flop is positive edge triggered, and the truth table is:

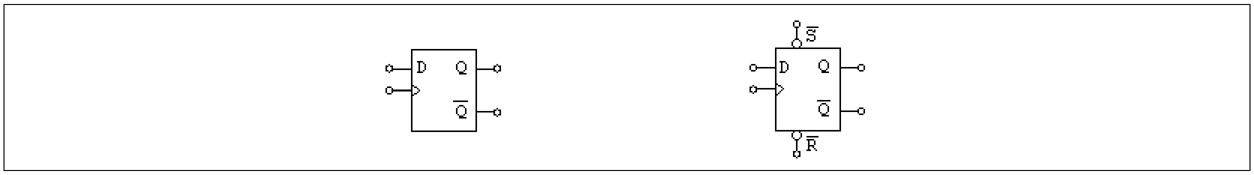
S	R	J	K	Clock	Q	Q
0	1	x	x	x	1	0
1	0	x	x	x	0	1
0	0	x	x	x	0	0
1	1	0	0	↑	no change	
1	1	0	1	↑	0	1
1	1	1	0	↑	1	0
1	1	1	1	↑	Toggle	

x: Do not care

5.4.4 D Flip-Flops

Two types of D flip-flops are provided: One without the set/reset inputs, and the other with the set input ($\overline{S}$) and reset input ($\overline{R}$). For the element without the set and reset inputs, it is assumed that both set/reset inputs are high (1).

Images:



A D flip-flop is positive edge triggered, and the truth table is:

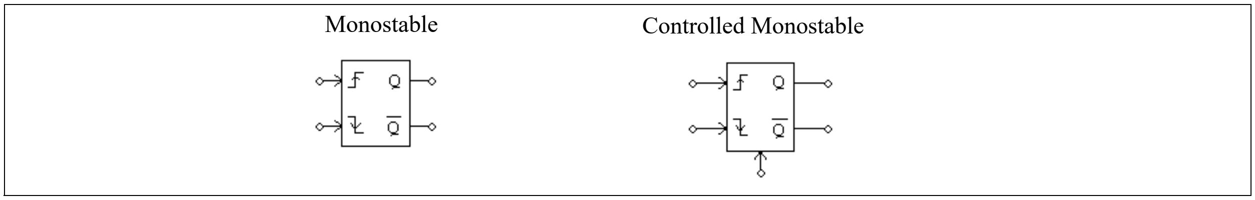
S	R	D	Clock	Q	Q
0	1	x	x	1	0
1	0	x	x	0	1
0	0	x	x	0	0
1	1	0	↑	0	1
1	1	1	↑	1	0

5.4.5 Monostable Multivibrator

In a monostable multivibrator, the positive (or negative) edge of the input signal triggers the monostable. A pulse, with the specified pulse width, will be generated at the output.

The output pulse width can be either fixed or adjusted through another input variable. The latter type of monostables is referred to as controlled monostables. Its on-time pulse width, in second, is determined by the control input.

Images:



Attribute:

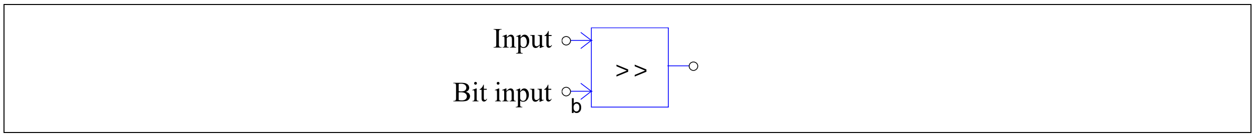
Parameter	Description
Pulse Width	On-time pulse width, in sec.

The input node at the bottom of the controlled monostable block is for the pulse width input.

5.4.6 Bit Shift

A bit shift block performs left shift or right shift depending on the sign of the bit input.

Image:



The lower left input node with the letter b is for the bit input. If the bit input is not an integer, it is rounded to the nearest integer.

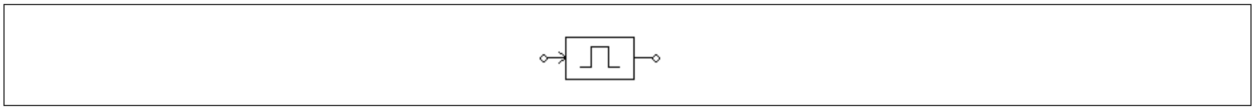
When the bit input is positive, the block performs left shift. For example, if the bit input is 3, the output is equal to the input multiplied by 2^3.

When the bit input is negative, the block performs right shift. For example, if the bit input is 3, the output is equal to the input divided by 2^3.

5.4.7 Pulse Width Counter

A pulse width counter measures the width of a pulse. The rising edge of the input activates the counter. At the falling edge of the input, the output gives the width of the pulse (in sec.). During the interval of two falling pulse edges, the pulse width counter output remains unchanged.

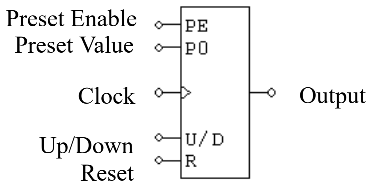
Image:



5.4.8 Up/Down Counter

An up/down counter increments or decrements by 1 at each rising edge of the clock.

Image:



Attribute:

Parameter	Description
No. of Bits	Number of bits N

When the Up/Down input is 0, the counter decrements, and when the Up/Down input is 1, the counter increments.

The Reset input resets the counter to 0 when it is high (1). The Preset Enable input sets the counter to the preset value when it is high.

The truth table of the counter is:

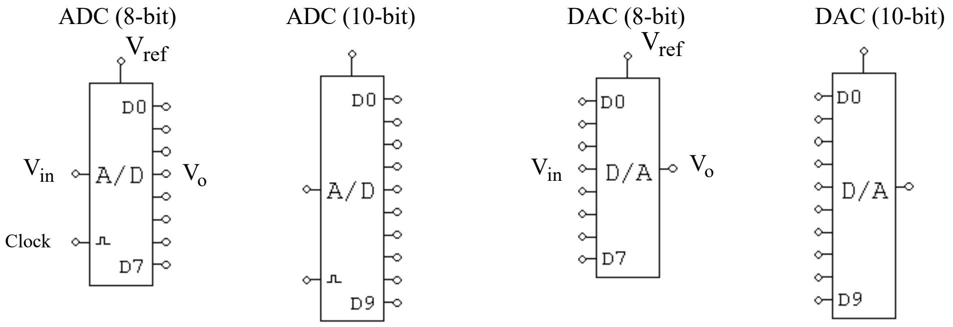
Up/Down	Preset Enable	Reset	Clock	Action
x	0	0	x	No count
1	0	0	↑	Count up
0	0	0	↑	Count down
x	1	0	x	Preset
x	x	1	x	Reset

x: Do not care

5.4.9 A/D and D/A Converters

A/D and D/A converters perform analog-to-digital and digital-to-analog conversion. Both 8-bit and 10-bit converters are provided.

Images:



Let *N* be the number of bits. The output of the A/D converter is calculated as:

$$V_o = \frac{2^N}{V_{ref}} \cdot V_{in}$$

The output will be clamped to 2^N-1.

For example, if V_{ref} = 5 V, V_{in} = 3.2 V, N = 8 bits, then

V_o = 256/5*3.2 = 163.84 = 10100011 (binary)

The output of the D/A converter is calculated as:

$$V_o = \frac{V_{ref}}{2^N} \cdot V_{in}$$

For example, if V_{ref} = 5 V, V_{in} = 10100011 (binary) = 163, N = 8 bits, then

V_o = 163/256*5 = 3.1836

5.5 Digital Control Module

The Digital Control Module is an add-on module to the basic PSIM program. It provides discrete elements, such as zero-order hold, z-domain transfer function blocks, digital filters, etc., for digital control system simulation.

In contrary to a s-domain circuit which is continuous, a z-domain circuit is discrete, and the calculation is only performed at the discrete sampling points. There is no calculation between two sampling points.

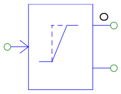
5.5.1 Motor Control Blocks

Elements in this subdivision are for digitally controlled motor drive systems. These blocks have similar functionality as the corresponding function blocks in the TI Digital Motor Control (DMC) library. Please refer to Chapter 11 of the *SimCoder User Manual* for more information.

5.5.1.1 Ramp Control

This block produces a ramp output when the input has a step change.

Image:



Attributes:

Parameters	Description
Upper Limit	Maximum ramp output, in per unit.
Lower Limit	Minimum ramp output, in per unit
Step Duration	Number of sampling period N at each step. It must be an integer equal or greater than 1.
Step Size	The maximum ramp step change Vstep at the output, in per unit.
Sampling Frequency	Sampling frequency, in Hz

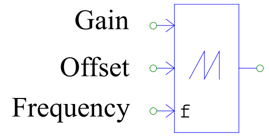
In the image, the output node with a small dot is the ramp output, and the other output node is the flag indicating that the output equals the input.

If the input is greater than the output, the output will increment by Vstep after a duration of N sampling periods. If the input is less than output, the output will decrement by Vstep after a duration of N sampling periods. If the input is equal to the output, the flag will be set to 1. Otherwise it will be 0.

5.5.1.2 Ramp Generator

This block generates a ramp output based on input settings.

Image:

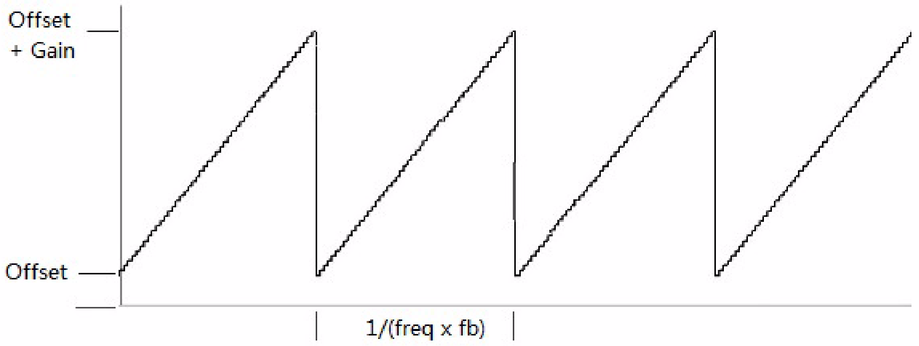


Attributes:

Parameters	Description
Base Frequency	Base frequency fb, in Hz.
Sampling Frequency	Sampling frequency, in Hz

In the image, the first input is the Gain; the second input is the Offset; and the third input is the desired frequency freq for output ramp.

The ramp output waveform is shown below.

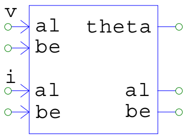


The output ramp frequency is the multiple of the input value and the base frequency fb. The output range is limited from -1 to +1. Therefore, the range of Offset and Gain must be limited as: Offset + Gain <= 1.

5.5.1.3 Sliding Mode Observer (PMSM)

The function of this sliding mode observer block is to estimate the rotor position of a permanent-magnet synchronous motor (PMSM). This block behaviors in the same way as the SMOPOS block of the TI DMC Library.

Image:



Attributes:

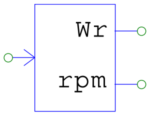
Parameters	Description
Base Voltage	Motor base phase voltage, in V
Base Current	Motor base phase current, in A

Stator Resistance	Motor stator resistance, in Ohm
Stator Inductance	Motor stator inductance, in H
Gain Kslide	Sliding mode control gain
Gain Kslf	Sliding mode control filter gain
Coefficient E0	Coefficient to limit the error integral saturation. The default value is 0.5.
Sampling Frequency	Sampling frequency, in Hz

5.5.1.4 Speed Calculator

This block calculates the motor speed based on the rotor angle theta.

Image:



Attributes:

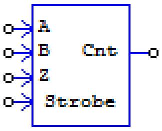
Parameters	Description
Base Frequency fb	Base frequency, in Hz
Cut-off Frequency	First-order low-pass filter cut-off frequency, in Hz
Number of Poles	Number of poles of the motor
Sampling Frequency	Sampling frequency, in Hz

The input signal *theta* is the motor rotor electrical angle, in per unit. The output Wr is the estimated motor speed, in per unit. The output rpm is the estimated motor speed, in rpm. A first-order low-pass filter is used at the output variables.

5.5.1.5 Encoder

This block provides the function of the encoder with the ABZ signals.

Image:



Attributes:

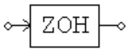
Parameters	Description
Counting Direction	Counting direction (0: counting up; 1: counting down)
Z Signal Polarity	Trigger polarity of the Z signal (0: active high; 1: active low)
Encoder Resolution	Resolution of the encoder. If it is 0, the encoder counter will keep on counting and will not reset. If, for example, the resolution is 4096, the counter will be reset to 0 after it reaches 4095.
Sampling Frequency	Sampling frequency, in Hz

The output of the encoder gives the counter value.

5.5.2 Zero-Order Hold

A zero-order hold samples the input at the point of sampling. The output remains unchanged between two sampling points.

Image:



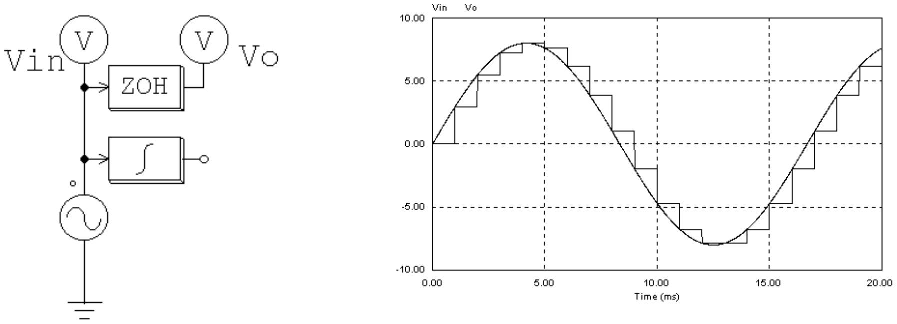
Attribute:

Parameter	Description
Sampling Frequency	Sampling frequency of the zero-order hold, in Hz

Like all other discrete elements, the zero-order hold has a free-running timer which determines the moment of sampling. The sampling moment is synchronized with the origin of the simulation time. For example, if the zero-order hold has a sampling frequency of 1000 Hz, the input will be sampled at 0, 1 msec., 2 msec., 3 msec., and so on.

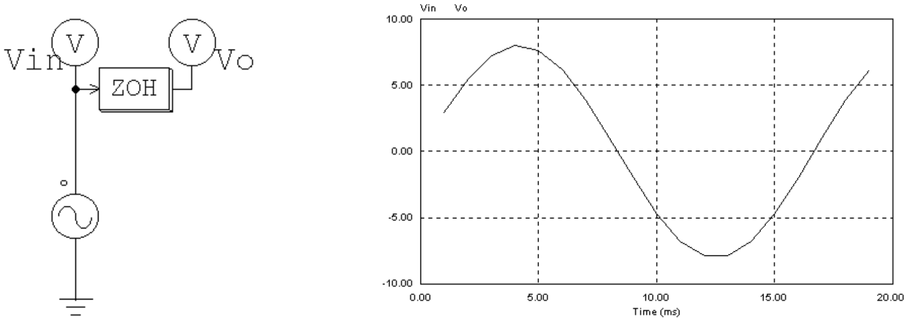
Example:

In the following circuit, the zero-order hold sampling frequency is 1000 Hz. The input and output waveforms are shown on the left.



Note that in above circuit, a continuous-domain integrator is also connected to the input sine source. This makes it a mixed continuous-discrete circuit, and a simulation time step selected for the continuous circuit will be used. With this time step, the familiar staircase-like waveform can be observed at the zero-order hold output.

Without the integrator, the circuit becomes a discrete circuit. Since only the calculation at the discrete sampling points is needed, the simulation time step will be equal to the sampling period, and only the results at the sampling points are available. The waveforms, as shown below, appear continuous. In fact the waveforms are discrete, and the connection between two sampling points makes it look like continuous.



5.5.3 z-Domain Transfer Function Block

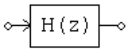
A z-domain transfer function block is expressed in polynomial form as:

$$H(z) = \frac{b_0 \cdot z^N + b_1 \cdot z^{N-1} + \ldots + b_{N-1} \cdot z + b_N}{a_0 \cdot z^N + a_1 \cdot z^{N-1} + \ldots + a_{N-1} \cdot z + a_N}$$

If $a_0 = 1$, the expression $Y(z) = H(z) * U(z)$ can be expressed in difference equation as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \ldots + b_N \cdot u(n-N) -$$
$$[a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + \ldots + a_N \cdot y(n-N)]$$

Image:



Attributes:

Parameters	Description
Order N	Order N of the transfer function
Coeff. $b_0 ... b_N$	Coefficients of the numerator (from b_0 to b_N)
Coeff. $a_0 ... a_N$	Coefficients of the denominator (from a_0 to a_N)
Sampling Frequency	Sampling frequency, in Hz

Example:

The following is a second-order transfer function:

$$H(z) = \frac{400.e^3}{z^2 + 1200 \cdot z + 400.e^3}$$

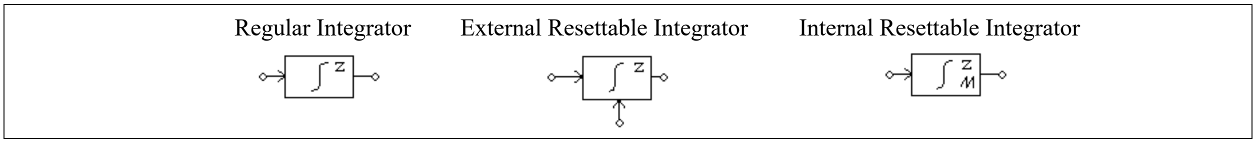
Assuming a sampling frequency of 3 kHz, the specification will be:

Order N	2
Coeff. $b_0 ... b_N$	0. 0. 400.e3
Coeff. $a_0 ... a_N$	1. 1200. 400.e3
Sampling Frequency	3000.

5.5.3.1 Integrator

There are three types of integrators: regular integrator, external resettable integrator, and internal resettable integrator.

Images:


Regular Integrator
External Resettable Integrator
Internal Resettable Integrator
z
z
z
M

Attribute:

Parameters	Description
Algorithm Flag	Flag for integration algorithm 0: trapezoidal rule 1: backward Euler 2: forward Euler
Initial Output Value	Initial output value
Reset Flag	Reset flag (0: edge reset; 1: level reset) (for *external resettable integrator* only)
Lower Output Limit	Lower limit of the output (for *internal resettable integrator* only)
Upper Output Limit	Upper limit of the output (for *internal resettable integrator* only)
Sampling Frequency	Sampling frequency, in Hz

The output of an external resettable integrator can be reset by an external control signal (at the bottom of the block). With the edge reset (reset flag = 0), the integrator output is reset to zero at the rising edge of the control signal. With the level reset (reset flag = 1), the integrator output is reset to zero as long as the control signal is high (1).

The output of an internal resettable integrator is reset to 0 whenever the output reaches either the lower limit or the upper limit. The integrator works in the same way as the external resettable integrator with the edge reset, except that users do not need to set up the external reset circuit in this case.

If we define $u(t)$ as the input, $y(t)$ as the output, T_s as the sampling period, and $H(z)$ as the discrete transfer function, the input-output relationship of an integrator can be expressed under different integration algorithms as follows.

With trapezoidal rule:

$$H(z) = \frac{T_s}{2} \cdot \frac{z+1}{z-1}$$

$$y(n) = y(n-1) + \frac{T_s}{2} \cdot (u(n) + u(n-1))$$

With backward Euler:

$$H(z) = T_s \cdot \frac{z}{z-1}$$

$$y(n) = y(n-1) + T_s \cdot u(n)$$

With forward Euler:

$$H(z) = T_s \cdot \frac{1}{z-1}$$

$$y(n) = y(n-1) + T_s \cdot u(n-1)$$

5.5.3.2 Differentiator

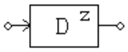
The transfer function of a discrete differentiator is:

$$H(z) = \frac{1}{T_s} \cdot \frac{z-1}{z}$$

where T_s is the sampling period. The input-output relationship can be expressed in difference equation as:

$$y(n) = \frac{1}{T_s} \cdot (u(n) - u(n-1))$$

Image:



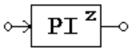
Attribute:

Parameter	Description
Sampling Frequency	Sampling frequency, in Hz

5.5.3.3 Digital PI Controller

A digital PI controller is defined below.

Image:



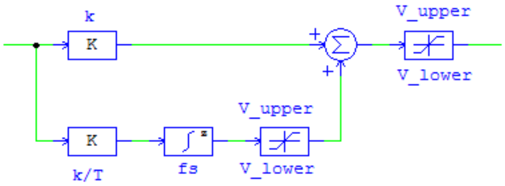
Attribute:

Parameters	Description
Gain	Gain k of the PI controller
Time Constant	Time constant T of the PI controller
Lower Output Limit	Lower limit V_lower of the output
Upper Output Limit	Upper limit V_upper of the output
Sampling Frequency	Sampling frequency f_s, in Hz

The transfer function of an analog PI controller in s-domain is defined as:

$$G(s) = k \cdot \frac{1 + sT}{sT}$$

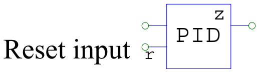
The digital PI controller is obtained by discretizing the analog PI controller using the backward Euler method. The implementation of the controller is shown below:



5.5.3.4 Digital PID Controller with Reset

The digital PID controller with reset is defined below. Depending on the setting of the gains, the block can be used as a PI or PD controller.

Image:



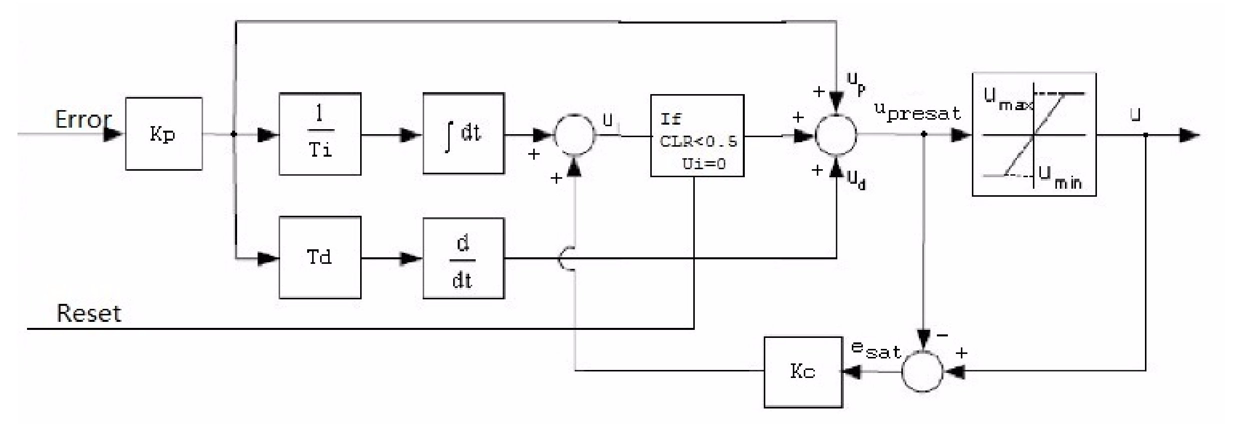
Attribute:

Parameters	Description
Sampling Frequency	Sampling frequency, in Hz
Proportional Gain	Proportional loop gain Kp
Integral Gain	Integral gain Ki
Derivative Gain	Derivative gain Kd
Integral Correction Gain	Integral correction gain Kc
Upper Saturation	Maximum output limit, in per unit
Lower Saturation	Minimum output limit, in per unit

In the image, the input with "r" is for the reset signal. The other input, please note, is the error between the reference and the feedback. When the value of the input Reset is greater than 0.5, integral output is clamped to zero. This function is useful to prevent the integrator from entering saturation before the feedback signal is available.

In the PID controller, the differential equation is transformed to the difference equation by means of the backward Euler approximation.

The block diagram of this conventional PID controller with anti-windup correction is shown below.



Let the sampling period be Ts, the integral gain and derivative gain are defined as:

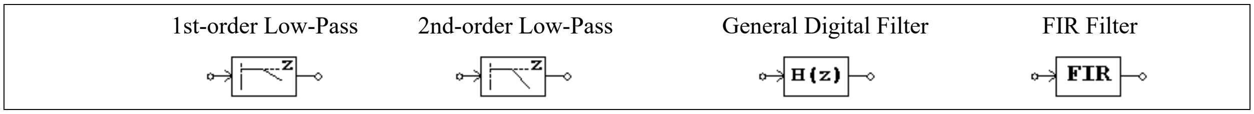
$$K_i = \frac{T_s}{T_i}$$

$$K_d = \frac{T_d}{T_s}$$

5.5.3.5 Digital Filters

Four digital filter blocks are provided: 1st-order low-pass filter, 2nd-order low-pass filter, general digital filter, and general finite impulse response (FIR) filter. For both general filters, the filter coefficients can either be entered directly through the element property window, or be specified through a text file.

Images:



Attributes:

For *1st-order low-pass filter* and *2nd-order low-pass filter*:

Parameters	Description
Gain	Gain k of the filter
Cut-off Frequency	Cut-off frequency f_c ($f_c = \frac{\omega_c}{2\pi}$), in Hz
Damping Ratio	Damping ratio ξ of the filter (for 2nd-order low-pass filter only)
Sampling Frequency	Sampling frequency f_s, in Hz

For *General Digital Filter* or *FIR Filter* that reads the coefficients directly:

Parameters	Description
Order N	Order N of the transfer function
Coeff. $b_0 ... b_N$	Coefficients of the numerator (from b_0 to b_N)
Coeff. $a_0 ... a_N$	Coefficients of the denominator (from a_0 to a_N)
Sampling Frequency	Sampling frequency f_s, in Hz

For *General Digital Filter* or *FIR Filter* that reads the coefficients from a text file:

Parameters	Description
File for Coefficients	Name of the file storing the filter coefficients
Sampling Frequency	Sampling frequency f_s, in Hz

The transfer functions of the low-pass filters are listed below.

For first-order low-pass analog filter in s-domain:

$$G(s) = k \cdot \frac{\omega_c}{s + \omega_c}$$

For second-order low-pass analog filter in s-domain:

$$G(s) = k \cdot \frac{\omega_c^2}{s^2 + 2\xi\omega_c s + \omega_c^2}$$

The 1st-order and 2nd-order digital filters are obtained by discretizing the analog filter using the Backward Euler method.

The transfer function of a general digital filter is expressed in polynomial form as:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + ... + b_{N-1} \cdot z^{-(N-1)} + b_N \cdot z^{-N}}{a_0 + a_1 \cdot z^{-1} + ... + a_{N-1} \cdot z^{-(N-1)} + a_N \cdot z^{-N}}$$

If $a_0 = 1$, the output y and input u can be expressed in difference equation form as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \ldots + b_N \cdot u(n-N) -$$
$$[a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + \ldots + a_N \cdot y(n-N)]$$

If the denominator coefficients $a_0..a_N$ are not zero, this type of filter is called infinite impulse response (IIR) filter.

The transfer function of a FIR filter is expressed in polynomial form as:

$$H(z) = b_0 + b_1 \cdot z^{-1} + \ldots + b_{N-1} \cdot z^{-(N-1)} + b_N \cdot z^{-N}$$

If $a_0 = 1$, the output y and input u can be expressed in difference equation form as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \ldots + b_N \cdot u(n-N)$$

The coefficient file for block *Digital Filter* (*file*) and *FIR Filter* (*file*) has the following format:

For *FIR Filter* (*file*):

N
b_0
b_1
... ...
b_N

For *Digital Filter* (*file*), the format can be either one of the following:

N	or	N
b_0		b_0, a_0
b_1		b_1, a_1
... ...		... ...
b_N		b_N, a_N
a_0		
a_1		
... ...		
a_N		

Example:

To design a 2nd-order low-pass Butterworth digital filter with the cut-off frequency fc = 1 kHz, assuming the sampling frequency fs = 10 kHz, using MATLAB, we have:

Nyquist frequency fn = fs / 2 = 5 kHz
Normalized cut-off frequency fc* = fc/fn = 1/5 = 0.2
[B,A] = butter (2, fc*)

which will give:

B = [0.0201 0.0402 0.0201] = [b_0 b_1 b_2]
A = [1 -1.561 0.6414] = [a_0 a_1 a_2]

The transfer function is:

$$H(z) = \frac{0.0201 + 0.0402 \cdot z^{-1} + 0.0201 \cdot z^{-2}}{1 - 1.561 \cdot z^{-1} + 0.6414 \cdot z^{-2}}$$

The input-output difference equation is:

$$y(n) = 0.0201 \cdot u(n) + 0.0402 \cdot u(n-1) + 1.561 \cdot y(n-1) - 0.6414 \cdot y(n-2)$$

The parameter specification of the filter in PSIM will be:

Order *N*	2
Coeff. $b_0 ... b_N$	0.0201 0.0402 0.0201
Coeff. $a_0 ... a_N$	1. -1.561 0.6414
Sampling Frequency	10000.

If the coefficients are stored in a file, the file content will be:

```
2
0.0201
0.0402
0.0201
1.
-1.561
0.6414
```

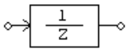
Or the file can also have the content as follows:

```
2
0.0201,  1
0.0402,  -1.561
0.0201, 0.6414
```

5.5.4 Unit Delay

A unit delay block provides one sampling period delay to the input.

Image:



Attribute:

Parameter	Description
Initial Output Value	Initial value of the output of the block
Sampling Frequency	Sampling frequency, in Hz

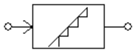
The difference between the unit delay block and the time delay block is that the unit delay block is a discrete element and it delays the sampled points by one sampling period, whereas the time delay block is a continuous element and it delays the whole waveform by the delay time specified.

5.5.5 Quantization Blocks

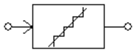
Quantization blocks simulate the quantization process of an A/D converter. Two quantization blocks are provided: One with 1 LSB (least significant bit) quantization error, and the other with input offset with 0.5 LSB quantization error.

Images:

Quantization Block



Quantization Block (with offset)

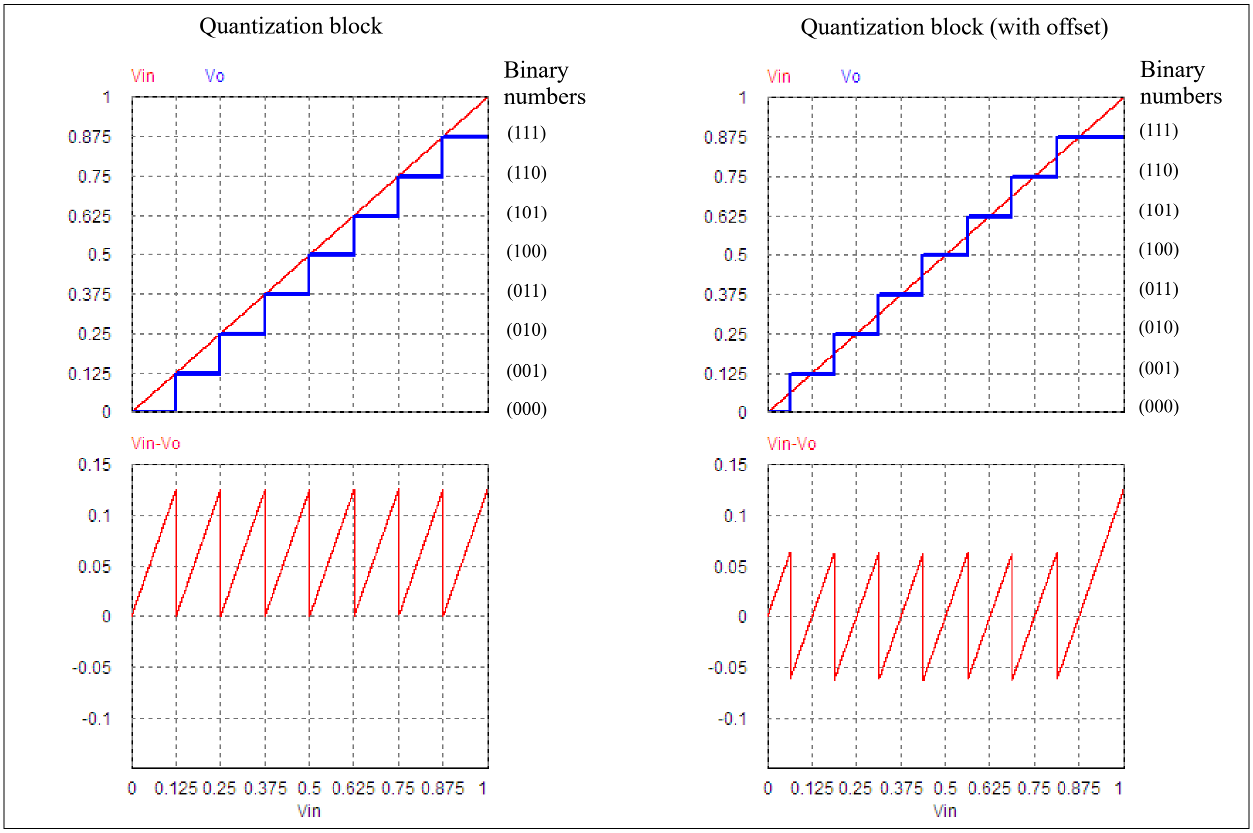


Attributes:

Parameters	Description
No. of Bits	Number of bits N
Vin_min	Lower limit of the input value $V_{in,min}$
Vin_max	Upper limit of the input value $V_{in,max}$
Vo_min	Lower limit of the output value $V_{o,min}$
Vo_max	Output value $V_{o,max}$ that corresponds to the input value $V_{in,max}$
Sampling Frequency	Sampling frequency, in Hz

To understand how the quantization blocks work, we will look at a special case where $N = 3$, $V_{in,min} = 0$, and $V_{in,max} = 1$. The input will be divided into 2^3 (or 8) stages. Depending on the input level, the output will be one of the 3-bit discrete numbers 000, 001, 010, 011, 100, 101, 110, and 111. The output will have 2^3 (or 8) levels, and 2^3-1 (or 7) steps. If the binary number 000 corresponds to the decimal number 0, and the binary number 111 corresponds to the decimal number 0.875, and $V_{o,min} = 0$, and $V_{o,max} = 1$, we can plot the waveforms of the input, output, and the quantization error V_{in}-V_o of the two quantization blocks as below.

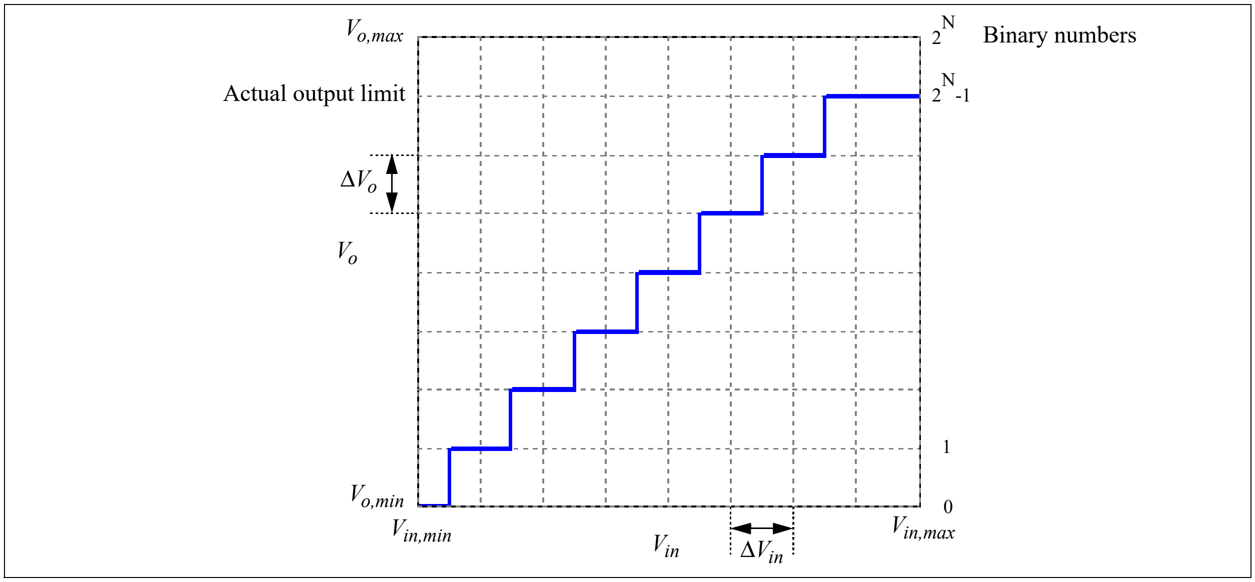
The waveforms on the left are from the quantization block without the offset, and the waveforms on the right are from the quantization block with the offset.



As shown from the waveforms on the left, the quantization error of the quantization block is from 0 to 1 LSB (least significant bit), or $1/2^3$ (or 0.125).

For the quantization block with offset, a value of 0.5*LSB is added to the input before performing the quantization. This reduces the quantization error to -0.5 LSB to +0.5 LSB (except when the input is close to the maximum value), as shown from the waveforms on the right.

The figure below shows the input-output relationship of a general quantization block with offset.



The number of bits determines the quantization resolution. The input range, $V_{in,max}$ - $V_{in,min}$, is divided into 2^N

stages, with the width of each stage as:

$$\Delta V_{in} = \frac{V_{in,\,max} - V_{in,\,min}}{2^N}$$

except for the 1st stage where the width is 0.5*ΔV_{in}, and the last stage where the width is 1.5*ΔV_{in}. Note that for the quantization block without the offset, the width of the 1st and last stage is ΔV_{in}.

If the input falls in the k_{th} stage, the output will be calculated as:

$$V_o = V_{o,\,min} + (k-1) \cdot \Delta V_o$$

where k is from 1 to 2^N, and the output step is calculated as:

$$\Delta V_o = \frac{V_{o,\,max} - V_{o,\,min}}{2^N}$$

Note that the value $V_{o,max}$ corresponds to the output value when the input is at $V_{in,max}$. But because of the quantization, the output can be represented in only N levels, from 0 to 2^N-1. As a result, the actual limit of the output is not $V_{o,max}$, but $V_{o,max}$ - ΔV_o, as shown in the figure above.

Example:

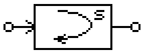
For a quantization block with the offset, let N = 3, $V_{in,min}$ = 0, $V_{in,max}$ = 1, $V_{o,min}$ = 0, and $V_{o,min}$ = 1. We have $\Delta V_{in} = \Delta V_o$ = 1/8. If V_{in} = 0.25, it is in the 3rd stage (k = 3), and it corresponds to V_o = 0 + (3 - 1) * 1/8 = 0.25. If V_{in} = 0.6, it is in the 6th stage (k = 6), and it corresponds to V_o = 0 + (6 - 1) * 1/8 = 0.625.

5.5.6 Circular Buffers

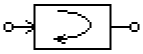
A circular buffer is a memory location that can store an array of data. Two types of circular buffers are provided in PSIM: One with a single output, and the other with a vector output.

Images:

Circular Buffer (single output)



Circular Buffer (vector output)



Attributes:

Parameters	Description
Buffer Length	The length of the buffer
Sampling Frequency	Sampling frequency, in Hz

A circular buffer stores data in a buffer. When the pointer reaches the end of the buffer, it will start again from the beginning.

The circular buffer with single output is a type of first-in-first-out memory storage device, and the output is equal to the value being pushed out.

The output of the circular buffer with vector output is a vector array with the length equal to the buffer length. To access each memory location, use the memory read block.

Example:

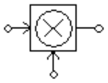
If a circular buffer has a buffer length of 4 and a sampling frequency of 10 Hz, we have the buffer storage at different time as follows:

Time	Input	Value at Memory Location				Output (for single-output buffer)
		1	2	3	4	
0	0.11	0.11	0	0	0	0
0.1	0.22	0.11	0.22	0	0	0
0.2	0.33	0.11	0.22	0.33	0	0
0.3	0.44	0.11	0.22	0.33	0.44	0
0.4	0.55	0.55	0.22	0.33	0.44	0.11
0.5	0.66	0.55	0.66	0.33	0.44	0.22
... ...						

5.5.7 Convolution Block

A convolution block performs the convolution of two input vectors. The output is also a vector.

Image:



Let the two input vectors be:

$$A = [\ a_m\ a_{m-1}\ a_{m-2}\ ...\ a_1]$$

$$B = [\ b_n\ b_{n-1}\ b_{n-2}\ ...\ b_1]$$

We have the convolution of *A* and *B* as:

$$C = A \otimes B = \left[c_{m+n-1}\ c_{m+n-2}\ ...\ c_1\right]$$

where

$$c_i = \Sigma[\ a_{k+1} * b_{j-k}], \quad k=0, ..., m+n-1; j=0, ..., m+n-1; i=1, ..., m+n-1$$

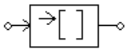
Example:

If *A* = [1 2 3] and *B* = [4 5], we have m = 3; n = 2; and the convolution of *A* and *B* is: *C* = [4 13 22 15].

5.5.8 Memory Read Block

A memory read block is used to read the value of a memory location of a vector.

Image:



Attribute:

Parameter	Description
Memory Index Offset	Offset from the starting memory location

A memory read block allows one to access the memory location of elements such as convolution block, vector array, and circular buffer. The index offset defines the offset from the starting memory location.

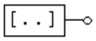
Example:

Let a vector be A = [2 4 6 8]. If index offset is 0, the memory read block output will be 2. If the index offset is 2, the output will be 6.

5.5.9 Data Array

This is a one-dimensional array. The output is a vector. The data are either entered directly (the element is called *Array* in the PSIM library) or specified in a file (the element is called *Array* (*file*) in the PSIM library).

Image:



Attributes:

Parameters	Description
Array Length	The length of the data array *N* (for the element *Array* only)
Values	Values of the array (for the element *Array* only)
File for Coefficients	Name of the file storing the array (for the element *Array* (*file*) only)

If the array is read from a file, the file will have the following format:

N
a_1
... ...
a_N

where *N* is the length of the array, and $a_1..a_N$ are the array values.

Example:

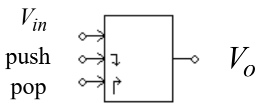
To define an array A = [2 4 6 8], we will have: Array Length = 4; Values = 2 4 6 8. If the array is to be read from a file, the file will be:

4
2.
4.
6.
8.

5.5.10 Stack

A stack is a first-in-last-out register.

Image:



Attribute:

Parameter	Description
Stack Depth	The stack depth

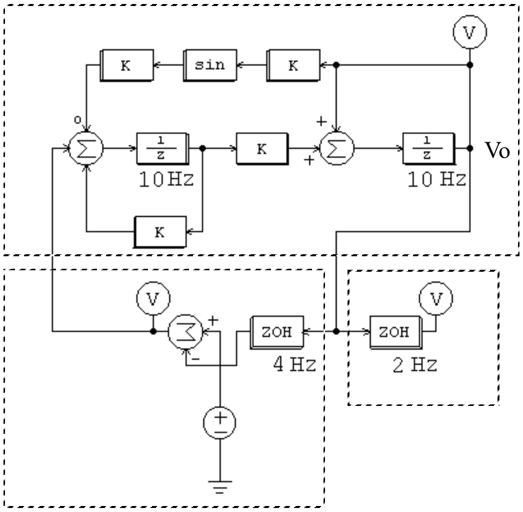
The rising edge triggers the push or pop action. When a pop action is performed and the stack is empty, the output remains unchanged. When a push action is performed and the stack is already full, the data at the bottom of the stack will be pushed out and will be lost.

5.5.11 Multi-Rate Sampling System

A discrete system can have more than one sampling rate. The following system is used to illustrate this.

The system below has 3 sections. The first section has a sampling rate of 10 Hz. The output, Vo, fed back to the system and is sampled at 4 Hz in the second section. In the third section, the output is displayed at a sampling rate of 2 Hz.

It should be noted that a zero-order hold must be used between two elements with different sampling rates.

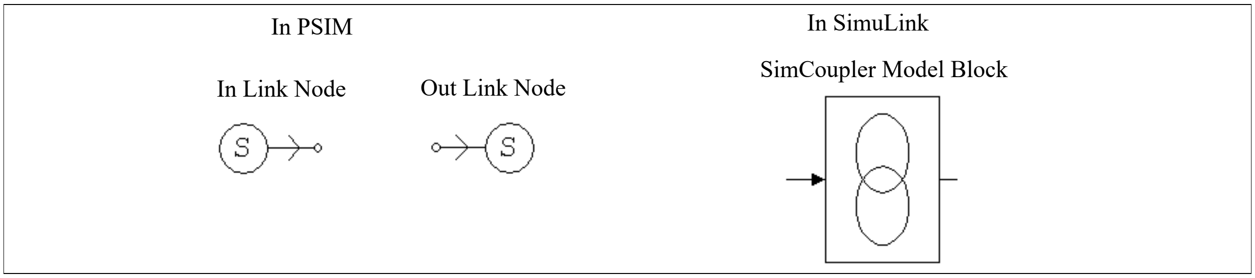


5.6 SimCoupler Module

The SimCoupler Module is an add-on module to the PSIM software. It provides interface between PSIM and Matlab/Simulink for co-simulation. With the SimCoupler Module, part of a system can be implemented and simulated in PSIM, and the rest of the system in Simulink. One can therefore make full use of PSIM's capability in power simulation and Matlab/Simulink's capability in control simulation in a complementary way.

The SimCoupler interface consists of two parts: the link nodes in PSIM, and the SimCoupler model block in Simulink. The images are shown below.

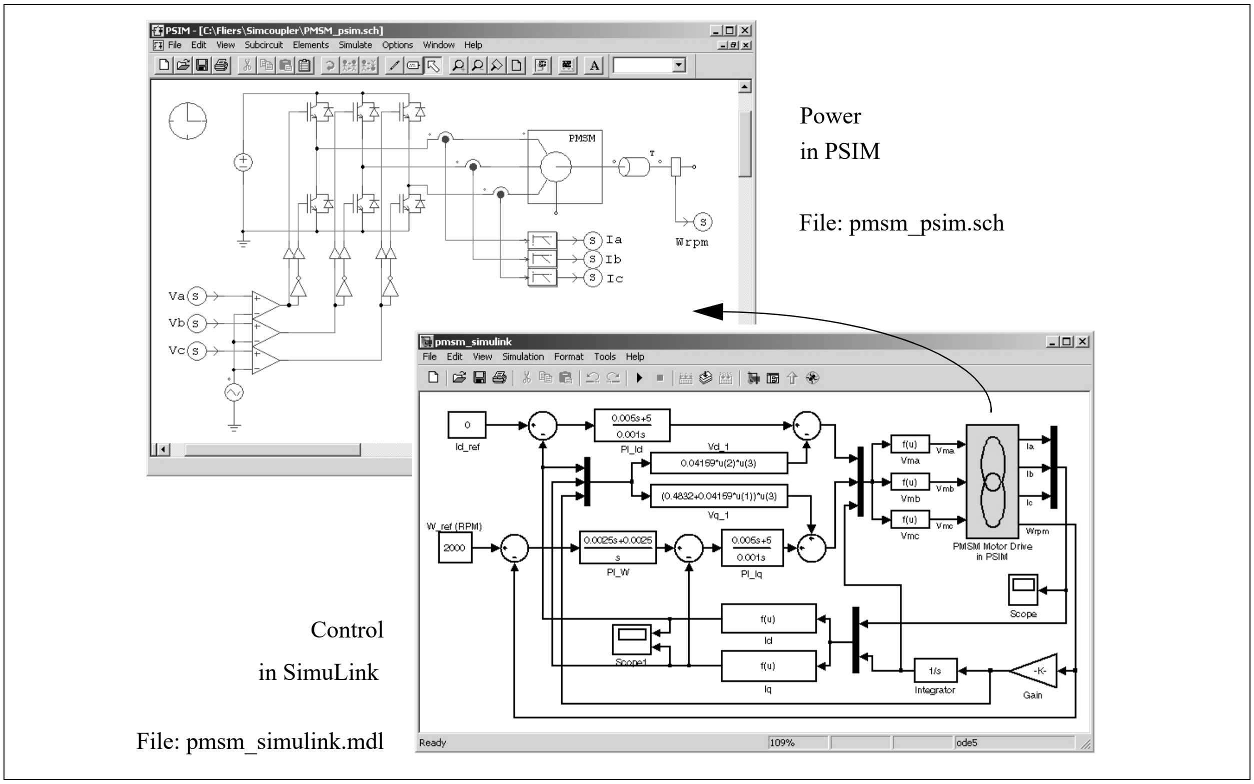
Images:



In PSIM, the In Link nodes receive values from Simulink, and the Out Link nodes send the values to Simulink. They are all control elements and can be used in the control circuit only. In Simulink, the SimCoupler model block is connected to the rest of the system through input/output ports.

5.6.1 Set-up in PSIM and Simulink

The use of the SimCoupler Module is easy and straightforward. As an example, the following shows a permanent-magnet synchronous motor (PMSM) drive system with the power stage implemented in PSIM, and the control in Simulink.



Power
in PSIM

File: pmsm_psim.sch

Control
in SimuLink

File: pmsm_simulink.mdl

The following are the steps to set up SimCoupler for PSIM-Matlab/Simulink co-simulation for the example above.

Adding the SimCoupler Block to the Simulink Library:

Run the program "SetSimPath.exe" to add the SimCoupler block to the Simulink library and set up the SimCoupler Module for co-simulation of PSIM and Matlab/Simulink. After the execution, the SimCoupler block will appear as "S-function SimCoupler" in the Simulink Library Browser.

Note that this step is necessary, otherwise Simulink will not be able to find PSIM. With this, it is also not necessary to manually add the PSIM folder to the Matlab path.

Also, this program needs to be run only once. It needs to be run again only if the PSIM folder or Matlab folder has changed.

In PSIM:

- After the rest of the power circuit is created, connect three Out Link nodes to the low-pass filters of Phase *A*, *B*, and *C* currents, and rename them as “Ia”, “Ib”, and “Ic”; and connect one Out Link node to the speed sensor output and rename it as “Wrpm”.
- Connect three In Link nodes to the positive inputs of the comparators, and rename them as “Va”, “Vb”, and “Vc”.
- Go to the **Simulate** menu, and select **Arrange SLINK Nodes**. A dialog window will appear. Arrange the order of the In Link nodes (SLINK In) and Out Link nodes (SLINK Out) to be the same as how the input/output ports would appear in the SimCoupler model block in Simulink (the order of the ports is from the top to the bottom). In this example, the order will be “Va”, “Vb”, and “Vc” for the In Link nodes, and “Ia”, “Ib”, “Ic”, and “Wrpm” for the Out Link nodes.
- Save the schematic file. In this example, we assume that the file is saved to “C:\PSIM\pmsm_psim.sch”.

In Simulink:

- Start Matlab.
- Launch Simulink. Open an existing file or create a new file. After the rest of the system is created, go to the menu "S-function SimCoupler" in the Simulink Library Browser, select the SimCoupler block, and place it on the schematic.
- In the PMSM example file, double click on the SimCoupler block, and click on the **Browser** button to locate and select the PSIM schematic file “C:\PSIM\pmsm_psim.sch”. Then click on **Apply**. The number of input and output ports of the SimCoupler model block will automatically match those defined in the PSIM netlist. In this case, there will be 3 input ports and 4 output ports.
- Go to the **Simulation** menu and select **Simulation Parameters**. Under **Solver Options**, set the **Type** to “Fixed-step”. Set **Fixed step size** to be the same as or close to PSIM’s time step. In this case, the time step is set to 0.1ms. More discussion on the selection of the solver option and the time step is given in the next section.
- The setup is now complete. Go to Simulink and start the simulation.

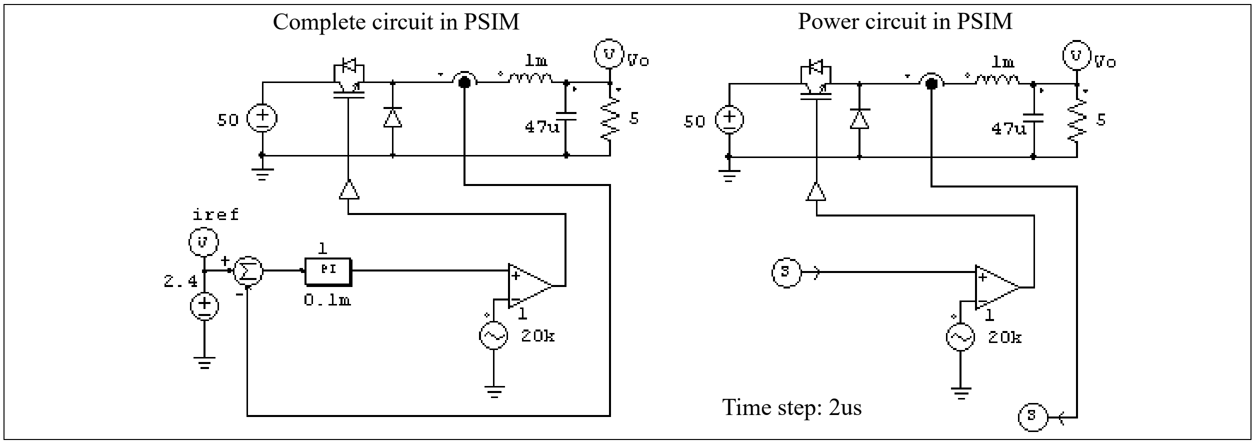
The SimCoupler Module supports Matlab/Simulink Release 13 and higher.

Please also note that when the SimCoupler model block is used in a feedback system in Simulink, the SimCoupler model block may be part of an algebraic loop (please refer to Matlab Help for more information on algebraic loops). Some versions of Matlab/Simulink can not solve a system containing algebraic loops, and other can solve the system but with degraded performance. To break an algebraic loop, place a memory block at each output of the SimCoupler model block. The memory block introduces one integration time step delay.

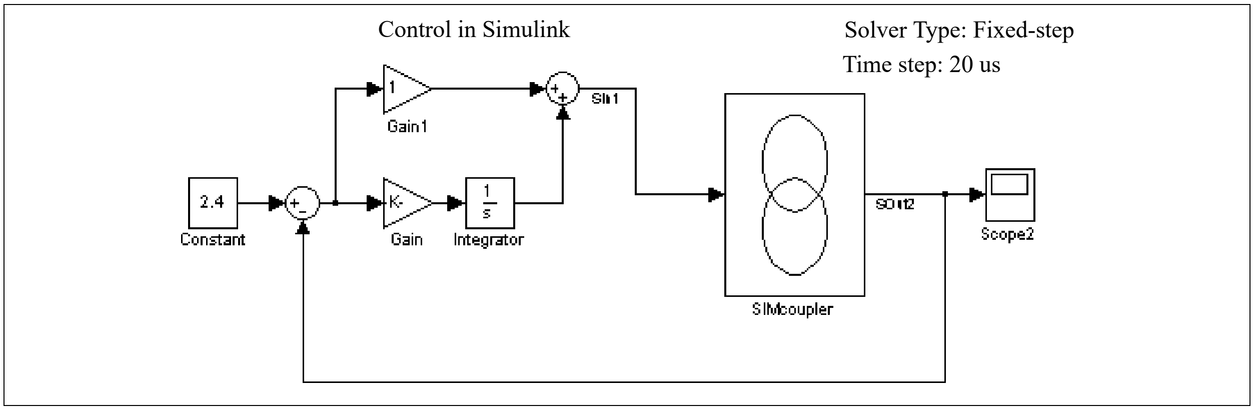
5.6.2 Solver Type and Time Step Selection in Simulink

There are certain restrictions on the selection of the solver type and the time step in Simulink when performing the co-simulation. To illustrate this, we use the following one-quadrant chopper circuit with average current mode control as an example.

The circuit on the left is all implemented and simulated in PSIM. The circuit on the right has the power stage implemented in PSIM, and the control implemented in Simulink. In both circuits, the PSIM time step is 2 us.

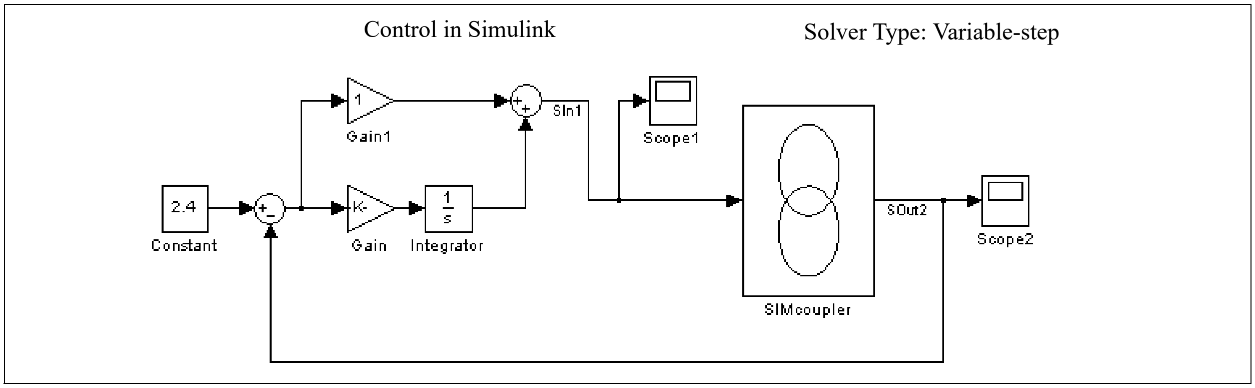


There are different ways of setting up Simulink to perform co-simulation. The recommend approach is to set the Solve Type to **Fixed-step** and define the **Fixed step size** to be the same or close to PSIM's time step. The figure below shows this option.

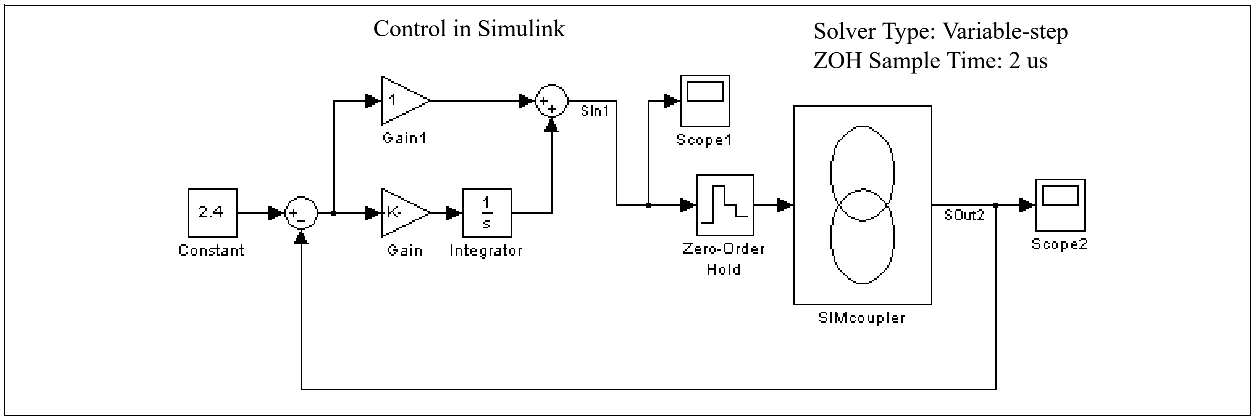


It is recommended that Simulink use the same time step as PSIM, although we have found that, even if the Simulink time step is slightly larger than PSIM time step, satisfactory results are obtained. In this case, for example, the time step is set to 20 us, 10 times larger than the PSIM time step.

If the Simulink Solver type is instead set to **Variable-step**, the simulation results will not be correct. The figure below shows this option.



When the Simulink Solver type is set to **Variable-step**, in order to obtain correct results, a zero-order-hold must be placed at the input of the SimCoupler model block. Moreover, the zero-order-hold sample time must be the same or close to PSIM time step. The figure below shows the configuration.



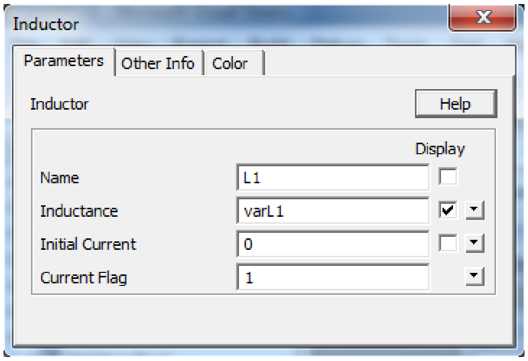
Control in Simulink

Solver Type: Variable-step
ZOH Sample Time: 2 us

Therefore, Simulink must be set up to have the Solver Type as **Fixed-step** with the time step the same or close to the PSIM time step, or if the Solver Type is **Variable-step**, a zero-order-hold must be used with the sample time the same or close to PSIM time step

5.6.3 Passing Parameters from Simulink to PSIM

One can set parameter values in Simulink, and pass them to PSIM. For example, to set the inductance of the inductor L1 in Simulink, change the "Inductance" to *varL1* in PSIM, as shown below.

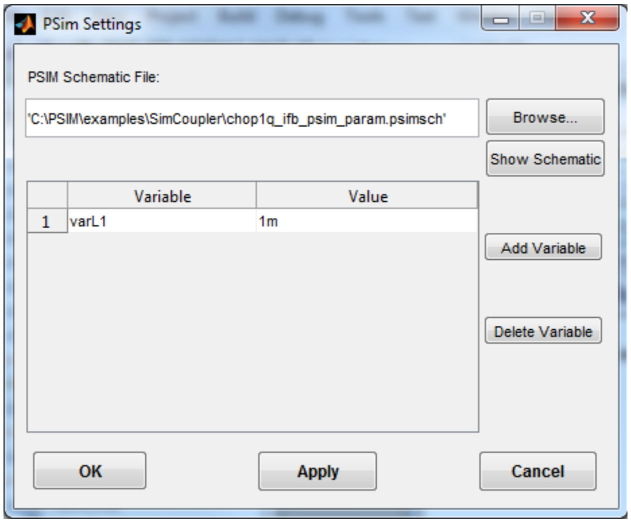


Inductor

Inductor

		Display
Name	L1	□
Inductance	varL1	☑
Initial Current	0	□
Current Flag	1	

In Simulink, double click on the SimCoupler block to open the property dialog, and click on **Add Variable**. Click on the newly added row in the list and type the variable name and value, as shown below.



PSim Settings

PSIM Schematic File:

'C:\PSIM\examples\SimCoupler\chop1q_ifb_psim_param.psimsch'

	Variable	Value
1	varL1	1m

The variable *varL1* will now be accessible from PSIM.

5.7 CosiMate Links

CosiMate links provide the link between PSIM and CosiMate, a co-simulation platform that supports various software. Through the links, one can perform co-simulation between PSIM and the software that CosiMate supports. For more information on CosiMate, please visit www.chiastek.com.

The links consist of two parts: Input Port and Output Port. An input port represents a control signal input from CosiMate to PSIM, and an output port represents a control signal output from PSIM to CosiMate. During co-simulation, data are exchanged between PSIM and other software through these ports.

Note that the links will be functional only if one has CosiMate installed. For more details on how to set up the links, please refer to relevant CosiMate documents.

5.8 Design Suite Blocks

This section contains blocks that consist of basic power and control elements for specific motor drive system applications.

The following control blocks are provided:

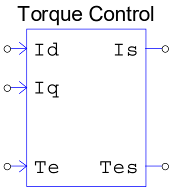
- Torque Control (PMSM)
- Dynamic Torque Limit Control (PMSM)
- Voltage Control (PMSM)
- DC-DC Charging Control
- DC-DC Discharging Control
- DC-DC Regeneration Control

These blocks are part of the HEV Design Suite.

5.8.1 Torque Control (PMSM)

The Torque Control (PMSM) block is for linear PMSM machines only. It is defined as follow.

Image:



Attributes:

Parameters	Description
Ld (d-axis inductance)	d-axis inductance of the machine, in H
Lq (q-axis inductance)	q-axis inductance of the machine, in H
Vpk / krpm	Peak line-to-line back emf constant of the machine, in V/krpm (mechanical speed)
Number of Poles	Number of poles of the machine
Integrator Gain	Gain of the torque loop integrator controller
Maximum Motor Torque	Maximum motor torque, in Nm
Base Current Value	Base current value of the system, in A
Base Torque Value	Base torque value, in Nm
Sampling Frequency	Sampling frequency of the torque loop integrator controller, in Hz

The torque control block has the following inputs and outputs:

Id: d-axis current feedback
Iq: q-axis current feedback
Te: Torque reference
Is: Current amplitude reference
Tes: Estimated torque developed by the motor

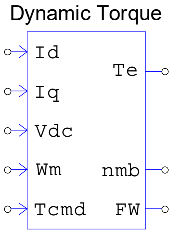
This block estimates the motor developed torque from the current feedback and motor parameters. A control loop based on a discrete integrator is used to regulate the motor torque and generate the motor current reference.

The block requires the parameters of the PMSM controlled and the inverter current rating.

5.8.2 Dynamic Torque Limit Control (PMSM)

The Dynamic Torque Limit Control (PMSM) block is for linear PMSM machines only.

Image:



Attributes:

Parameters	Description
Ld (d-axis inductance)	d-axis inductance of the PMSM machine, in H
Lq (q-axis inductance)	q-axis inductance of the PMSM machine, in H
Vpk / krpm	Peak line-to-line back emf constant of the machine, in V/krpm (mechanical speed)
Number of Poles	Number of poles of the machine
Maximum Motor Torque	Maximum motor torque, in N*m
Maximum Motor Speed (rpm)	Maximum motor speed, in rpm
Maximum Motor Power	Maximum motor power, in W
Base Voltage Value	Base voltage value for the per unit system, in V
Base Current Value	Base current value for the per unit system, in A
Base Mechanical Speed	Base mechanical speed for the per unit system, in rad/sec
Base Torque Value	Base torque value for the per unit system, in N*m

The torque control block has the following inputs and outputs:

Id: d-axis current feedback
Iq: q-axis current feedback
Vdc: DC bus voltage feedback
Wm: Motor mechanical speed, in rad/sec.
Tcmd: Torque command
Te: Torque reference
nmb: Calculated speed limit of the maximum torque region, in rpm
FW: Flag of field weakening (1: in field weakening; 0: not in field weakening)

This block calculates the speed limit of the maximum torque region. When the motor speed is less than this speed limit, the motor operates in the maximum torque region. Otherwise, it operates in the maximum power region with field weakening control where the inverter voltage amplitude is the limiting factor.

The block requires the parameters of the PMSM controlled and the inverter voltage rating.

When base values are all 1, all input and output quantities are in real values.

Note that this block assumes that the inverter connected to the motor is a 6-pulse inverter with space vector PWM. The maximum value of the ac phase-to-neutral voltage (peak) is calculated as:

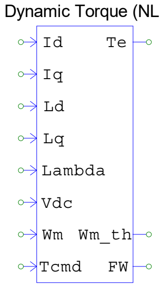
$$V_{\text{ph\_max}} = \frac{V_{dc}}{\sqrt{3}}$$

Internally, a safety factor of 0.9 is used when determining the threshold to transition to field weakening. To remove this factor, divide the dc voltage voltage by 0.9 before the Vdc input port.

5.8.3 Dynamic Torque Limit Control (Nonlinear PMSM)

The Dynamic Torque Limit Control (Nonlinear PMSM) block is for nonlinear PMSM machines only.

Image:



Attributes:

Parameters	Description
Number of Poles	Number of poles of the machine
Filter Cut-Off Frequency	Cut-off frequency fc of the internal low-pass filter, in Hz
Maximum Motor Torque	Maximum motor torque, in N*m
Maximum Motor Speed	Maximum motor speed, in rpm
Maximum Motor Power	Maximum motor power, in W
Base Voltage Value	Base voltage value for the per unit system, in V
Base Current Value	Base current value for the per unit system, in A
Base Mechanical Speed	Base mechanical speed for the per unit system, in rad/sec
Base Torque Value	Base torque value for the per unit system, in N*m
Sampling Frequency	Sampling frequency of the internal low-pass filter, in Hz

The dynamic torque limit control block is for the control of nonlinear PMSM machines only. It has the following inputs and outputs, all in per unit value except Ld, Lq, and Lambda. When base values Vb, Ib, and Wmb are all 1, all input and output quantities are in real values.

Input Signals:
Id: d-axis current feedback
Iq: q-axis current feedback
Ld: d-axis inductance Ld, in H
Lq: q-axis inductance Lq, in H

Lambda: Peak stator phase flux linkage, in Weber
Vdc: DC bus voltage feedback
Wm: Motor mechanical speed
Tcmd: Torque reference input

Output Signals:
Te: Torque reference output
Wm_th: Calculated threshold speed of the constant torque region
FW: Flag of field weakening (1: in field weakening; 0: not in field weakening)

This block calculates the threshold speed of the constant torque region. When the motor speed is less than this speed, the motor operates in the constant torque region. Otherwise, it operates in the constant power region with field weakening control where the inverter voltage amplitude is the limiting factor.

A second-order low-pass digital filter, with a cut-off frequency of fc and a damping ratio of 0.7, is used to smooth out the change of the calculated threshold speed.

When base values are all 1, all input and output quantities are in real values.

Note that this block assumes that the inverter connected to the motor is a 6-pulse inverter with space vector PWM. The maximum value of the ac phase-to-neutral voltage (peak) is calculated as:

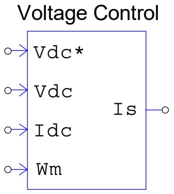
$$V_{\text{ph\_max}} = \frac{V_{dc}}{\sqrt{3}}$$

Internally, a safety factor of 0.9 is used when determining the threshold to transition to field weakening. To remove this factor, divide the dc voltage voltage by 0.9 before the Vdc input port.

5.8.4 Voltage Control (PMSM)

The Voltage Control block is for linear PMSM machines only.

Image:



Attributes:

Parameters	Description
Vpk / krpm	Peak line-to-line back emf constant of the machine, in V/krpm (mechanical speed)
Number of Poles	Number of poles of the machine
Maximum Motor Torque	Maximum motor torque, in N*m
Number of Cells in Parallel	Number of cells Np in parallel in the battery pack
PI Gain	Gain of the voltage loop PI controller
PI Time Constant	Time constant of the voltage loop PI controller, in sec.
Sampling Frequency	Sampling frequency of the voltage PI controller, in Hz

The voltage control block has the following inputs and outputs:

Vdc*: DC bus voltage reference
Vdc: DC bus voltage feedback

Idc: DC bus current feedback

Wm: Motor mechanical speed in rad/sec.

Is: Current amplitude reference

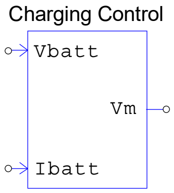
This block uses a digital PI controller to regulate the dc bus voltage. Together with the dc bus current and the machine speed, it generates the machine current reference.

The block requires the parameters of the PMSM machine controlled.

5.8.5 DC-DC Charging Control

The DC-DC Charging Control block is defined as below.

Image:



Attributes:

Parameters	Description
Converter Rated Power	Rated power of the dc-dc converter, in W
Battery-side Rated Voltage	Converter rated voltage on the battery side, in V
Number of Cells in Series	Number of cells Ns in series in the battery pack
Number of Cells in Parallel	Number of cells Np in parallel in the battery pack
Voltage Derating Factor	Voltage derating factor Ks, from 0 (100% derating) to 1 (no derating)
Full Battery Voltage	Full (or maximum) voltage of the battery cell, in V
Battery Resistance	Internal resistance of the battery cell, in Ohm
Current PI Gain	Gain of the current loop PI controller
Current PI Time Constant	Time constant of the current loop PI controller, in sec.
Voltage PI Gain	Gain of the voltage loop PI controller
Voltage PI Time Constant	Time constant of the voltage loop PI controller, in sec.
Control Block Output Limit	The upper limit of the control block output Vm. The lower limit is 0.
Sampling Frequency	Sampling frequency of the voltage and current PI controller, in Hz

The dc-dc charging control block has the following inputs and outputs:

Vbatt: Battery-side voltage

Ibatt: Current flowing into the battery

Vm: Modulation signal output

This block implements constant-voltage-constant-current charging control to a battery. When the battery terminal voltage is less than the battery float voltage (the float voltage is defined as the internal battery full voltage plus the voltage drop across the battery resistance), it is constant current charging, in which case the voltage is disabled and the current loop changes the battery at a constant current rate. The charging current amplitude is set to the converter rated current on the battery side.

On the other hand, when the battery terminal voltage reaches the float voltage, it is constant voltage charging.

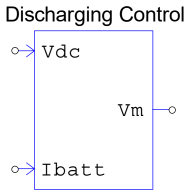
The voltage loop generates the reference for the current loop.

Digital PI controllers are used in the voltage and current loops.

5.8.6 DC-DC Discharging Control

The DC-DC Discharging Control block is defined as below.

Image:



Attributes:

Parameters	Description
Current/Voltage Mode	Operation mode of the discharge control (0: current mode; 1: voltage mode)
Converter Rated Power	Rated power of the dc-dc converter, in W
Battery-side Rated Voltage	Converter rated voltage on the battery side, in V
DC Bus Voltage Reference	DC bus voltage reference, in V
Current PI Gain	Gain of the current loop PI controller
Current PI Time Constant	Time constant of the current loop PI controller, in sec.
Voltage PI Gain	Gain of the voltage loop PI controller
Voltage PI Time Constant	Time constant of the voltage loop PI controller, in sec.
Control Block Output Limit	The upper limit of the control block output Vm. The lower limit is 0.
Sampling Frequency	Sampling frequency of the voltage and current PI controller, in Hz

The dc-dc discharging control block has the following inputs and outputs:

Vdc:	DC bus voltage feedback
Ibatt:	Current flowing into the battery
Vm:	Modulation signal output

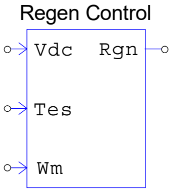
This block implements constant-voltage or constant-current discharging control to a battery. When the operation mode is set to Voltage Mode (1), the converter regulates the dc bus voltage, and the voltage loop generates the reference for the current loop. When the operation mode is set to Current Mode (0), the converter regulates the current flowing into the dc bus to the maximum current that the converter can deliver.

Digital PI controllers are used in the voltage and current loops.

5.8.7 DC-DC Regeneration Control

The DC-DC Regeneration Control block is defined as below.

Image:



Attributes:

Parameters	Description
Regen Power Threshold	Threshold of the motor power level. Beyond the threshold, regeneration is enabled.
Regen Enable Time	Time from which regeneration is enabled. Before this time, regeneration is disabled.
Sampling Frequency	Sampling frequency of the low-pass filter for speed feedback, in Hz

The dc-dc regeneration control block has the following inputs and outputs:

Vdc:	DC bus voltage feedback
Tes:	Estimated developed torque of the traction motor
Wm:	Mechanical speed of the traction motor in rad/sec.
Rgn:	Regeneration flag (0: no regeneration; 1: regeneration)

This block generates a regeneration flag based on the dc bus voltage and the traction motor power. When the motor power is negative (in generation mode) and the power amplitude exceeds the threshold, the regeneration flag is set to 1 to enable regeneration.

To avoid false triggering due to start-up transient, a regeneration enable time is defined. Regeneration is possible only after this time has passed.

6
Other Components

This chapter provides descriptions for the components in PSIM element library's Other section.

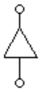
6.1 Switch Controllers

A switch controller has the same function as a switch gate/base drive circuit in an actual circuit. It receives the input from the control circuit, and controls switches in the power circuit. One switch controller can control multiple switches simultaneously.

6.1.1 On-Off Switch Controller

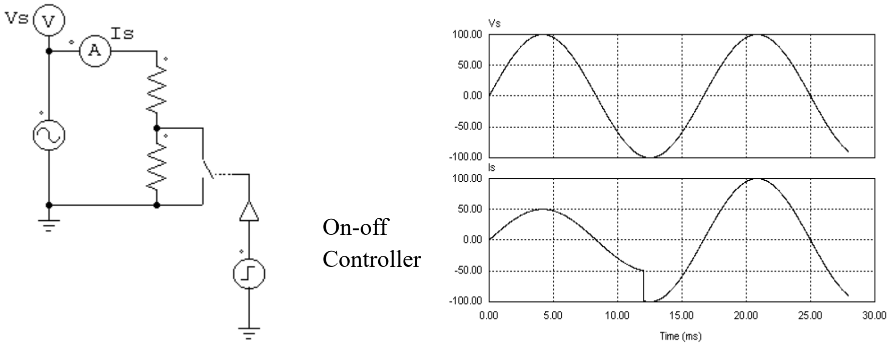
An on-off switch controller is used as the interface between a control gating signal and a power switch. The input, which is a logic signal (either 0 or 1) from the control circuit, is passed to the power circuit as the gating signal.

Image:



Example:

The circuit below implements the step change of a load. In the circuit, the on-off switch controller is used to control the bi-directional switch. The step voltage source, which is connected to the controller input, changes from 0 to 1 at the time of 12 ms. The closure of the switch results in the short-circuit of the resistor across the switch and the increase of the current.



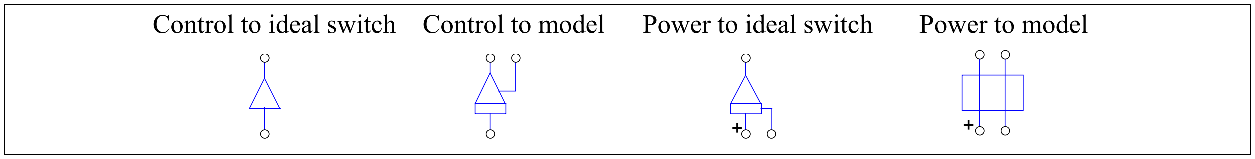
6.1.2 On-Off Switch Controller (Multi-Level)

An on-off switch controller (multi-level) is used to control a multi-level switch such as MOSFET or IGBT. Depending on the input/output (I/O) signal type, this element can connect a control signal to an ideal switch or to a switch model (such as Level-1 or Level-2 or SPICE model). Similarly, this element can connect a power circuit to an ideal switch or to a switch model.

The image is different depending on the I/O signal type. There are four I/O signal types:

- *Control to ideal switch*: Input is control signal (0/1) and the output is the gate node of an ideal switch
- *Control to model*: Input is control signal and outputs are transistor gate node and source node
- *Power to ideal switch*: Inputs are power circuit and the output is the gate node of an ideal switch
- *Power to model*: Inputs are power circuit and outputs are transistor gate node and source node

Images:



Attributes for *Control to model*:

Parameters	Description
I/O Signal Type	It is set to *Control to model*.
Gate Voltage High	The high level of the gate voltage, in V
Gate Voltage Low	The low level of the gate voltage, in V
Gate Resistance	Gate resistance, in Ohm
Gate Current Flag	Flag for gate current display

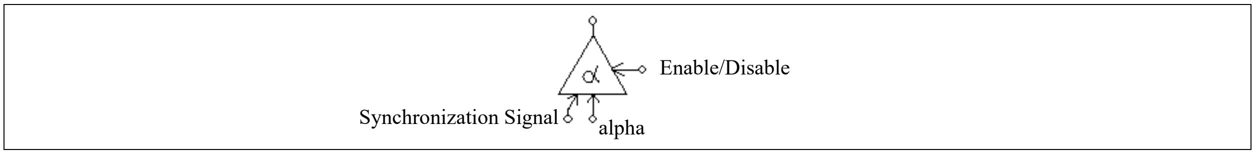
Attributes for *Power to ideal switch*:

Parameters	Description
I/O Signal Type	It is set to *Power to ideal switch*.
Gate Threshold Voltage	When the input voltage is above the gate threshold voltage, the output is 1. Otherwise, the output is 0.

6.1.3 Alpha Controller

An alpha controller is used for delay angle control of thyristor switches or bridges. There are three input for the controller: the alpha value, the synchronization signal, and the gating enable/disable signal. The transition of the synchronization signal from low to high (from 0 to 1) provides the synchronization and this corresponds to the moment when the delay angle alpha equals zero. A gating with a delay of alpha degrees is generated and sent to the thyristors. The alpha value is updated instantaneously.

Image:



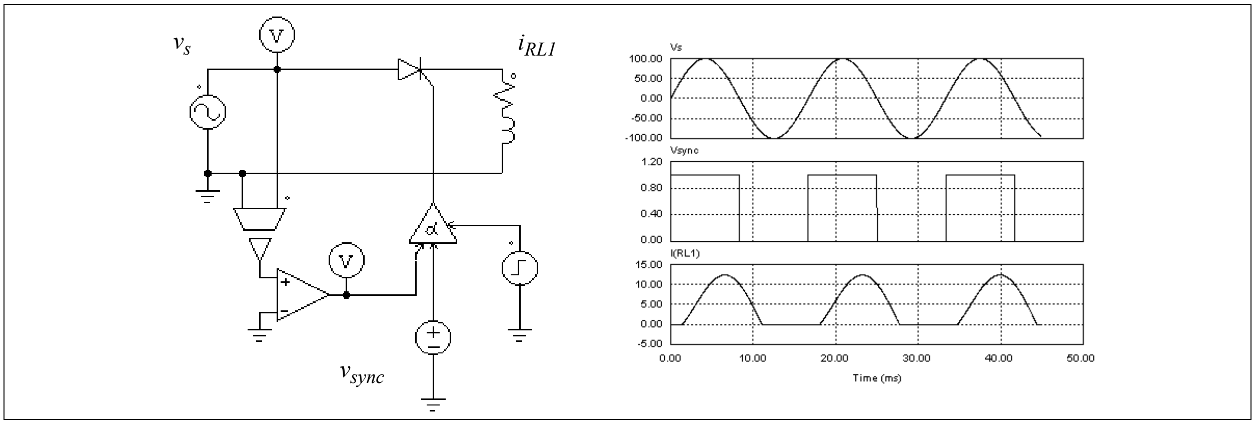
Attributes:

Parameters	Description
Frequency	Operating frequency of the controlled switch/switch module, in Hz
Pulse Width	On-time pulse width of the switch gating, in deg.

The input for the delay angle alpha is in deg.

Example:

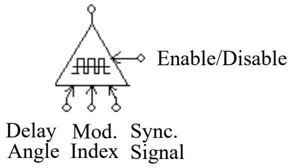
The figure below shows a thyristor circuit using delay angle control. In the circuit, the zero-crossing of v_s, which corresponds to the moment that the thyristor would start conducting naturally, is used to provide the synchronization. The delay angle is set at 30^o. The gating signal is delayed from the rising edge of the synchronization signal by 30^o.



6.1.4 PWM Lookup Table Controller

There are four input signals in a PWM lookup table controller: the modulation index, the delay angle, the synchronization signal, and the gating enable/disable signal. The gating pattern is selected based on the modulation index. The synchronization signal provides the synchronization to the gating pattern. The gating pattern is updated when the synchronization signal changes from low to high. The delay angle defines the relative angle between the gating pattern and the synchronization signal. For example, if the delay angle is 10 deg., the gating pattern will be leading the synchronization signal by 10 deg.

Image:



Attributes:

Parameters	Description
Frequency	Switching frequency, in Hz
Update Angle	Update angle, in deg., based on which the gatings are internally updated. If the angle is 360^o, the gatings are updated at every cycle. If it is 60^o, the gatings are updated at every 60^o.
File Name	Name of the file storing the PWM gating pattern

A lookup table, which is stored in a file, contains the gating patterns. It has the following format:

n, m_1, m_2, ..., m_n
k_1
$G_{1,1}$, $G_{1,2}$, ..., $G_{1,k1}$
... ...
k_n
$G_{n,1}$, $G_{n,2}$, ..., $G_{n,kn}$

where n is the number of gating patterns; m_i is the modulation index correspondent to Pattern i; and k_i is the number of switching points in Pattern i. The modulation index array m_1 to m_n should be monotonically increasing. The output will select the i_{th} pattern if the input is smaller than or equal to m_i. If the input exceeds m_n, the last pattern will be selected.

The table below shows a sample PWM pattern file with 4 modulation index levels and 14 switching points.

4, 0.901, 0.910253, 0.920214, 1.199442

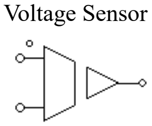
```
14
7.736627   72.10303   80.79825   99.20176   107.8970   172.2634   180.
187.7366   252.1030   260.7982   279.2018   287.8970   352.2634   360.
14
7.821098   72.27710   80.72750   99.27251   107.7229   172.1789   180.
187.8211   252.2771   260.7275   279.2725   287.7229   352.1789   360.
14
7.902047   72.44823   80.66083   99.33917   107.5518   172.0979   180.
187.9021   252.4482   260.6608   279.3392   287.5518   352.0980   360.
14
10.186691   87.24225   88.75861   91.24139   92.75775   169.8133   180.
190.1867   267.2422   268.7586   271.2414   272.7578   349.8133   360.
```

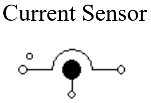
In this example, if the modulation index input is 0.8, the controller will select the first gating pattern. If the modulation index is 0.915, the controller will select the third pattern.

6.2 Sensors

Voltage/current sensors measure the voltages/currents of the power circuit and send them to the control circuit. The current sensor has an internal resistance of 1 μΩ.

Images:





In the images, the dot indicates the positive terminal.

Attribute:

Parameter	Description
Gain	Gain of the sensor

6.3 Probes, Meters, and Scopes

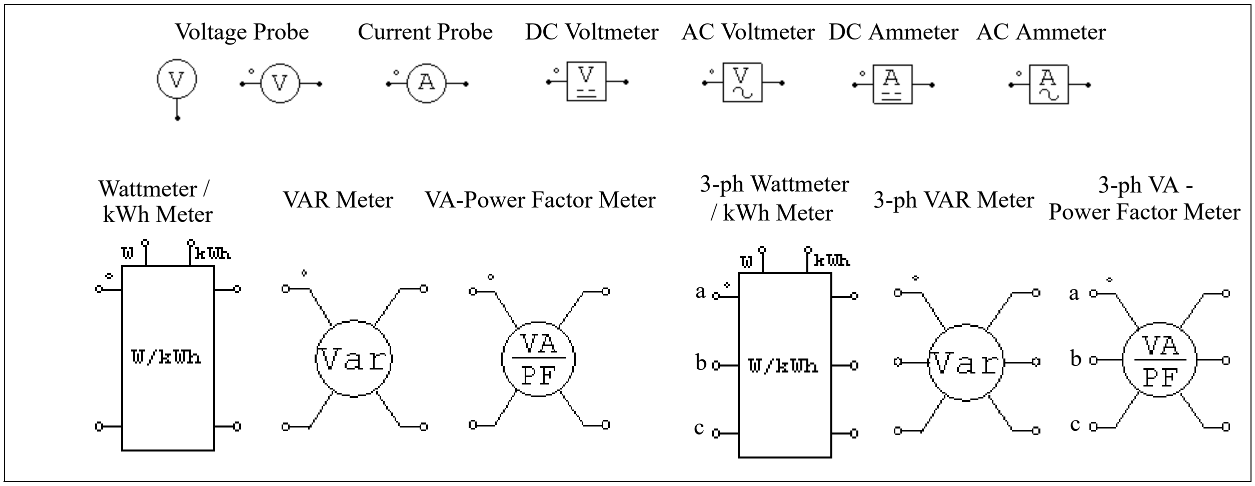
Probes and meters are used to measure voltages, currents, power, or other quantities, while scopes are used to display voltage and current waveforms.

6.3.1 Probes and Meters

Below find the images of the ac ammeter, dc ammeter, voltage probe, current probe, ac voltmeter, dc voltmeter, single-phase and 3-phase wattmeters and kilowatt hour (kWh) meters, VAR meters, and VA-Power Factor meters. A voltage probe measures a node voltage with respect to ground. A two-terminal voltage probe measures the voltage between two nodes. A current probe or meter measures the current flowing into the meter from the dotted side. Similarly, a power meter measures the power flowing into the meter from the dotted side. Note that all the probes and meters, except the node-to-ground probe, are allowed in the power circuit only.

While voltage/current probes measure the instantaneous voltage or current, meters measure the dc or ac quantities. A small resistor of 1 μΩ is used in the current probe internally to measure the current.

Images:



Attributes:

Parameters	Description
Operating Frequency	Operating frequency or fundamental frequency of the ac meter, in Hz
Cut-off Frequency	Cut-off frequency of the low-pass/high-pass filter, in Hz
kWh Meter Start Time	Time from which the kilowatt hour meter starts, in sec. (for W/kWh meters only)
kWh Meter Stop Time	Time at which the kilowatt hour meter stops, in sec. (for W/kWh meters only)
VA Display Flag	Display flag for apparent power (0: no display; 1: display) (for VA-Power Factor meters only)
PF Display Flag	Display flag for power factor (0: no display; 1: display) (for VA-Power Factor meters only)
DPF Display Flag	Display flag for displacement power factor (0: no display; 1: display) (for VA-Power Factor meters only)

In the images for the single-phase/3-phase wattmeter / kWh meter, the node with the letter "W" is for the real power output, in watt, and the node with the letter "kWh" is for the kilowatt hour output, in kilowatt-hour.

A low-pass filter is used in the dc meter and wattmeter models to filter out high-frequency components, whereas a high-pass filter is used in the ac meter models to filter out the dc component. The cut-off frequency determines the transient response of the filter.

Except the voltage and current probes, the readings of all the meters are meaningful only when the readings reach the steady state.

The single-phase and 3-phase VAR meter measure the reactive power (VAR) at the fundamental frequency; and the VA-Power Factor meters measure the apparent power (VA), the total power factor (PF), and the displacement power factor (DPF). The kilowatt hour meter reading gives the integration of the real power over a time interval defined by the start time and the stop time.

For a single-phase circuit, the real power *P*, reactive power *Q*, apparent power *S*, total power factor *PF*, and the displacement power factor *DPF* are defined as follows.

Assume both the voltage and current contains harmonics, i.e.

$$v(t) = \sqrt{2}V_1 \sin(\omega_1 t + \phi_1) + \sqrt{2}V_2 \sin(\omega_2 t + \phi_2) + \ldots$$

$$i(t) = \sqrt{2}I_1 \sin(\omega_1 t + \theta_1) + \sqrt{2}I_2 \sin(\omega_2 t + \theta_2) + \ldots$$

where ω_1 is the fundamental frequency and all others are harmonic frequencies. We have the rms values of the voltage and current as:

$$V_{rms} = \sqrt{V_1^2 + V_2^2 + \dots}$$

$$I_{rms} = \sqrt{I_1^2 + I_2^2 + \dots}$$

The real power (or average power) *P* is defined as:

$$P = \frac{1}{T}\int_0^T (v(t) \cdot i(t))dt$$

where *T* is the fundamental period.

The reactive power *Q* is defined as:

$$Q = V_1 \cdot I_1 \cdot \sin(\phi_1 - \theta_1)$$

Note that the reactive power is for the fundamental component only.

The apparent power *S* is defined as:

$$S = V_{rms} \cdot I_{rms}$$

The total power factor *PF* and the displacement power factor *DPF* are then defined as follow:

$$PF = \frac{P}{S}$$

$$DPF = \cos(\phi_1 - \theta_1)$$

For the 3-phase circuit, the definitions are similar.

Note that all the 3-phase meters, including 3-phase wattmeter/kWh meter, VAR meter, VA-Power Factor meter, are for 3-phase 3-wire circuit only, and cannot be used in 3-phase 4-wire circuit. The models of the meters are based on the assumption that the summation of 3-phase voltages and currents must be equal to zero, that is:

$$v_a + v_b + v_c = 0$$

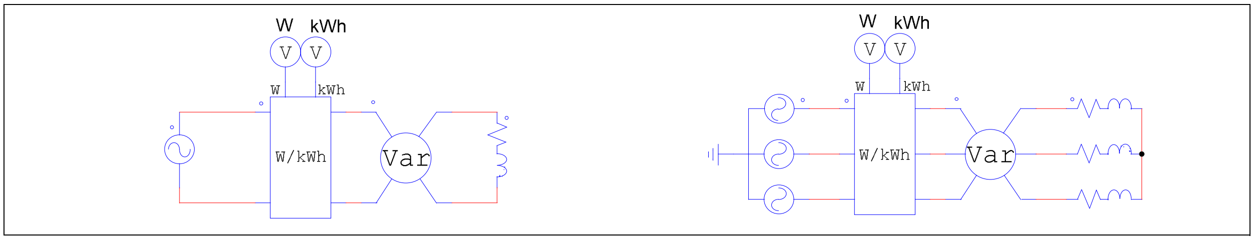
$$i_a + i_b + i_c = 0$$

For 3-phase 4-wire circuit, single-phase meters should be used instead.

To use the single-phase or 3-phase wattmeter/VAR meters, insert the meters into the circuit.

Example:

This example shows how single-phase and 3-phase meters are used. The circuit on the left shows the use of the wattmeter and VAR meter, and the circuit on the right shows the use of the three-phase wattmeter and VAR meter.



6.3.2 Voltage/Current Scopes

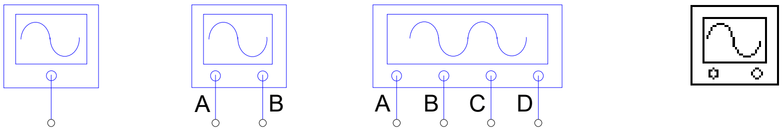
While voltage/current probes and meters save the simulation results for post waveform processing, voltage/current scopes allow users to view simulation waveforms at runtime in the middle of the simulation.

Three scopes are provided: 1-channel voltage scope, 2-channel voltage scope, 4-channel voltage scope, and current scope.

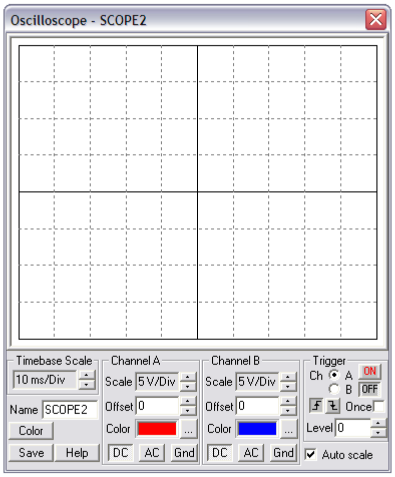
Below are the images of the voltage and current scopes and their expanded view.

1-channel, 2-channel, and 4-channel voltage scopes

Current scope



Interface for 2-channel scope


Oscilloscope - SCOPE2

Timebase Scale: 10 ms/Div

Name: SCOPE2

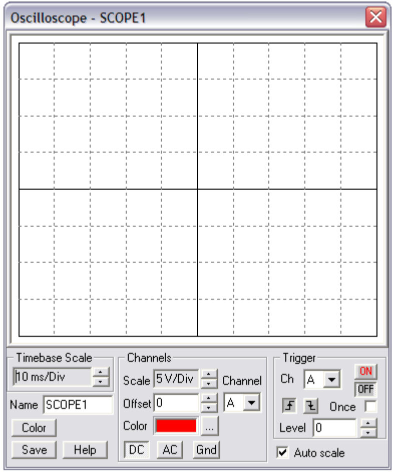
Channel A: Scale 5 V/Div, Offset 0, Color 🟥

Channel B: Scale 5 V/Div, Offset 0, Color 🟦

Trigger: Ch ● A ○ B, Once □, Level 0

☑ Auto scale

Interface for current scope, and 1-channel and 4-channel voltage scopes


Oscilloscope - SCOPE1

Timebase Scale: 10 ms/Div

Name: SCOPE1

Channels: Scale 5 V/Div, Offset 0, Color 🟥, Channel A

Trigger: Ch A, Once □, Level 0

☑ Auto scale

The 1-channel and 4-channel voltage scope and the current scope have the same interface. For the 4-channel scope, the channel can be selected through the drop-down channel selection, and the settings apply that the channel selected.

The scope is designed to operate in a similar way as the actual oscilloscope in the lab. It has 3 main sections: *Timebase* section, *Channel* section, and *Trigger* section.

In the *Timebase* section, the scale of time (x axis) is defined.

In the *Channel* section, the scale of the Y axis, as well as the offset and the color of the waveform, are defined. The channel display mode can be either *DC*, *AC*, or *Gnd* (for ground). When the display mode is in *DC*, the entire waveform is displayed. If the display mode is in *AC*, only the ac portion of the waveform is displayed. When the display mode is *Gnd*, the waveform will show 0.

In the *Trigger* section, the trigger conditions are defined. The trigger can be set to either *ON* or *OFF*. When the trigger is off, the waveform is free-running, and the display of the waveform in the scope may vary from one frame to another. If the trigger is on, the waveform display will only start when the trigger conditions are met. This will lead to a steady waveform display.

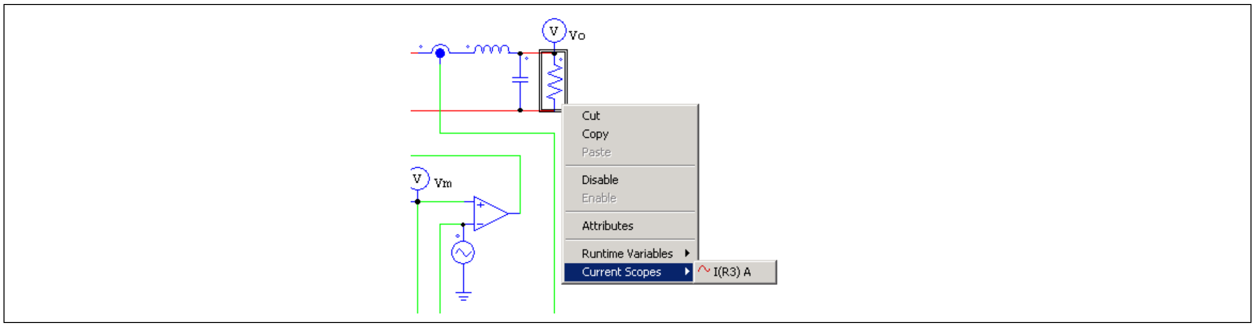
There are three trigger modes: rising-edge triggering, falling-edge triggering, and one-shot triggering (if the *once* checkbox is checked, the one-shot triggering is selected). The one-shot triggering will trigger only once, and it is useful, for example, in situations where a transient needs to be captured.

The trigger level sets the level at which the triggering occurs. For example, if Channel A is selected with the rising-edge triggering and the trigger level of 0V, whenever the Channel A input crosses over 0 from negative to positive, triggering will occur and the waveform display will start from that instant.

On the scope, if the *Auto scale* checkbox is checked, the scale of all the channels will be automatically adjusted so that the waveforms are within the display of the scope.

Note that voltage scopes have connecting terminals which can be connected to either power circuit nodes or control circuit nodes. The scopes will display the node-to-ground voltages at these nodes.

The current scope, on the other hand, has no connecting terminals. It is associated with any element that has the parameter of the current flag, and it is enabled by right clicking on top of the element, and selecting the branch current under the *Current Scopes*, as shown below. After the branch current is selected, a check mark will appear in front of the branch current name.



If the element has multiple current flags, under the *Current Scopes* menu, there will be multiply branch currents, one corresponding to each current flag.

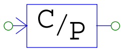
For example, for a 3-phase resistor R1, under the *Current Scopes* menu, there will be three branch currents: I(R1) A, I(R1) B, and I(R1) C. The letter "A", "B", and "C" refer to Channel A, B, and C, respectively. For example, if "I(R1) A", "I(R1) B", and "I(R1) C) are all selected, in the current scope, one can go to the *Channel* pull-down menu in the *Channel* section, and select one of the channels for display. If Channel A is selected, the scope will show the Phase A branch current I(R1).

6.4 Function Blocks

6.4.1 Control-Power Interface Block

A control-to-power interface block passes a control circuit value to the power circuit. It is used as a buffer between the control and power circuit. The output of the interface block is treated as a voltage source when the power circuit is solved. With this block, some of the functions that can only be generated in the control circuit can be passed to the power circuit.

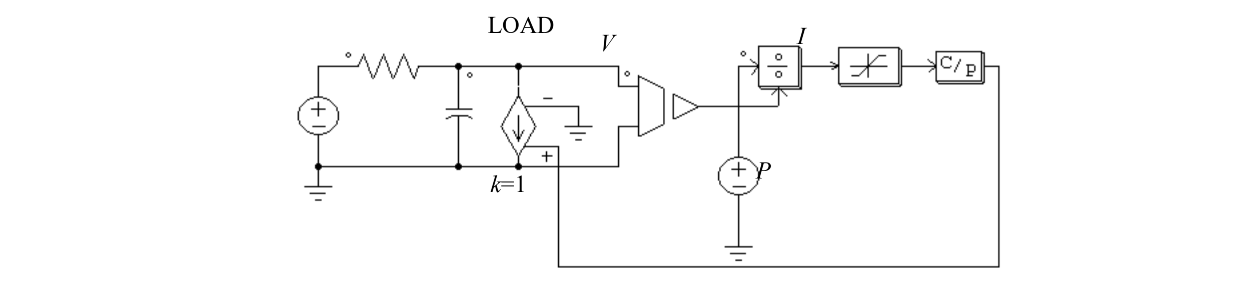
Image:



Example: A Constant-Power Load Model

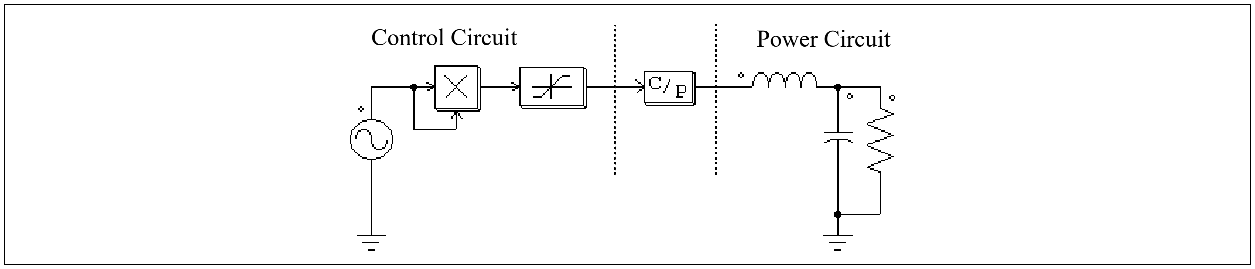
In a constant-power dc load, the voltage *V*, current *I*, and power *P* have the relationship as *P=V*I*. Given the voltage and the power, the current can be calculated as *I=P/V*. This can be implemented using the circuit as shown below.

The load voltage is measured through a voltage sensor and is fed to a divider. The output of the divider gives the current value *I*. Since the voltage could be zero or a low value at the initial stage, a limiter is used to limit the current amplitude. This value is converted into the load current quantity through a voltage-controlled current source.



Example:

The following circuit illustrates how a control circuit signal can be passed to the power circuit. As seen from the power circuit, the CTOP block behaviors as a grounded voltage source.



6.4.2 Transformation Blocks

Function blocks of the following transformations are provided:

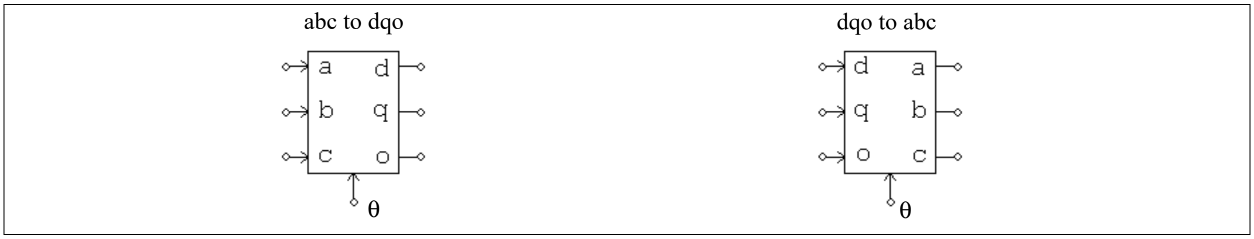
- abc-dqo transformation
- abc-αβ transformation
- αβ-dq transformation
- Cartesian-polar transformation

All these blocks can be used in either power circuit or control circuit.

6.4.2.1 abc-dqo Transformation

The abc-dqo function blocks perform the transformation between the abc coordinate and the dqo coordinate. The transformation is often referred to as the Park transformation.

Images:



Attributes:

Parameters	Description
Transformation Flag	Transformation flag (0: The q-axis leads the d-axis; 1: The q-axis lags the d-axis)

The angle θ at the bottom of the blocks is in rad.

It should be noted that, in power circuit, currents must first be converted into voltage quantities (using current-controlled voltage sources) before they can be transformed. Also, if an input terminal is not used (such as in the dqo-to-abc transformation block where only Phase d and q are not used, and Phase o is not used), it must be connected to ground.

Depending on the transformation flag, the transformation equations are as follows.

From abc to dqo, with the transformation flag set to 0:

$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

From abc to dqo, with the transformation flag set to 1:

$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

Note that the coefficient 2/3 in front of the transformation matrix means that the transformation is amplitude invariant, not power invariant. That is, the dq vector amplitude is equal to the peak amplitude of the three-phase sinusoidal waveforms. But the power in the dq frame must be multiplied by 3/2 to have the same power, i.e. *Power* = $v_a{*}i_a + v_b{*}i_b + v_c{*}i_c$ = 3/2 *($v_d{*}i_d{+}v_q{*}i_q$). To have the power invariant transformation, replace 2/3 with $\sqrt{\frac{2}{3}}$, and 1/2 with $\sqrt{\frac{1}{2}}$.

From dqo to abc, with the transformation flag set to 0:

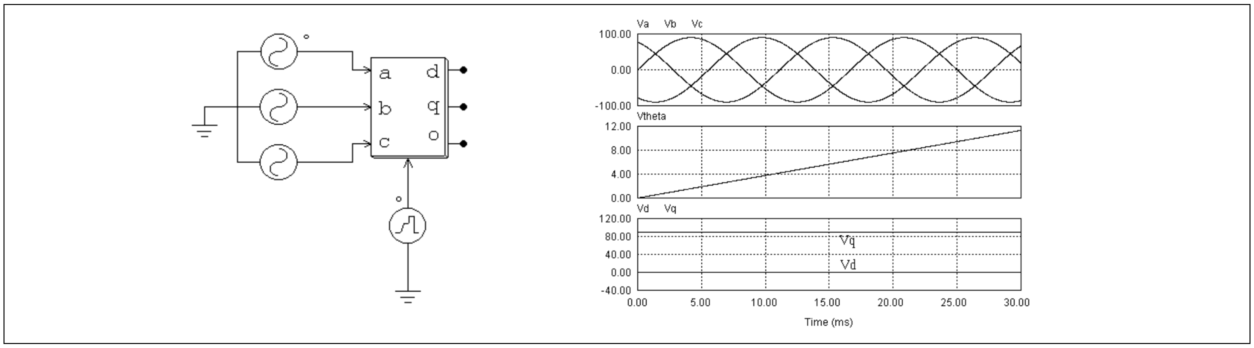
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix}$$

From dqo to abc, with the transformation flag set to 1:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix}$$

Example:

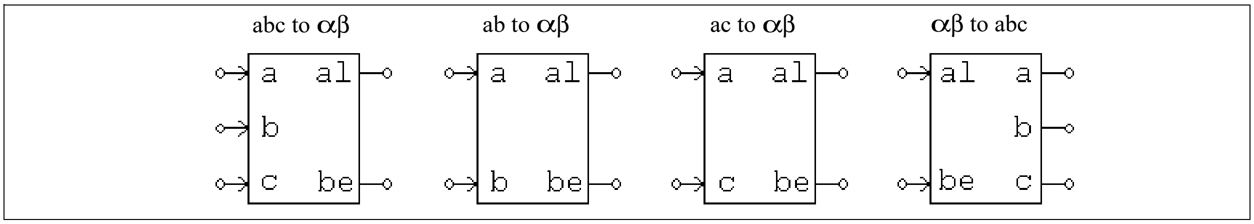
In this example, three symmetrical ac waveforms are transformed into dqo quantities. The angle θ is defined as θ = ωt where ω = 2π*60. Since the angle θ changes linearly with time, a piecewise linear voltage which has a ramp waveform is used to represent θ. The simulation waveforms show the three-phase ac (top), the angle θ (middle), and the dqo output. In this example, the “q” component is constant, and both the “d” and the “o” components are zero.



6.4.2.2 abc-αβ Transformation

The abc-αβ function blocks perform the transformation between the abc coordinate and the αβ coordinate. It is often referred to as the Clarke transformation.

Images:



In the images, the letter "al" refers to α, and the letter "be" refers to β. The transformation equations are:

From abc to αβ:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

From ab to αβ:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ \frac{\sqrt{2}}{2} & \sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \end{bmatrix}$$

From ac to αβ:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ -\frac{\sqrt{2}}{2} & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_c \end{bmatrix}$$

From αβ to abc:

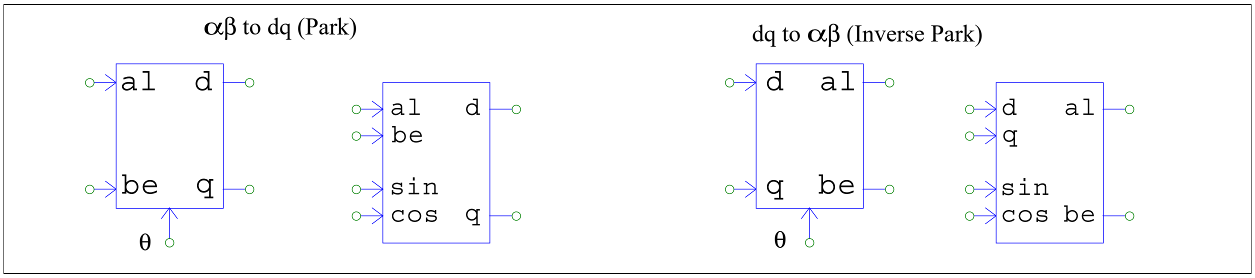
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

For the ab-to-αβ and ac-to-αβ transformation, it is assumed that $v_a + v_b + v_c = 0$.

6.4.2.3 αβ-dq Transformation

The αβ-dq function blocks perform the transformation from αβ to dq coordinate which is often referred as Park Transformation. The dq-αβ function blocks perform the transformation from dq to αβ coordinate which is often referred as Inverse Park Transformation.

Images:



In the images, the letter "al" refers to α, and the letter "be" refers to β. The phase angle input can be either θ in radian or sin(θ) and cos(θ).

The transformation equations are:

From αβ to dq:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

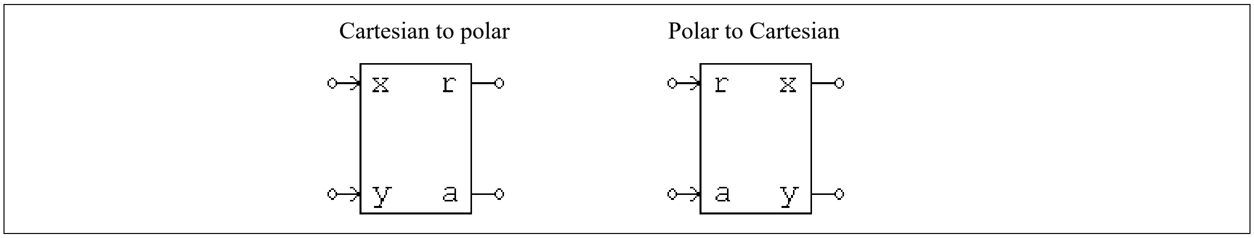
From dq to αβ:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \end{bmatrix}$$

6.4.2.4 Cartesian-Polar Transformation

The Cartesian-polar function blocks perform the transformation between the Cartesian coordinate and the polar coordinate.

Images:



In the images, the letter "r" refers to the amplitude, and "a" refers to the phase angle θ, in rad.

The transformation equations are:

From Cartesian to polar:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \operatorname{atan}\left(\frac{y}{x}\right)$$

From polar to Cartesian:

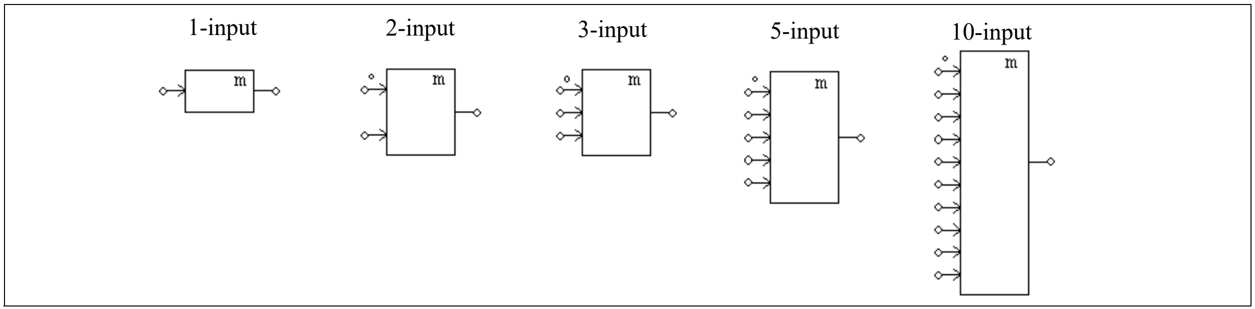
$$x = r \cdot \cos\theta$$

$$y = r \cdot \sin\theta$$

6.4.3 Math Function Blocks

The output of a math function block is expressed as the mathematical function of the inputs. With this block, one can implement complex and nonlinear relationship easily. Blocks with 1, 2, 3, 5, and 10 inputs are provided.

Images:



Attributes:

Parameters	Description
Expression f(x_1,x_2,...,x_n)	Expression of the output versus inputs where *n* is the number of inputs
Expression *df/dx*$_i$	Expression of the derivative of the function *f* versus the i_{th} input

The derivative df/dx_i can be set to zero.

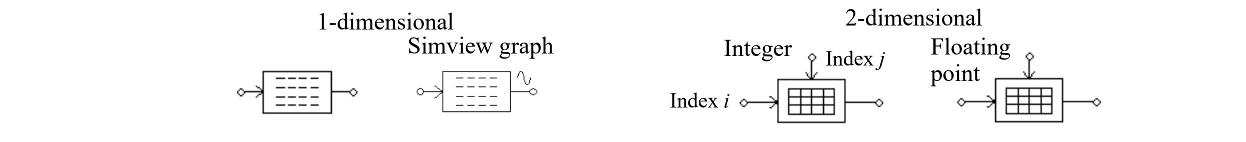
The variables that are allowed in the expression are: *T* or *t* for time, and x_i (*i* from 1 to *n*) which represents the i_{th} input. For example, for the 3-input math function block, the allowed variables are: *T*, *t*, x_1, x_2, and x_3. For the 1-input math function block, the variable *x*, which refers to the only input, is also allowed.

6.4.4 Lookup Tables

There are four lookup table blocks: One-dimensional regular lookup table and lookup table with data from Simview graph, and 2-dimensional lookup tables with integer input and floating-point input.

All four lookup tables can be used in both power circuit and control circuit.

Images:



Attributes for *one-dimensional lookup table (Simview graph)*:

Parameter	Description
Graph File	Name of the Simview file to be used as the lookup table data
Input Column	Input column of the lookup table, selected from a drop-down menu after a graph file is loaded.
Output Column	Output column of the lookup table, selected from a drop-down menu after a graph file is loaded.

For 2-dimensional lookup tables, the node on the left is for the row index input, and the node on the top is for the column index input.

A one-dimensional lookup table has one input and one output. Two data arrays, corresponding to the input and output arrays, are stored in a lookup table.

A 2-dimensional lookup table with *integer input* has two inputs. The output data is stored in a 2-dimensional matrix. The two inputs correspond to the row and column indices of the matrix. For example, if the row index is 3 and the column index is 4, the output will be *A*(3,4) where *A* is the data matrix.

A 2-dimensional lookup table with *floating-point input* is similar to the lookup table with integer input. The difference is that inputs are floating-point values, and interpolation is used to calculate the output.

Except the one-dimensional lookup table (Simview graph) where data is defined in a Simview file, there are two ways to define lookup table data. One is to enter the data directly in the dialog window in the **Lookup Table** tab. Another is to prepare the lookup table externally using a text editor, and define the file in the dialog window.

To enter the data directly in the dialog window, define the number of rows (and columns for 2-dimensional lookup tables), and click on **Set**. Then enter the values in the data cells provided. For 2-dimensional lookup tables with floating-point input, also enter row input array in the left most column and column input array in the top most row.

To prepare the data externally in a text file, define the data in the format as described below. Then click on **Open File...** to load the file. Click on **Reload Data** if the text file is modified after it is loaded. Click on S**ave As...** to save the data to an external file.

Note that the external text file is optional. When an external text file is not defined, the internal data is used. But if an external text file is defined, the text file takes the precedence and will overwrite the internal data. It is important to note that, if the dialog window is closed, the external file will be saved automatically when one closes the dialog window (by clicking on X at the upper right corner of the dialog window). Also, a copy of the external text file is saved into the schematic file. In the case where the schematic file is moved to a different computer and the external text file does not exist, PSIM will recreate the text file from the last time.

The data format for one-dimensional lookup tables is:

$V_{in}(1)$, $V_o(1)$
$V_{in}(2)$, $V_o(2)$
...
$V_{in}(\mathrm{N})$, $V_o(\mathrm{N})$

Data points can be separated by either comma or space, such as:

$V_{in}(1)$ $V_o(1)$
$V_{in}(2)$ $V_o(2)$
...
$V_{in}(\mathrm{N})$ $V_o(\mathrm{N})$

The input array V_{in} must be monotonically increasing. Between two points, linear interpolation is used to obtain the output. When the value of the input is less than $V_{in}(1)$ or greater than $V_{in}(\mathrm{N})$, the output will be clamped to $V_o(1)$ or $V_o(\mathrm{N})$.

The data format for 2-dimensional lookup tables with integer input is:

M, N
A_{11}, A_{12}, ..., $A_{1\mathrm{N}}$
A_{21}, A_{22}, ..., $A_{2\mathrm{N}}$
... ...
$A_{\mathrm{M}1}$, A_{M2}, ..., A_{MN}

where *M* and *N* are the number of rows and columns, respectively. Since the row or column index must be an integer, the input value is automatically converted to an integer. If either the row or column index is out of the range (for example, the row index is less than 1 or greater than m), the output will be zero.

The data format for 2-dimensional lookup tables with floating-point input is:

M, N
$V_{\mathrm{r}1}$ $V_{\mathrm{r}2}$ ... V_{rM}
$V_{\mathrm{c}1}$ $V_{\mathrm{c}2}$ ... V_{cN}
A_{11}, A_{12}, ..., $A_{1\mathrm{N}}$

$A_{21}, A_{22}, ..., A_{2\mathrm{N}}$
... ...
$A_{\mathrm{M}1}, A_{M2}, ..., A_{MN}$

where *m* is the number of rows and *n* is the number of columns; V_r is the row vector and V_c is the column vector; and *A*(i,j) is the output value at Row *i* and Column *j*. Note that Vectors V_r and V_c must be monotonically increasing.

If an input falls between two points, interpolation is used to calculate the value. If an input is less than the minimum or greater than the maximum value, the input will be limited to the minimum or maximum value.

Examples:

The following shows a one-dimensional lookup table:

1., 10.
2., 30.
3., 20.
4., 60.
5., 50.

If the input is 0.99, the output will be 10. If the input is 1.5, the output will be $10 + \frac{(1.5-1)\cdot(30-10)}{2-1}$=20.

The following shows a 2-dimensional lookup table with integer inputs:

3, 4
1., -2., 4., 1.
2., 3., 5., 8.
3., 8., -2., 9.

If the row index is 2 and the column index is 4, the output will be 8. If the row index is 5, regardless of the column index, the output will be 0.

The following shows a 2-dimensional lookup table with floating-point inputs:

3, 4
1.1 2.2 3.3
1.2 2.3 3.4 4.5
1., -2., 4., 1.
2., 3., 5., 8.
3., 8., -2., 9.

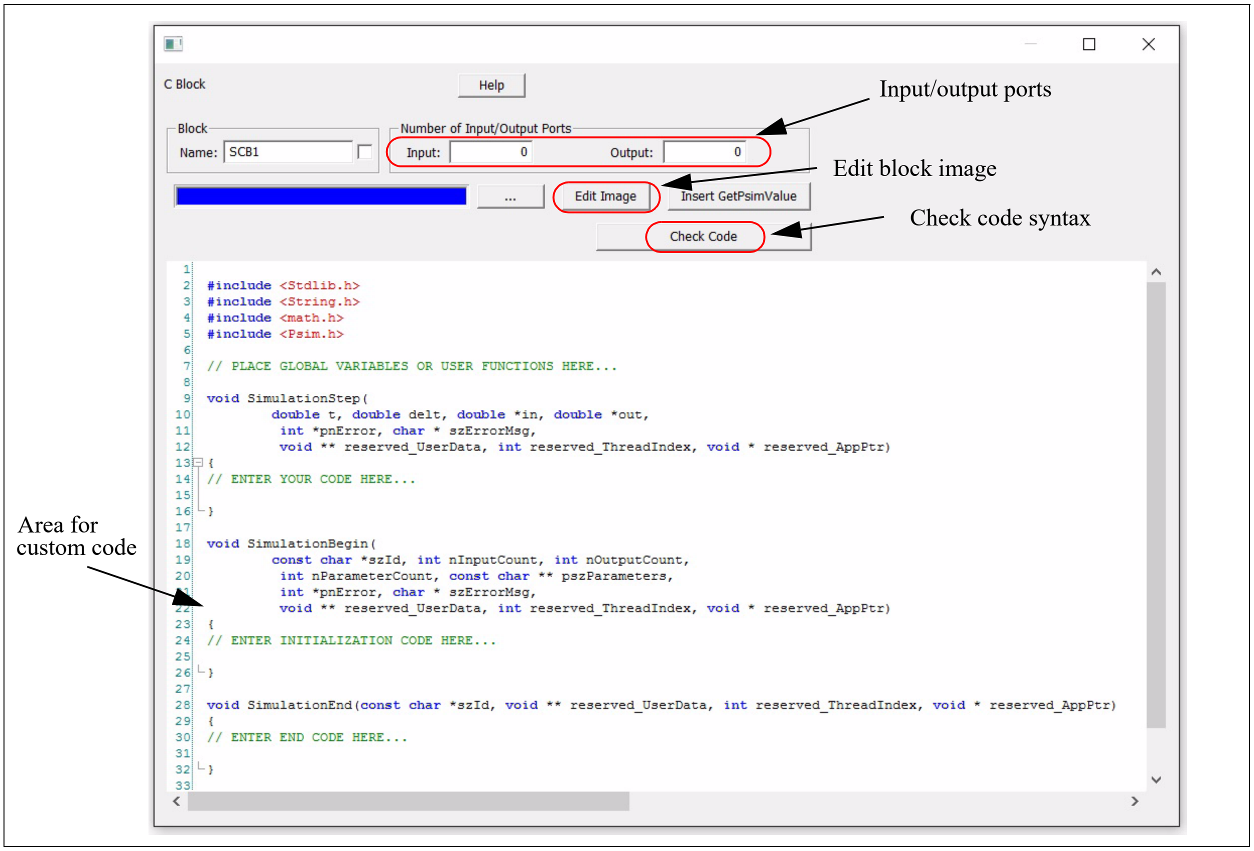
If the row input is 2 and the column input is 3, the following table shows the four points that enclose the input point, and the output value of 3.826 through interpolation:

		Column		
		2.3	3	3.4
	1.1	-2		4
Row	2	2.091	3.826	4.818
	2.2	3		5

6.4.5 C Block

A C block allows users to enter C code directly without compiling the code, unlike in external DLL blocks where the code needs to be compiled into a DLL using an external compiler. The code of the C block will be interpreted and executed at runtime by a built-in C interpreter in PSIM.

The interface of the C block dialog window is shown below.



Name: SCB1

Input: 0   Output: 0

Input/output ports

Edit block image

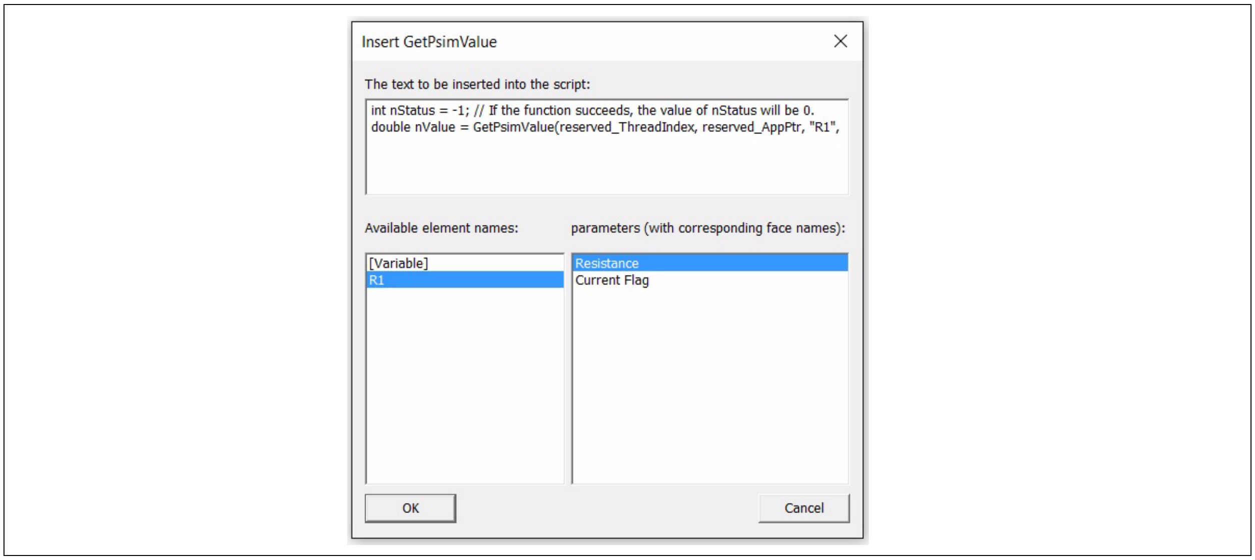
Check code syntax

Area for custom code

```

#include <Stdlib.h>
#include <String.h>
#include <math.h>
#include <Psim.h>

// PLACE GLOBAL VARIABLES OR USER FUNCTIONS HERE...

void SimulationStep(
        double t, double delt, double *in, double *out,
         int *pnError, char * szErrorMsg,
         void ** reserved_UserData, int reserved_ThreadIndex, void * reserved_AppPtr)
{
// ENTER YOUR CODE HERE...

}

void SimulationBegin(
        const char *szId, int nInputCount, int nOutputCount,
         int nParameterCount, const char ** pszParameters,
         int *pnError, char * szErrorMsg,
         void ** reserved_UserData, int reserved_ThreadIndex, void * reserved_AppPtr)
{
// ENTER INITIALIZATION CODE HERE...

}

void SimulationEnd(const char *szId, void ** reserved_UserData, int reserved_ThreadIndex, void * reserved_AppPtr)
{
// ENTER END CODE HERE...

}

```

The number of input and output ports of the C block is defined in the *Number of Input/Output Ports* section. If the number of ports is changed, the image of the block in the schematic will change accordingly.

The function **Insert GetPsimValue** allows the C block to access element parameters and parameters defined in a parameter file. Assuming that the circuit has a resistor called R1, to access the resistance value of R1, click on **Insert GetPsimValue**. The following dialog window will appear.



The text to be inserted into the script:

```
int nStatus = -1; // If the function succeeds, the value of nStatus will be 0.
double nValue = GetPsimValue(reserved_ThreadIndex, reserved_AppPtr, "R1",
```

Available element names:

- [Variable]
- R1

parameters (with corresponding face names):

- Resistance
- Current Flag

Click on OK, and the following code will be inserted to the code:

```
int nStatus = -1; // If the function succeeds, the value of nStatus will be 0.
double nValue = GetPsimValue(reserved_ThreadIndex, reserved_AppPtr, "R1", "Resistance", &nStatus);
```

The resistance of the resistor R1 will be obtained and placed in the variable *nValue*.

To access a variable, for example "alpha", defined in a parameter file, use the following code:

```
int nStatus = -1;
double nValue = GetPsimValue(reserved_ThreadIndex, reserved_AppPtr, "", "alpha", &nStatus);
if(nStatus != 0)
{       //optional error handling code
   printf("Error: alpha = %f, nStatus = %i \n", value, nStatus);
}
```

The value of the variable alpha will be obtained and placed in the variable *nValue*.

For faster simulation, the function GetPsimValue should be used in *SimulationBegin*, and the variable nValue should be defined as a global variable so that it can be accessed in the *SimulationStep* function.

The code area includes four main sections:

Variable/Function Definitions:This section includes statements and global variable definition.

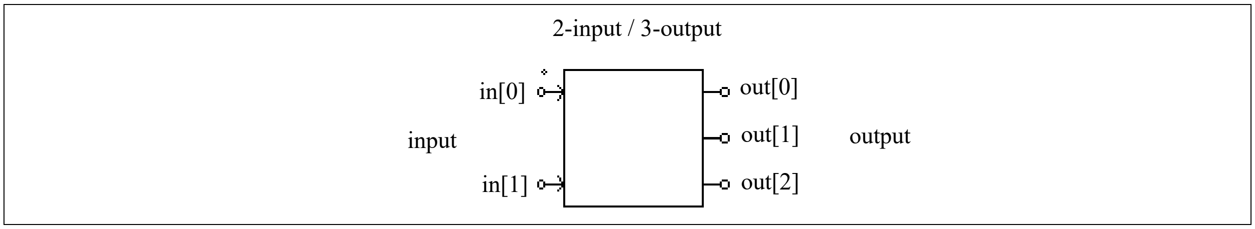
Simulation Step Function:This function is called at each simulation step.

SimulationBegin Function:This function is called only once at the beginning of simulation for initialization.

SimulationEnd Function:This function is called only once at the end of simulation for termination.

Click on the **Check Code** button to check if the code has any compiling errors. Click on the **Edit Image** button to customize the image of the C block.

For example, assume that a C block has 2 inputs and 3 outputs. After numbers of input/output ports are defined, the C block will have an image in PSIM as follows:



The node sequence is from the top to the bottom.

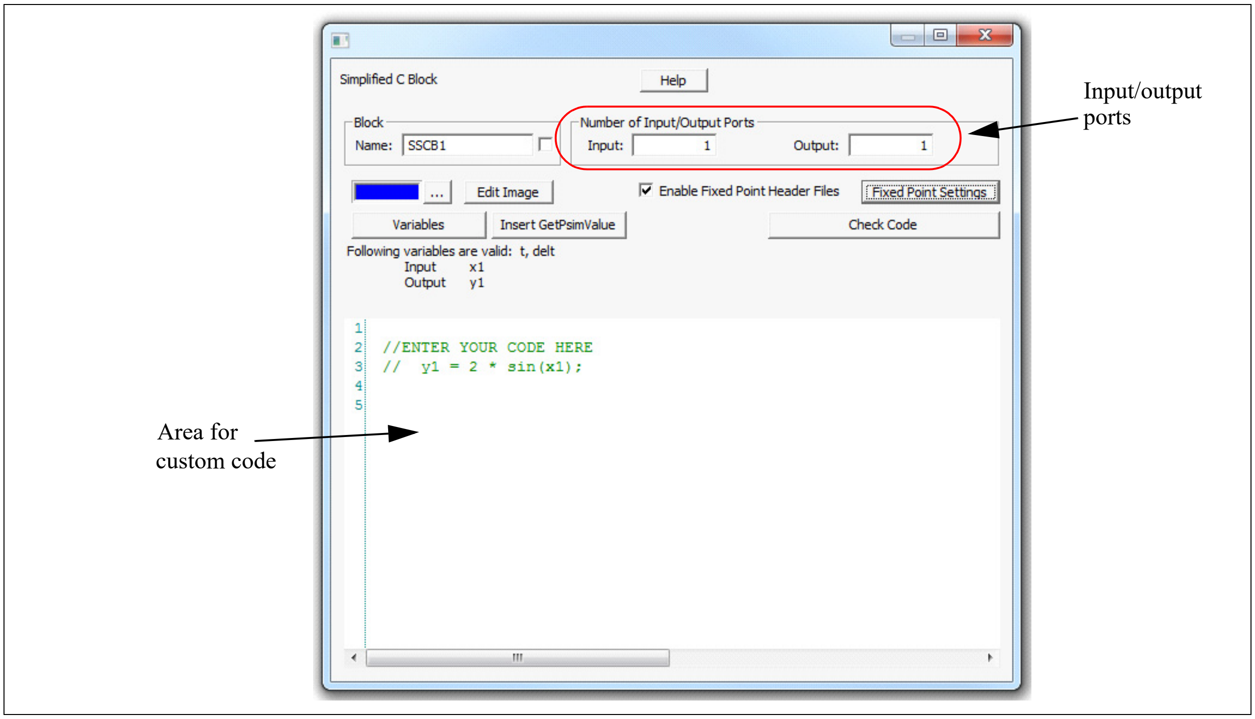
In C code, the *in* array is used to pass the values from PSIM into the C block, and the *out* array is used to pass the values from the C block back to PSIM. In this example, the first input, which corresponds to in[0] in the code, is the node at the top left, and the first output, which corresponds to out[0] in the code, is at the top right.

The difference between the C block and the external DLL block is that, even though the C block is easier to use, it does have the disadvantage that the custom code in the C block can not be debugged, while in the external DLL block it is possible to set break points and trace/step through the code for debugging.

6.4.6 Simplified C Block

A simplified C block is a special case of the C block. Unlike the C block that contains a variable/function definition section and three functions, the simplified C block contains only the C code that is called and ran at every time step, corresponding to the code for the *SimulationStep* function.

The interface of the simplified C block dialog window is shown below.



Name: SSCB1

Input: 1

Output: 1

☑ Enable Fixed Point Header Files

Following variables are valid: t, delt
Input x1
Output y1

```

//ENTER YOUR CODE HERE
//  y1 = 2 * sin(x1);


```

Input/output ports

Area for custom code

The number of input and output ports of the ports is defined in the *Number of Input/Output Ports* section. After the number of ports is changed, the image of the block in the schematic will change accordingly.

Variables that can be used in the code include:

t: Time passed from PSIM
delt:Time step passed from PSIM
x1, x2, ... :Inputs 1, 2, etc.
y1, y2, ... :Outputs 1, 2, etc.

When the checkbox "Enable Fixed Point Header Files" is checked, by clicking on **Fixed Point Settings**, one can define the fixed point data types of the C block inputs and outputs.

By clicking on the **Variables** button, one can define variables that can be passed from the main circuit into the C block. The variable can be a numerical value or a formula, for example:

k1 = 15.2
k1 = Freq/sqrt(2)

where Freq is a variable defined in a parameter file.

By clicking on the **Insert GetPsimValue** button, one can get the variables defined elsewhere and use them in the C code. The usage of this function is the same as in the C block.

Unlike the C block, the simplified C block can be used for automatic code generation.

6.4.7 External DLL Blocks

An external DLL (dynamic link library) block allows users to write code in C/C++, compile it into DLL using Microsoft Visual C/C++, and link it with PSIM. These blocks can be used in either the power circuit or the control circuit.

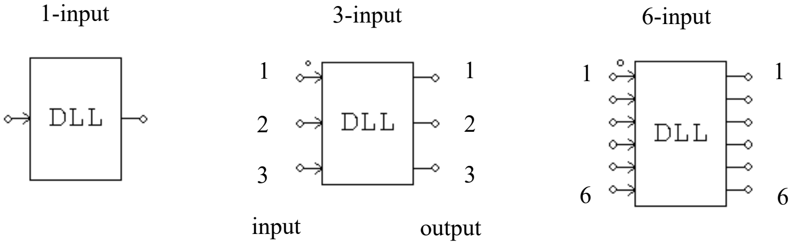
A DLL block receives values from PSIM as inputs, performs the calculation, and sends the results back to PSIM. PSIM calls the DLL routine at each simulation time step. However, when the inputs of the DLL block are connected to one of these discrete elements (zero-order hold, unit delay, discrete integrators and differentiators, z-domain transfer function blocks, and digital filters), the DLL block is called only at the discrete sampling times.

Two types of DLL blocks are provided: Simple DLL Block and General DLL Block. The simple DLL block has a fixed number of inputs and outputs, and the DLL file name is the only parameter that needs to be defined. On the other hand, the general DLL block allows users to define arbitrary number of inputs/outputs and additional parameters. Users can also customize the DLL block image.

The simple DLL block is easier to program and use.

Simple DLL blocks with 1 input/out up to 25 inputs/outputs are provided. The images and parameters of DLL blocks with 1, 3, and 6 inputs/outputs are shown below.

Images:



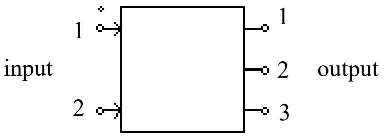
Attribute:

Parameter	Description
File Name	Name of the DLL file

The node with a dot is for the first input (in[0]). The input/output node sequence is from the top to the bottom.

The images and parameters of a general DLL block are shown below.

Image (for a block with 2 inputs and 3 outputs):



Attribute:

Parameter	Description
DLL File	Name of the DLL file
Input Data File	Name of the input data file that DLL reads (optional)
Number of Input Nodes	Number of input nodes (optional)
Number of Output Nodes	Number of output nodes (optional)
IN Nodes	List of input nodes (optional)

OUT Nodes	List of output nodes (optional)
Parameter 1	Parameter to be passed from PSIM into DLL (optional)
Parameter 2	Parameter to be passed from PSIM into DLL (optional)
Edit Image (button)	Click this button to edit and customize the image of the DLL block.
Display File (button)	Click this button to display the content of the *Input Data File* (optional).
Read File (button)	If the *Input Data File* is modified, click this button to reload the data file (optional).

The node with a dot is the first input (in[0]). The input/output node sequence is from the top to the bottom.

By default, users define the number of inputs and outputs. But the number of inputs and outputs, the node names, as well as the number of parameters and the parameter names can all be defined inside the DLL routine. For more details on defining and programming for the general DLL block, please refer to the help file "Help General DLL Block.pdf" and related examples.

The name of the DLL file can be arbitrary. The DLL file can be placed in one of the two places, in the order of precedence: in the PSIM directory, and in the same directory as the schematic file that uses the DLL file.

Note: When a DLL file is used in multiple DLL blocks in a schematic, if global or static variables are declared and used in the DLL code, these global/static variables will be the same and be shared among all the DLL blocks. If this is not what users intended, the DLL calculation can be incorrect, and users should avoid using global/static variables in the code in this case.

Several examples using the simple DLL blocks and the general DLL blocks are provided in the "examples\custom DLL" sub-folder in the PSIM directory.

6.4.8 Embedded Software Block

An Embedded Software Block is a special type of the external DLL blocks. It is intended for modeling embedded software devices such as microcontrollers and DSP.

Attribute:

Parameter	Description
DLL File	Name of the DLL file that defines the functionality and the interface of the block
Number of Nodes	Total number of input and output nodes

An Embedded Software Block has similar functionality as a general external DLL block. However, unlike the general DLL block whose connection nodes are predefined as either inputs or outputs, the Embedded Software Block allows the node types to be programmed as needed. Also, additional information, such as the exact instant at which the state of a variable changes, can be calculated and passed to and from PSIM.

The Embedded Software Block is a control circuit element, and can be used in the control circuit only.

For more information on how to use the Embedded Software Block, please refer to the document "Help Embedded Software Block.pdf".

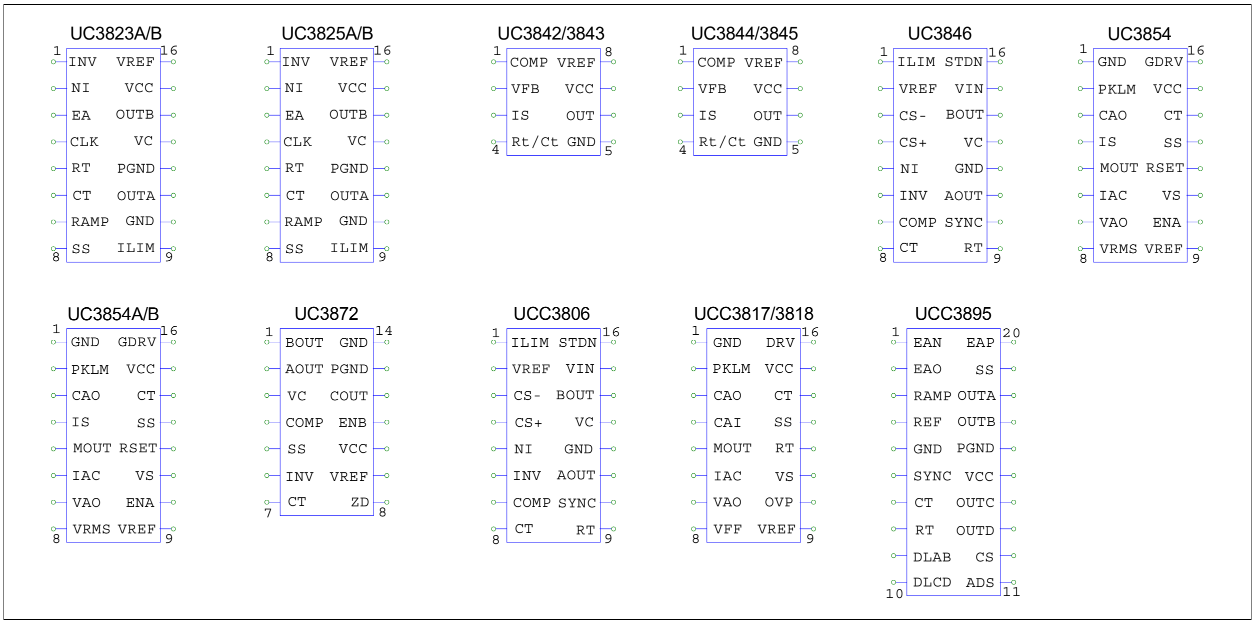
6.5 IC Models

Several PWM IC and driver IC models are provided.

6.5.1 PWM IC

The following PWM IC models are provided:

Images:



Attribute:

Parameter	Description
Model Level	All these IC models have two levels: - Level 1: The PWM gating outputs are control signals with logic level of 1 (high) and 0 (low) - Level 2: the PWM gating outputs are power signals which can directly drive three-state switches.

PWM IC Descriptions:

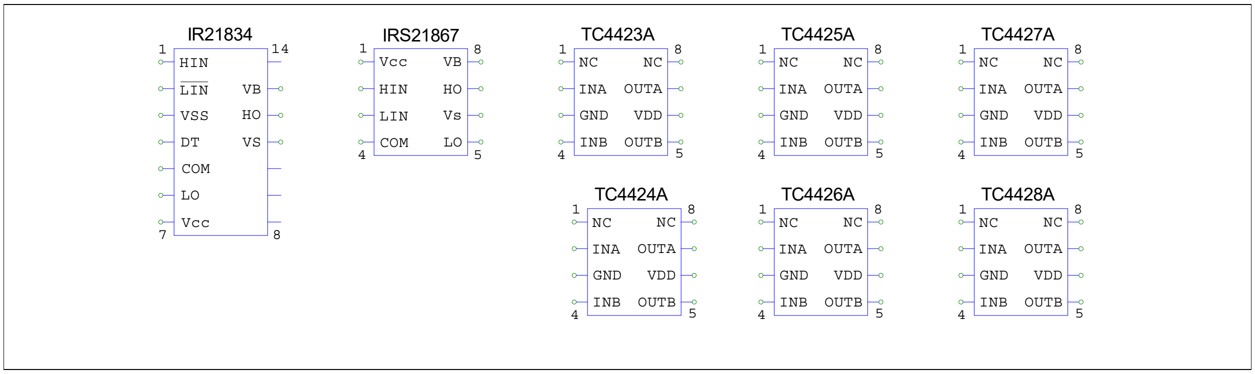
UC3823A/B, UC3825A/B	High speed PWM controller for high frequency switch-mode power supplies. The UC3823A and UC3823B and the UC3825A and UC3825B family of PWM controllers are improved versions of the standard UC3823 and UC3825 family.
UC3842/3843, UC3844/3845	Current mode fixed frequency PWM controller for off-line or dc-dc converters. The UC3842/3/4/5 family of control devices provides the necessary functions to implement off-line or dc-to-dc fixed frequency current mode control schemes.
UC3846	Current mode fixed frequency PWM controller for off-line or dc-dc converters. The UC3846 family of control ICs provides the necessary functions to implement fixed frequency, current mode control schemes.
UC3854, UC3854A/B	High power factor preregulator providing active power factor correction. UC3854A/B products are pin compatible enhanced versions of the UC3854. It provides all the functions necessary for active power factor corrected preregulators.

UC3872	Resonant lamp ballast controller. The UC3872 is a resonant lamp ballast controller optimized for driving cold cathode fluorescent, neon, and other gas discharge lamps.
UCC3806	Low power, dual output, current mode PWM controller. With the same block diagram and pinout of the popular UC3846 series, the UCC3806 line features increased switching frequency capability while reducing the bias current used within the device.
UCC3817/3818	BiCMOS power factor preregulator with average current mode control. The UCC3817/18 family provides all the functions necessary for active power factor corrected preregulators. The controller achieves near unity power factor by shaping the ac input line current waveform to correspond to that of the ac input line voltage.
UCC3895	BiCMOS advanced phase-shift PWM controller for full bridge power stage. The UCC3895 is a phase-shift PWM controller that implements control of a full-bridge power stage by phase shifting the switching of one half-bridge with respect to the other. It allows constant frequency pulse-width modulation in conjunction with resonant zero-voltage switching to provide high efficiency at high frequencies.

6.5.2 Driver IC

PSIM has implemented the following power switching devices' driver models:

Images:



Driver IC Description:

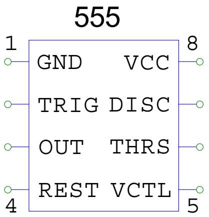
IR21834	Half-bridge driver for high voltage, high speed power MOSFET and IGBT with dependent high and low side referenced output channels.
IRS21867	High and low side driver for voltage, high speed power MOSFET and IGBT driver with independent high and low side referenced output channels.
TC4423A, TC4424A, TC4425A	3A dual output high-speed power MOSFET drivers. TC4423A: dual inverting. TC4424A: dual non-inverting. TC4425A: one inverting and one non-inverting.
TC4426A, TC4427A, TC4428A	1.5A dual output high-speed power MOSFET drivers. TC4426A: dual inverting. TC4427A: dual non-inverting. TC4428A: one inverting and one non-inverting.

6.5.3 555 Timer

The 555 Timer is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by external resistor and capacitor.

In PSIM model, the OUT signal of the 555 Timer is a control signal node.

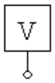
Images:



6.6 Initial Values

The initial node voltages of the power circuit and control circuit can be defined using this block. This value will not be kept as soon as the simulation started.

Image:



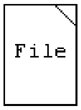
Attribute:

Parameter	Description
Initial Value	Initial value of the node voltage

6.7 Parameter File

A parameter file defines the variables used in a circuit. For example, the resistance of a resistor can be specified as R1, and R1 is defined in a parameter file. Also, a parameter file can be used by itself as a computational tool.

Image:



A parameter file is a text file created by a text editor. It supports the following format:

```
// Text from the character // to the end of the line is treated as comment
k1 = 12.3          // Define the value of the variable k1
k1 = a + b*c       // Define k1 in a math expression
(global) k1 = 12.3// Define k1 as a global variable (for use in SimCoder)
L1 = 3m            // power-of-ten suffix is supported. L1 = 3e-3
C1 = 100uF         // C1 = 100e-6. The unit "F" is ignored.
```

Note that, as compared to Version 9.1, the following format is no longer supported:

```
LIMIT var1 Vlow Vupper // This should be replaced by the "if...else if..." statement
% as comment. Now comments must start with double slash //. % is now used as modulo operation.
var1 value     (for example "R1 2.5"). Now an equal sign must be used to define a variable.
```

The definition "(global)" is used in SimCoder only for automatic code generation. Please refer to the SimCoder User Manual on how it is used. For PSIM simulation, this definition is ignored. That is, a parameter definition "(global) K1 = 1.2" is treated the same as "K1 = 1.2".

Conditional statements and error/warning message functions are supported, as shown below:

```
if (k1 > 10)
{
        a = 10
        b = 1
}
else if (k1 <= 20)
{
        a = 20
        b = 2
}
else
{
        Error {"Error: The value of k1 is out of the range. k1 = %f", k1}
}
if (k1 == 20)
{
        Warning {"Warning: The value of k1 is equal to %.0f", k1}
}
```

When writing the conditional statements "if...else if...else...", the standard C syntax applies (without the semicolon ; at the end of each statement though).

In addition, the while loop is supported. For example,

```
iflag = 1
a = 0
b = 0.1
while (iflag == 1)
{
        a = a + b
        if (a > 10)
                iflag = 0
}
```

Supported operators and mathematical functions in a parameter file are:

```
+                       // addition
-                       // subtraction
*                       // multiplication
/                       // division
%                       // modulo operator that returns the remainder after division (e.g. 5 % 2 = 1)
^                       // to the power of
=                       // equal assignment
==                      // conditional equal
!=                      // not equal
>                       // greater than
>=                      // equal or greater than
<                       // less than
<=                      // equal or less than
!                       // not
&&                      // and
||                      // or
sin(x)                  // sine
cos(x)                  // cosine
```

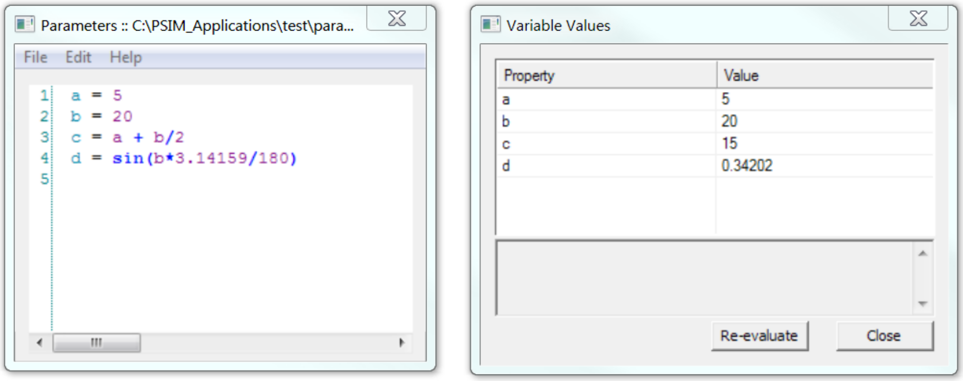
```
tan(x)                  // tangent
asin(x)                 // arcsine
acos(x)                 // arccosine
atan(x)                 // arctangent
atan2(y,x)              // arctangent with x and y defined
sinh(x)                 // hyperbolic sine
cosh(x)                 // hyperbolic cosine
tanh(x)                 // hyperbolic tangent
pow(x,y)                // x to the power of y
sqrt(x)                 // square root
exp(x)                  // base-e exponential of x, i.e. e^x
ln(x) (or log(x))       // natural logarithm of x (base e)
log10(x)                // common logarithm of x (base 10)
abs(x)                  // absolute
sign(x)                 // sign function that returns 1 if x > 0, -1 if x < 0, and 0 if x = 0
ceil(x)                 // function that returns the integer larger than x
floor(x)                // function that returns the integer smaller than x
if (...){...} else if (...){...} else {...}          // conditional if statement
iif (condition, value1, value2)                     // inline if statement (note it is "iif" not "if")
while (...) {...}                                    // while loop
Error {"Error text %f, %f", var1, var2}              // Error statement. Up to 5 variables are supported.
Warning {"Warning text %f, %f", var1, var2}  // Warning statement. Up to 5 variables are supported
```

Note that the input or output of all trigonometric functions is in rad. Also, the Error or Warning reporting functions can control the variable number format, as shown below.

Error {"Error message"}
Error {"Error message %.nf", k1}// Print the value of k1, where n is the number of digits after the
// decimal point. For example, if k1 = 12.34, %.1f will give 12.3
Warning {"Warning message"}
Warning {"Warning message %.nf", k1}// Print the value of k1, where n is the number of digits after
// the decimal point. For example, if k1 = 12.34, %.1f will give 12.3

An Error function will stop the simulation, while a Warning function will allow the simulation to proceed.

To display the variable values of a parameter file, go to **Edit** >> **Show Values**. For example, the figure below shows a parameter file on the left, and the values on the right.



```
a = 5
b = 20
c = a + b/2
d = sin(b*3.14159/180)
```

Property	Value
a	5
b	20
c	15
d	0.34202

The function **Show Values** provides a very convenient way of checking the calculation.

Parameter File in Circuit Schematic:

When a parameter file element is used in a circuit schematic, it has the same hierarchy as the circuit in which it resides. A variable defined in a parameter file in a higher-level circuit can be used in all lower circuits, but not the other way around. Also, a variable define in a parameter file in circuit cannot be used in another circuit of the same level.

For example, assume that a main circuit contain a parameter file "main_param.txt", and the main circuit contains two subcircuits S1 and S2, with a parameter file "sub1_param.txt" in S1 and another parameter file "sub2_param.txt" in S2. A variable defined in "main_param.txt" can be used in both S1 and S2 circuits as well as in parameter files "sub1_param.txt" and "sub2_param.txt". However, a variable defined in "sub1_param.txt" or "sub2_param.txt" can not be used in the higher level main circuit and in "main_param.txt".

Similarly, since Subcircuits S1 and S2 are parallel, a variable defined in "sub1_param.txt" cannot be used in Subcircuit S2 and "sub2_param.txt", and vice versa.

In addition, when PSIM saves a parameter file, it will save the information of both the relative path and absolute path of the parameter file, as well as a copy of the content of the parameter file.

When PSIM loads a schematic with a parameter file element, it will search and load the parameter file in the following order: relative path (relative to the schematic file), absolute path, and the local path (the local folder where the schematic file is). If PSIM still can not find the parameter file, it will create the parameter file in the local schematic folder from the copy that it saved in the schematic file.

Parameter Tool:

With the capability to show variable values and handle conditional statements, the parameter file element becomes a very useful computational tool in its own right. To facilitate its usage, a function is provided to open a parameter file without the need of a schematic. To access this function, go to Utilities >> Parameter Tool.

6.8 AC Analysis

6.8.1 AC Sweep

The frequency response of a circuit or a control loop can be obtained with the ac analysis. A key feature of the ac analysis in PSIM is that, a circuit can be in its original switchmode form, and no average model is required. Nevertheless, with the average model, it takes much shorter time to perform the ac analysis.

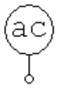
The following are the steps to set up the ac analysis:

- Identify a sinusoidal voltage source as the excitation source for the ac sweep.
- Place ac sweep probes at the desired output location. To measure the loop response of a closed control loop, use the node-to-node probe.
- Place the AC Sweep block on the schematic, and define the parameters of the ac sweep.
- Run the simulation.

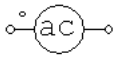
Below are the images of the ac sweep probes and the AC Sweep block.

Images:

AC Sweep Probe



AC Sweep Probe (loop)



AC Sweep

Attributes:

Parameters	Description
Start Frequency	Start frequency of the ac sweep, in Hz
End Frequency	End frequency of the ac sweep, in Hz
No. of Points	Number of data points
Flag for Points	Flag to define how the data points is generated. Flag = 0: Points are distributed linearly in LOG10 scale Flag = 1: Points are distributed linearly in linear scale
Source Name	Name of the excitation source

Start Amplitude	Excitation source amplitude at the start frequency
End Amplitude	Excitation source amplitude at the end frequency
Freq. for extra Points	Frequencies of additional data points. If the frequency-domain characteristics change rapidly at a certain frequency range, one can add extra points in this region to obtain better data resolution.

The principle of the ac analysis is that a small ac excitation signal is injected into the system as the perturbation, and the signal at the same frequency is extracted at the output. To obtain accurate ac analysis results, the excitation source amplitude must be set properly. The amplitude must be small enough so that the perturbation stays in the linear region. On the other hand, the excitation source amplitude must be large enough so that the output signal is not affected by numerical errors.

In general, a physical system has low attenuation in the low frequency range and high attenuation in the high frequency range. A good selection of the excitation source amplitude would be to have a relatively small amplitude at the low frequency, and a relatively large amplitude at the high frequency.

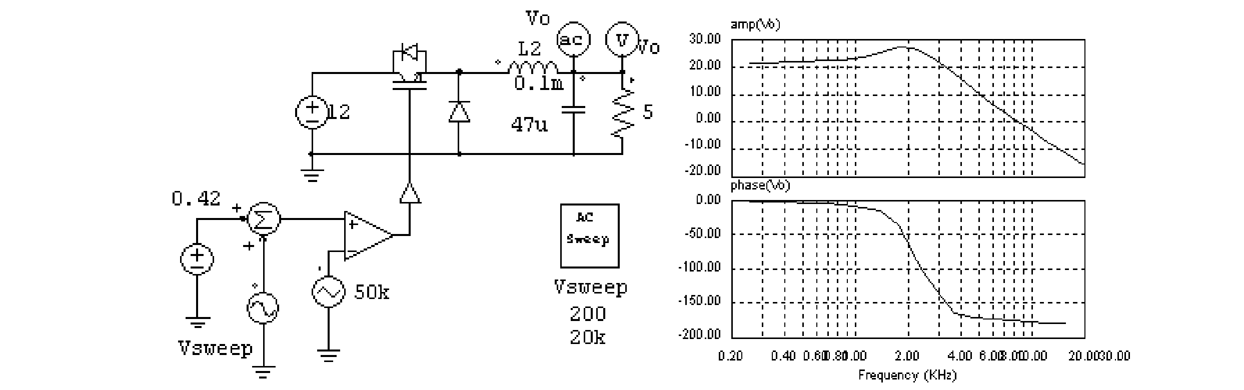
Sometimes, after ac analysis is complete, a warning message is displayed as follows:

Warning: The program did not reach the steady state after 60 cycles. See File “message.txt” for more details.

This message occurs when the software fails to detect the steady state at the ac sweep output after 60 cycles. To address this problem, one may increase damping in the circuit (by including parasitic resistances, for example), or adjust the excitation source amplitude, or reduce simulation time step. The file “message.txt” gives the information on the frequency at which this occurs and the relative error. The relative error will indicate how far the data point is from reaching the steady state.

Example: Open-Loop Response of a Buck Converter

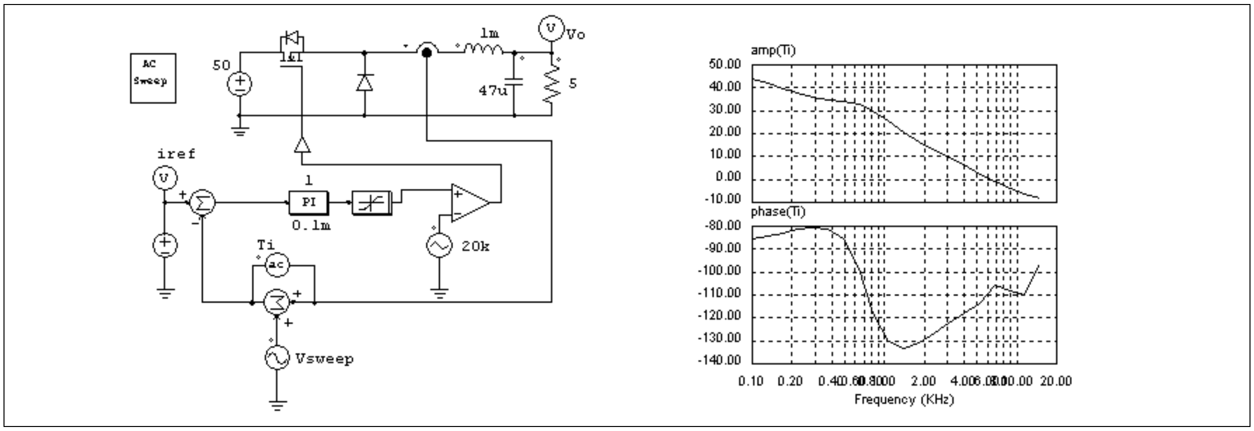
The circuit on the left is an one-quadrant buck converter. An excitation source is injected to the modulation signal, and the output voltage is measured. The result of the ac analysis, on the right, shows the open-loop response of the output voltage versus the modulation signal.



Example: Loop Transfer Function of a Closed-Loop Circuit

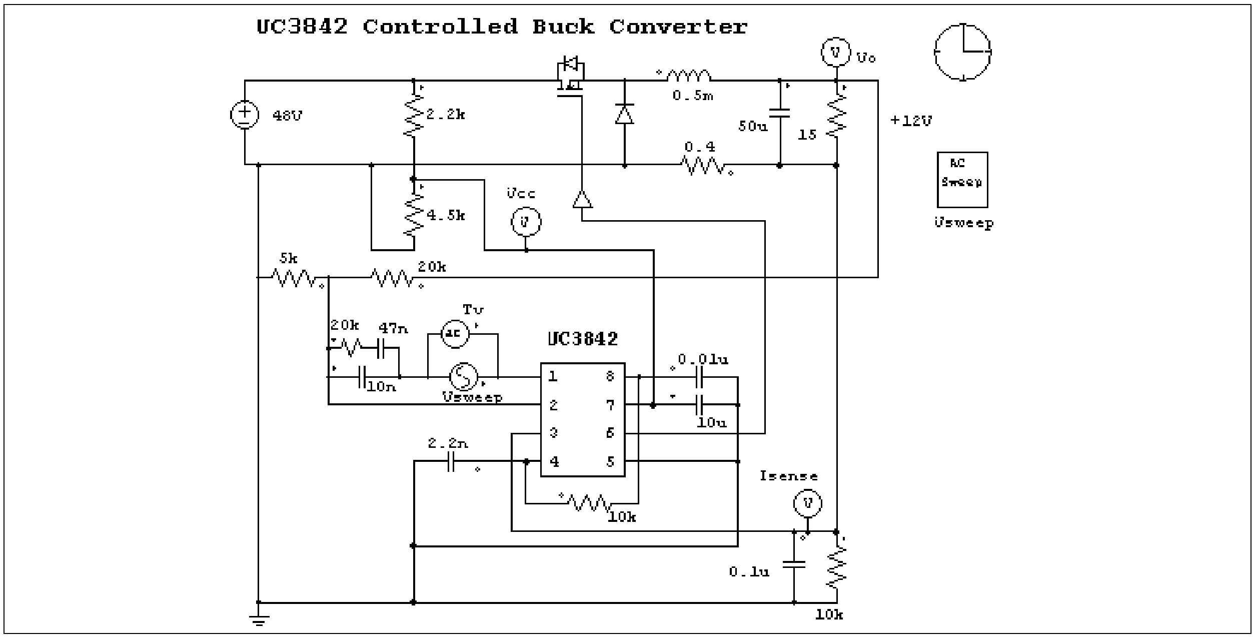
The ac analysis can be used to find out the loop response of a closed-loop system. The circuit below shows a buck converter with average current mode control. By injecting the excitation signal into the current feedback path, and using the node-to-node ac sweep probe, we can obtain the loop transfer function directly. With the loop transfer function, one can determine the bandwidth of the control loop and the phase margin.

Please note that the ac sweep probe should be connected such that the dotted side is connected to the node after the excitation source injection.



Example: Loop Transfer Function of a Switchmode Power Supply

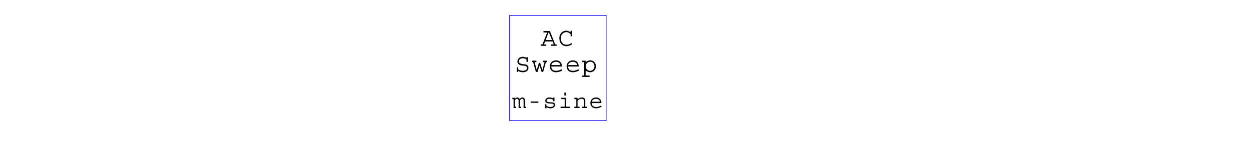
The loop transfer function of a switchmode power supply controlled by a PWM IC can also be determined in a similar way. The figure below shows a buck converter controlled by TI UC3842. The excitation source can be inserted in the feedback path, before the op. amp. output.



6.8.2 AC Sweep Multisine

The AC Sweep (multisine) block has the same functionality as the AC sweep. Its advantage is to simulate multiple frequency of sine-wave in only one cycle of the lowest frequency wave, resulting faster simulation.

Image:



Attributes:

Parameters	Description
Start Frequency	Start frequency of the ac sweep, in Hz
End Frequency	End frequency of the ac sweep, in Hz
No. of Points	Number of data points
Flag for Points	Flag to define how the data points is generated. Flag = 0: Points are distributed linearly in LOG10 scale Flag = 1: Points are distributed uniformly in LOG10 scale
Source Name	Name of the excitation source
Source Amplitude	Excitation source peak amplitude at the start frequency.
Number of Cycles	The number of cycles of the excitation source amplitude to be simulated
Steady State Time	The time predicted when the circuit has completed transient and reached steady state, in second. User may make approximated estimation of this value or run a single time simulation to determine it.

6.9 Parameter Sweep

Parameter sweep can be performed on any parameters:

The image and parameters of the parameter sweep block are shown below.

Image:

Attributes:

Parameters	Description
Name	Name of the parameter to be swept
Start	Starting value of the parameter
End	End value of the parameter
Increment	Increment step value of the parameter
Enabled	When checked, the parameter line will be enabled.

For example, let the resistance of a resistor be “Ro”. To sweep the resistance from 2 Ohm to 10 Ohm, with an increment step value of 2 Ohm, check the Enabled checkbox, and define the following:

Name	Ro
Start	2
End	10
Increment	2

Parameter sweep will produce two plots: One is the outputs versus time, and the other is the outputs at the last simulation point versus the swept parameter.

For example, a circuit has two outputs V1 and V2, and the resistance Ro is swept. The total simulation time is

0.1 sec. After the simulation, there will be two plots in Simview. One is V1 and V2 versus time. The other is V1 and V2 versus Ro. The V1 and V2 values used in the second plot are the values at the last simulation point at 0.1 sec.

7 Sources

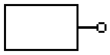
Several types of independent voltage/current sources are available in PSIM. The notation of a current source direction is: the current flows out of the higher-potential node, through the external circuit, and back into the lower-potential node of the source.

Note that current sources can be used in the power circuit only.

7.1 Constant

The constant element allows users to define a constant. It behaviors as a grounded voltage source.

Image:



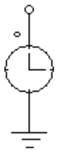
Attribute:

Parameter	Description
Value	Value of the constant

7.2 Time

The Time element is a special case of the piecewise linear voltage source. It is treated as a grounded voltage source, and the value is equal to the simulation time, in sec.

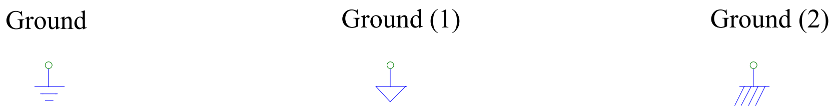
Image:



7.3 Ground

There are three different ground elements in PSIM library. Although they are in different shape, they are electrically connected together. The different shape of these grounds provide the convenience for user to separate the grounding in different functional sections of the circuits.

Images:

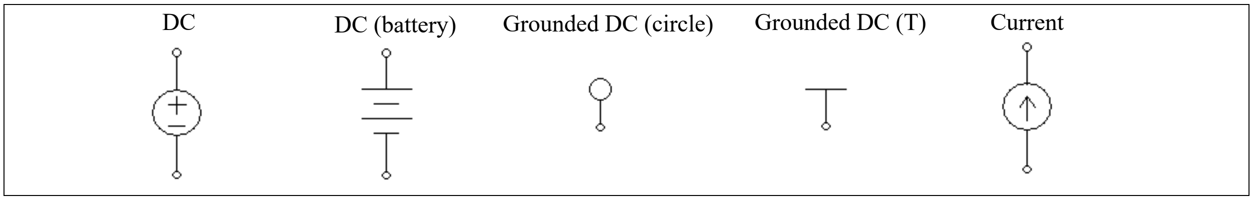


7.4 Voltage and Current Sources

7.4.1 DC Source

A dc source has a constant amplitude. The reference of the grounded dc voltage sources is the ground.

Images:



Attribute:

Parameter	Description
Amplitude	Amplitude of the source
Series Resistance	Source series resistance, in Ohm (for *DC* and *DC* (*battery*) voltage sources only)
Series Inductance	Source series Inductance, in H (for *DC* and *DC* (*battery*) voltage sources only)

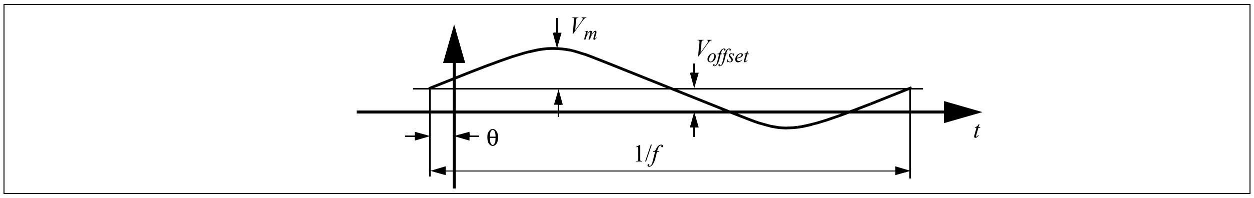
The series resistance and inductance represent the source impedance.

7.4.2 Sinusoidal Source

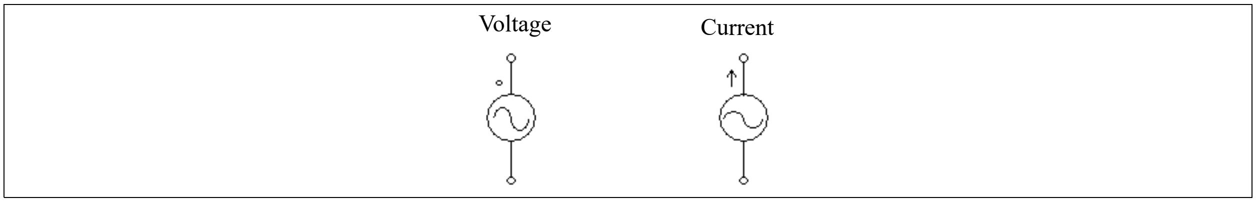
A sinusoidal source is defined as:

$$v_o = V_m \cdot \sin(2\pi \cdot f \cdot t + \theta) + V_{offset}$$

The specifications can be illustrated as follows.



Images:



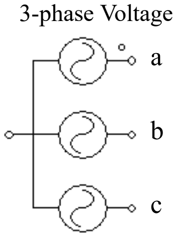
Attributes:

Parameters	Description
Peak Amplitude	Peak amplitude V_m
Frequency	Frequency *f*, in Hz
Phase Angle	Initial phase angle θ, in deg.
DC Offset	DC offset V_{offset}

Series Resistance	Source series resistance, in Ohm (for voltage source only)
Series Inductance	Source series inductance, in H (for voltage source only)
Tstart	Starting time, in sec. Before this time, the source is 0.

To facilitate the setup of three-phase circuits, a symmetrical three-phase Y-connected sinusoidal voltage source is provided. The dotted phase of the source refers to Phase A.

Image:



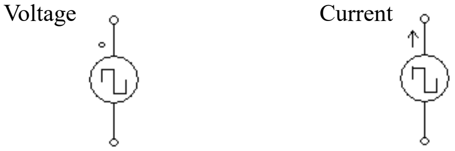
Attributes:

Parameters	Description
V (line-line rms)	Line-to-line rms voltage amplitude
Frequency	Frequency *f*, in Hz
Initial Angle (phase A)	Initial angle for Phase A, in deg.
Series Resistance	Source series resistance, in Ohm
Series Inductance	Source series inductance, in H

7.4.3 Square-Wave Source

A square-wave voltage source or current source is defined by peak-to-peak amplitude, frequency, duty-cycle, and DC offset. The duty cycle is defined as the ratio between the high-potential interval versus the period.

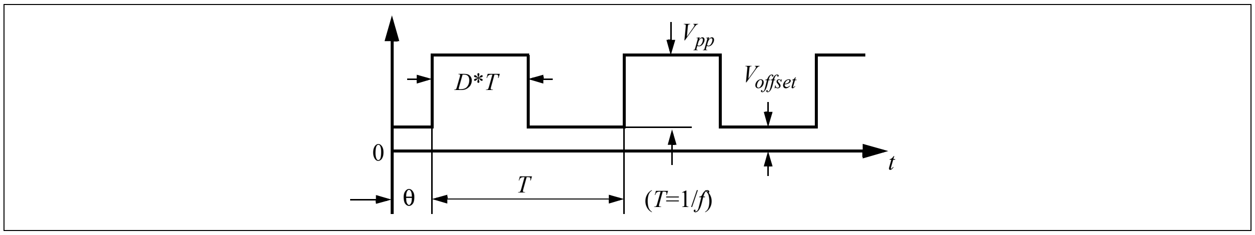
Images:



Attributes:

Parameters	Description
Vpeak-peak	Peak-to-peak amplitude V_{pp}
Frequency	Frequency, in Hz
Duty Cycle	Duty cycle *D* of the high-potential interval
DC Offset	DC offset V_{offset}
Phase Delay	Phase delay θ of the waveform, in deg.
Rise Time (SPICE)	Rise time from low to high, in sec. (for SPICE only)
Fall Time (SPICE)	Fall time from high to low, in sec. (for SPICE only)

The specifications of a square wave source are illustrated as follows.

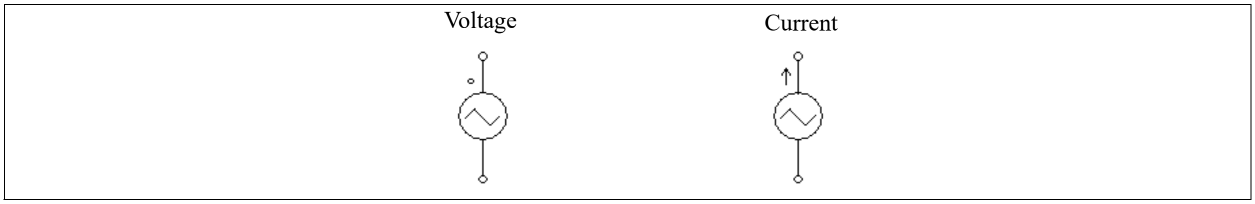


When the phase delay θ is positive, the waveform is shifted to the right along the time axis.

7.4.4 Triangular/Sawtooth Sources

A triangular-wave voltage source or current source is defined by peak-to-peak amplitude, frequency, duty-cycle, and DC offset. The duty cycle is defined as the ratio between the rising-slope interval versus the period.

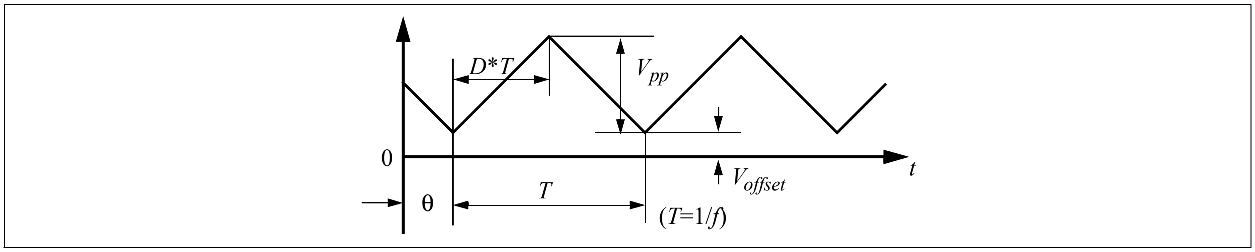
Images:



Attributes:

Parameters	Description
Vpeak-peak	Peak-to-peak amplitude V_{pp}
Frequency	Frequency, in Hz
Duty Cycle	Duty cycle D of the rising slope interval
DC Offset	DC offset V_{offset}
Phase Delay	Phase delay θ of the waveform, in deg.

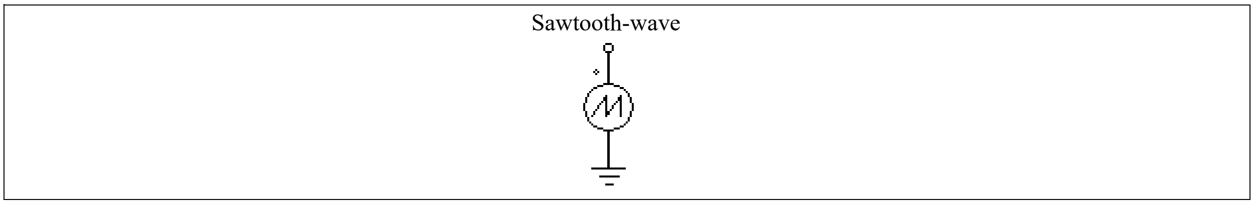
The specifications of a triangular wave source are illustrated as:



When the phase delay θ is positive, the waveform is shifted to the right along the time axis.

A sawtooth-wave voltage source is a special case of the triangular source with the duty cycle of 1 and dc offset and phase delay of 0, and with one node grounded. It is defined as below:

Image:



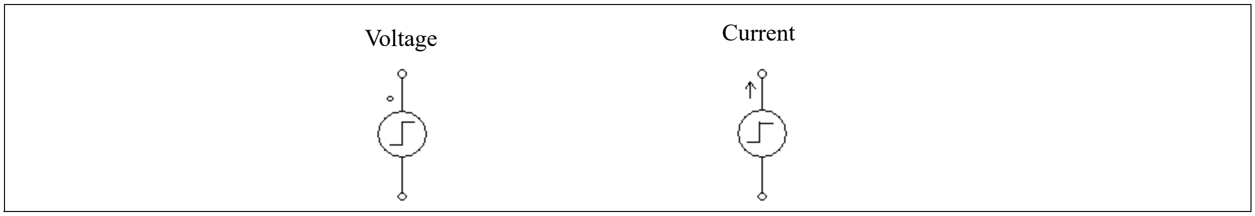
Attributes:

Parameters	Description
Vpeak	Peak amplitude
Frequency	Frequency, in Hz
Fall Time (SPICE)	Fall time, in sec. (for SPICE only)

7.4.5 Step Sources

A step voltage/current source changes from one level to another at a given time. Two types of step sources are provided: one that changes from 0 to a certain level (refer to as *Step* in the library), and the other that changes from one level to another level (referred to as *Step* (*2-level*) in the library).

Images:



Attributes:

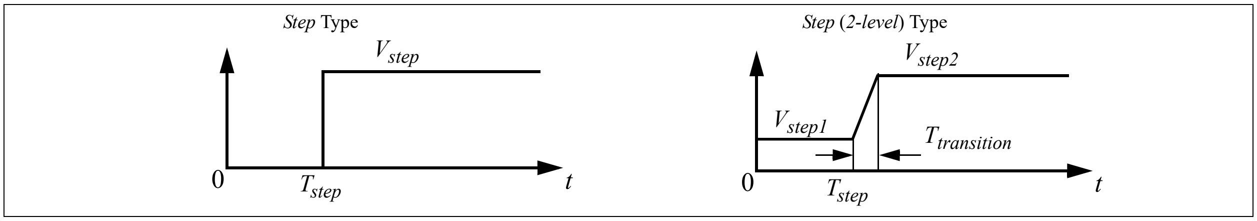
For the *Step* type source:

Parameters	Description
Vstep	Value V_{step} after the step change
Tstep	Time T_{step} at which the step change occurs
Transition Time (SPICE)	Transition time from low to high, in sec. (for SPICE only)

For the *Step* (*2-level*) type source:

Parameters	Description
Vstep1	Value V_{step1} before the step change
Vstep2	Value V_{step2} after the step change
Tstep	Time T_{step} at which the step change occurs
Transition Time	Transition time $T_{transition}$ from V_{step1} to V_{step2}

The specifications of the voltage step sources are illustrated as follows:

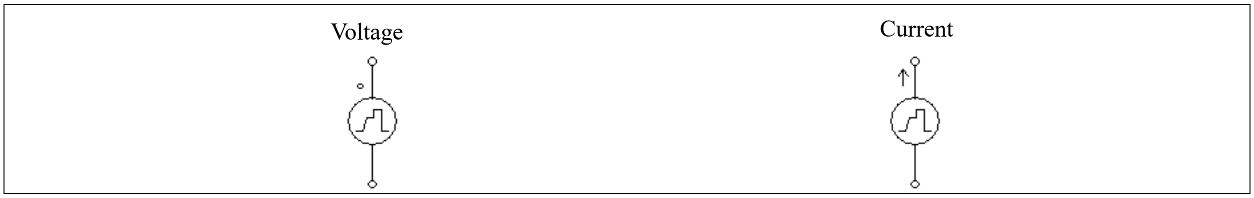


For SPICE, the transition time cannot be 0, and must be a non-zero value.

7.4.6 Piecewise Linear Source

The waveform of a piecewise linear source consists of piecewise linear segments. It is defined by the number of points, the values and the corresponding time (in sec.). The values and times can be entered either separately, or in pair.

Images:



Attributes:

For the sources that define the values and times separately:

Parameters	Description
Frequency	Frequency of the waveform, in Hz
No. of Points n	No. of points
Values V1...Vn	Values at each point
Time T1...Tn	Time at each point, in sec.

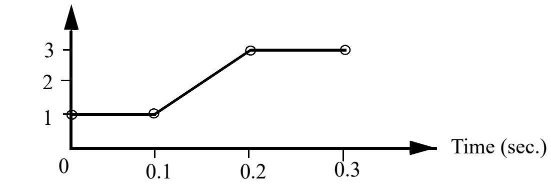
For the sources that define the values and times in pair:

Parameters	Description
Frequency	Frequency of the waveform, in Hz
Times, Values (t1,v1) ...	Time and value at each point

The time and value pair must be enclosed by left and right brackets. The time and value can be separated by either a comma, such as (1.2m,5.5), or a space, such as (1.2m 5.5), or both, such as (1.2m, 5.5).

Example:

The following is a non-periodic piecewise linear source. It has 3 segments which can be defined by four points (marked in the figure).



The specification for the *piecewise linear* voltage source will be:

Frequency	0.
No. of Points n	4
Values V1...Vn	1. 1. 3. 3.
Times T1...Tn	0. 0.1 0.2 0.3

The specification for the *piecewise linear* (*in pair*) voltage source will be:

Frequency	0.
Times, Values (t1,v1)...	(0., 1) (0.1, 1) (0.2, 3) (0.3, 3)

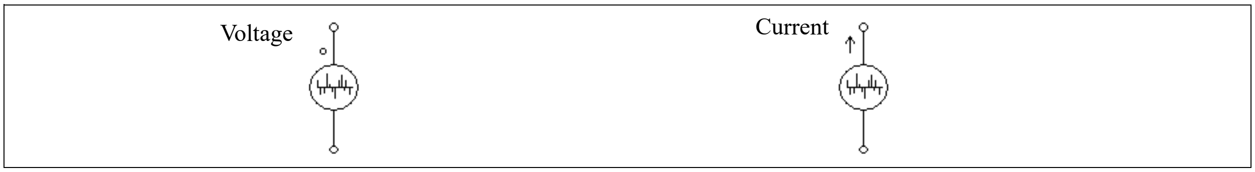
7.4.7 Random Source

The amplitude of a random voltage source (VRAND) or current source (IRAND) is determined randomly at each simulation time step. A random source is defined as:

$$v_o = V_m \cdot n + V_{offset}$$

where V_m is the peak-to-peak amplitude of the source, n is a random number in the range of 0 to 1, and V_{offset} is the dc offset.

Images:



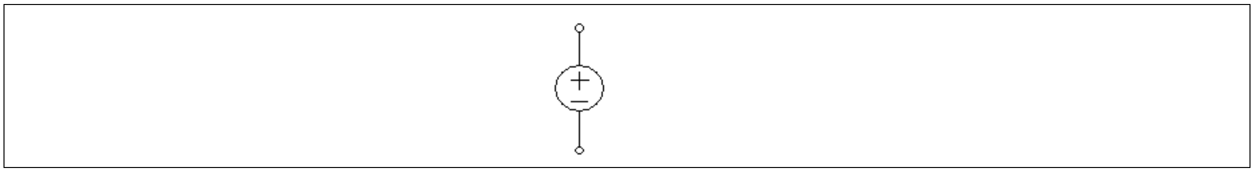
Attributes:

Parameters	Description
Peak-Peak Amplitude	Peak-to-peak amplitude of the source
DC Offset	DC offset
Duration (SPICE)	Duration of an individual voltage value, in sec. (for SPICE only)

7.4.8 Math Function Source

A math function source allows one to define the source in a mathematical expression.

Image:



Attributes:

Parameters	Description
Expression	The mathematical expression of the source
Tstart	Start time of the source

In the expression, “T” or “t” represents time. For example, to implement a sinusoidal source, the expression will be: sin(2*3.14159*60*t+2.09).

7.4.9 Voltage/Current-Controlled Sources

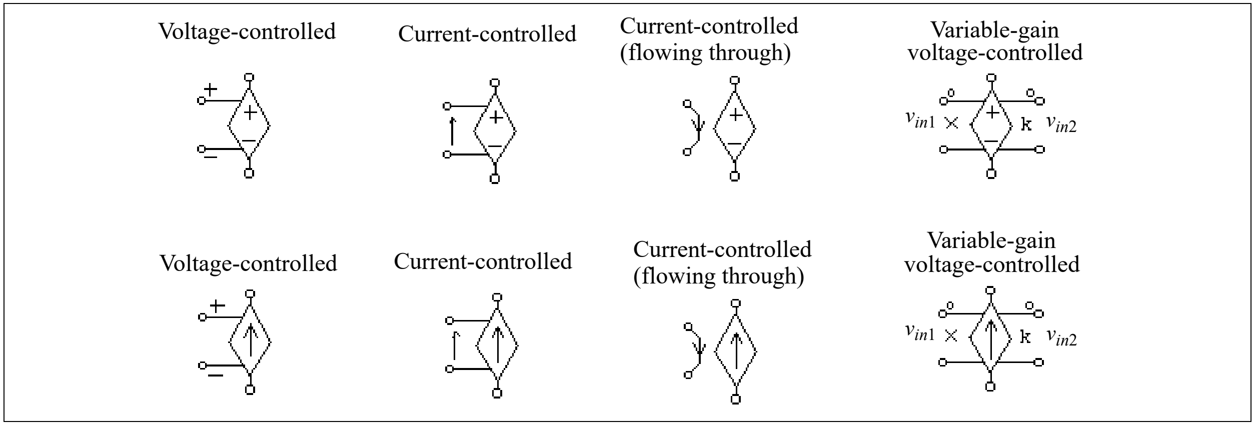
The following types of controlled sources are available:

- Voltage controlled voltage source
- Current controlled voltage source
- Voltage controlled current source
- Current controlled current source
- Variable-gain voltage controlled voltage source
- Variable-gain voltage controlled current source

The controlling current of a current controlled source must come from a RLC branch. Also, for a controlled current source, the controlling voltage or current can not be an independent source.

Note that controlled sources can be used in the power circuit only.

Images:


Voltage-controlled
Current-controlled
Current-controlled (flowing through)
Variable-gain voltage-controlled
v_{in1} × k v_{in2}
Voltage-controlled
Current-controlled
Current-controlled (flowing through)
Variable-gain voltage-controlled
v_{in1} × k v_{in2}

Attribute:

Parameter	Description
Gain	Gain of the source

For voltage-controlled sources, the controlling voltage is from the positive node (+) to the negative node (-).

For current-controlled sources (with an arrow pointing from one node to another), the control nodes are connected across a RLC branch, and the direction of the controlling current is indicated by the arrow.

For current-controlled sources (with a wire connecting the two nodes), the controlling current flows from one control node to another, as indicated by the arrow. A 10-uOhm resistor is used to sense the controlling current.

The output of a controlled source, except variable-gain controlled sources, is equal to the gain multiplied by the controlling voltage or current. For the variable-gain controlled sources, the output is equal to the following:

$$v_o = (k \cdot v_{in2}) \cdot v_{in1}$$

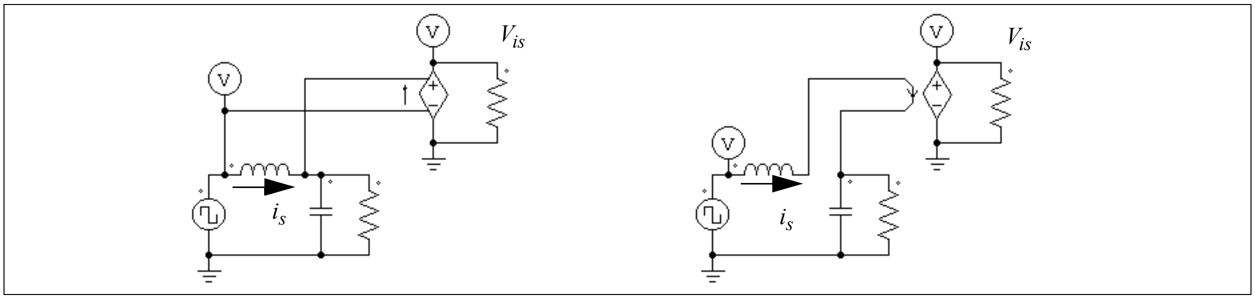
$$i_o = (k \cdot v_{in2}) \cdot v_{in1}$$

Input 1 is on the side with the multiplication sign, and Input 2 is on the side with the letter *k*.

The difference between a variable-gain controlled source and a nonlinear source with multiplication is that, for the nonlinear source with multiplication, values of both v_{in1} and v_{in2} at the current time step are used to calculate the output and are updated in each iteration. But for the variable-gain controlled source, it is assumed that the change of v_{in2} is small from one time step to the next, and the value of v_{in2} at the previous time step is used at the current time step. This assumption is valid as long as v_{in2} changes at a much slower rate as compared to v_{in1} and the time step is small as compared to the change of v_{in2}. Variable-gain controlled sources can be used in circuits which may otherwise have convergence problem with nonlinear sources with multiplication.

Example:

The circuits below illustrates the use of current controlled voltage sources. In the circuit on the left, the voltage source is controlled by the inductor current i_s. With a gain of 1, the waveform of the voltage v_{is} is equal to that of i_s. In this way, a current quantity can be converted to a voltage quantity. These two circuits ares equivalent.



7.4.10 Nonlinear Voltage-Controlled Sources

The output of a nonlinear voltage-controlled source is either the multiplication, division, or square-root of the inputs. They are defined as:

Nonlinear (multiplication):Output $v_o = k \cdot v_{in1} \cdot v_{in2}$ or $i_o = k \cdot v_{in1} \cdot v_{in2}$

Nonlinear (division):Output $v_o = k \cdot \frac{v_{in1}}{v_{in2}}$ or $i_o = k \cdot \frac{v_{in1}}{v_{in2}}$

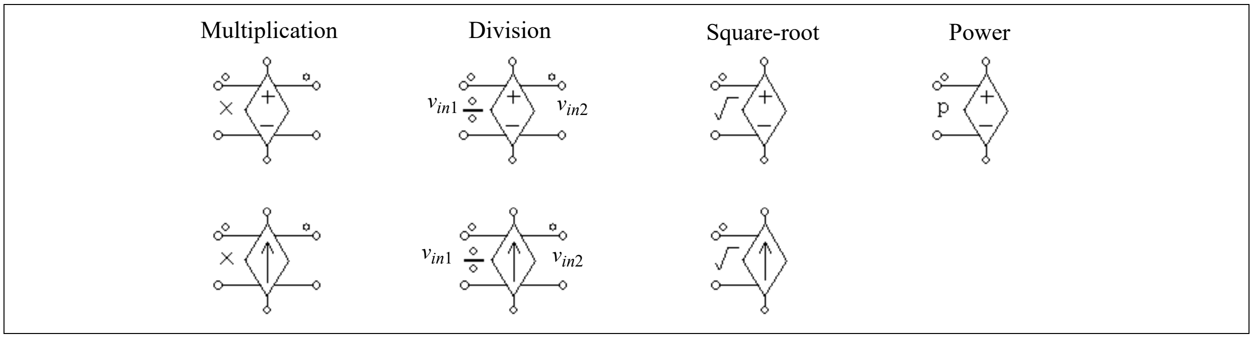
Nonlinear (square-root):Output $v_o = k \cdot \sqrt{v_{in1}}$ or $i_o = k \cdot \sqrt{v_{in1}}$

Nonlinear (power):Output $v_o = sign(v_{in}) \cdot k \cdot (k_1 \cdot v_{in})^{k_2}$

In the nonlinear power source, the term sign(v_{in}) is 1 if v_{in} is positive, and it is -1 if v_{in} is negative.

Note that these nonlinear sources can be used in the power circuit only.

Images:



Attributes:

For all the sources except the nonlinear power source:

Parameter	Description
Gain	Gain *k* of the source

For the nonlinear power source:

Parameters	Description
Gain	Gain *k* of the source
Coefficient k_1	Coefficient k_1
Coefficient k_2	Coefficient k_2

For the nonlinear (division) source, Input 1 is on the side of the division sign.

8

Error/Warning Messages and Other Simulation Issues

8.1 Simulation Issues

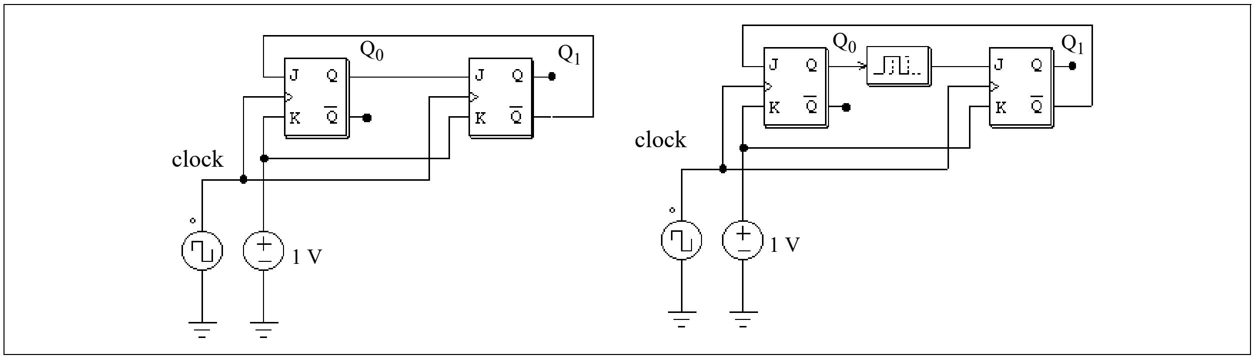
8.1.1 Time Step Selection

PSIM uses the fixed time step in the simulation. In order to assure accurate results, the simulation time step should be properly chosen. The factors that limit the time step in a circuit include the switching period, widths of pulses or square waveforms, and intervals of fast transients. It is recommended that the time step should be at least one magnitude smaller than the smallest of the above.

8.1.2 Propagation Delays in Logic Circuits

The logic elements in PSIM are ideal, i.e. there is no propagation delay. If a logic circuit uses the propagation delays for its operation, a function block in PSIM, called the Time Delay block, needs to be added to represent the effect of the propagation delay.

To illustrate this, take a two-bit counter circuit as an example.



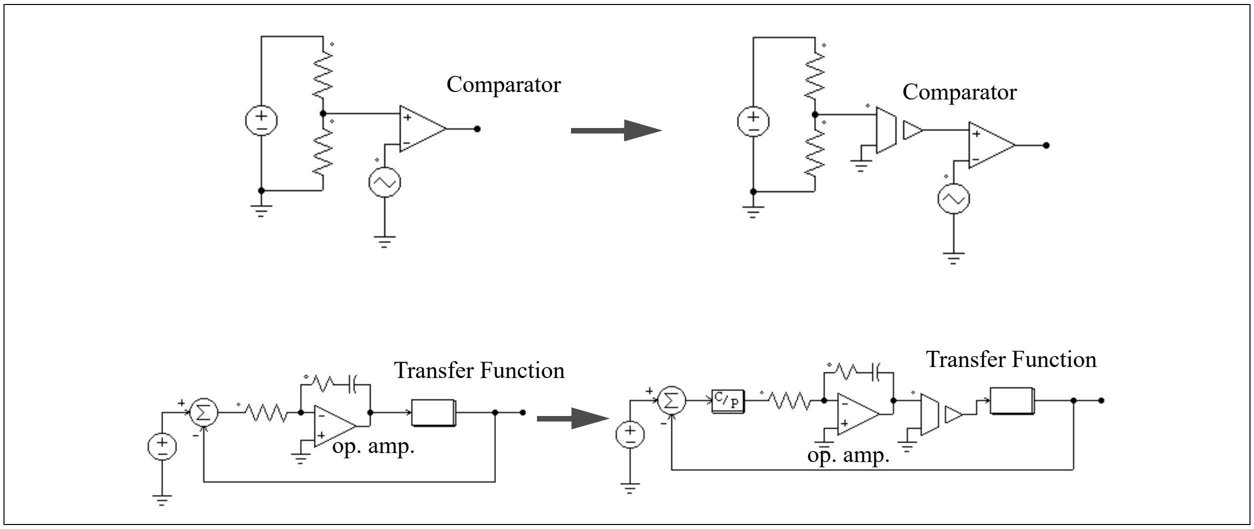
In the circuit on the left, the initial values of both Q_0 and Q_1 are assumed to be zero. At the clock rising edge, Q_0 will change to 1. Without delay, the position of Q_1, which should remain at 0, will toggle to 1 at the same time.

To prevent this, a time delay element with the delay period of one time step needs to be inserted between Q_0 and the input (J) of the second flip-flop.

8.1.3 Interface Between Power and Control Circuits

In PSIM, power circuits are represented in the discrete circuit form, and control circuits are represented in function block diagram. Power circuit components, such as RLC branches, switches, transformers, mutual inductors, current sources, floating voltage sources, and all types of controlled sources are not allowed in the control circuit. Similarly, control circuit components, such as logic gates, PI controllers, lookup tables, and other function blocks, are not allowed in the power circuit.

If there is a direct connection between the power circuit and the input of a control circuit element, a voltage sensor will be automatically inserted by the program. Similarly, if there is a direct connection between the output of a control circuit element and the power circuit, a control-power interface block will be automatically inserted. This is illustrated in the examples below.



It should be noted that, in PSIM, the power circuit and the control circuit are solved separately. There is one time step delay between the power and the control circuit solutions.

8.1.4 FFT Analysis

When using FFT for the harmonic analysis, one should make sure that the following requirements are satisfied:

- The waveforms have reached the steady state;
- The length of the data selected for FFT should be the multiple integer of the fundamental period.

For a 60-Hz waveform, for example, the data length should be restricted to 16.67 msec. (or multiples of 16.67 msec.). Otherwise, the FFT results will be incorrect. The data is selected by clicking on **X Axis** in SIMVIEW, de-selecting **Auto-scale** in **Range**, and specifying the starting time and the final time. The FFT analysis is only performed on the data that are displayed on the screen.

Note that the FFT results are discrete. The FFT results are determined by the time interval between two consecutive data points, Δt, and the data length T_{length}. The data point interval Δt is equal to the simulation time step multiplied by the print step. In the FFT results, the frequency incremental step will be $1/T_{length}$, and the maximum frequency will be 1/(2*Δt).

For example, if you take the FFT of a 1-kHz square waveform with a data length of 1 ms and a data point interval of 10 us, that is, T_{length} = 1 ms, and Δt = 10 us, the frequency incremental step will be: Δf = $1/T_{length}$ = 1 kHz. The maximum frequency will be: f_{max} = 1/(2*Δt) = 50 kHz.

8.2 Debugging

Some of the approaches in debugging a circuit is discussed in the following.

Symptom:

Simulation results show sudden changes (discontinuity) of inductor currents and capacitor voltages.

Solution:

This may be caused by the interruption of inductor current path and short-circuit of capacitor (or capacitor-voltage source) loops. Check the switch gating signals. If necessary, include overlap or dead time pulses to avoid open-circuit or shooting-through.

If an initial current is assigned to an inductor, initial switch positions should be set such that a path is provided for the current flow. Otherwise, the inductor current will be forced to start from zero.

Symptom:

Simulation waveforms look incorrect or inaccurate, or the waveform resolution is poor.

Solution:

This may be caused by two reasons. One is the time step. Since PSIM uses the fixed time step during the entire simulation, one should make sure that the time step is sufficiently small. As a rule of thumb, the time step should be several tens times smaller than the switching period.

Another reason is the problem of waveform display. One should make sure that the print step is not too big. To display all the data points, set the print step to 1.

8.3 Error/Warning Messages

The error and warning messages are listed in the following.

E-1 Input format errors occurred in the simulation.

It may be caused by one of the following:

- Incorrect/Incomplete specifications
- Wrong input for integers and character strings

Make sure that the PSIM library is not modified, and the PSIM simulator is up-to-date.

In the circuit file, character strings should be included between two apostrophes (like ‘test’). Also, make sure an integer is specified for an integer variable. The specification of a real number (like 3. instead of 3) for an integer will trigger the error message.

E-2 Error message: The node of an element is floating.

This can also be caused by a poor connection in PSIM. When drawing a wire between two nodes, make sure that the wire is connected to the terminal of the element.

W-1 Warning!!! The program failed to converge after 10 iterations when determining switch positions. The computation continues with the following switch positions: ... ...

This warning occurs when the program fails to converge when determining switching positions. Since the computation continues based on the switch positions at the end of the 10th iteration, results could be inaccurate. One should be cautious when analyzing the results.

There are many factors that cause this problem. The following measures can be taken to isolate and solve the problem:

- Check the circuit and make sure the circuit is correct.
- Check the switch gating signals.
- Connect small resistors/inductors in series with switches and voltage sources.

W-2 Warning!!! The program did not reach the steady state after 60 cycles when performing the ac sweep.

This warning occurs when the program fails to reach the steady state after 60 cycles when performing the ac sweep. The cause of the problem could be that the system is poorly damped at that particular frequency or the signal amplitude is too small.

You may try the following to isolate and solve the problem:

- Run the time-domain simulation with the excitation source at that frequency and see if time-domain waveforms are oscillatory.
- Increase the excitation voltage amplitude for larger signal level, or
- Reduce the time step for better accuracy and resolution.

Index