

# **PSIM**

---

## **User's Guide**

**Powersim Inc.**

# **PSIM User's Guide**

**Version 7.0**

Release 4

June 2006

Copyright © 2001-2006 Powersim Inc.

All rights reserved. No part of this manual may be photocopied or reproduced in any form or by any means without the written permission of Powersim Inc.

## **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](mailto:info@powersimtech.com)

<http://www.powersimtech.com>

---

# Contents

## 1 General Information

- 1.1 Introduction 1
- 1.2 Circuit Structure 2
- 1.3 Software/Hardware Requirement 3
- 1.4 Installing the Program 3
- 1.5 Simulating a Circuit 4
- 1.6 Component Parameter Specification and Format 4

## 2 Power Circuit Components

- 2.1 Resistor-Inductor-Capacitor Branches 7
  - 2.1.1 Resistors, Inductors, and Capacitors 7
  - 2.1.2 Rheostat 8
  - 2.1.3 Saturable Inductor 9
  - 2.1.4 Nonlinear Elements 10
- 2.2 Switches 11
  - 2.2.1 Diode, DIAC, and Zener Diode 12
  - 2.2.2 Thyristor and TRIAC 13
  - 2.2.3 GTO, Transistors, and Bi-Directional Switch 15
  - 2.2.4 Linear Switches 17
  - 2.2.5 Switch Gating Block 19
  - 2.2.6 Single-Phase Switch Modules 21
  - 2.2.7 Three-Phase Switch Modules 22
- 2.3 Coupled Inductors 24
- 2.4 Transformers 26
  - 2.4.1 Ideal Transformer 26

2.4.2	Single-Phase Transformers	26
2.4.3	Three-Phase Transformers	29
2.5	Magnetic Elements	30
2.5.1	Winding	30
2.5.2	Leakage Flux Path	31
2.5.3	Air Gap	32
2.5.4	Linear Core	34
2.5.5	Saturable Core	34
2.6	Other Elements	36
2.6.1	Operational Amplifier	36
2.6.2	dv/dt Block	37
2.6.3	Power Modeling Block	38
2.7	Thermal Module	39
2.7.1	Device Database Editor	39
2.7.2	Diode Device in the Database	48
2.7.3	Diode Loss Calculation	49
2.7.4	IGBT Device in the Database	51
2.7.5	IGBT Loss Calculation	53
2.7.6	MOSFET Device in the Database	56
2.7.7	MOSFET Loss Calculation	58
2.8	Motor Drive Module	61
2.8.1	Reference Direction of Mechanical Systems	61
2.8.2	DC Machine	64
2.8.3	Induction Machine	66
2.8.4	Induction Machine with Saturation	70
2.8.5	Brushless DC Machine	72
2.8.6	Synchronous Machine with External Excitation	77
2.8.7	Permanent Magnet Synchronous Machine	80
2.8.8	Permanent Magnet Synchronous Machine with Saturation	83
2.8.9	Switched Reluctance Machine	87
2.9	MagCoupler Module	90
2.10	MagCoupler-RT Module	95

2.11	Mechanical Elements and Sensors	98
2.11.1	Mechanical Elements and Sensors	98
2.11.1.1	Constant-Torque Load	98
2.11.1.2	Constant-Power Load	99
2.11.1.3	Constant-Speed Load	100
2.11.1.4	General-Type Load	100
2.11.1.5	Externally-Controlled Load	101
2.11.2	Gear Box	101
2.11.3	Mechanical Coupling Block	102
2.11.4	Mechanical-Electrical Interface Block	102
2.11.5	Speed/Torque Sensors	105
2.11.6	Position Sensors	107
2.11.6.1	Absolute Encoder	108
2.11.6.2	Incremental Encoder	108
2.11.6.3	Resolver	109
2.11.6.4	Hall-Effect Sensor	110

### **3 Control Circuit Components**

3.1	Transfer Function Blocks	111
3.1.1	Proportional Controller	112
3.1.2	Integrator	113
3.1.3	Differentiator	115
3.1.4	Proportional-Integral Controller	115
3.1.5	Built-in Filter Blocks	116
3.2	Computational Function Blocks	117
3.2.1	Summer	117
3.2.2	Multiplier and Divider	118
3.2.3	Square-Root Block	118
3.2.4	Exponential/Power/Logarithmic Function Blocks	119
3.2.5	Root-Mean-Square Block	119
3.2.6	Absolute and Sign Function Blocks	120
3.2.7	Trigonometric Functions	120

3.2.8	Fast Fourier Transform Block	121
3.3	Other Function Blocks	122
3.3.1	Comparator	122
3.3.2	Limiter	122
3.3.3	Gradient (dv/dt) Limiter	123
3.3.4	Trapezoidal and Square Blocks	123
3.3.5	Sampling/Hold Block	124
3.3.6	Round-Off Block	125
3.3.7	Time Delay Block	126
3.3.8	Multiplexer	127
3.3.9	THD Block	128
3.4	Logic Components	129
3.4.1	Logic Gates	129
3.4.2	Set-Reset Flip-Flop	130
3.4.3	J-K Flip-Flop	131
3.4.4	D Flip-Flop	131
3.4.5	Monostable Multivibrator	132
3.4.6	Pulse Width Counter	132
3.4.7	Up/Down Counter	133
3.4.8	A/D and D/A Converters	134
3.5	Digital Control Module	135
3.5.1	Zero-Order Hold	135
3.5.2	z-Domain Transfer Function Block	136
3.5.2.1	Integrator	138
3.5.2.2	Differentiator	139
3.5.2.3	Digital Filters	140
3.5.3	Unit Delay	143
3.5.4	Quantization Block	143
3.5.5	Circular Buffer	144
3.5.6	Convolution Block	145
3.5.7	Memory Read Block	146
3.5.8	Data Array	146
3.5.9	Stack	148

- 3.5.10 Multi-Rate Sampling System 148
- 3.6 SimCoupler Module 150
  - 3.6.1 Set-up in PSIM and Simulink 150
  - 3.6.2 Solver Type and Time Step Selection in Simulink 153

## **4 Other Components**

- 4.1 Parameter File 157
- 4.2 Sources 158
  - 4.2.1 Time 158
  - 4.2.2 DC Source 158
  - 4.2.3 Sinusoidal Source 159
  - 4.2.4 Square-Wave Source 160
  - 4.2.5 Triangular Source 161
  - 4.2.6 Step Sources 162
  - 4.2.7 Piecewise Linear Source 163
  - 4.2.8 Random Source 165
  - 4.2.9 Math Function Source 165
  - 4.2.10 Voltage/Current-Controlled Sources 166
  - 4.2.11 Nonlinear Voltage-Controlled Sources 168
- 4.3 Voltage/Current Sensors 169
- 4.4 Probes and Meters 169
- 4.5 Voltage/Current Scopes 172
- 4.6 Switch Controllers 174
  - 4.6.1 On-Off Switch Controller 174
  - 4.6.2 Alpha Controller 175
  - 4.6.3 PWM Lookup Table Controller 176
- 4.7 Function Blocks 179
  - 4.7.1 Control-Power Interface Block 179
  - 4.7.2 ABC-DQO Transformation Block 180
  - 4.7.3 Math Function Blocks 181

- 4.7.4 Lookup Tables 182
- 4.7.5 C Script Block 185
- 4.7.6 External DLL Blocks 186
- 4.7.7 Embedded Software Block 191

## **5 Analysis Specification**

- 5.1 Transient Analysis 193
- 5.2 AC Analysis 194
- 5.3 Parameter Sweep 198

## **6 Circuit Schematic Design**

- 6.1 Creating a Circuit 202
- 6.2 Editing a Circuit 203
- 6.3 Saving a File 206
- 6.4 Subcircuit 207
  - 6.4.1 Creating Subcircuit - In the Main Circuit 208
  - 6.4.2 Creating Subcircuit - Inside the Subcircuit 208
  - 6.4.3 Connecting Subcircuit - In the Main Circuit 210
  - 6.4.4 Other Features of the Subcircuit 210
    - 6.4.4.1 Passing Variables from the Main Circuit to Subcircuit 211
    - 6.4.4.2 Customizing the Subcircuit Image 212
    - 6.4.4.3 Including Subcircuits in the PSIM Element List 213
- 6.5 Running the Simulation 214
- 6.6 Managing the PSIM Library 218
- 6.7 Other Options 223
  - 6.7.1 Generate and View the Netlist File 223
  - 6.7.2 Set Path 223
  - 6.7.3 Settings 224
  - 6.7.4 Printing the Circuit Schematic 224



## **7    Waveform Processing**

- 7.1    File Menu   226
- 7.2    Edit Menu   226
- 7.3    Axis Menu   227
- 7.4    Screen Menu   228
- 7.5    View Menu   230
- 7.6    Option Menu   231
- 7.7    Label Menu   231
- 7.8    Exporting Data   231

## **8    Error/Warning Messages and Other Simulation Issues**

- 8.1    Simulation Issues   233
  - 8.1.1   Time Step Selection   233
  - 8.1.2   Propagation Delays in Logic Circuits   233
  - 8.1.3   Interface Between Power and Control Circuits   234
  - 8.1.4   FFT Analysis   234
- 8.2    Error/Warning Messages   235
- 8.3    Debugging   236



---

# General Information

## 1.1 Introduction

PSIM is a simulation software specifically designed for power electronics and motor drives. With fast simulation and friendly user interface, PSIM provides a powerful simulation environment for power electronics, analog and digital control, magnetics, and motor drive system studies.

This manual covers both PSIM<sup>1</sup> and the following add-on Modules:

- Motor Drive Module
- Digital Control Module
- SimCoupler Module
- Thermal Module
- MagCoupler Module
- MagCoupler-RT Module

The Motor Drive Module has built-in machine models and mechanical load models for motor drive system studies.

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

The SimCoupler Module provides interface between PSIM and Matlab/Simulink<sup>2</sup> for co-simulation.

The Thermal Module provides the capability to calculate semiconductor devices losses.

The MagCoupler Module provides interface between PSIM and the electromagnetic field analysis software JMAG<sup>3</sup> for co-simulation.

The MagCoupler-RT Module links PSIM with JMAG-RT<sup>3</sup> data files.

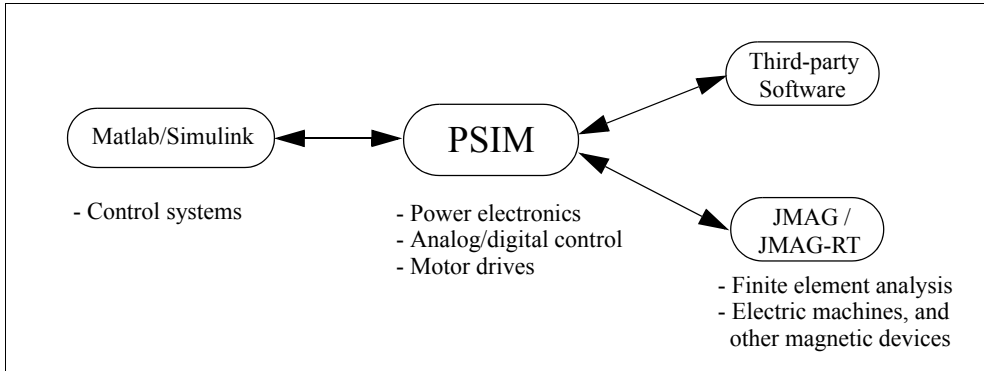
In addition, PSIM supports links to third-party software through custom DLL blocks. The overall PSIM environment is shown below.

---

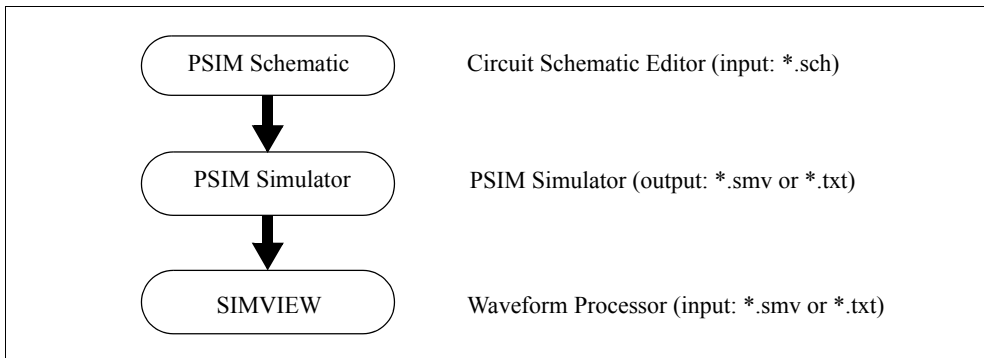
1. PSIM and SIMVIEW are copyright by Powersim Inc., 2001-2006

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

3. JMAG and JMAG-RT are copyright by the Japan Research Institute, Ltd., 1997-2006



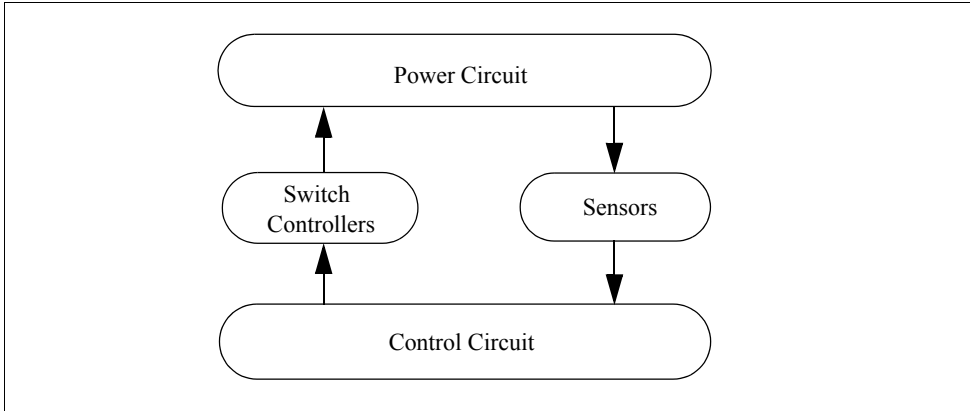
The PSIM simulation environment consists of the circuit schematic program PSIM, the simulator engine, and the waveform processing program SIMVIEW<sup>1</sup>. The simulation process is illustrated as follows.



Chapter 1 of this manual describes the circuit structure, software/hardware requirement, and parameter specification format. Chapter 2 through 4 describe the power and control circuit components. Chapter 5 describes the specifications of the transient analysis and ac analysis. The use of the PSIM schematic program and SIMVIEW is discussed in Chapter 6 and 7. Finally, error/warning messages are discussed in Chapter 8.

## 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 environment 2000/XP on personal computers. The minimum RAM memory requirement is 128 MB.

### 1.4 Installing the Program

A quick installation guide is provided in the flier “PSIM - Quick Guide” and on the CD-ROM.

Some of the files in the PSIM directory are shown in the table below.

Files	Description
PSIM.exe	PSIM circuit schematic editor
SIMVIEW.exe	Waveform display program SIMVIEW
PcdEditor.exe	Device database editor
s2z_converter.exe	s-domain to z-domain converter

SetSimPath.exe	Program to set up the SimCoupler Module
psim.lib, psimimage.lib	PSIM library files

File extensions used in PSIM are:

*.sch	PSIM schematic file
*.txt	PSIM simulation output file (text)
*.fra	PSIM ac analysis output file (text)
*.dev	Device database file
*.smv	SIMVIEW data file

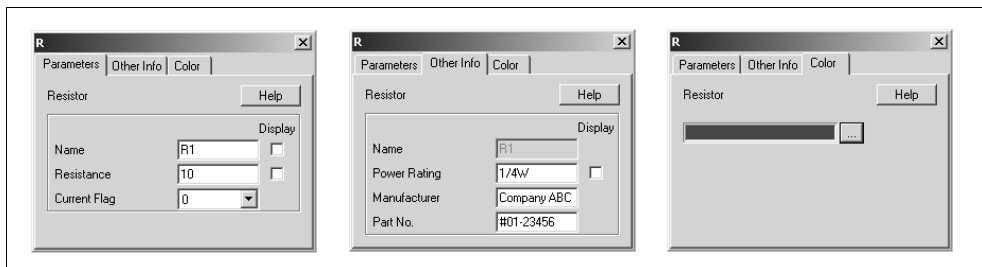
## 1.5 Simulating a Circuit

To simulate the sample one-quadrant chopper circuit “chop.sch”:

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

## 1.6 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.



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: $2^3 = 2*2*2$ ]
SQRT	square-root function
SIN	sine function
COS	cosine function

ASIN	sine inverse function
ACOS	cosine inverse function
TAN	tangent function
ATAN	inverse tangent function
ATAN2	inverse tangent function [ $-\pi \leq \text{atan2}(y,x) \leq \pi$ ]
SINH	hyperbolic sine function
COSH	hyperbolic cosine function
EXP	exponential (base e) [Example: $\text{EXP}(x) = e^x$ ]
LOG	logarithmic function (base e) [Example: $\text{LOG}(x) = \ln(x)$ ]
LOG10	logarithmic function (base 10)
ABS	absolute function
SIGN	sign function [Example: $\text{SIGN}(1.2) = 1$ ; $\text{SIGN}(-1.2) = -1$ ]



## Power Circuit Components

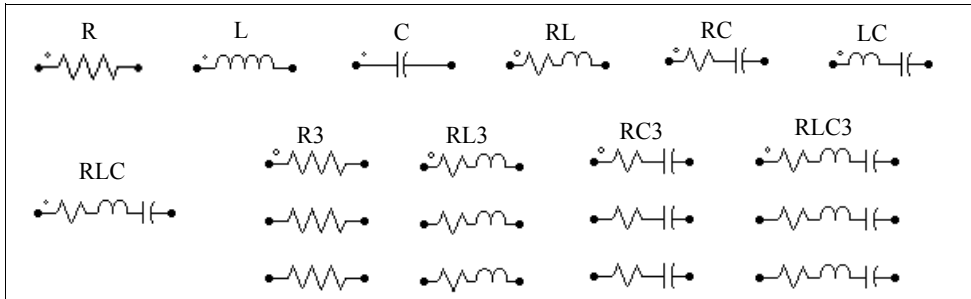
### 2.1 Resistor-Inductor-Capacitor Branches

#### 2.1.1 Resistors, Inductors, and Capacitors

Both individual resistor, inductor, capacitor, and lumped RLC branches are provided in PSIM. Initial conditions of inductor currents and capacitor voltages can be defined.

To facilitate the setup of three-phase circuits, symmetrical three-phase RLC branches, “R3”, “RL3”, “RC3”, “RLC3”, are provided. Initial inductor currents and capacitor voltages of the three-phase branches are all zero.

**Images:**



For three-phase branches, the phase with a dot is Phase A.

**Attributes:**

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

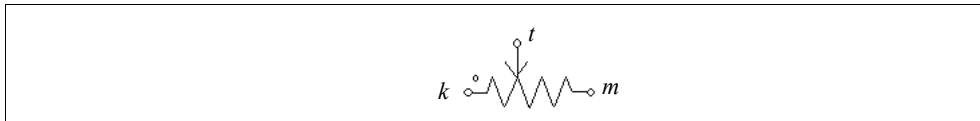
Current Flag	<p>Flag for branch current output.</p> <p>If the flag is zero, there is no current output. If the flag is 1, the current will be available for display in the runtime graphs (under <b>Simulate</b> -&gt; <b>Runtime Graphs</b>). It will also be saved to the output file for display in SIMVIEW.</p> <p>The current is positive when it flows into the dotted terminal of the branch.</p>
Current Flag_A; Current Flag_B; Current Flag_C	Current flags for Phase A, B, and C of three-phase branches, respectively.

The resistance, inductance, or capacitance of a branch can not be all zero. At least one of the parameters has to be a non-zero value.

### 2.1.2 Rheostat

A rheostat is a resistor with a tap.

**Image:**



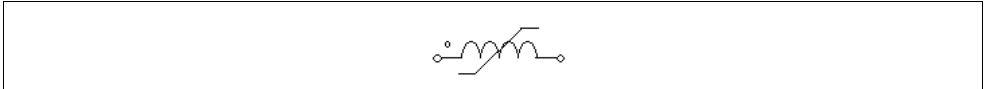
**Attributes:**

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 $k$ .

### 2.1.3 Saturable Inductor

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

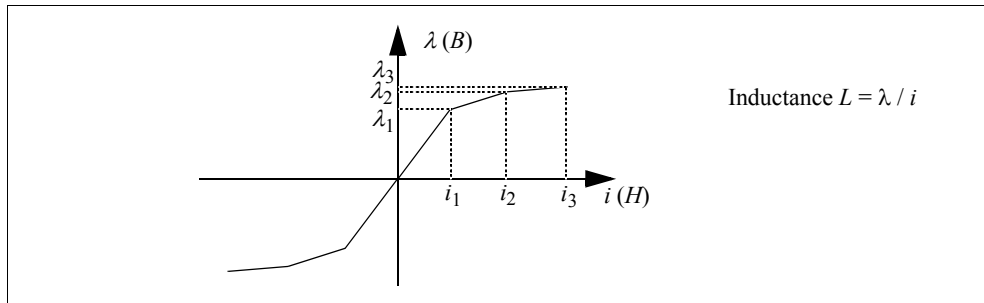
Image:



Attributes:

Parameters	Description
Current v.s. 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  $\lambda$  and the magnetizing force  $H$  is proportional to the current  $i$ , the  $B$ - $H$  curve can be represented by the  $\lambda$ - $i$  curve instead, as shown below.



The inductance is defined as:  $L = \lambda / i$ , the ratio of  $\lambda$  v.s.  $i$  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  $\lambda$  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.

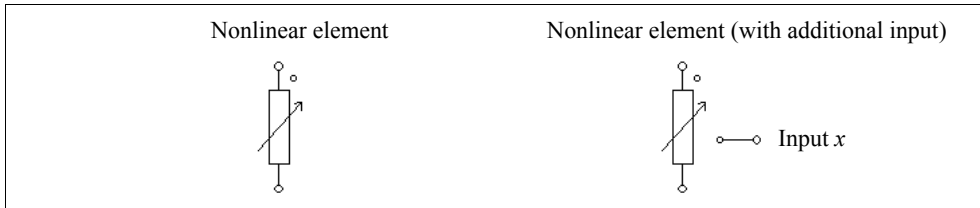
## 2.1.4 Nonlinear Elements

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:**



**Attributes:**

For resistance-type elements:

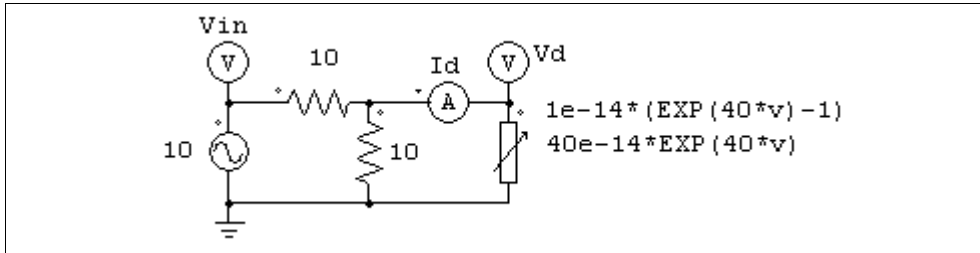
Parameters	Description
Expression $f(i)$ or $f(i, x)$	Expression of $v$ in terms of $i$ and $x$ [ $v = f(i)$ or $v = f(i, x)$ ]
Expression $df/di$	The derivative of the voltage $v$ versus current $i$ , i.e. $df(i)/di$
Initial Value $i_0$	The initial value of the current $i$
Lower Limit of $i$	The lower 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 $v_0$	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 $v_0$	0
Lower Limit of $v$	-1e3
Upper Limit of $v$	1

## 2.2 Switches

There are two basic types of switches 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 operate in either cut-off, linear, or saturation region.

Switches in switchmode include the following:

- Diode and DIAC
- Thyristor and TRIAC
- Self-commutated switches, specifically:
  - Gate-Turn-Off switch
  - npn bipolar junction transistor
  - pnp bipolar junction transistor
  - Insulated-Gate Bipolar Transistor (IGBT)
  - n-channel Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) and p-channel MOSFET
- Bi-directional switch

Switch models are ideal. That is, both turn-on and turn-off transients are neglected. A switch has an on-resistance of  $10\mu\Omega$  and an off-resistance of  $10M\Omega$ . Snubber circuits are not required for switches.

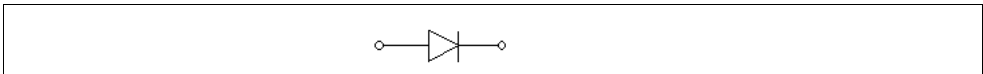
Linear switches include the following:

- npn bipolar junction transistor
- pnp bipolar junction transistor

## 2.2.1 Diode, DIAC, and Zener Diode

The conduction of a diode is determined by circuit operating conditions. A diode is turned on when it is positively biased, and is turned off when the current drops to zero.

Image:

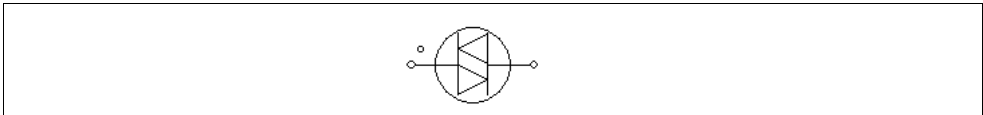


Attributes:

Parameters	Description
Diode Voltage Drop	Diode conduction voltage drop, in V
Initial Position	Flag for the initial diode position. If the flag is 0, the diode is off. If it is 1, the diode is on.
Current Flag	Current flag of the diode.

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:



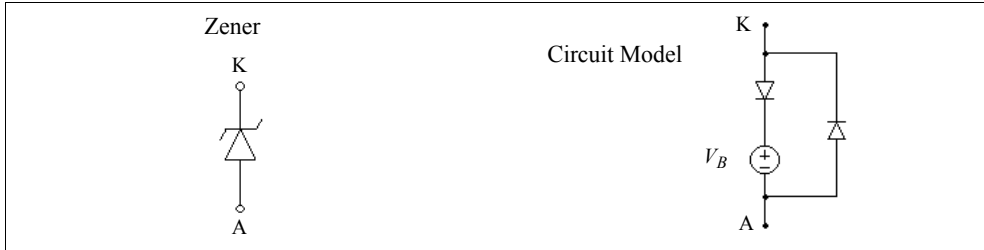
Attributes:

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

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

**Images:**



**Attributes:**

Parameters	Description
Breakdown Voltage	Breakdown voltage $V_B$ of the zener diode, in V
Forward Voltage Drop	Voltage drop of the forward conduction (diode voltage drop from anode to cathode), in V
Current Flag	Flag for zener current output (from anode to cathode)

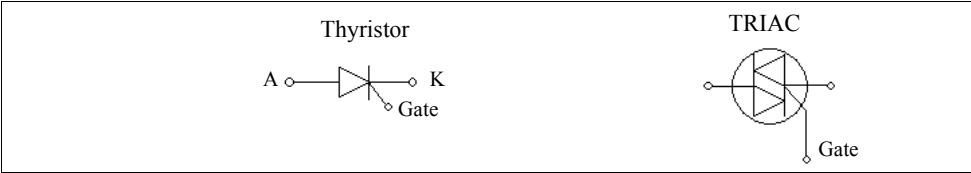
When the zener diode is positively biased, it behaves as a regular diode. When it is reverse biased, it will block the conduction as long as the cathode-anode voltage  $V_{KA}$  is less than the breakdown voltage  $V_B$ . When  $V_{KA}$  exceeds  $V_B$ , the voltage  $V_{KA}$  will be clamped to  $V_B$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

## 2.2.2 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 behaves in the same way as two opposite thyristors connected in parallel.

**Images:**



**Attributes:**

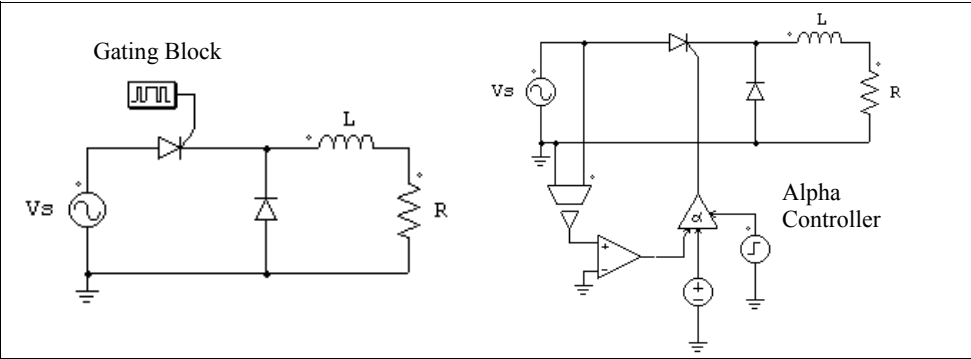
Parameters	Description
Voltage Drop	Thyristor conduction voltage drop, in V
Holding Current	Minimum conduction current below which the device stops conducting and returns to the OFF state (for thyristor only)
Latching Current	Minimum ON state current required to keep the device in the ON state after 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.

## 2.2.3 GTO, Transistors, and Bi-Directional Switch

Self-commutated switches in the switchmode, except pnp bipolar junction transistor (BJT) and p-channel MOSFET, are 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 or drain-source voltage is positive). It is turned off whenever the gating signal is low or the current drops to zero.

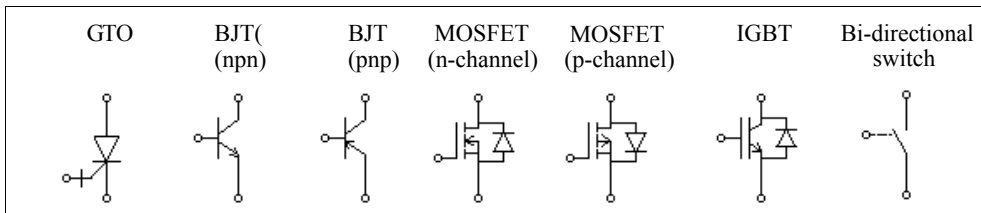
For pnp BJT and p-channel MOSFET, switches are turned on when the gating signal is low and switches are negatively biased (collector-emitter or drain-source voltage is negative).

A GTO switch is a symmetrical device with both forward-blocking and reverse-blocking capabilities. An IGBT or MOSFET switch consist of an active switch with an anti-parallel diode.

A bi-directional switch conducts currents in both directions. It is on when the gating signal is high and is off when the gating signal is low, regardless of the voltage bias conditions.

Note that a limitation of the BJT switch model in PSIM, in contrary to the device behavior in the real life, is that a BJT switch in PSIM will block reverse voltage (in this sense, it behaviors like a GTO). Also, it is controlled by a voltage signal at the gate node, not a current.

**Images:**

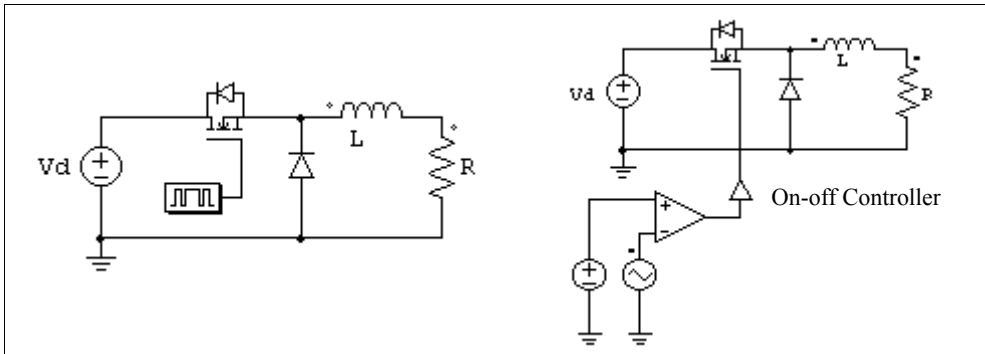


### Attributes:

Parameters	Description
Initial Position	Initial switch position flag. For MOSFET and IGBT, this flag is for the active switch, not for the anti-parallel diode.
Current Flag	Switch current flag. For MOSFET and IGBT, the current through the whole module (the active switch plus the diode) will be displayed.

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. The following examples illustrate the control of a MOSFET switch.

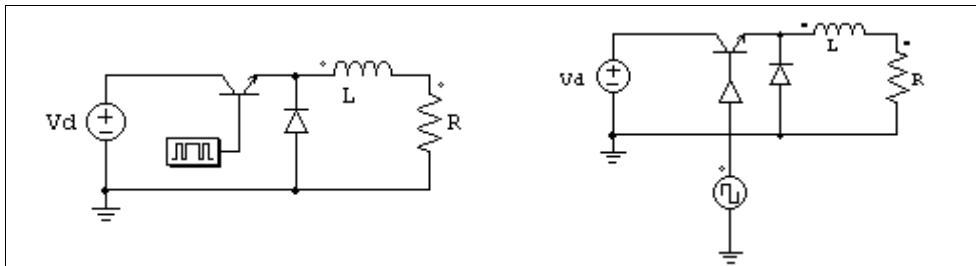
### Examples: Control of a MOSFET Switch



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.

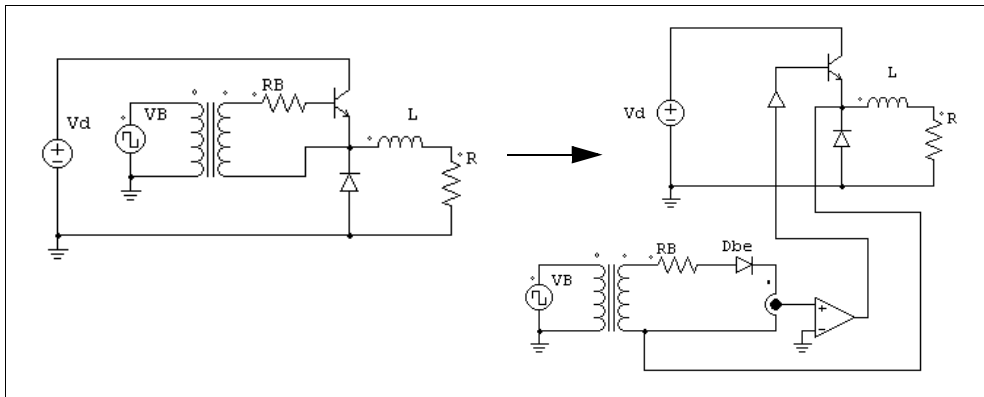
### Example: Control of a npn Bipolar Junction Transistor

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



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.

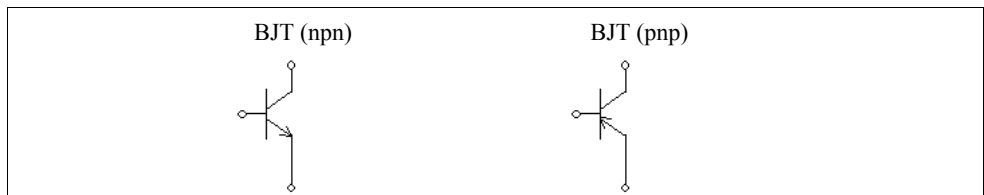
This circuit can be modelled and implemented in PSIM as shown on the right. A diode,  $D_{be}$ , with a conduction voltage drop of 0.7V, is used to model the pn junction between the base and the emitter. When the base current exceeds 0 (or a certain threshold value, in which case the base current will 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.



## 2.2.4 Linear Switches

Linear switches include npn bipolar junction transistor and pnp bipolar junction transistor. They can operate in either cut-off, linear, or saturation region.

**Images:**



**Attributes:**

Parameters	Description
Current Gain $\beta$	Transistor current gain $\beta$ , 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.

A linear BJT switch is controlled by the base current  $I_b$ . It can operate in one of the three regions: cut-off (off state), linear, and saturation region (on state). The properties of these regions for the npn transistor 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:  $V_{be} = V_r$ ;  $I_c < \beta * I_b$ ;  $V_{ce} = V_{ce,sat}$

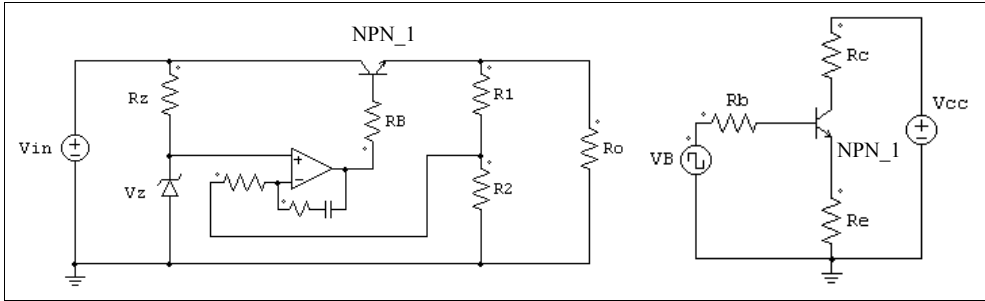
where  $V_{be}$  is the base-emitter voltage,  $V_{ce}$  is the collector-emitter voltage, and  $I_c$  is the collector current.

Note that for both the npn and pnp transistors, the gate node (base 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.

**WARNING:** It has been found that the linear model for npn and pnp transistors works well in simple circuits, but may not work when circuits are complex. Please use this model with caution.

**Examples: Circuits Using the Linear BJT Switch**

Examples below illustrate the use of linear switches. 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.

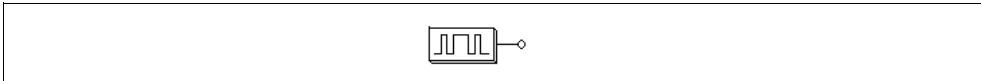


## 2.2.5 Switch Gating Block

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 (1)* 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 <i>Gating Block</i> element only)
Switching Points	Switching points, in deg. If the frequency is zero, the switching points is in second. (for the <i>Gating Block</i> element only)
File for Gating Table	Name of the file that stores the gating table (for the <i>Gating Block (1)</i> 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.

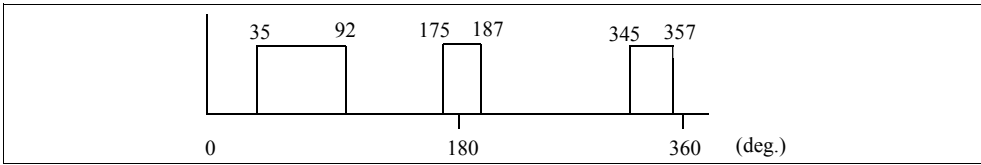
For the *Gating Block (1)* element, the file for the gating table must be in the same directory as the schematic file. The gating table file has the following format:

```
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°, 92°, 175°, 187°, 345°, and 357°, respectively.

If the *Gating Block (1)* element is used instead, the specification will be:

Frequency	2000.
File for Gating Table	test.tbl

The file “test.tbl” will contain the following:

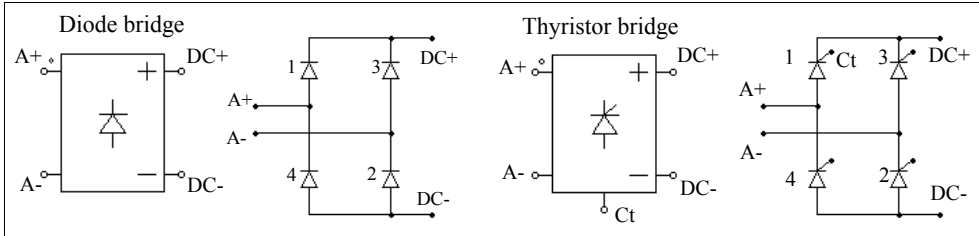
```
6
35.
92.
175.
187.
```

345.  
357.

## 2.2.6 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:**

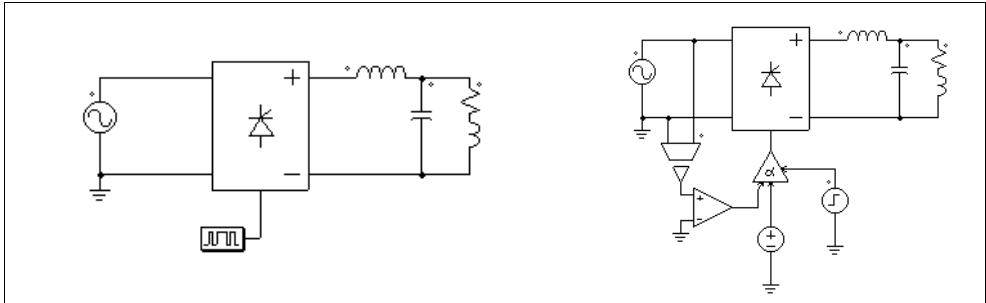
Parameters	Description
Diode Voltage Drop <i>or</i> Voltage Drop	Forward voltage drop of each diode or thyristor, in V
Init. Position_ <i>i</i>	Initial position for Switch <i>i</i>
Current Flag_ <i>i</i>	Current flag for Switch <i>i</i>

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

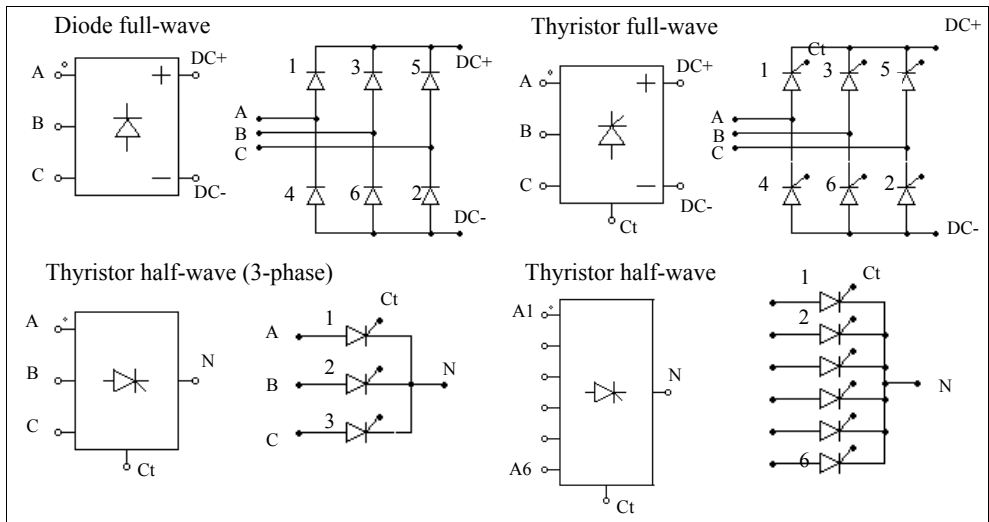
The gating signal for the circuit on the left is specified through a gating block, and the gating signal for the circuit on the right is provided through an alpha controller. A major advantage of the alpha controller is that the delay angle  $\alpha$  of the thyristor bridge, in deg., can be directly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



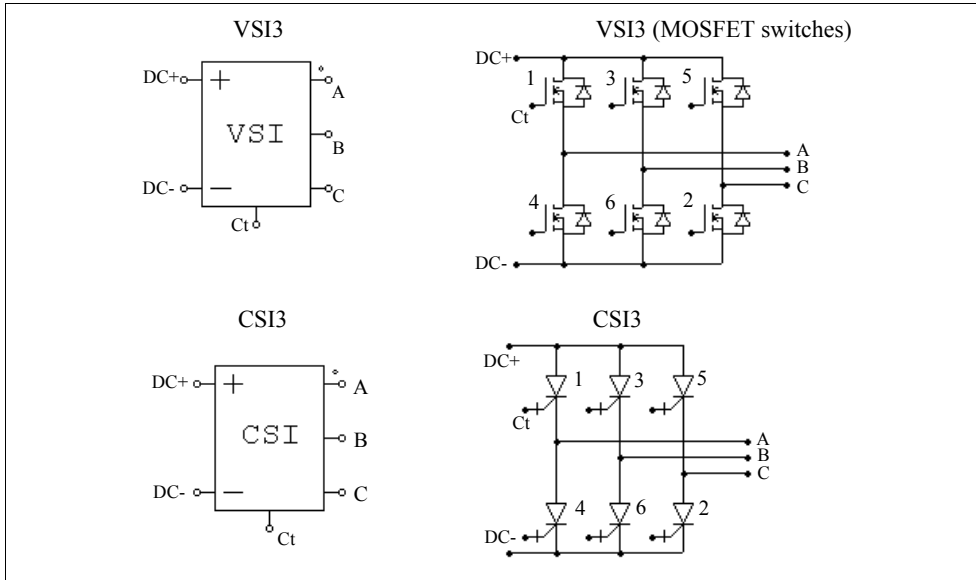
## 2.2.7 Three-Phase Switch Modules

The following figure shows three-phase switch modules and the internal circuit connections. A three-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:

Parameters	Description
On-Resistance	On resistance of the MOSFET switch during the on state, in Ohm (for MOSFET-type switches only)
Saturation Voltage	Conduction voltage drop of the IGBT switch, in V (for IGBT-type switches only)
Voltage Drop	Conduction voltage drop of the switch, in V (for CSI3 only)
Diode Voltage Drop	Conduction voltage drop of the anti-parallel diode, in V (for VSI3 only)
Init. Position <sub><i>i</i></sub>	Initial position for Switch <i>i</i>
Current Flag <sub><i>i</i></sub>	Current flag for Switch <i>i</i>

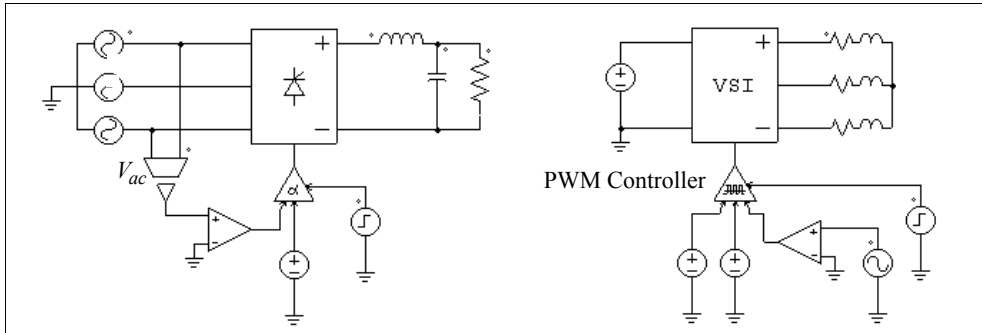
Similar to single-phase modules, only the gating signal for Switch 1 need to be specified for three-phase modules. Gating signals for other switches will be automatically derived. For the 3-phase half-wave thyristor bridge, the phase shift between two consecutive switches is 120°. For all other bridges,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

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 Three-Phase Thyristor and VSI Modules



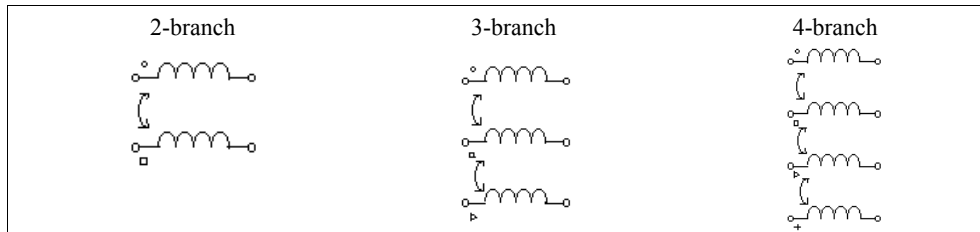
The thyristor circuit on the left uses an alpha controller. For a three-phase circuit, the zero-crossing of the voltage  $V_{ac}$  corresponds to the moment when the delay angle alpha is equal to zero. This signal is used to provide synchronization to the controller.

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.

## 2.3 Coupled Inductors

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

Images:

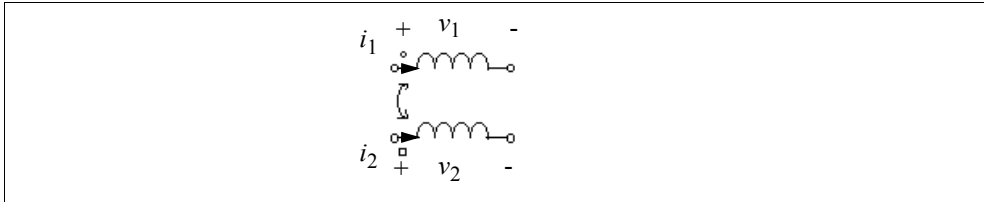


### Attributes:

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
$i_{i\_initial}$	Initial current in Inductor $i$
$Iflag\_i$	Flag for the current printout in Inductor $i$

In the images, the circle, square, triangle, and plus marks refer to Inductor 1, 2, 3, and 4, respectively.

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.,  $L_{12} = L_{21}$ .

### 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 specification of this element will be:

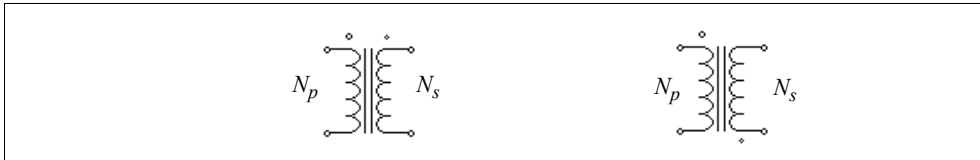
$L_{11}$ (self)	1m
$L_{12}$ (mutual)	0.9m
$L_{22}$ (self)	1.1m

## 2.4 Transformers

### 2.4.1 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
$N_p$ (primary)	No. of turns of the primary winding
$N_s$ (secondary)	No. of turns of the secondary winding

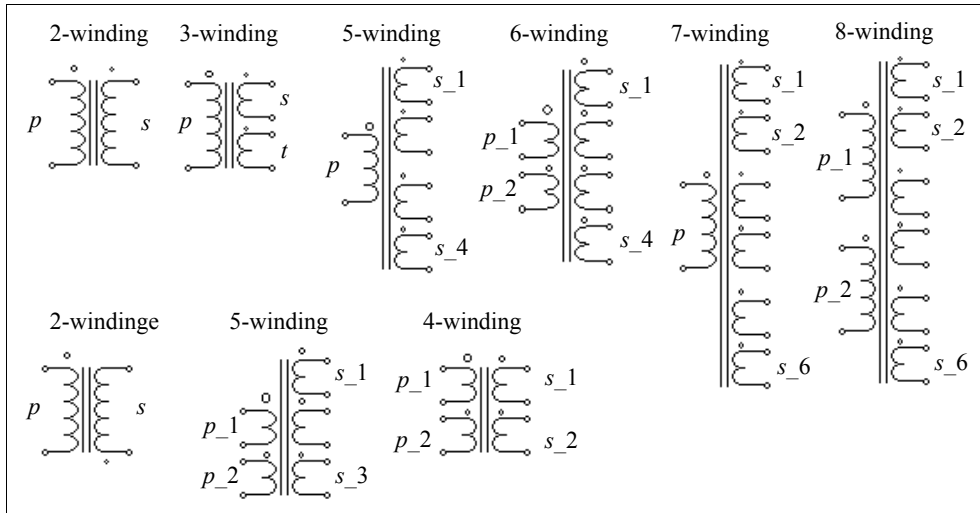
Since the turns ratio is equal to the ratio of the rated voltages, the number of turns can be replaced by the rated voltage at each side.

### 2.4.2 Single-Phase Transformers

The following single-phase transformer modules are provided:

- Transformer with 1 primary and 1 secondary windings
- Transformer with 1 primary and 2 secondary windings
- Transformer with 2 primary and 2 secondary windings
- Transformer with 1 primary and 4 secondary windings
- Transformer with 2 primary and 4 secondary windings
- Transformer with 1 primary and 6 secondary windings
- Transformer with 2 primary and 6 secondary windings

## 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 the transformers with more than 1 primary winding or more than 3 secondary windings, the attributes are as follows.

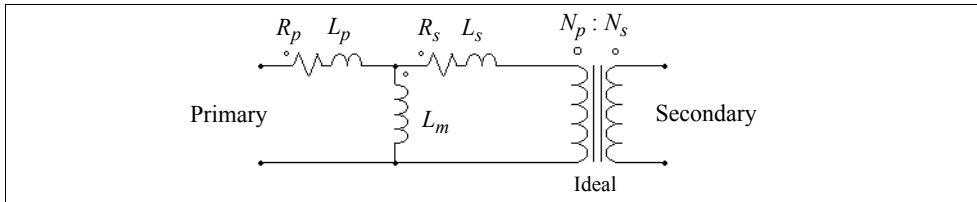
#### Attributes:

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:

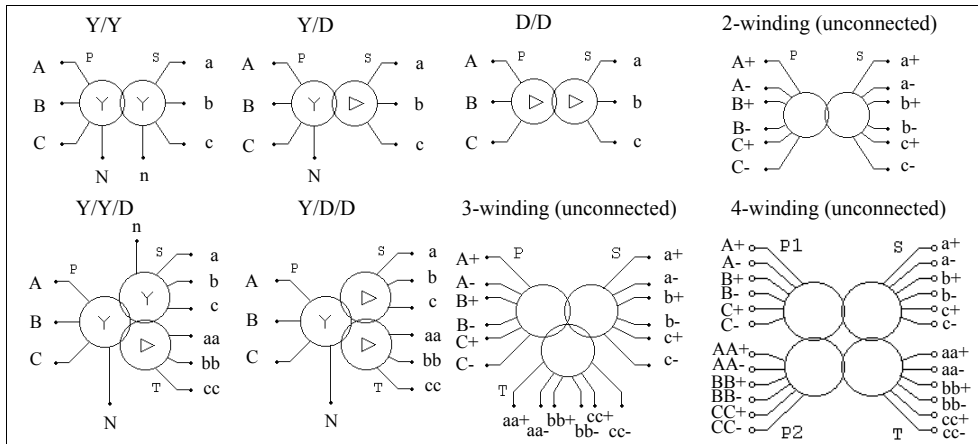
$R_p$ (primary)	2m
$R_s$ (secondary)	2m
$L_p$ (primary)	1m
$L_s$ (secondary)	1m
$L_m$ (magnetizing)	100m
$N_p$ (primary)	220
$N_s$ (secondary)	440

### 2.4.3 Three-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/ $\Delta$  connected transformer
- 3-phase 3-winding transformer (windings unconnected)
- 3-phase 3-winding Y/Y/ $\Delta$  and Y/ $\Delta$ / $\Delta$  connected transformer
- 3-phase 4-winding transformer (windings unconnected)

Images:



**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
$L_m$ (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

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.

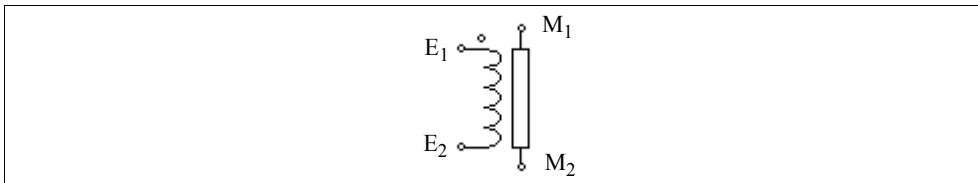
## 2.5 Magnetic Elements

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.

### 2.5.1 Winding

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

**Image:**





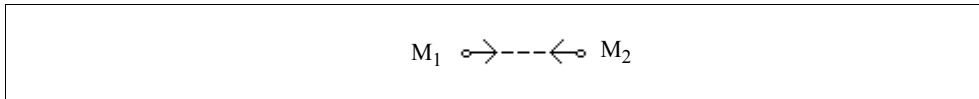
**Attributes:**

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).

## 2.5.2 Leakage Flux Path

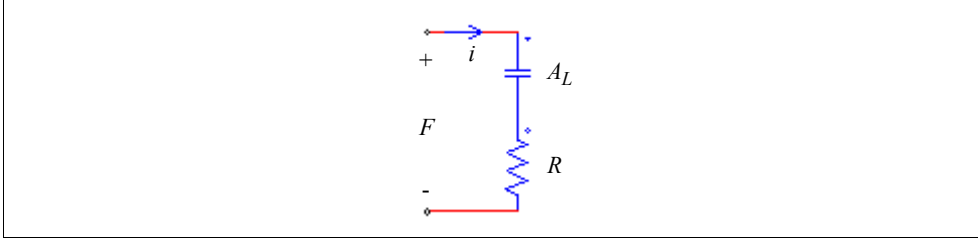
This element models the flow path of the leakage flux.

**Image:****Attributes:**

Parameters	Description
Inductance Factor $A_L$	Inductance factor $A_L$ , 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:



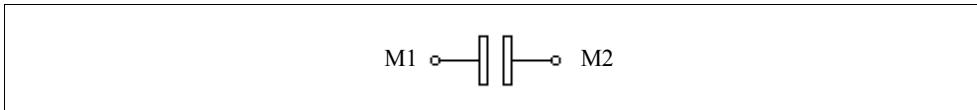
The mmf, in the form of a voltage source, applies across the capacitor (the capacitance is  $A_L$ ) and the resistor  $R$ . Let the current flowing through this branch be  $i$ , and the rms value be  $I_{rms}$ , the relationship between the losses due to the leakage flux and the resistance  $R$  is:

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

### 2.5.3 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:**

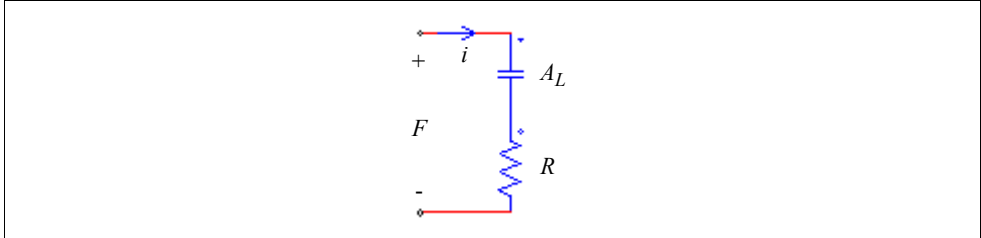
For the element *Air Gap*:

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 <sup>2</sup>
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 (1)*:

Parameters	Description
Inductance Factor $A_L$	Inductance factor $A_L$ , 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  $\mu_o = 4\pi \cdot 10^{-7}$ .

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

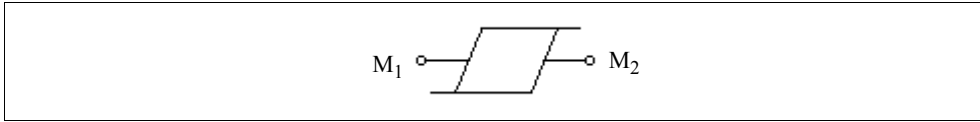
$$P_{loss} = I_{rms}^2 \cdot R$$

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

## 2.5.4 Linear Core

This element represents a linear lossless core.

**Image:**



**Attributes:**

Parameters	Description
Inductance Factor $A_L$	Inductance factor $A_L$ of the core, 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:

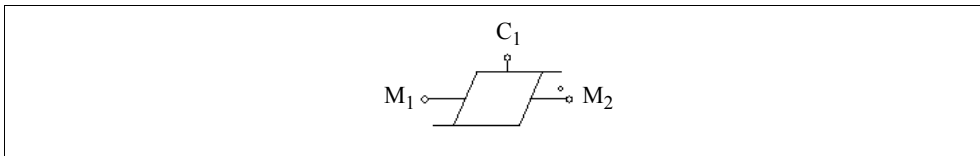
$$A_L = \frac{\mu_o \cdot \mu_r \cdot A_c}{L_{length}}$$

where  $\mu_r$  is the relative permeability of the core material.

## 2.5.5 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, defined as the inductance per turn squared
Resistance for Losses	Resistance $R$ , in Ohm, that represents the core losses
Coefficient $\phi_{sat}$	Coefficient $\Phi_{sat}$ for the core B-H curve, in Weber
Coefficient $K_1$	Coefficient $K_1$ for the core B-H curve
Coefficient $K_{exp1}$	Coefficient $K_{exp1}$ for the core B-H curve
Coefficient $K_2$	Coefficient $K_2$ for the core B-H curve
Coefficient $K_{exp2}$	Coefficient $K_{exp2}$ for the core B-H curve
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  $M_1$ .

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  $\Phi_{sat}$ ,  $K_1$ ,  $K_{exp1}$ ,  $K_2$ , and  $K_{exp2}$  are used to fit the B-H curve of an actual magnetic material. A good initial guess of  $\Phi_{sat}$  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:

$$K_2 > 2$$

$$K_{exp2} > 20$$

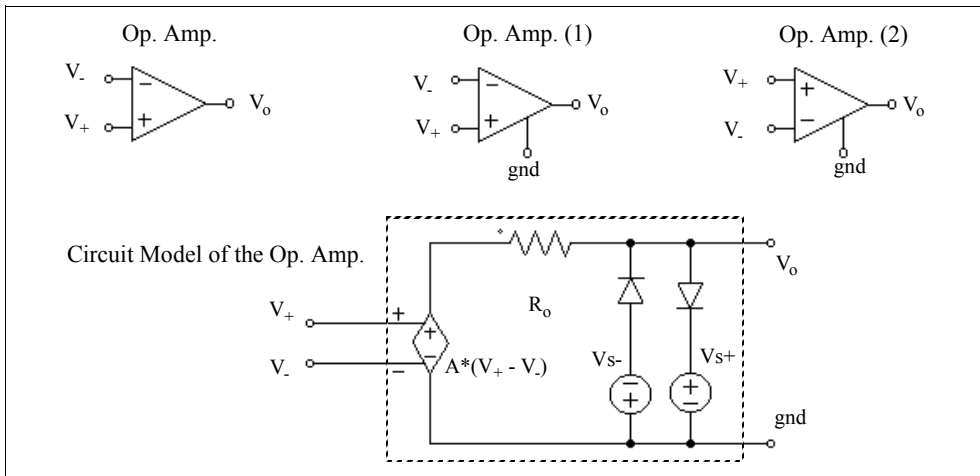
## 2.6 Other Elements

### 2.6.1 Operational Amplifier

Three op. amp. elements are provided in the PSIM Library: *Op. Amp.*, *Op. Amp. (1)*, and *Op. Amp. (2)*.

An ideal operational amplifier (op. amp.) is modelled using power circuit elements, as shown below.

**Images:**



where

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

**Attributes:**

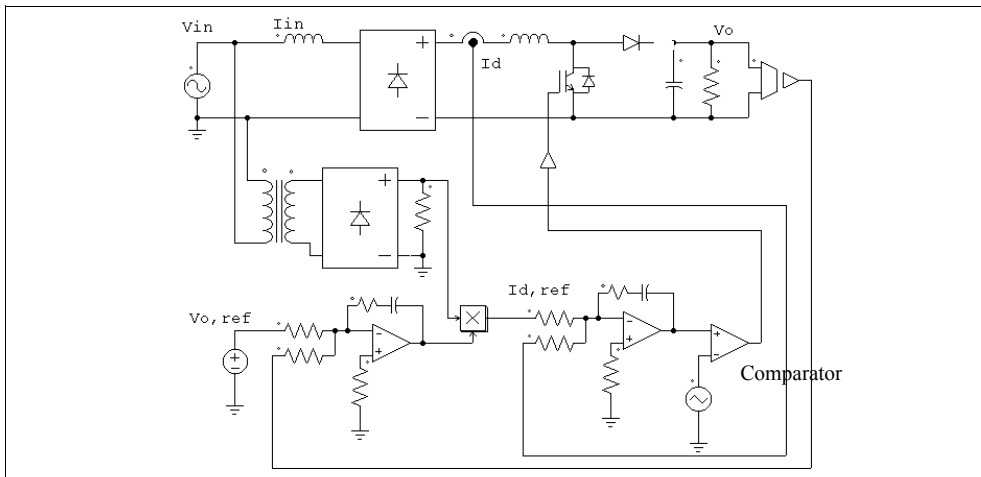
Parameters	Description
Voltage $V_{s+}$	Upper voltage source level of the op. amp.
Voltage $V_{s-}$	Lower voltage source levels of the op. amp.

The difference between the element *Op. Amp.* and *Op. Amp. (1)* or *Op. Amp. (2)* 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. (1)* or *Op. Amp. (2)*, 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

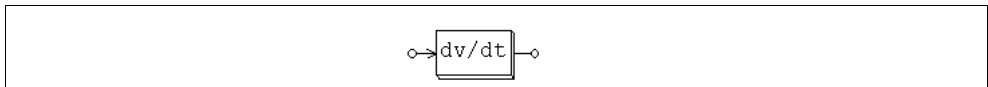
The figure below shows a boost power factor correction circuit. The PI regulators of both the inner current loop and the outer voltage loop are implemented using op. amp.



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

### 2.6.3 Power Modeling Block

The Power Modeling Block is a type of external DLL block that allows users to define algebraic and differential equations for a device, and to build a model in the power circuit. Unlike conventional DLL blocks that have signal inputs and signal outputs with no consideration of the power conservation (no input and output power balance), the Power Modeling Block allows electric currents to flow in and out of the terminals, and maintains the power balance.

The Power Modeling Block provides a very powerful way of modeling power devices. It can have power terminals, control input/output terminals, and mechanical shaft terminals, and equations can be either algebraic or differential, linear or nonlinear. A significant feature of the Power Modeling Block is that these equations are assembled and solved simultaneously with the other equations from the rest of the PSIM circuit, resulting a very robust, stable, and efficient solution.

For more information on how to use the Power Modeling Block, refer to the document "*Help Power Modeling Block.pdf*".



## 2.7 Thermal Module

The Thermal Module is an add-on module to the basic PSIM program. It provides a quick way of estimating the losses of semiconductor devices (diodes, IGBT, and MOSFET).

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.

The following illustrates the process of how a device in the database is used in the simulation and how the losses are calculated:

- The behavior model of the device is used in the simulation. The behavior model takes into account the static characteristics of the device (such as conduction voltage drop, on-state resistance, etc.), but not the dynamic characteristics (such as turn-on and turn-off transients).
- Based on the voltage and current values from the simulation, 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.

Please note that the loss calculation is only approximation and the accuracy of the results depends on the accuracy of the device data, as well as 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 the hardware setup.

The following sections describe how a device is added to the database, and how it is used in the simulation.

### 2.7.1 Device Database Editor

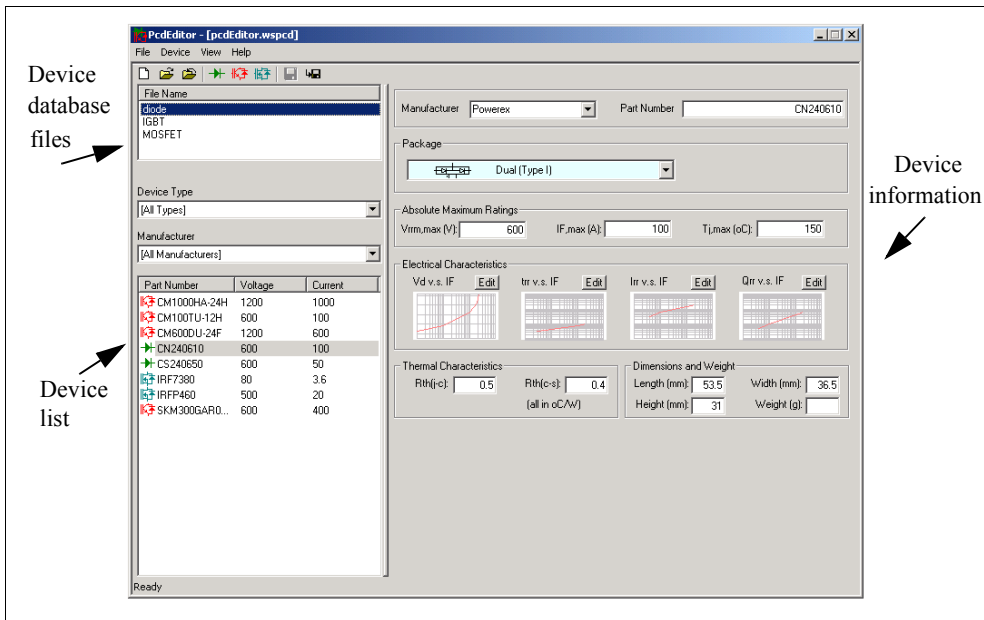
The device database editor, PcdEditor.exe, provides an easy and convenient way of adding, editing, and managing devices. An image of the database editor is shown below.

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.

On the right is the information of each device. In general, the following information is defined for the device:

- Manufacturer and Part Number
- Package type
- Absolute maximum ratings

- Electrical characteristics
- Thermal characteristics
- Dimension and weight



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**.

Three types of devices can be added to a device files: diode, IGBT, and MOSFET. However, since dual IGBT-diode modules have a different set of parameters as compared to the regular IGBT devices, they are treated as a separate type (referred to as the IGBT-DIODE type). The sections that follow describe in more details each type of devices.

To create a new device, go to the **Device** menu, and choose either **New Diode**, **New IGBT**, **New IGBT-Diode**, or **New MOSFET**.

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**.

### Adding a Device to the Database:

To illustrate how to add a device to a database file, below is the step-by-step procedure to add the Powerex discrete diode CS240650 (600V, 50A) into the device database file "diode\_new.dev".

- 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.
- 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".
- Obtain the datasheet of Powerex diode CS240650 from the web site [www.pwr.com](http://www.pwr.com). Show the PDF file of the datasheet on the screen.
- 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*:

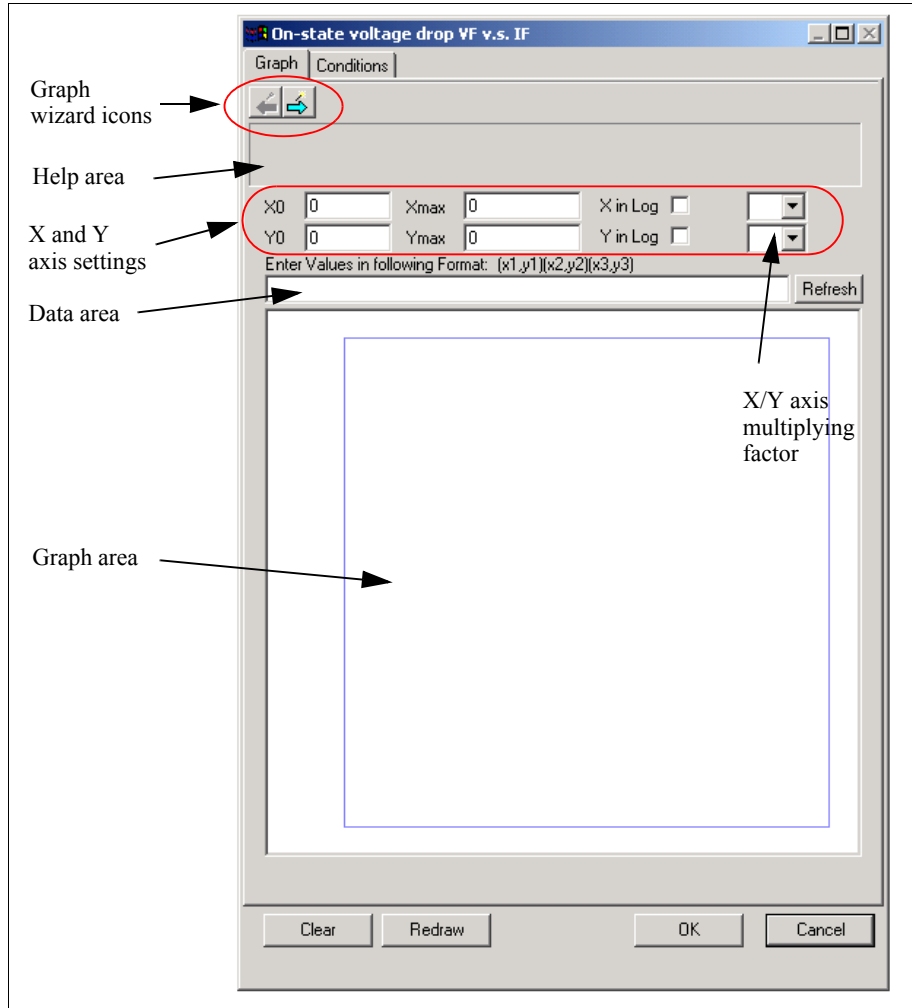
$V_{rrm,max}$ (V):	600
$I_{F,max}$ (A):	50
$T_{j,max}$ (°C):	150

- Define the forward voltage characteristics  $V_d$  v.s.  $I_F$  under *Electrical Characteristics* by clicking on the **Edit** button on top of the  $V_d$  v.s.  $I_F$  graph area.

The following dialog window will appear. The dialog window has two pages: *Graph* and *Conditions*.

The *Graph* page contains the 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. One is to enter the graph data points manually. Another is to use the Graph Wizard to capture the graph directly from the datasheet image. Defining the graph manually is preferred if there is only one data point or there are just a few data points. However, if the graph image is available, it is easier with the Graph Wizard.

#### **To Define the Graph Manually:**

- Refer to the "Maximum On-State Characteristics" graph of the datasheet, 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:

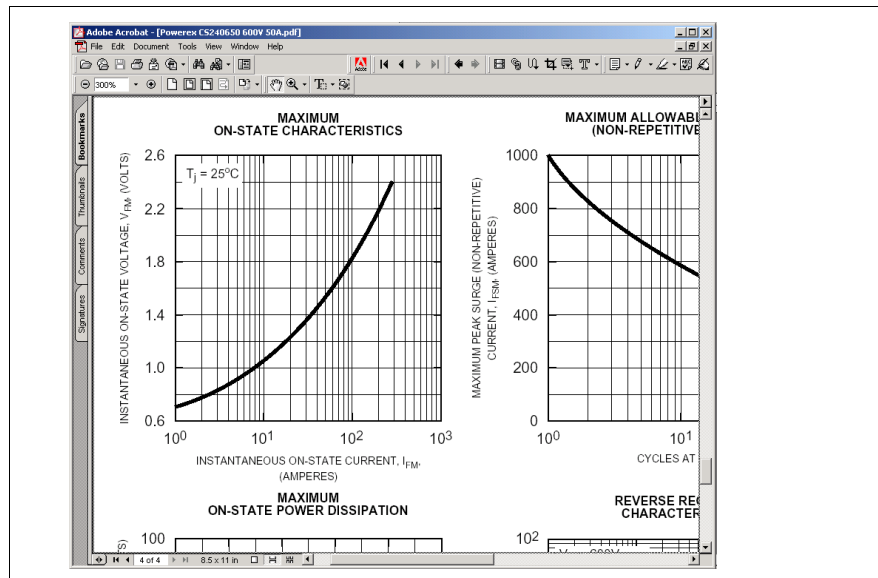
(1,0.7) (10,1.05) (100,1.8) (200,2.2) (300,2.4)

Then click on the **Refresh** button to display the graph.


- 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 forward wizard icon  to start the Graph Wizard.
- Display the graph of the datasheet on the screen as follows:

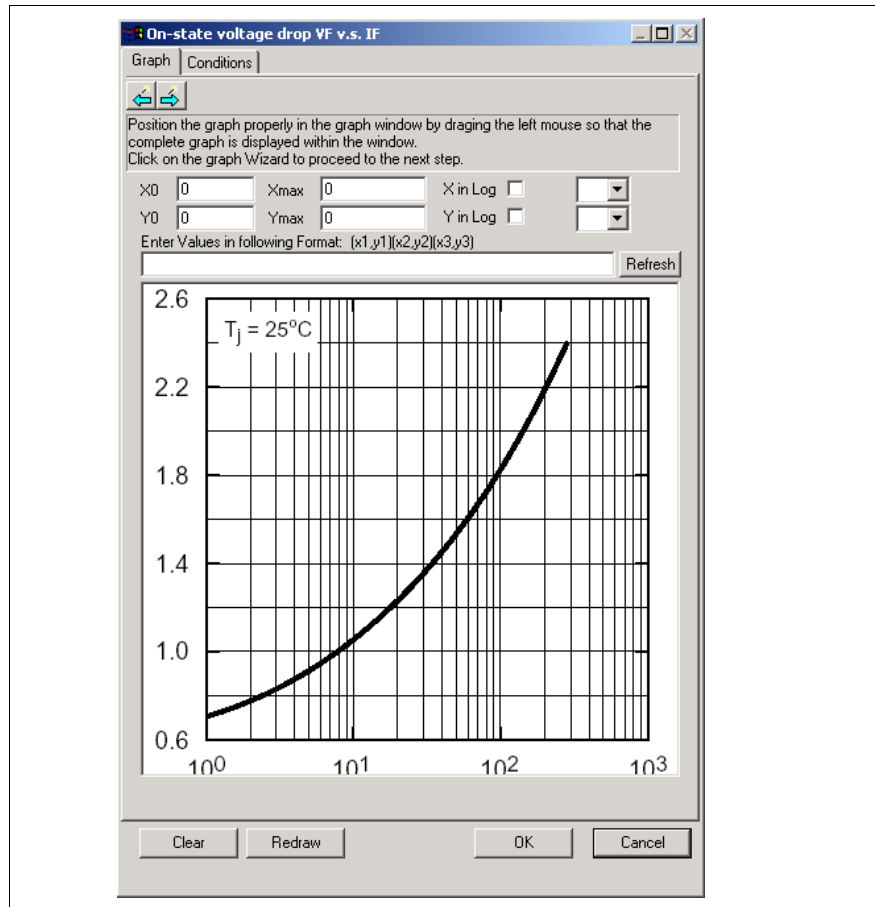


Then press the Print Screen key (the key is labeled as "Prt Scr" on the keyboard) to copy the screen image to the clipboard.

- 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.

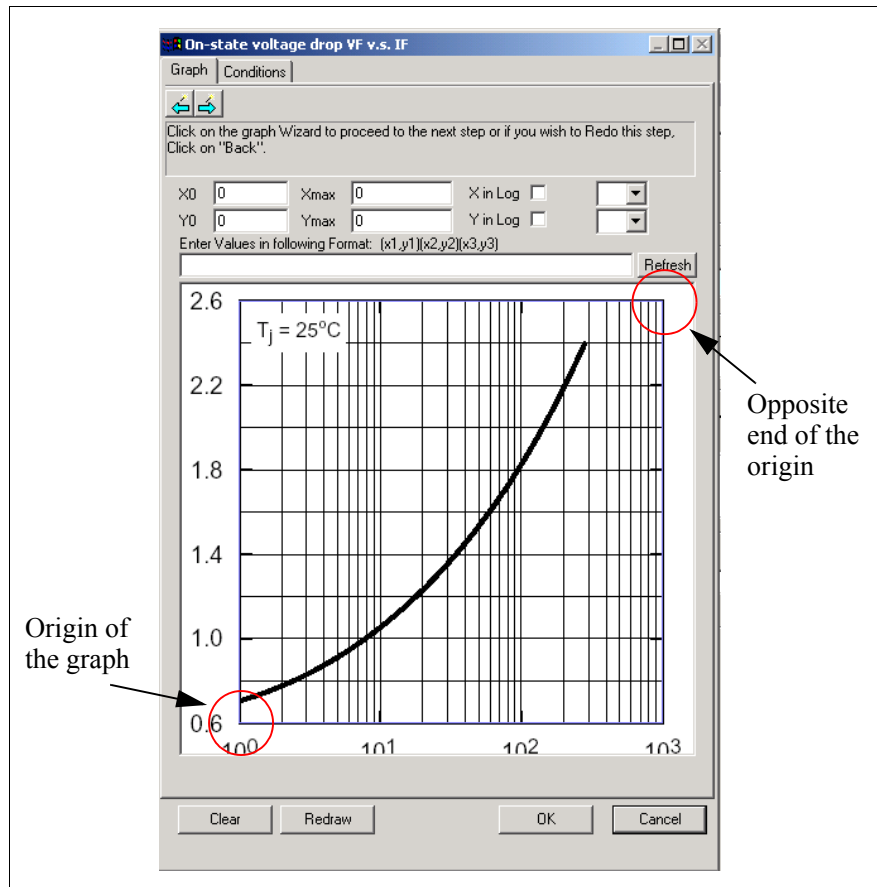


Click on the forward wizard icon  to move on to the next step.

- 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.



Then click on the forward wizard icon  to move on to the next step.

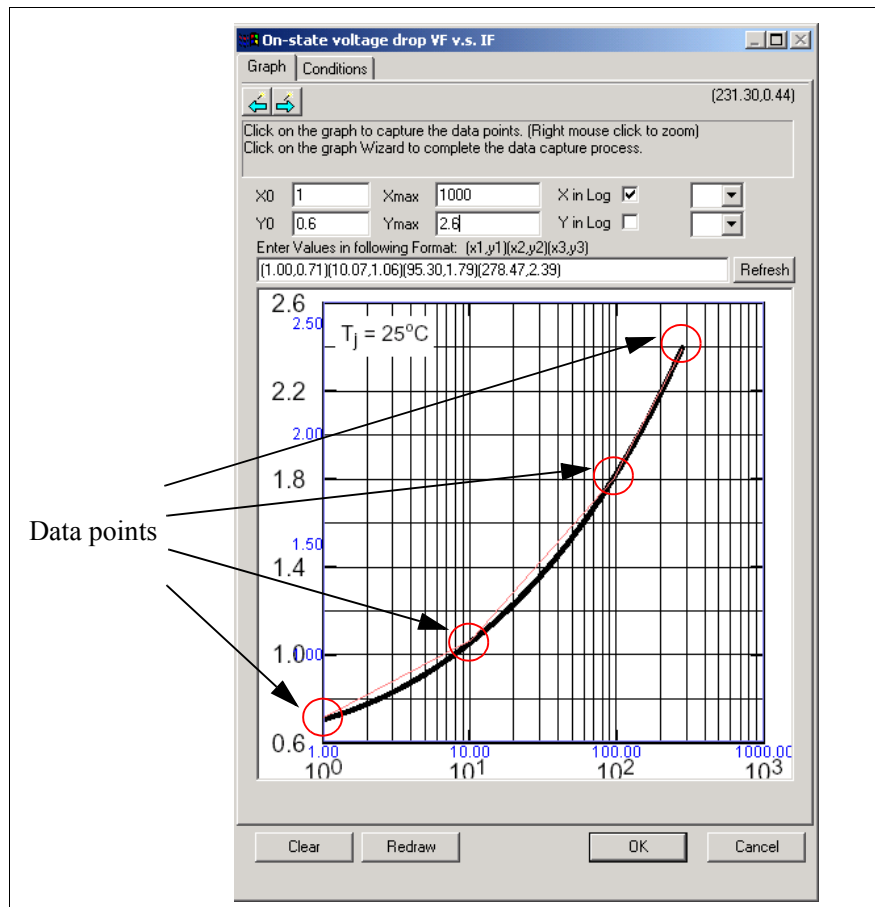
- In this step, the x and y axis settings will be defined. Enter the settings as

follows:

X0:	1
Xmax:	1000
Y0:	0.6
Ymax:	2.6
X in log:	checked


Click on the forward wizard icon  to move on to the next step.

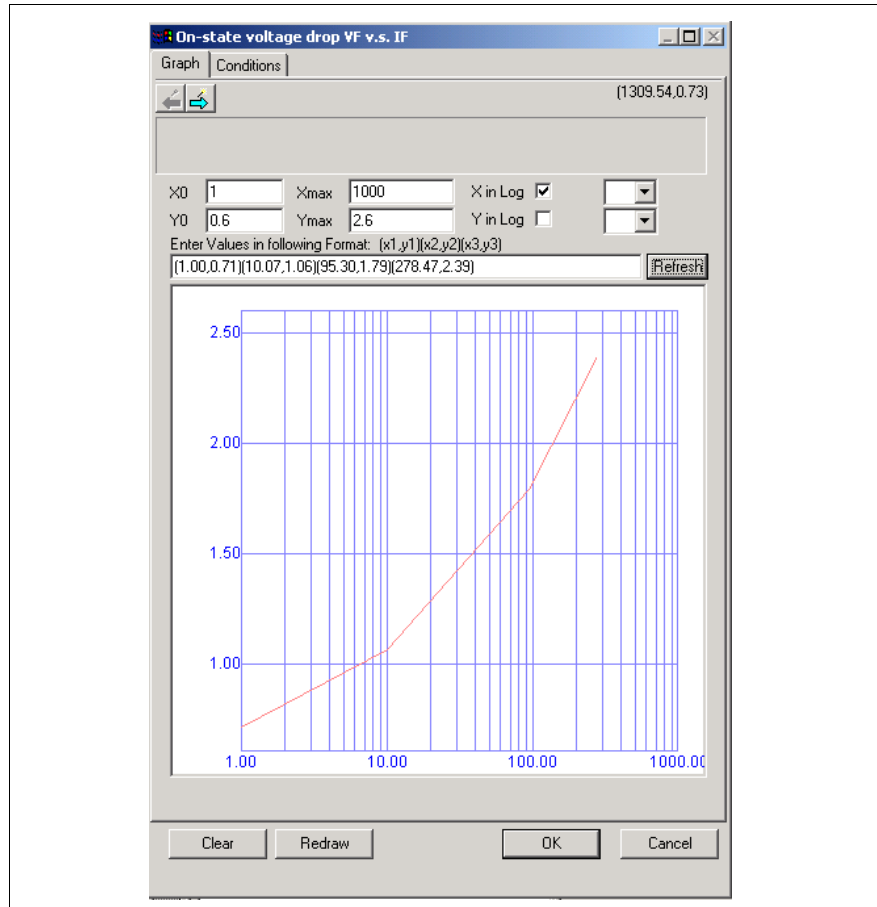
- Left click on top of the graph to capture the data points. In this case, for example, four data points at the current values of around 1A, 10A, 100A, and 300A are captured. Again, right click to zoom in.





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.



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.

- With the same process, define the reverse recovery characteristics  $t_{rr}$  v.s.  $I_F$ ,  $I_{rr}$

v.s.  $I_F$ , and  $Q_{rr}$  v.s.  $I_F$ .

- 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

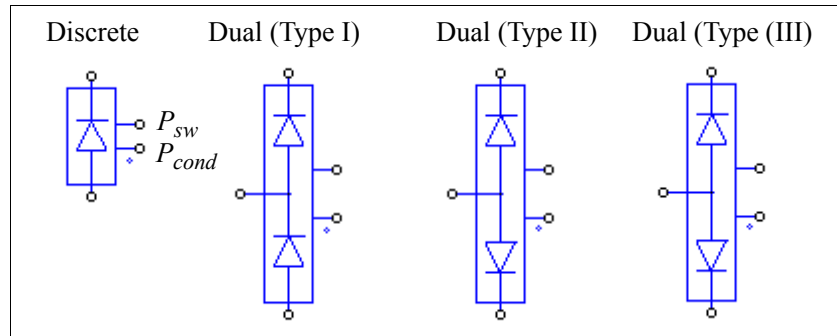
Choose **Device** -> **Save Device** to save the device information. This completes the process of adding the diode into the database.

## 2.7.2 Diode Device in the 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 or dual packages, 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 with no dot is for the diode switching losses  $P_{sw}$ .

*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:*

$V_d$  v.s.  $I_F$ : Forward conduction voltage drop  $V_d$  v.s. forward current  $I_F$

$t_{rr}$  v.s.  $I_F$ : Reverse recovery time  $t_{rr}$  v.s. current  $I_F$

$I_{rr}$  v.s.  $I_F$ : Peak reverse recovery current  $I_{rr}$  v.s. current  $I_F$

$Q_{rr}$  v.s.  $I_F$ : Reverse recovery charge  $Q_{rr}$  v.s. current  $I_F$

*Thermal Characteristics:*

$R_{th(j-c)}$ : Junction-to-case thermal resistance, in °C/W

$R_{th(c-s)}$ : Case-to-sink thermal resistance, in °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

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.

### 2.7.3 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 $P_{sw}$

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<sub>cond</sub> Calibration Factor* is the correction factor for the conduction losses. For the example, if the calculated conduction losses before the correction is *P<sub>cond\_cal</sub>*, then

$$P_{cond} = K_{cond} * P_{cond\_cal}$$

Similarly, the parameter *P<sub>sw</sub> Calibration Factor* is the correction factor for the switching losses. For the example, if the calculated switching losses before the correction is *P<sub>sw\_cal</sub>*, then

$$P_{sw} = K_{sw} * P_{sw\_cal}$$

### **Conduction Losses:**

The diode conduction losses is calculated as:

$$\text{Conduction Losses} = V_d * I_F$$

where  $V_d$  is the diode voltage drop, and  $I_F$  is the diode forward current.

### **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:

$$\text{Turn-off Losses} = 1/4 * Q_{rr} * V_R * f$$

where  $Q_{rr}$  is the reverse recovery charge,  $V_R$  is the reverse blocking voltage, and  $f$  is the frequency as defined in the input parameter *Frequency*. The reverse recovery charge  $Q_{rr}$  is defined as:

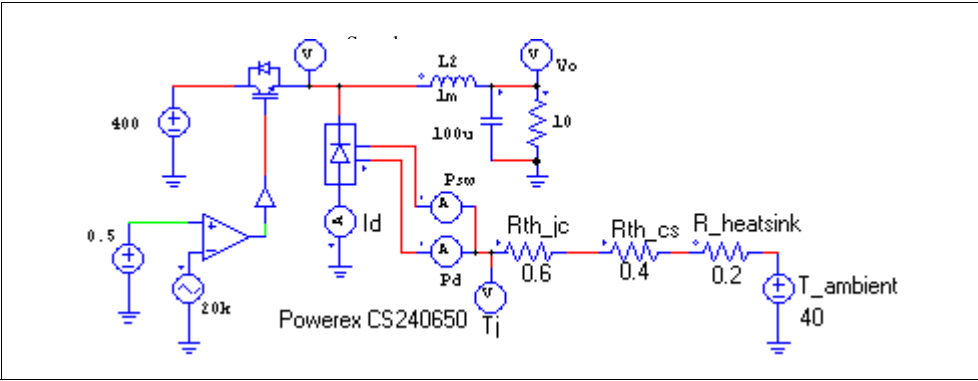
$$Q_{rr} = 1/2 * t_{rr} * I_{rr}$$

Whenever  $Q_{rr}$  is given in the device database, 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.

### **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. The circuit shows a thermal circuit without considering the thermal transient.



### 2.7.4 IGBT Device in the Database

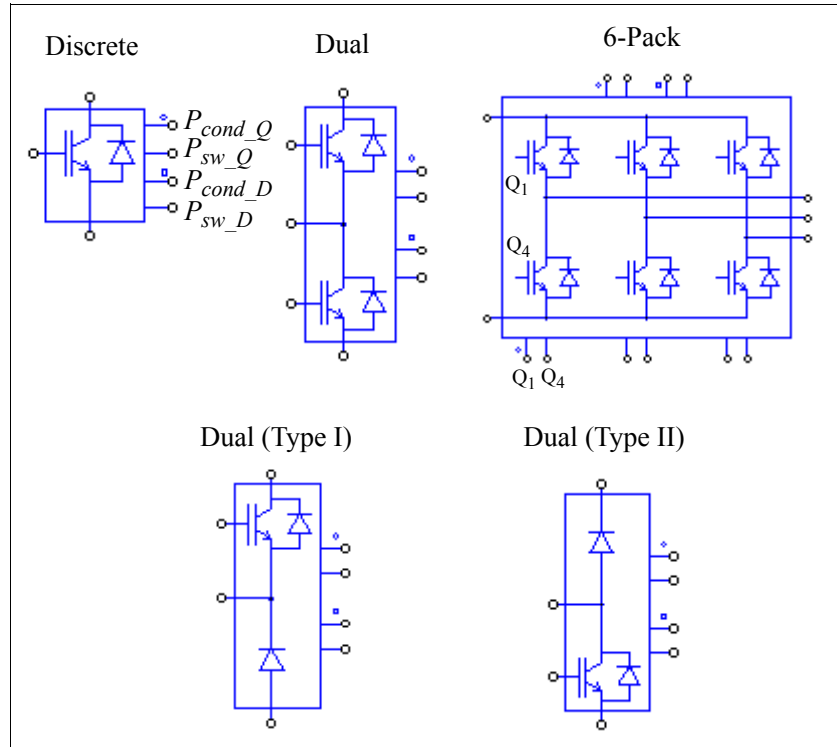
An IGBT device has three types of packages: discrete, dual, or 6-pack.

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:



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.

#### *Absolute Maximum Ratings:*

$V_{ce,max}$ (V):	Maximum collector-emitter voltage
$I_{c,max}$ (A):	Maximum collector current
$T_{j,max}$ (°C):	Maximum junction temperature

#### *Electrical Characteristics - Transistor:*

$V_{ce(sat)}$ v.s. $I_c$ :	Collector-emitter saturation voltage $V_{ce(sat)}$ v.s. collector current $I_c$
----------------------------	---

$E_{on}$ v.s. $I_c$ :	Turn-on energy losses $E_{on}$ v.s. collector current $I_c$
$E_{off}$ v.s. $I_c$ :	Turn-off energy losses $E_{off}$ v.s. collector current $I_c$

*Electrical Characteristics - Diode (or Anti-Parallel Diode):*

$V_d$ v.s. $I_F$ :	Forward conduction voltage drop v.s. forward current $I_F$
$t_{rr}$ v.s. $I_F$ :	Reverse recovery time $t_{rr}$ v.s. current $I_F$
$I_{rr}$ v.s. $I_F$ :	Peak reverse recovery current $I_{rr}$ v.s. current $I_F$
$Q_{rr}$ v.s. $I_F$ :	Reverse recovery charge $Q_{rr}$ v.s. current $I_F$

*Electrical Characteristics - Free-Wheeling Diode (for IGBT-Diode device only):*

$V_d$ v.s. $I_F$ :	Forward conduction voltage drop v.s. forward current $I_F$
$t_{rr}$ v.s. $I_F$ :	Reverse recovery time $t_{rr}$ v.s. current $I_F$
$I_{rr}$ v.s. $I_F$ :	Peak reverse recovery current $I_{rr}$ v.s. current $I_F$
$Q_{rr}$ v.s. $I_F$ :	Reverse recovery charge $Q_{rr}$ v.s. current $I_F$

*Thermal Characteristics:*

$R_{th(j-c)}$ (transistor):	Transistor junction-to-case thermal resistance, in $^{\circ}C/W$
$R_{th(j-c)}$ (diode):	Diode junction-to-case thermal resistance, in $^{\circ}C/W$
$R_{th(c-s)}$ :	Case-to-sink thermal resistance, in $^{\circ}C/W$

*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

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.

## 2.7.5 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
$P_{cond\_Q}$ 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 $P_{sw\_D}$

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.

**Conduction Losses:**

The transistor conduction losses is calculated as:

$$\text{Transistor Conduction Losses} = V_{ce(sat)} * I_c$$



where  $V_{ce(sat)}$  is the transistor collector-emitter saturation voltage, and  $I_c$  is the collect current.

### **Switching Losses:**

The transistor turn-on losses is calculated as:

$$\text{Transistor Turn-on Losses} = E_{on} * f$$

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:

$$\text{Transistor Turn-off Losses} = E_{off} * f$$

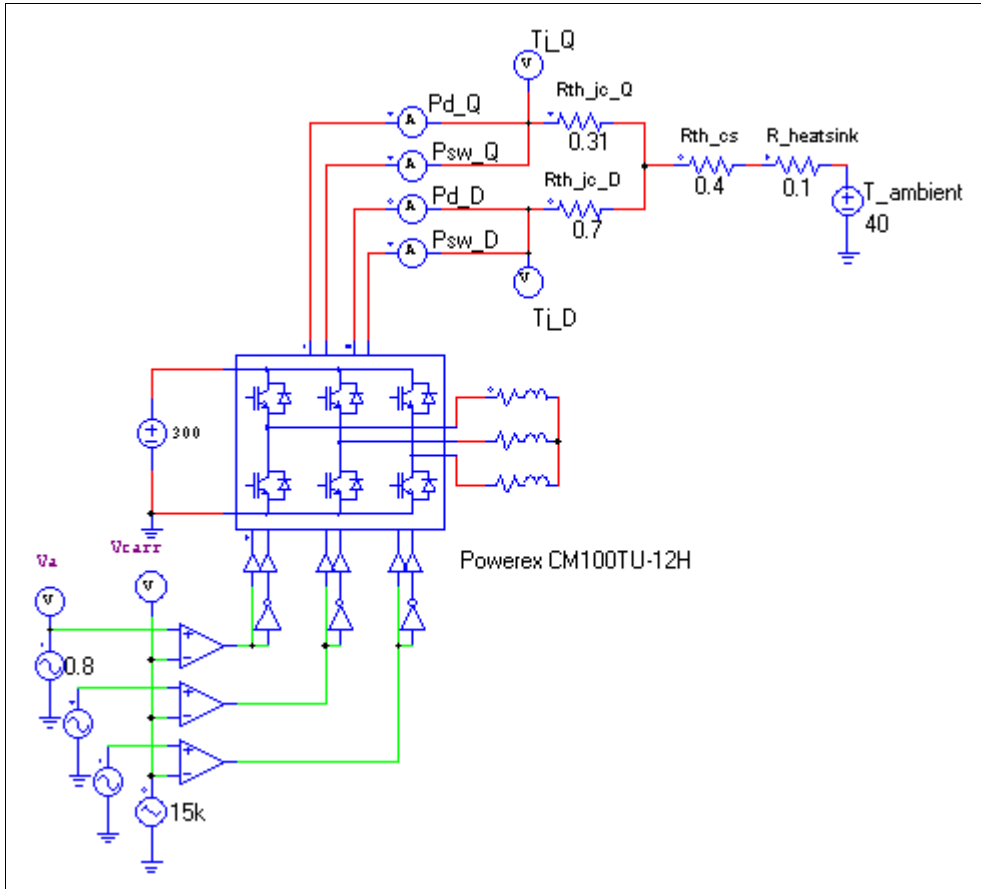
where  $E_{off}$  is the transistor turn-off energy losses.

The loss calculation for the anti-parallel diode or free-wheeling diode is the same as described in the section for the diode device.

### **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.

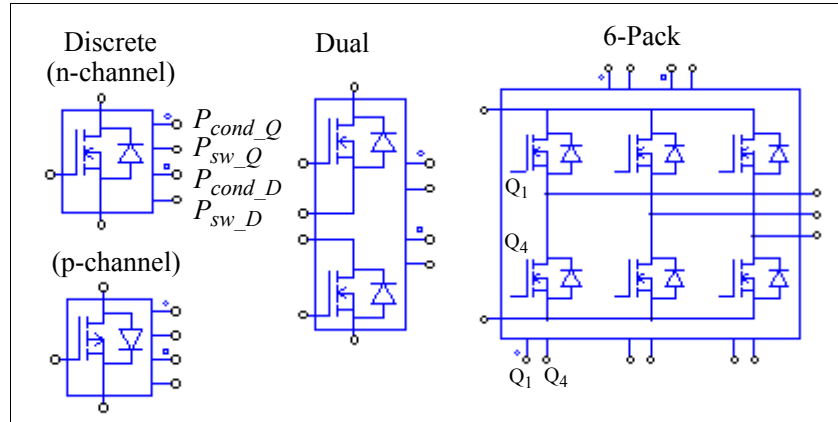


## 2.7.6 MOSFET Device in the Database

The following information is defined for a MOSFET device in the database:

### *General Information:*

Manufacturer:	Device manufacture
Part Number:	Manufacturer's part number
Package:	It can be discrete, dual, or 6-pack, as shown in the figure below:



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.

#### *Absolute Maximum Ratings:*

$V_{DS,max}$ (V):	Maximum drain-to-source voltage
$I_{D,max}$ (A):	Maximum continuous drain current
$T_{j,max}$ (°C):	Maximum junction temperature

#### *Electrical Characteristics - Transistor:*

$R_{DS(on)}$ (ohm):	Static drain-to-source on-resistance (test conditions: gate-to-source voltage $V_{GS}$ , in V, and drain current $I_D$ , in A)
$V_{GS(th)}$ (V):	Gate threshold voltage $V_{GS(th)}$ (test condition: drain current $I_D$ , in A)
$g_{fs}$ (S):	Forward transconductance $g_{fs}$ (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:*

$V_d$ v.s. $I_F$ :	Forward conduction voltage drop $V_d$ v.s. 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 °C)

*Thermal Characteristics:*

$R_{th(j-c)}$ :	Junction-to-case thermal resistance, in °C/W
$R_{th(c-s)}$ :	Case-to-sink thermal resistance, in °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

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.

## 2.7.7 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 $P_{sw\_D}$

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.

### **Conduction Losses:**

The transistor conduction losses is calculated as:

$$\text{Conduction Losses} = I_D^2 * R_{DS(on)}$$

where  $I_D$  is the drain current, and  $R_{DS(on)}$  is the static on-resistance.

### **Switching Losses:**

The transistor turn-on losses is calculated as:

$$\text{Transistor Turn-on Losses} = E_{on} * f$$

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:

$$\text{Transistor Turn-off Losses} = E_{off} * f$$

where  $E_{off}$  is the transistor turn-off energy losses. The energy losses  $E_{on}$  and  $E_{off}$  are calculated based on the information of the gate current, input/output/reverse transfer capacitances, and gate charges of the MOSFET devices.

The loss calculation for the anti-parallel diode or free-wheeling diode is the same as described in the diode device section.

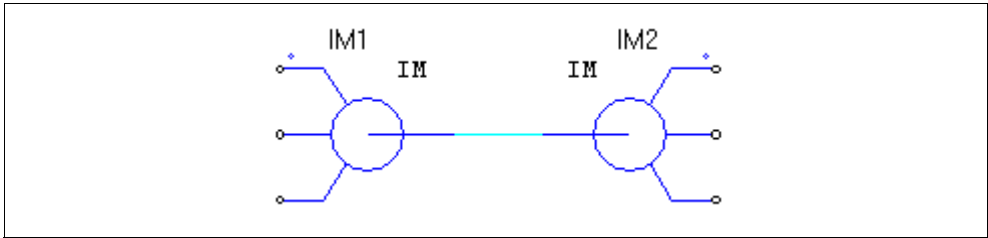
## 2.8 Motor Drive Module

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 2.11.

### 2.8.1 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.

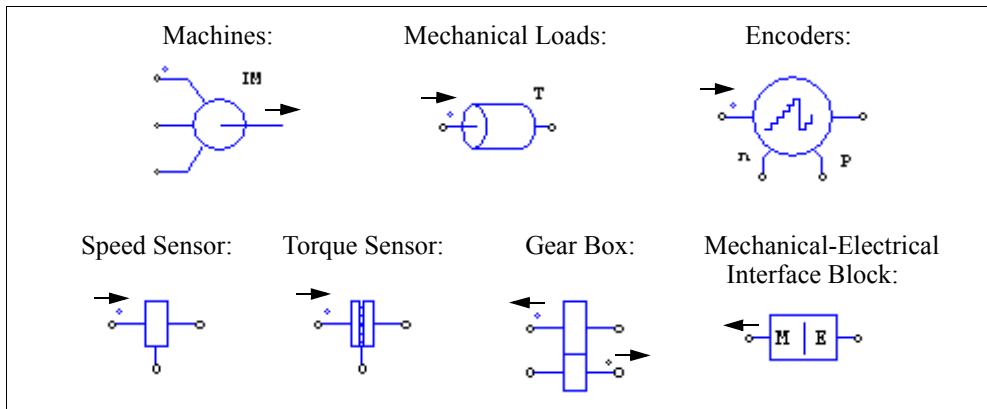
In a mechanical system, one element is designated as the master unit (this element is considered to operate in the master mode), and the rest of the elements are in the slave

mode. Elements that can be master units are: Electric machines, mechanical-to-electrical interface blocks, gear boxes, and devices modeled by Power Modeling Blocks.

The master unit defines the reference direction of the mechanical system. The direction is defined 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:



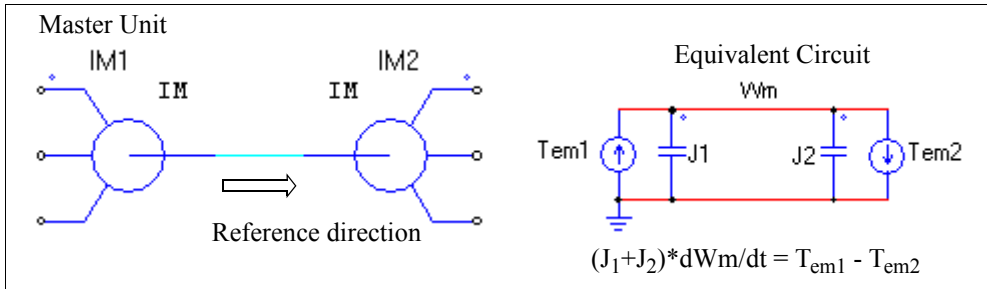
The reference direction of each element and the reference direction of the overall mechanical system determine how the element interacts with the mechanical system.

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.

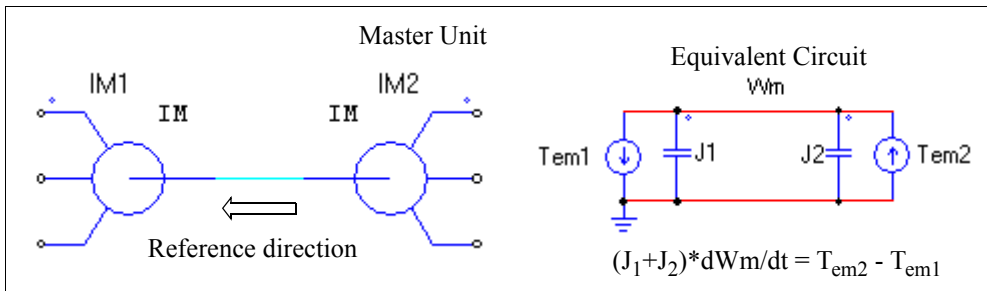
In the two-machine example above, using the notation of the "reference direction", if we define the machine IM1 as the master unit, the reference direction of the overall mechanical system will be from left to right, as shown below. Based on this direction, the machine IM1 will be along the reference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



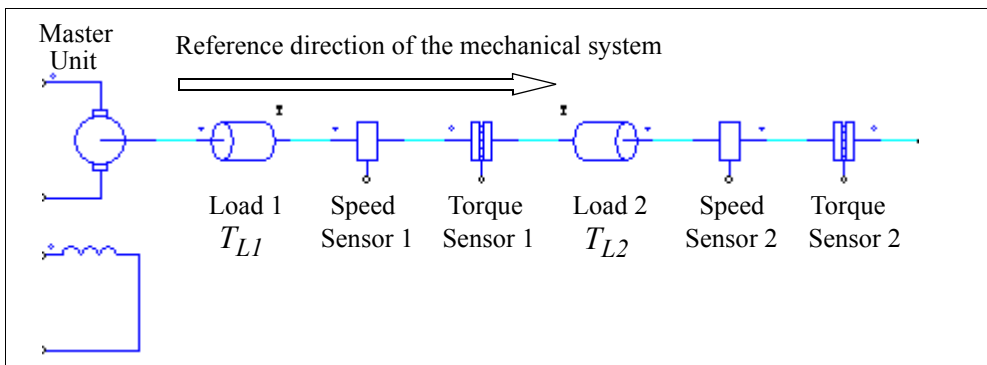
opposite to the reference direction. This leads to the equivalent circuit of the mechanical system as shown on the right.



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.



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.

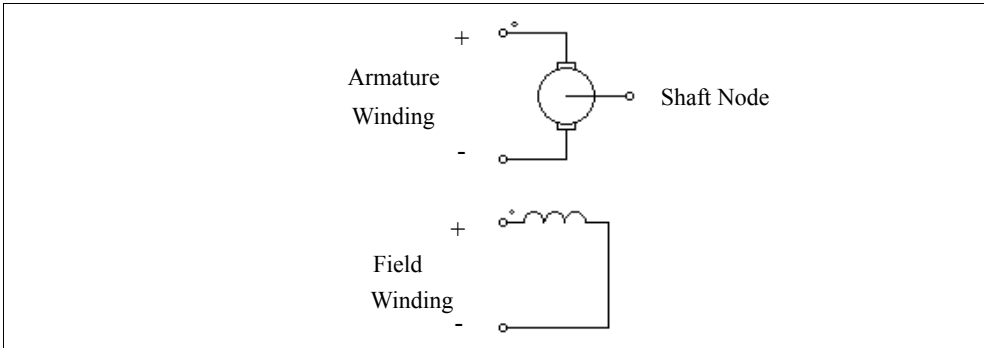
Therefore, if the speed of the machine is positive, Speed Sensor 1 reading will be positive, and Speed Sensor 2 reading will be negative.

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  $-T_{L2}$ .

### 2.8.2 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 <sup>2</sup>
$V_t$ (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 2.8.1.

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,  $\omega_m$  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:

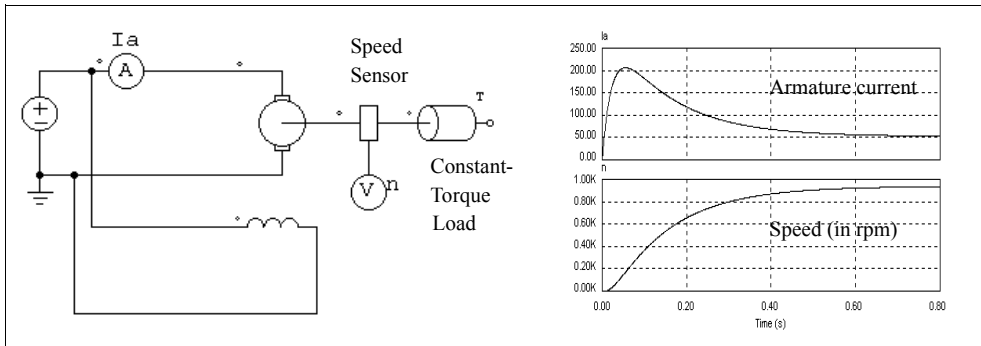
$$L_{af} = \frac{(V_t - I_a \cdot R_a)}{I_f \cdot \omega_m}$$

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.



## 2.8.3 Induction Machine

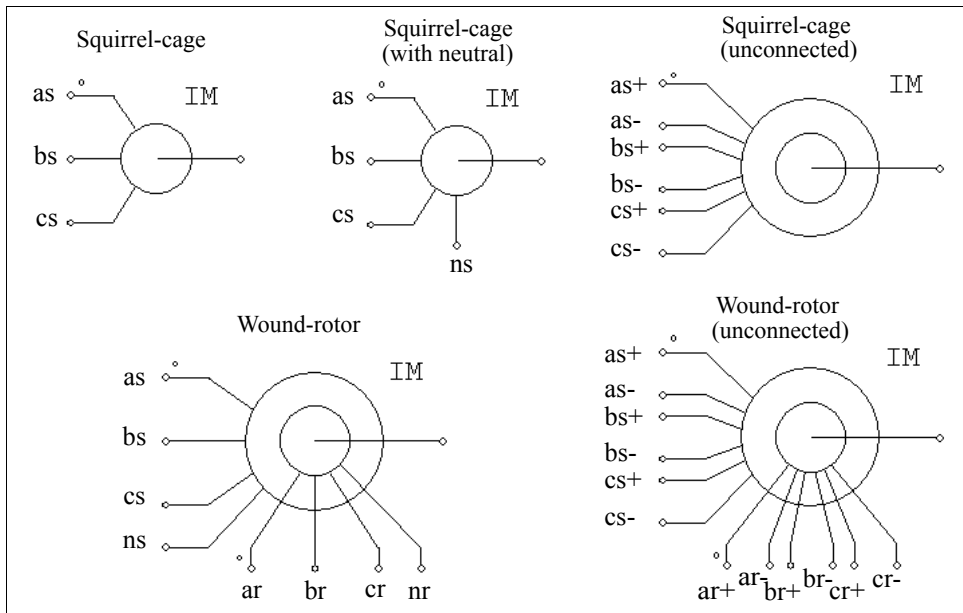
Linear and nonlinear models are provided for squirrel-cage and wound-rotor induction machines. The linear model is further divided into general type and symmetrical type. This section describes the linear models.

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, in Ohm
$L_r$ (rotor)	Rotor winding leakage inductance, 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 <sup>2</sup>
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 2.8.1.

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:

$$\begin{aligned} \begin{bmatrix} v_{abc,s} \end{bmatrix} &= \begin{bmatrix} R_s \end{bmatrix} \cdot \begin{bmatrix} i_{abc,s} \end{bmatrix} + \begin{bmatrix} L_s \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,s} \end{bmatrix} + \begin{bmatrix} M_{sr} \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,r} \end{bmatrix} \\ \begin{bmatrix} v_{abc,r} \end{bmatrix} &= \begin{bmatrix} R_r \end{bmatrix} \cdot \begin{bmatrix} i_{abc,r} \end{bmatrix} + \begin{bmatrix} L_r \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,r} \end{bmatrix} + \begin{bmatrix} M_{sr} \end{bmatrix}^T \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,s} \end{bmatrix} \end{aligned}$$

where

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

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

$$\begin{aligned} \begin{bmatrix} R_s \end{bmatrix} &= \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} & \begin{bmatrix} R_r \end{bmatrix} &= \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \\ \begin{bmatrix} L_s \end{bmatrix} &= \begin{bmatrix} L_s + M_{sr} & -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & L_s + M_{sr} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} & L_s + M_{sr} \end{bmatrix} & \begin{bmatrix} L_r \end{bmatrix} &= \begin{bmatrix} L_r + M_{sr} & -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & L_r + M_{sr} & -\frac{M_{sr}}{2} \\ -\frac{M_{sr}}{2} & -\frac{M_{sr}}{2} & L_r + M_{sr} \end{bmatrix} \\ \begin{bmatrix} M_{sr} \end{bmatrix} &= M_{sr} \cdot \begin{bmatrix} \cos \theta & \cos \left( \theta + \frac{2\pi}{3} \right) & \cos \left( \theta - \frac{2\pi}{3} \right) \\ \cos \left( \theta - \frac{2\pi}{3} \right) & \cos \theta & \cos \left( \theta + \frac{2\pi}{3} \right) \\ \cos \left( \theta + \frac{2\pi}{3} \right) & \cos \left( \theta - \frac{2\pi}{3} \right) & \cos \theta \end{bmatrix} \end{aligned}$$

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

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

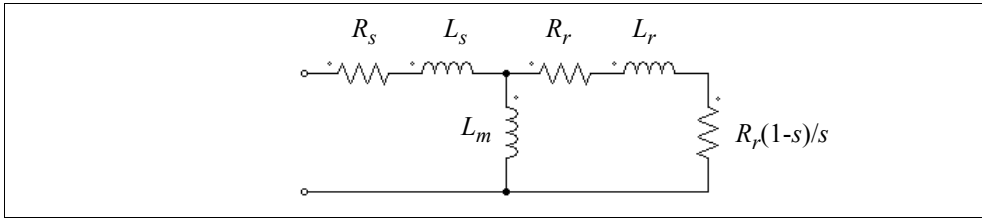
The mechanical equation is expressed as:

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

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

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

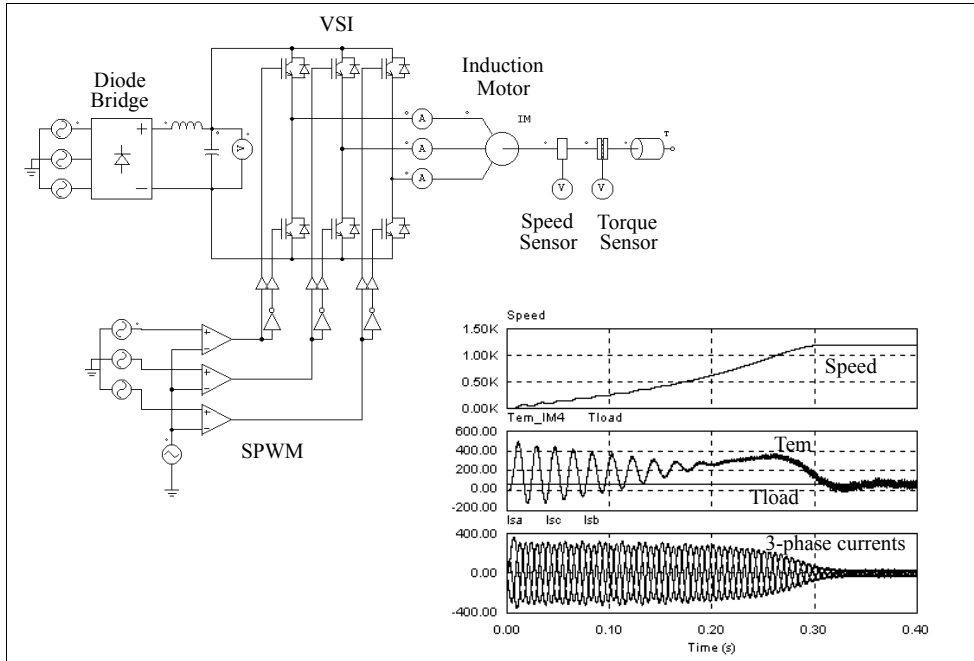
For a symmetrical squirrel-cage induction machine, the steady state equivalent circuit is shown below. In the figure,  $s$  is the slip.



### Example: A VSI Induction Motor Drive System

The figure below shows an open-loop induction motor drive system. The motor has 6 poles and is fed by a voltage source inverter with sinusoidal PWM. The dc bus is fed through a diode bridge.

The simulation waveforms of the mechanical speed (in rpm), developed torque  $T_{em}$  and load torque  $T_{load}$ , and 3-phase input currents show the start-up transient.

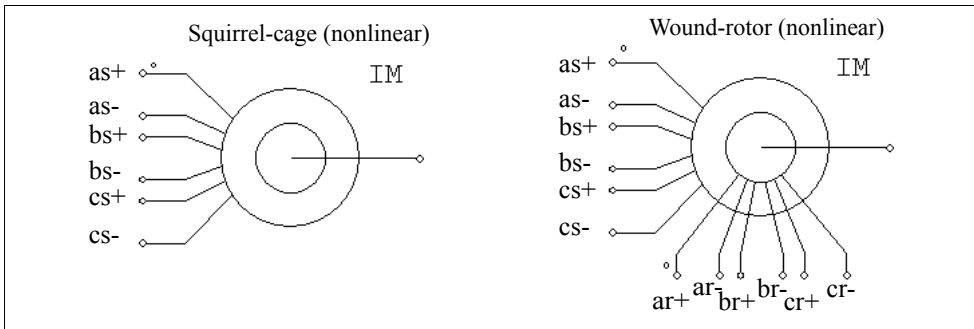


## 2.8.4 Induction Machine with Saturation

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, in Ohm
$L_r$ (rotor)	Rotor winding leakage 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 <sup>2</sup>
Torque Flag	Output flag for internal torque $T_{em}$
Master/Slave Flag	Master/slave flag of the machine (1: master; 0: slave)
$I_m$ v.s. $L_m$ ( $I_{m1}, L_{m1}$ ) ...	Characteristics of the magnetizing current $I_m$ versus the magnetizing inductance [ $(I_{m1}, L_{m1})$ ( $I_{m2}, L_{m2}$ ) ...]

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 2.8.1.

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

$$\begin{aligned} \begin{bmatrix} v_{abc,s} \end{bmatrix} &= \begin{bmatrix} R_s \end{bmatrix} \cdot \begin{bmatrix} i_{abc,s} \end{bmatrix} + L_s \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,s} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_{abc,s} \end{bmatrix} \\ \begin{bmatrix} v_{abc,r} \end{bmatrix} &= \begin{bmatrix} R_r \end{bmatrix} \cdot \begin{bmatrix} i_{abc,r} \end{bmatrix} + L_r \cdot \frac{d}{dt} \begin{bmatrix} i_{abc,r} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_{abc,r} \end{bmatrix} \end{aligned}$$

where

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

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

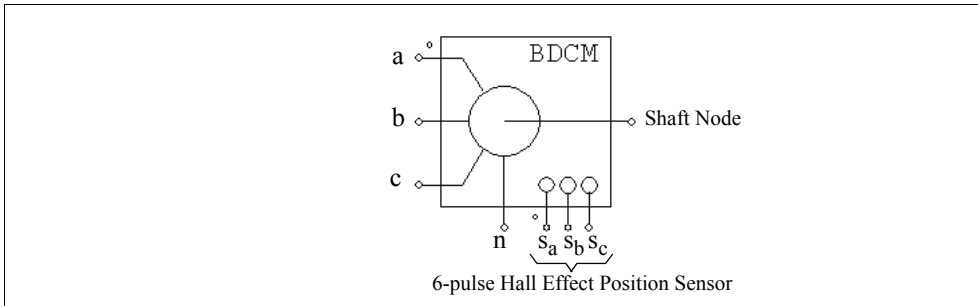
In this case, the inductance  $M_{sr}$  is no longer constant, but a function of the magnetizing current  $I_m$ .

### 2.8.5 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 and parameters of the 3-phase brushless dc machine are shown as follows.

**Image:**



**Attributes:**

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.)	<p>Stator mutual inductance <math>M</math>, in H</p> <p>The mutual inductance <math>M</math> is a negative value. Depending on the winding structure, the ratio between <math>M</math> and the stator self inductance <math>L</math> is normally between -1/3 and -1/2. If <math>M</math> is unknown, a reasonable value of <math>M</math> equal to <math>-0.4*L</math> can be used as the default value.</p>
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed)
Vrms / krpm	<p>RMS line-to-line back emf constant, in V/krpm (mechanical speed).</p> <p>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.</p>
No. of Poles $P$	Number of poles $P$
Moment of Inertia	Moment of inertia $J$ of the machine, in $\text{kg}\cdot\text{m}^2$
Mech. Time Constant	Mechanical time constant $\tau_{\text{mech}}$
theta_0 (deg.)	<p>Initial rotor angle <math>\theta_r</math>, in electrical deg.</p> <p>The initial rotor angle is the rotor angle at <math>t=0</math>. 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.</p>
theta_advance (deg.)	<p>Position sensor advance angle <math>\theta_{\text{advance}}</math>, in electrical deg.</p> <p>The advance angle is defined as such that, for a brushless dc machine with a <math>120^\circ</math> 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.</p>

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° 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)

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 2.8.1.

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.

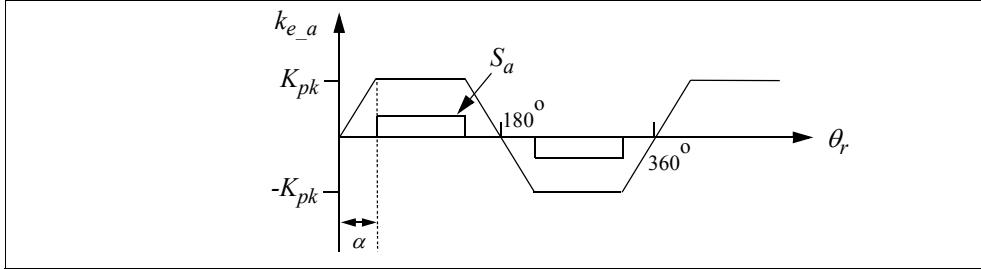
The back emf voltages are a function of the rotor mechanical speed  $\omega_m$  and the rotor electrical angle  $\theta_r$ , that is:

$$E_a = k_{e\_a} \cdot \omega_m$$

$$E_b = k_{e\_b} \cdot \omega_m$$

$$E_c = k_{e\_c} \cdot \omega_m$$

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  $V_{pk}/krpm$  and  $V_{rms}/krpm$ , the angle  $\alpha$  is determined automatically in PSIM.

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 a coefficient,  $T_{load}$  is the load torque, and  $P$  is the no. of poles. The coefficient  $B$  is calculated from the moment of inertia  $J$  and the mechanical time constant  $\tau_{mech}$  as below:

$$B = \frac{J}{\tau_{mech}}$$

### 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.

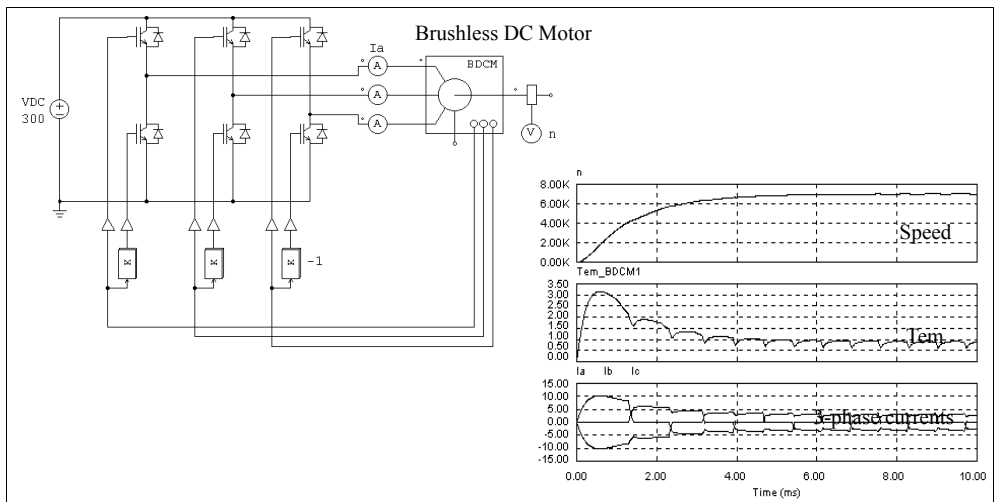
The hall switch is a semiconductor switch (e.g. MOSFET or BJT) that opens or closes when the magnetic field is higher or lower than a certain threshold value. It is based on the hall effect, which generates an emf proportional to the flux-density when the switch is carrying a current supplied by an external 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: 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 gating signals for the inverter, resulting a 6-pulse operation.

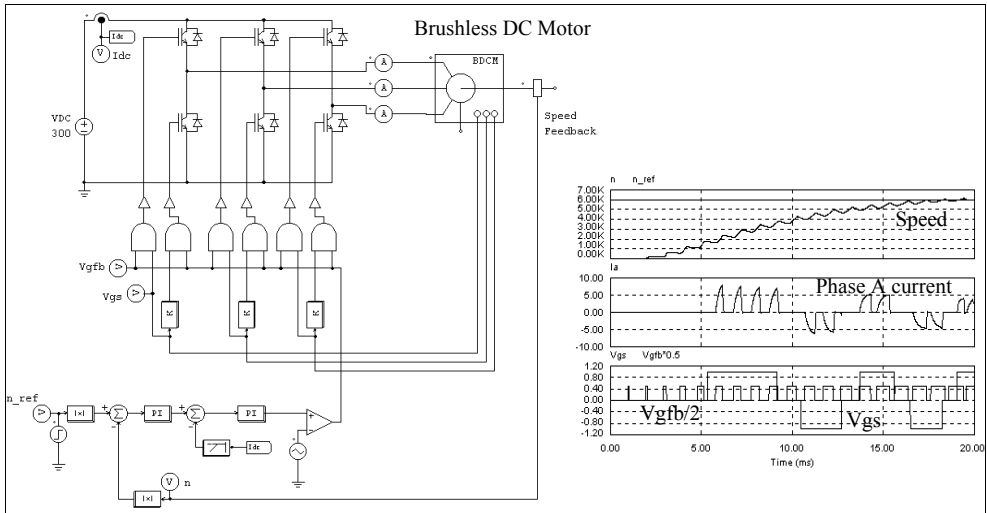
The simulation waveforms show the start-up transient of the mechanical speed (in rpm), developed torque  $T_{em}$ , and 3-phase input currents.



### Example: Brushless DC Motor with Speed Feedback

The figure below shows a brushless dc motor drive system with speed feedback. The speed control is achieved by modulating sensor commutation pulses ( $V_{gs}$  for Phase  $A$  in this case) with another high-frequency pulses ( $V_{gfb}$  for Phase  $A$ ). The high-frequency pulse is generated from a dc current feedback loop.

The simulation waveforms show the reference and actual mechanical speed (in rpm), Phase  $A$  current, and signals  $V_{gs}$  and  $V_{gfb}$ . Note that  $V_{gfb}$  is divided by half for display purpose.



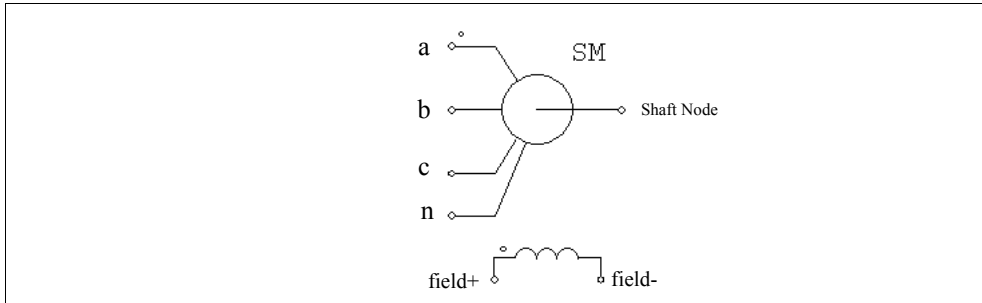
### 2.8.6 Synchronous Machine with External Excitation

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, in Ohm
$L_{fl}$ (field leakage ind.)	Field winding leakage inductance, in H
$R_{dr}$ (damping cage)	Rotor damping cage d-axis resistance, in Ohm
$L_{drl}$ (damping cage)	Rotor damping cage d-axis leakage inductance, in H
$R_{qr}$ (damping cage)	Rotor damping cage q-axis resistance, in Ohm
$L_{qrl}$ (damping cage)	Rotor damping cage q-axis leakage inductance, 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 $\text{kg}\cdot\text{m}^2$
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 2.8.1.

The equations of the synchronous machine can be expressed as follows:

$$[V] = [R] \cdot [I] + \frac{d}{dt}[\lambda]$$

where

$$[V] = [v_a \ v_b \ v_c \ v_f \ 0 \ 0]^T \quad [I] = [i_a \ i_b \ i_c \ i_f \ i_{dr} \ i_{qr}]^T$$

$$[R] = \text{diag}[R_s \ R_s \ R_s \ R_f \ R_{dr} \ R_{qr}] \quad [\lambda] = [\lambda_a \ \lambda_b \ \lambda_c \ \lambda_f \ \lambda_{dr} \ \lambda_{qr}]^T$$

and  $[\lambda] = [L][I]$ . 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  $\theta_r$  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:

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

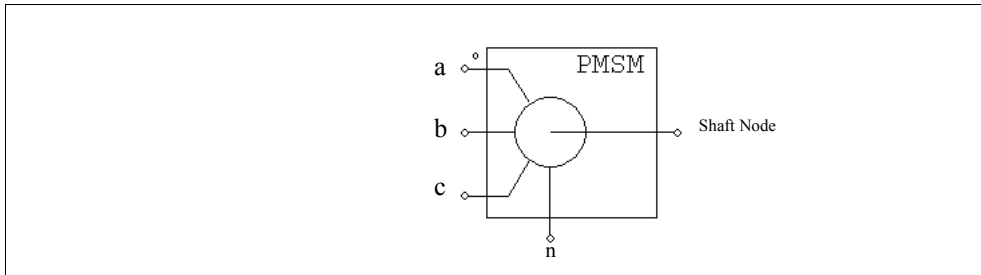
$$\frac{d\theta_r}{dt} = \frac{P}{2} \cdot \omega_m$$

## 2.8.7 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 <sup>2</sup>
Mech. Time Constant	Mechanical time constant $\tau_{\text{mech}}$
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 2.8.1.

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  $\lambda_a$ ,  $\lambda_b$ ,  $\lambda_c$  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  $\theta_r$  is the rotor electrical angle, and  $\lambda_{pm}$  is a coefficient which is 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:

$$\begin{aligned} L_{aa} &= L_s + L_o + L_2 \cdot \cos(2\theta_r) \\ L_{bb} &= L_s + L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ L_{cc} &= L_s + L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{ab} &= L_{ba} = -\frac{L_o}{2} + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{ac} &= L_{ca} = -\frac{L_o}{2} + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ L_{bc} &= L_{cb} = -\frac{L_o}{2} + L_2 \cdot \cos(2\theta_r) \end{aligned}$$

where  $L_s$  is the stator leakage inductance. The d-axis and q-axis inductances are associated with the above inductances as follows:

$$\begin{aligned} L_d &= L_s + \frac{3}{2}L_o + \frac{3}{2}L_2 \\ L_q &= L_s + \frac{3}{2}L_o - \frac{3}{2}L_2 \end{aligned}$$

The developed torque can be expressed as:

$$T_{em} = \frac{P}{2} \cdot L_2 \cdot \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \cdot \begin{bmatrix} \sin(2\theta_r) & \sin\left(2\theta_r - \frac{2\pi}{3}\right) & \sin\left(2\theta_r + \frac{2\pi}{3}\right) \\ \sin\left(2\theta_r - \frac{2\pi}{3}\right) & \sin\left(2\theta_r + \frac{2\pi}{3}\right) & \sin(2\theta_r) \\ \sin\left(2\theta_r + \frac{2\pi}{3}\right) & \sin(2\theta_r) & \sin\left(2\theta_r - \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} -$$

$$\frac{P}{2} \cdot \lambda_{pm} \cdot \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \cdot \begin{bmatrix} \sin(\theta_r) \\ \sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}$$

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 no. of poles. The coefficient  $B$  is calculated from the moment of inertia  $J$  and the mechanical time constant  $\tau_{mech}$  as below:

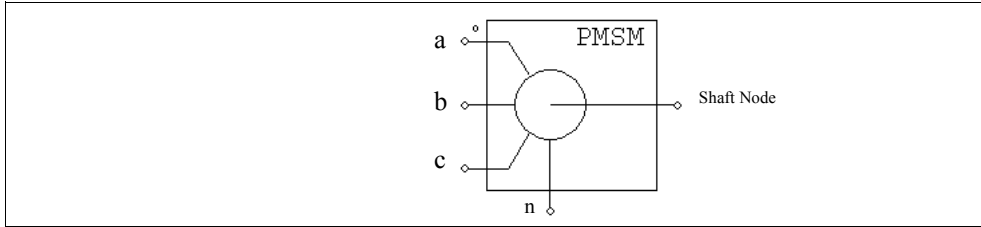
$$B = \frac{J}{\tau_{mech}}$$

### 2.8.8 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).  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 $\text{kg}\cdot\text{m}^2$
Mech. Time Constant	Mechanical time constant $\tau_{\text{mech}}$ , in sec. It is associated with the friction coefficient $B$ as: $B = J / \tau_{\text{mech}}$ .
$L_d$ 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 2.8.1.

The relationship between the d-axis/q-axis inductances  $L_d/L_q$  and the magnetizing inductances  $L_{dm}/L_{qm}$  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:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

$$I_m = \sqrt{I_d^2 + I_q^2}$$

$$\theta_m = \text{atan2}(I_q, I_d)$$

The current vector angle is in deg., and is from -180° to 180°.

When the *Transformation Flag* is 1:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

$$I_m = \sqrt{\frac{2}{3}} \cdot \sqrt{I_d^2 + I_q^2}$$

$$\theta_m = \text{atan2}(-I_d, I_q)$$

The current vector angle is in deg., and is from 0° to 360°.

The  $L_{dm}$  and  $L_{qm}$  lookup tables have the following format:

$$\begin{array}{c}
 m, n \\
 V_{r,1}, V_{r,2}, \dots, V_{r,m} \\
 V_{c,1}, V_{c,2}, \dots, V_{c,n} \\
 L_{1,1}, L_{1,2}, \dots, L_{1,n} \\
 L_{2,1}, L_{2,2}, \dots, L_{2,n} \\
 \dots \dots \dots \\
 L_{m,1}, L_{m,2}, \dots, L_{m,n}
 \end{array}$$

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.

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 be the same as 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

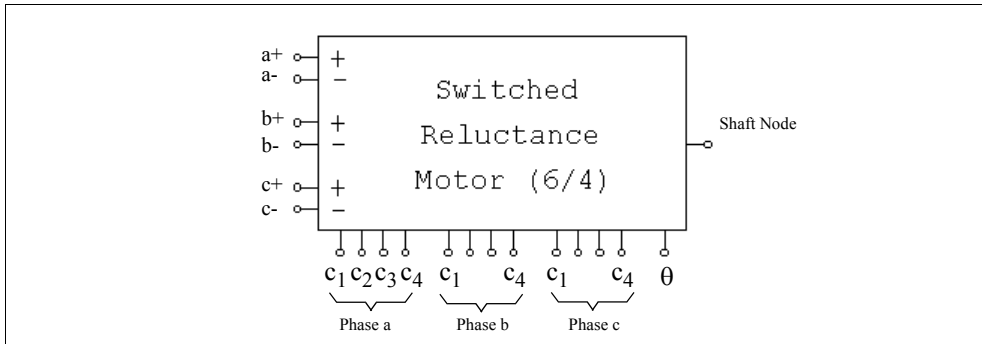
```



## 2.8.9 Switched Reluctance Machine

The model of a 3-phase switched reluctance machine with 6 stator teeth and 4 rotor teeth is provided. The images and parameters are shown as follows.

**Image:**



**Attributes:**

Parameters	Description
Resistance	Stator phase resistance $R$ , in Ohm
Inductance $L_{min}$	Minimum phase inductance, in H
Inductance $L_{max}$	Maximum phase inductance, in H
$\theta_r$	Duration of the interval where the inductance increases, in deg.
Moment of Inertia	Moment of inertia $J$ of the machine, in $\text{kg}\cdot\text{m}^2$
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 2.8.1.

The node assignments are: Nodes a+, a-, b+, b-, and c+, c- are the stator winding terminals for Phase a, b, and c, 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.

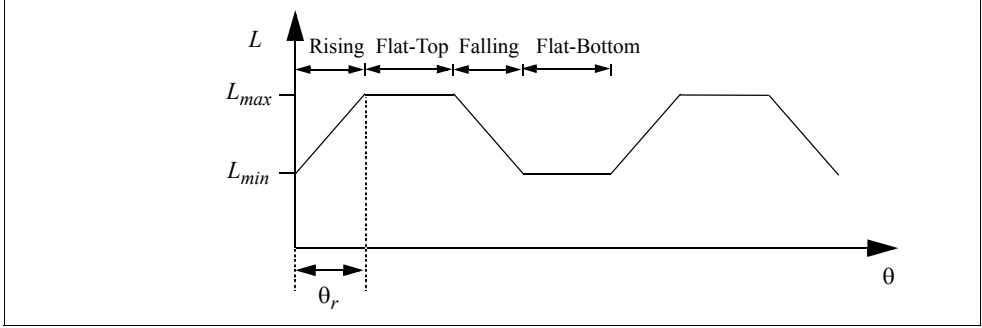
Node c1, c2, c3, and c4 are the control signals for Phase a, b, and c, respectively. The control signal value is a logic value of either 1 (high) or 0 (low). Node  $\theta$  is the

mechanical rotor angle. They are all control nodes and should be connected to the control 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  $\theta$ , as shown in the following figure.



The rotor angle is defined such that, when the stator and the rotor teeth are completely out of alignment,  $\theta = 0$ . 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:

$$k = \frac{L_{max} - L_{min}}{\theta_r}$$

we can express the inductance  $L$  as a function of the rotor angle  $\theta$ :

$$L = L_{min} + k * \theta \quad [\text{rising stage. Control signal } c_1=1]$$

$$L = L_{max} \quad [\text{flat-top stage. Control signal } c_2=1]$$

$$L = L_{max} - k * \theta \quad [\text{falling stage. Control signal } c_3=1]$$

$$L = L_{min} \quad [\text{flat-bottom stage. Control signal } c_4=1]$$

The selection of the operating state is done through control signals  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$  which are applied externally. For example, when  $c_1$  in Phase  $a$  is high (1), the rising stage is selected and Phase  $a$  inductance will be:  $L = L_{min} + k * \theta$ . Note that only one and at least one control signal out of  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$  in one phase must be high (1).

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:

$$T_{em} = i^2 * k / 2 \quad [\text{rising stage}]$$

$$T_{em} = 0 \quad [\text{flat-top stage}]$$

$$T_{em} = - i^2 * k / 2 \quad [\text{falling stage}]$$

$$T_{em} = 0 \quad [\text{flat-bottom stage}]$$

Note that saturation is not considered in this model.

## 2.9 MagCoupler Module

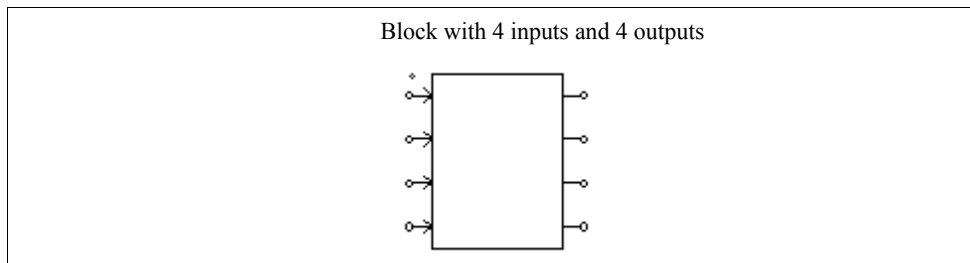
The MagCoupler Module provides interface for co-simulation between PSIM and the software JMAG. JMAG is an electromagnetic field analysis software for the development and design of electric machines, actuators, and other electrical and electronic devices and components. With the MagCoupler Module, one can perform power electronics and control analysis and simulation, as well as the electromagnetic field analysis, all in one integral environment.

The MagCoupler Module includes the MagCoupler interface block as described in this section, and mechanical elements and speed/torque/position sensors as described in Section 2.11.

To run the MagCoupler Module, the path of the JMAG directory must be included in PSIM so that PSIM can load JMAG DLL files. To set the JMAG directory path in PSIM, go to **Options** -> **Set Path**, and click on **Add Folder** to include the JMAG directory.

Also, the MagCoupler Module requires Microsoft Internet Explorer Version 6 or higher. It will not work with Internet Explorer Version 5.

### Image:




### 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:

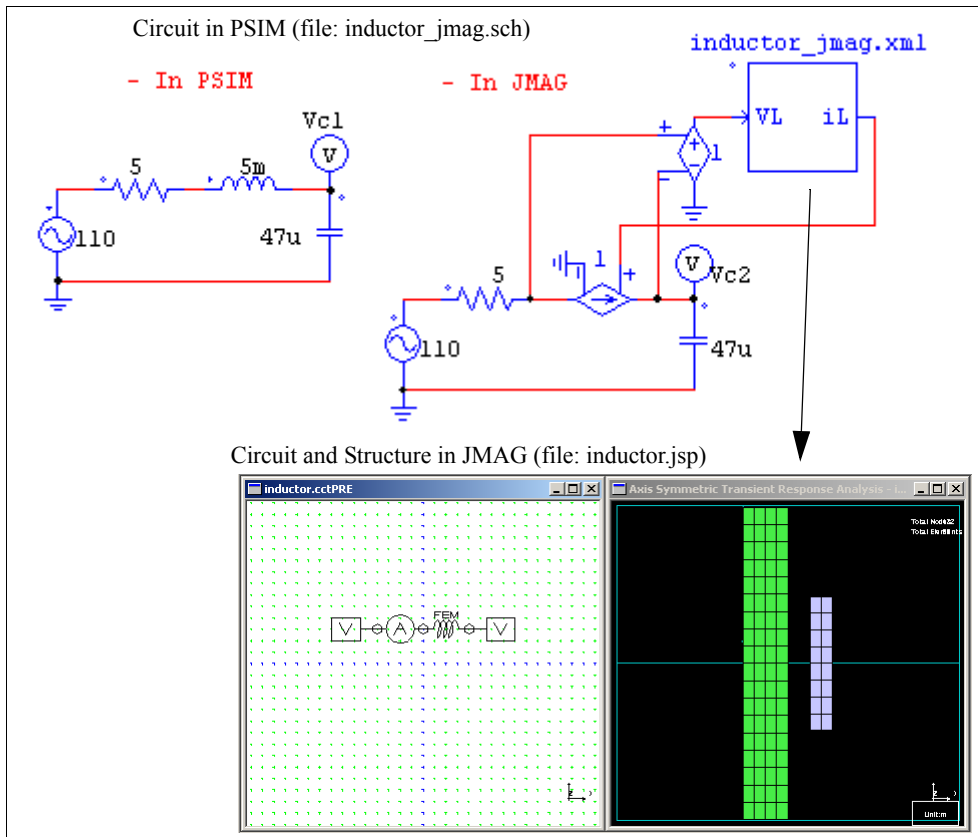
Using the MagCoupler block, it is easy to set up the link between JMAG and PSIM for co-simulation. It involves two main steps: setting up the circuit in JMAG and generating

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.



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.



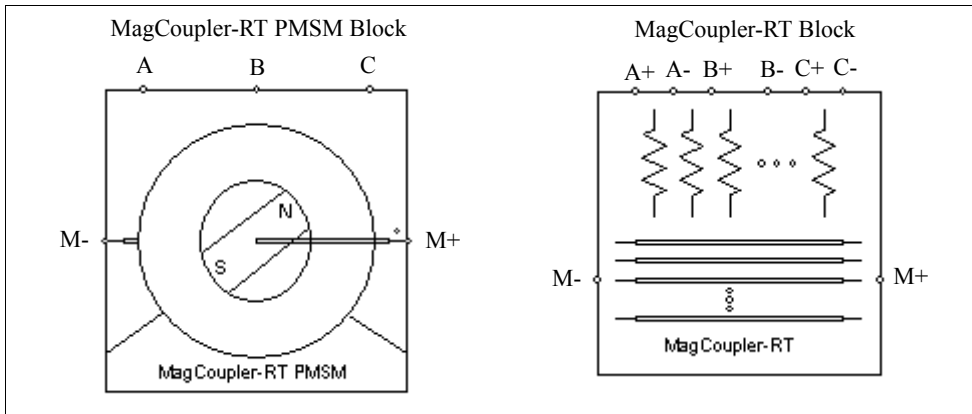
## 2.10 MagCoupler-RT Module

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 are stored in a lookup table form. During the PSIM simulation, JMAG is no longer 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 Module includes the MagCoupler-RT blocks as described in this section, and mechanical elements and speed/torque/position sensors as described in Section 2.11.

### 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 electric nodes (such as  $A$ ,  $B$ ,  $C$ ,  $A+$ ,  $A-$ ,  $B+$ ,  $B-$ ,  $C+$ , and  $C-$ , as shown above) are placed at the top of the block, arranged from the left to the right. The rotor 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, are defined in the *JMAG-RT Input 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.

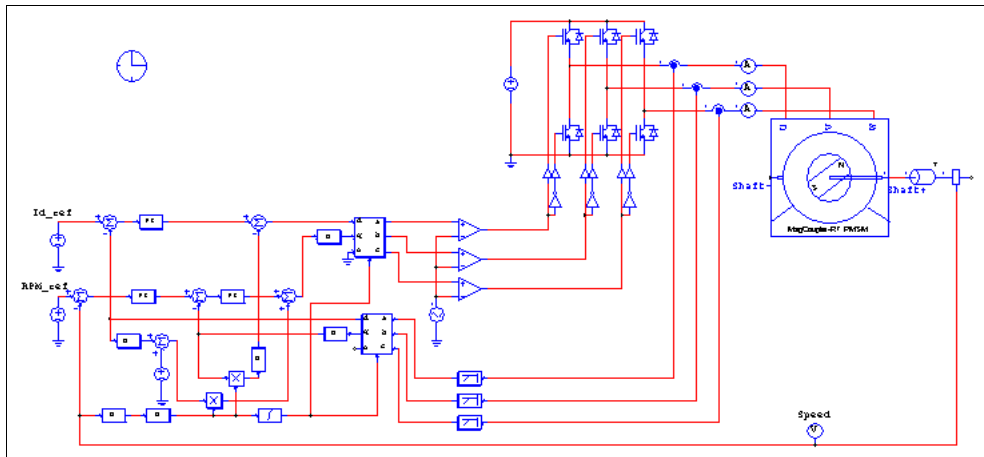
The *JMAG-RT Input File* is the JMAG-RT data file for the device modeled. The file has the .rtt extension and is defined in the JMAG-RT Input File. Note that the .rtt file and the .xml file must be in the same directory.

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 nodes on the top of the block are the power circuit nodes, and the nodes on the left and right of the blocks are the mechanical shaft nodes.

### Example: A PMSM motor drive system with PMSM modeled in JMAG-RT

The figure below shows a permanent-magnet synchronous motor (PMSM) drive system, with the PMSM modeled in JMAG-RT.



The figure below shows the property window of the MagCoupler-RT PMSM block. The .xml file in this example defines three electric nodes (Nodes  $U$ ,  $V$ , and  $W$ ), and two rotor shaft nodes (Nodes  $shaft+$  and  $shaft-$ ). The shaft nodes can be connected directly to

other mechanical elements in the PSIM library, as shown in this case.

Besides the information of the .rtt file and the terminal names, the property window also shows a set of parameters that allow users to modify the values of selected variables in the JMAG-RT data file. The last five parameters are the flags that, when set to 1, will display the currents, back emf, as well as the rotor angle, speed, and the developed torque of the machine. For definitions and the usage of these parameters, please consult the relevant JMAG-RT document.

**MagCoupler-RT Block**

MagCoupler-RT Block Help

Name	IPMY	Display	<input type="checkbox"/>
Netlist XML File	magRT_PMSM_Y.xml	Display	<input type="checkbox"/>
JMAG-RT Input File	Jes\PMSM\V15\ipm.rtt	Display	<input type="checkbox"/>
JMAG-RT Case Text	IPM Motor	Display	<input type="checkbox"/>

Terminal Names: ☒

- U
- V
- W
- Shaft+
- Shaft-

shaft1_MomentofInertia	0.0002	Display	<input type="checkbox"/>
shaft1_MechTimeConst	4.9266	Display	<input type="checkbox"/>
RU_resistance	3.3	Display	<input type="checkbox"/>
RV_resistance	3.3	Display	<input type="checkbox"/>
RW_resistance	3.3	Display	<input type="checkbox"/>
coef_flux	1.0	Display	<input type="checkbox"/>
coef_inductance	1.0	Display	<input type="checkbox"/>
coef_torque	1.0	Display	<input type="checkbox"/>
coef_magnet	1.0	Display	<input type="checkbox"/>
coef_material	1.0	Display	<input type="checkbox"/>

turns_coil1	35.0	Display	<input type="checkbox"/>
turns_coil2	35.0	Display	<input type="checkbox"/>
Current Flag	1	Display	<input type="checkbox"/>
Back emf Flag	1	Display	<input type="checkbox"/>
Rotor Angle Flag	1	Display	<input type="checkbox"/>
Speed Flag	1	Display	<input type="checkbox"/>
Torque Flag	1	Display	<input type="checkbox"/>

Edit Image Display File Read File

## 2.11 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.

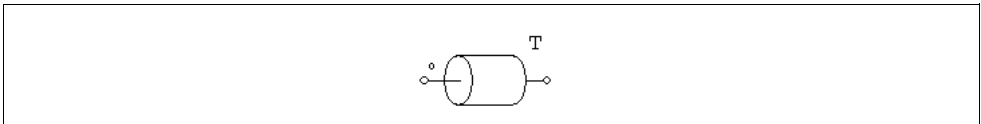
### 2.11.1 Mechanical Loads

Several mechanical load models are provided: constant-torque, constant-power, constant-speed, general-type, and externally controlled loads.

#### 2.11.1.1 Constant-Torque Load

The image of a constant-torque load is:

Image:



Attributes:

Parameters	Description
Constant Torque	Torque constant $T_{\text{const}}$ , in N*m
Moment of Inertia	Moment of inertia of the load, in kg*m <sup>2</sup>

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  $T_{\text{const}}$ . Otherwise the loading torque will be  $-T_{\text{const}}$ . See Section 2.6.1 for more detailed explanation on the reference direction.

A constant-torque load is expressed as:

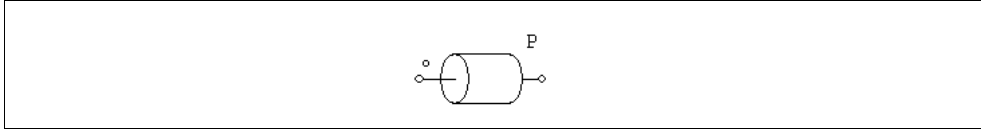
$$T_L = T_{\text{const}}$$

The torque does not depend on the speed direction.

### 2.11.1.2 Constant-Power Load

The image of a constant-power load is:

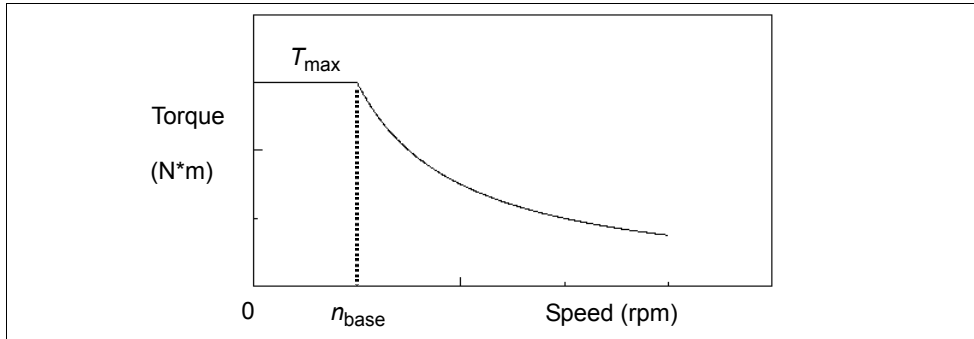
**Image:**



**Attributes:**

Parameters	Description
Maximum Torque	Maximum torque $T_{\max}$ of the load, in N*m
Base Speed	Base speed $n_{\text{base}}$ of the load, in rpm
Moment of Inertia	Moment of inertia of the load, in kg*m <sup>2</sup>

The torque-speed curve of a constant-power load is shown below:



When the mechanical speed is less than the base speed  $n_{\text{base}}$ , the load torque is:

$$T_L = T_{\max}$$

When the mechanical speed is above the base speed, the load torque is:

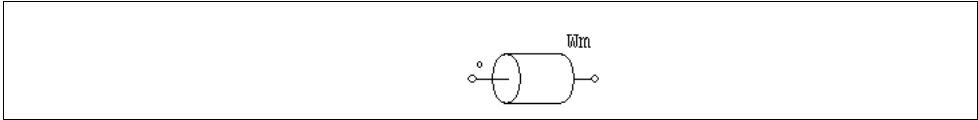
$$T_L = \frac{P}{|\omega_m|}$$

where  $P = T_{\max} * \omega_{\text{base}}$  and  $\omega_{\text{base}} = 2\pi * n_{\text{base}} / 60$ . The mechanical speed  $\omega_m$  is in rad./sec.

### 2.11.1.3 Constant-Speed Load

The image of a constant-torque load is:

**Image:**



**Attributes:**

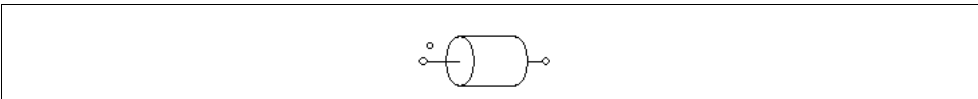
Parameters	Description
Constant Speed (rpm)	Speed constant, in rpm
Moment of Inertia	Moment of inertia of the load, in $\text{kg}\cdot\text{m}^2$

A constant-speed mechanical load defines the speed of a mechanical system, and the speed will remain constant, as defined by the speed constant.

### 2.11.1.4 General-Type Load

The image of a general-type mechanical load is as follows.

**Image:**



**Attributes:**

Parameters	Description
$T_c$	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 $\text{kg}\cdot\text{m}^2$

A general-type load is expressed as:

$$T_L = \text{sign}(\omega_m) \cdot (T_c + k_1 \cdot |\omega_m| + k_2 \cdot \omega_m^2 + k_3 \cdot |\omega_m|^3)$$

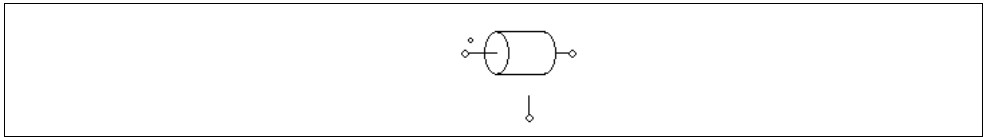
where  $\omega_m$  is the mechanical speed in rad./sec.

Note that the torque of the general-type load is dependent on the speed direction.

### 2.11.1.5 Externally-Controlled Load

An externally-controlled mechanical load is used to define a load of an arbitrary load profile.

**Image:**



**Attributes:**

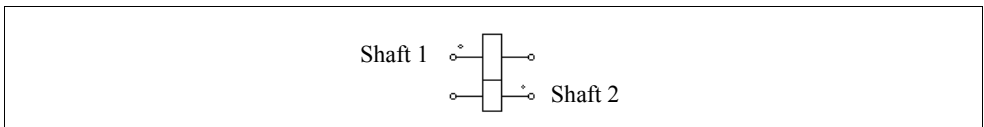
Parameters	Description
Speed Flag	Flag for speed dependency (Flag = 0: The load is frictional and is always against the rotational direction; Flag = 1: The load is independent of the rotational direction.)
Moment of Inertia	Moment of inertia of the load, in kg*m <sup>2</sup>

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.

### 2.11.2 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.

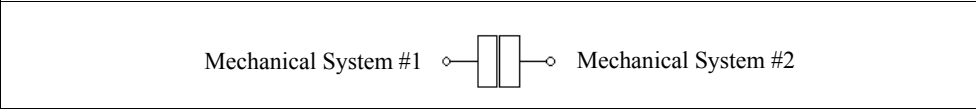
If the numbers of teeth of the first gear and the second gear are  $n_1$  and  $n_2$ , respectively, the gear ratio  $a$  is defined as:  $a = n_1 / n_2$ . Let the radius, torque, and speed of these two gears be:  $r_1, r_2, T_1, T_2, \omega_1$ , and  $\omega_2$ , we have:  $T_1 / T_2 = r_1 / r_2 = \omega_2 / \omega_1 = a$ .

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 2.8.1.

**2.11.3 Mechanical Coupling Block**

The mechanical coupler block is used to couple two mechanical systems.

**Image:**

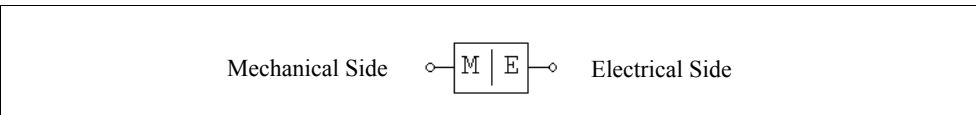


This block is used in situations where both mechanical systems have a device in the master mode, and they must be connected together to form one system. Based on the connection convention in PSIM, a mechanical system can have only one master device. In this case, the mechanical coupling block can be inserted in between, and the mechanical system on each side of the coupling block can have its own device in the master mode.

**2.11.4 Mechanical-Electrical Interface Block**

This block allows users to access the internal equivalent circuit of the mechanical system of a machine.

**Image:**





### Attribute:

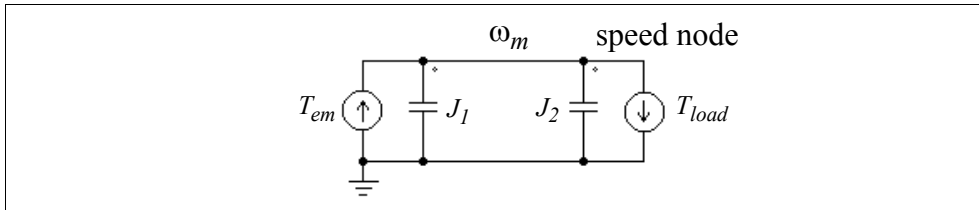
Parameter	Description
Master/Slave Flag	Flag for the master/slave mode (1: master, 0: slave)

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  $\omega_m$  is the shaft mechanical speed. In PSIM, this equation is modelled by an equivalent circuit as shown below.



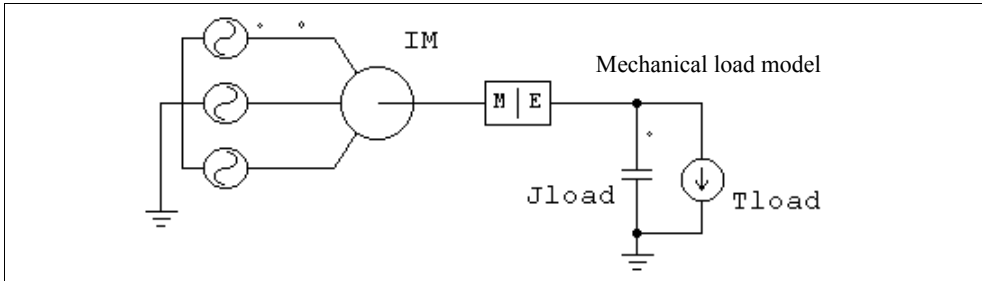
In this circuit, the two current sources have the values of  $T_{em}$  and  $T_{load}$ , and the capacitors have the values of  $J_1$  and  $J_2$ . The node-to-ground voltage (speed node voltage) represents the mechanical speed  $\omega_m$ . This is analogous to  $C \cdot dV/dt = i$  for a capacitor 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.

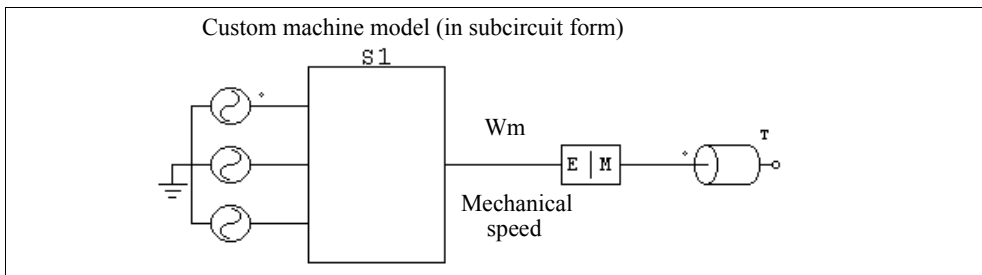
### Example: An induction machine with a custom mechanical load model

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

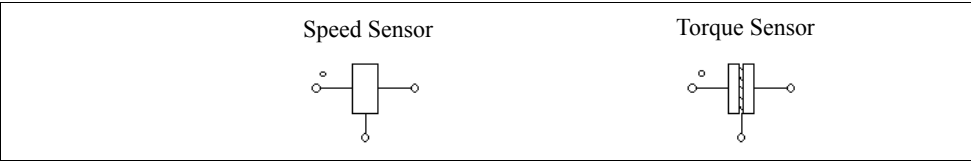
Similarly, one can build a custom machine model and connect it to the mechanical load in PSIM. The figure below shows such a circuit. The custom machine model must use the capacitor analogy to model the mechanical equation. The node representing the mechanical speed is then made available and is connected to the electrical side of the mechanical-electrical interface block.



### 2.11.5 Speed/Torque Sensors

A speed sensor or torque sensor is used to measure the mechanical speed or torque.

**Images:**



**Attribute:**

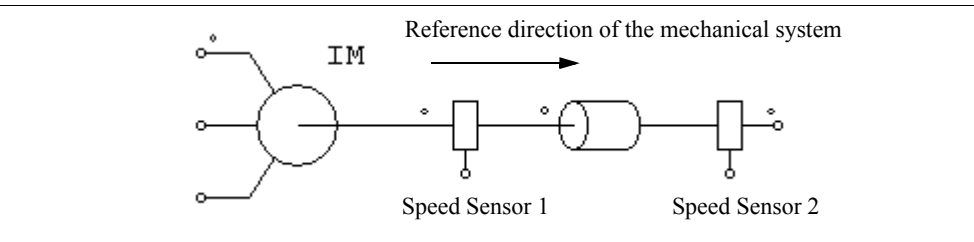
Parameter	Description
Gain	Gain of the sensor

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.

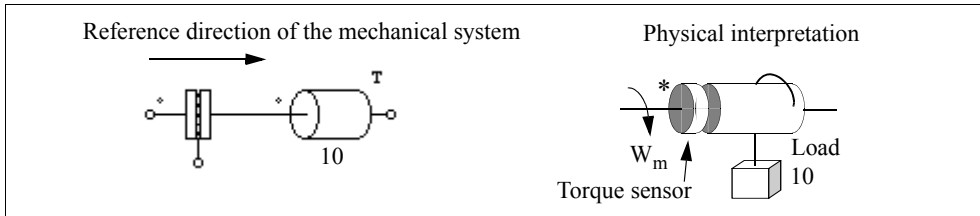
For the speed sensor, if the sensor is along the reference direction of the mechanical system (refer to Section 2.8.1 for more details on the definition and use of the reference direction), a positive mechanical speed would give a positive sensor output. Otherwise, if the sensor is opposite to the reference 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.

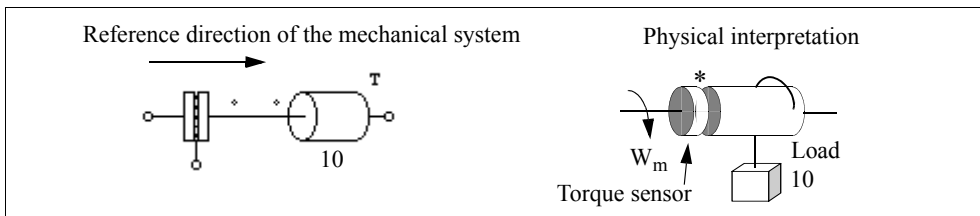
The figure on the left shows a torque sensor connected with a 10-N\*m mechanical load, and the reference direction of the mechanical system is from left to right. Based on the reference direction, if we use the right-hand method, by pointing the thumb in the reference direction and rotating the right 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.



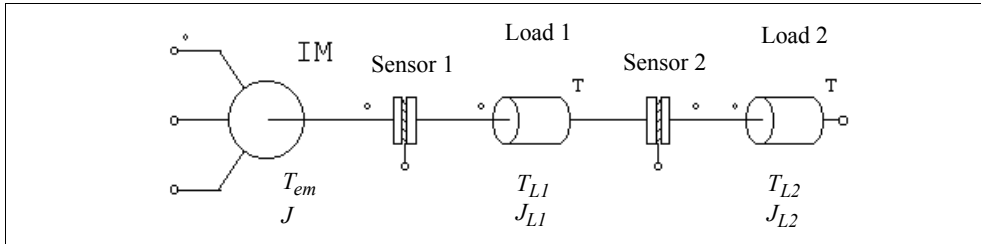
In this case, the direction of the positive speed and torque is in the clockwise direction. The dotted side of the sensor is on the left, and the load is in such a way that it tries to slow down the shaft (the load torque is in the counter-clockwise direction).

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.



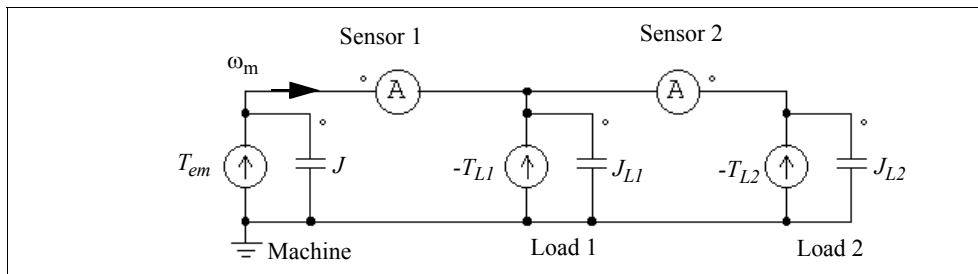
To understand how the torque sensor is modeled in the equivalent circuit of the mechanical system, we use the following system as an example.



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  $\omega_m$ . 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  $\omega_m$ .

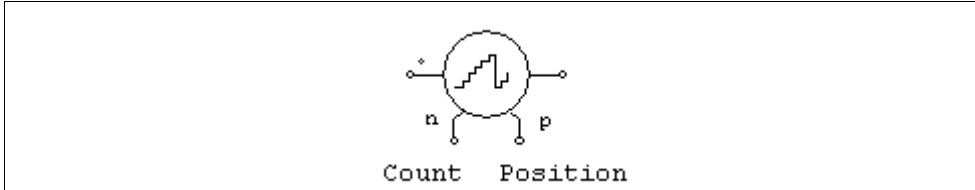
## 2.11.6 Position Sensors

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.

### 2.11.6.1 Absolute Encoder

An absolute encoder is a position sensor that provides the shaft position within a  $360^\circ$  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

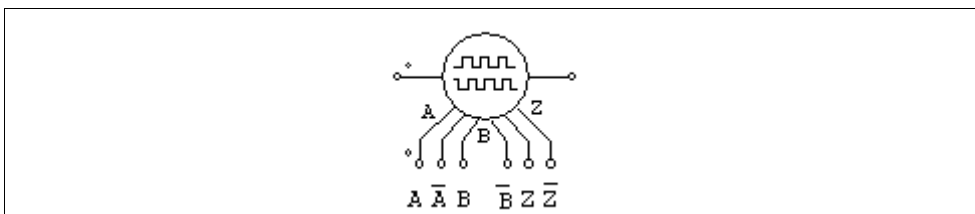
The encoder output resolution is determined by the number of bits N. The encoder has two outputs: one is the number of counts (the range is from 0 to  $2^N-1$ ), and the other is the position, in mechanical deg. (the range is from 0 to  $360^\circ$ ).

An example of a PMSM drive system using the absolute encoder is given in the sample file "Absolute Encoder PMSM Drive.sch".

### 2.11.6.2 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

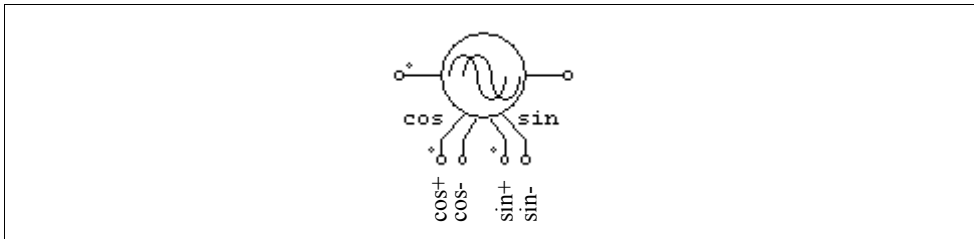
The two quadrature outputs are A and  $\bar{A}$  ( $\bar{A}$  is the inverse of A), and B and  $\bar{B}$ . They are offset by  $90^\circ$ . In addition, the encoder provides separate index signal output Z and  $\bar{Z}$  that provide one count per revolution.

An example of an induction motor drive system using the incremental encoder is given in the sample file "Incremental Encoder INDM Drive.sch".

### 2.11.6.3 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  $90^\circ$  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

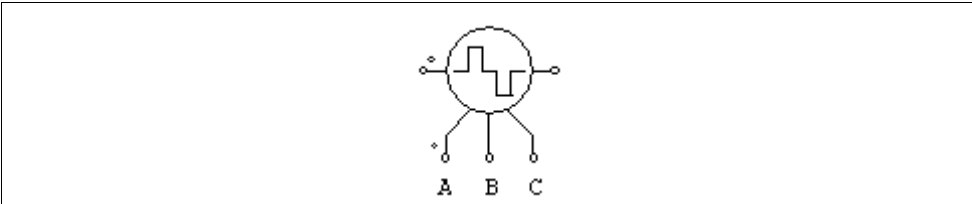
The resolver has four outputs:  $\cos^+$ ,  $\cos^-$  (the inverse of  $\cos^+$ ),  $\sin^+$ , and  $\sin^-$  (the inverse of  $\sin^+$ ). The peak amplitude of all the outputs is 1.

An example of a PMSM drive system using the resolver is given in the sample file "Resolver PMSM Drive.sch".

### 2.11.6.4 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.

An example of a BDCM motor drive system using the hall-effect sensor is given in the sample file "Hall-Effect Sensor BDCM\_Drive.sch".



## Control Circuit Components

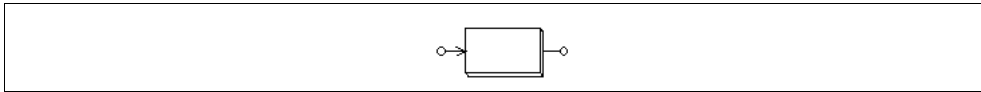
### 3.1 Transfer Function Blocks

A transfer function block is expressed in polynomial form as:

$$G(s) = k \cdot \frac{B_n \cdot s^n + \dots + B_2 \cdot s^2 + B_1 \cdot s + B_0}{A_n \cdot s^n + \dots + A_2 \cdot s^2 + A_1 \cdot s + A_0}$$

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 (I)* in the PSIM library).

**Images:**



**Attributes:**

Parameters	Description
Order n	Order $n$ of the transfer function
Gain	Gain $k$ of the transfer function
Coeff. $B_n \dots B_0$	Coefficients of the numerator (from $B_n$ to $B_0$ )
Coeff. $A_n \dots A_0$	Coefficients of the denominator (from $A_n$ to $A_0$ )
Initial Values $x_n \dots x_1$	Initial values of the state variables $x_n$ to $x_1$ (for the element <i>s-domain Transfer Function (I)</i> only)

Let  $Y(s) = G(s) \cdot U(s)$  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:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -A_0/A_n \\ 1 & 0 & 0 & \dots & 0 & -A_1/A_n \\ 0 & 1 & 0 & \dots & 0 & -A_2/A_n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & -A_{n-1}/A_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} + \frac{k}{A_n} \cdot \begin{bmatrix} B_0 - A_0 \cdot B_n/A_n \\ B_1 - A_1 \cdot B_n/A_n \\ B_2 - A_2 \cdot B_n/A_n \\ \dots \\ B_{n-1} - A_{n-1} \cdot B_n/A_n \end{bmatrix} \cdot u$$

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 (1)*.

### Example:

The following is a second-order transfer function:

$$G(s) = 1.5 \cdot \frac{400 \cdot e^3}{s^2 + 1200 \cdot s + 400 \cdot e^3}$$

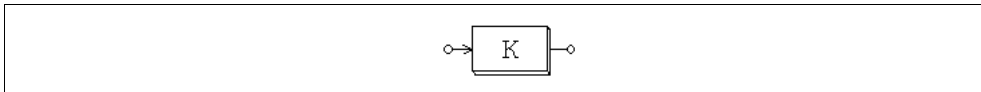
In PSIM, the specification will be:

Order n	2
Gain	1.5
Coeff. $B_n \dots B_0$	0. 0. 400.e3
Coeff. $A_n \dots A_0$	1. 1200. 400.e3

### 3.1.1 Proportional Controller

The output of a proportional (P) controller is equal to the input multiplied by a gain.

Image:



**Attribute:**

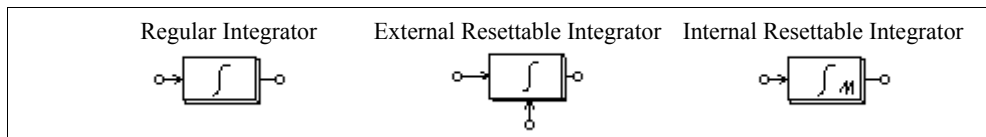
Parameter	Description
Gain	Gain $k$ of the transfer function

**3.1.2 Integrator**

The transfer function of an integrator is:

$$G(s) = \frac{1}{sT}$$

There are three types of integrators. regular integrator, external resettable integrator, and internal resettable integrator.

**Images:****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

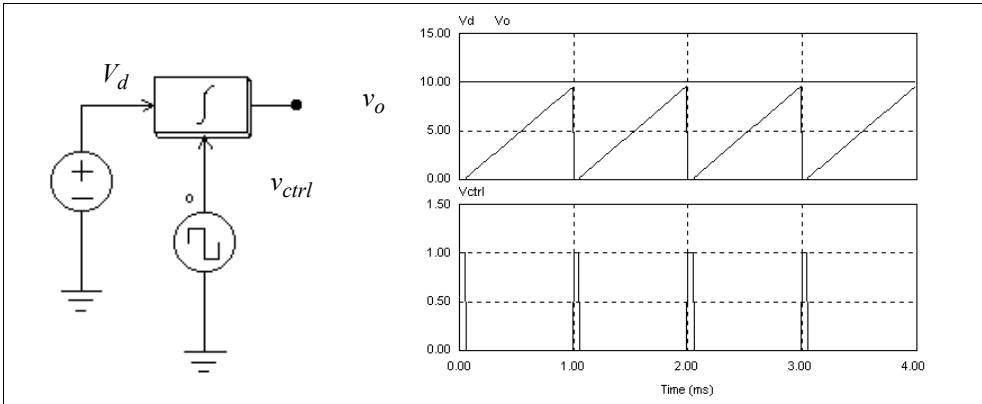
The output of the external resettable integrator can be reset by an external control signal (at the bottom of the block). For the edge reset (reset flag = 0), the integrator output is reset to zero at the rising edge of the control signal. For the level reset (reset flag = 1), the integrator output is 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.

To avoid over saturation, a limiter should be placed at the integrator output.

**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.



### 3.1.3 Differentiator

The transfer function of a differentiator is:

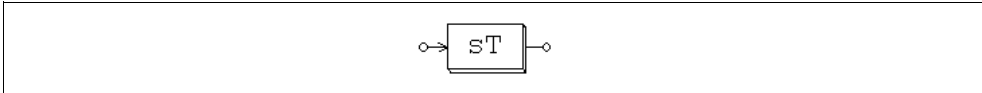
$$G(s) = sT$$

A differentiator is calculated as follows:

$$v_o(t) = T \cdot \frac{v_{in}(t) - v_{in}(t - \Delta t)}{\Delta t}$$

where  $\Delta t$  is the simulation time step,  $v_{in}(t)$  and  $v_{in}(t - \Delta t)$  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.

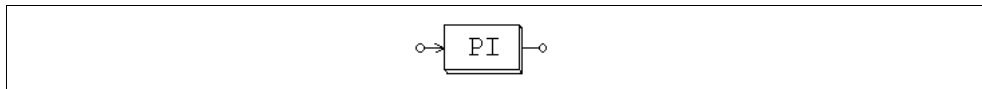
Since sudden changes of the input will generate spikes at the output, it is recommended that a low-pass filter be placed at the input of the differentiator.

### 3.1.4 Proportional-Integral Controller

The transfer function of a proportional-integral (PI) controller is defined as:

$$G(s) = k \cdot \frac{1 + sT}{sT}$$

**Image:**



### Attributes:

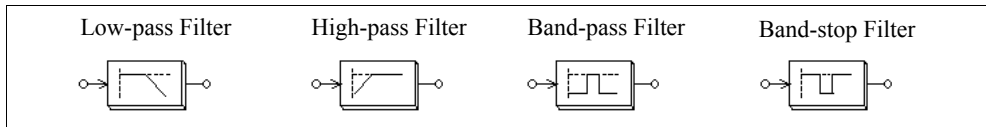
Parameters	Description
Gain	Gain $k$ of the PI controller
Time Constant	Time constant $T$ of the PI controller

To avoid over saturation, a limiter should be placed at the PI output.

### 3.1.5 Built-in Filter Blocks

Four second-order filters are provided as built-in modules in PSIM.

### Images:



### Attributes:

Parameters	Description
Gain	Gain $k$
Damping Ratio	Damping ratio $\xi$
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

The transfer function of these filters are listed below.

For a second-order low-pass filter:

$$G(s) = k \cdot \frac{\omega_c^2}{s^2 + 2\xi\omega_c s + \omega_c^2}$$

For a second-order high-pass filter:

$$G(s) = k \cdot \frac{s^2}{s^2 + 2\xi\omega_c s + \omega_c^2}$$

For a second-order band-pass filter:

$$G(s) = k \cdot \frac{B \cdot s}{s^2 + B \cdot s + \omega_o^2}$$

For a second-order band-stop filter:

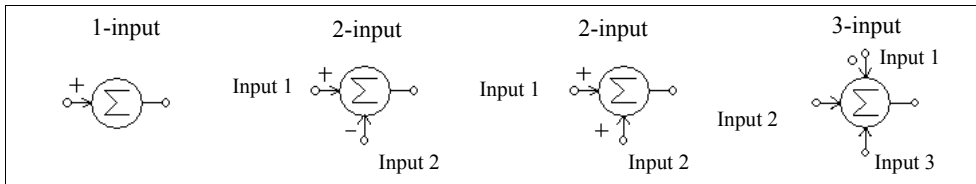
$$G(s) = k \cdot \frac{s^2 + \omega_o^2}{s^2 + B \cdot s + \omega_o^2}$$

## 3.2 Computational Function Blocks

### 3.2.1 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>i</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:

$$V_o = k_1 V_1 + k_2 V_2 + \dots + k_n V_n$$

If the input is a vector, the output of a two-input summer will also be a vector, which is defined as:

$$V_1 = [a_1 \ a_2 \ \dots \ a_n]$$

$$V_2 = [b_1 \ b_2 \ \dots \ b_n]$$

$$V_o = V_1 + V_2 = [a_1+b_1 \ a_2+b_2 \ \dots \ a_n+b_n]$$

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,  $V_o = a_1 + a_2 + \dots a_n$ .

### 3.2.2 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:

$$V_1 = [a_1 \ a_2 \ \dots \ a_n]$$

$$V_2 = [b_1 \ b_2 \ \dots \ b_n]$$

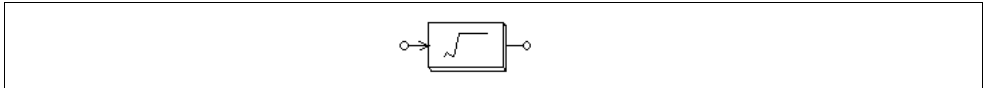
The output, which is a scalar, will be:

$$V_o = V_1 * V_2^T = a_1*b_1 + a_2*b_2 + a_n*b_n$$

### 3.2.3 Square-Root Block

A square-root function block calculates the square root of the input.

**Image:**





### 3.2.4 Exponential/Power/Logarithmic Function Blocks

The images and attributes of these function blocks are shown below.

**Images:**



**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.

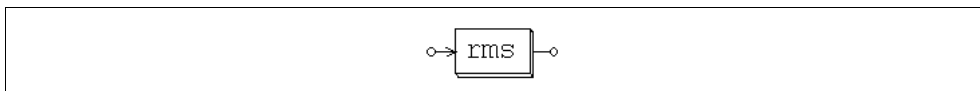
### 3.2.5 Root-Mean-Square Block

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

**3.2.6 Absolute and Sign Function Blocks**

An absolute value function block gives the absolute value of the input. A sign function block (SIGN) gives the sign of the input, i.e., the output is 1 if the input is positive, and the output is -1 if the input is negative.

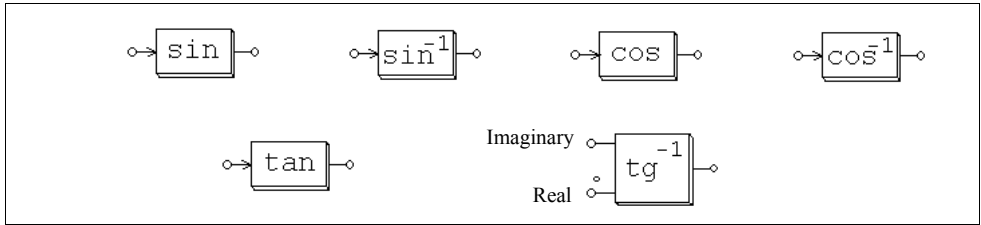
**Images:**



**3.2.7 Trigonometric Functions**

Six trigonometric functions are provided: sine (sin), arc sine ( $\sin^{-1}$ ), cosine (cos), arc cosine ( $\cos^{-1}$ ), tangent (tan), and arc tangent ( $\tan^{-1}$ ). The output is equal to the corresponding trigonometric function of the input. For the sin, cos, and tan blocks, the input is in deg., and for the  $\sin^{-1}$ ,  $\cos^{-1}$ , and  $\tan^{-1}$  blocks, the output is in deg.

**Images:**



The dotted note of the arc tangent block is for the real input and the other node is for the imaginary input. The output is the arc tangent of the ratio between the imaginary and the

real input, i.e.  $\theta = \tan^{-1}\left(\frac{V_{imaginary}}{V_{real}}\right)$ .

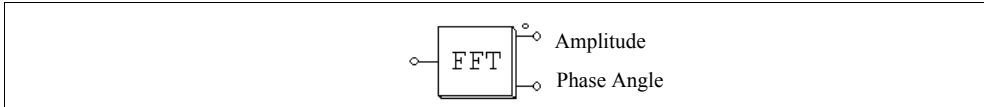
### 3.2.8 Fast Fourier Transform Block

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 amplitude (peak) and the phase angle of the input fundamental component. The output voltage (in complex form) is defined as:

$$v_o = \frac{2}{N} \cdot \sum_{n=0}^{\frac{N}{2}-1} \left[ v_{in}(n) - v_{in}\left(n + \frac{N}{2}\right) \right] \cdot e^{-j\frac{2\pi n}{N}}$$

**Image:**



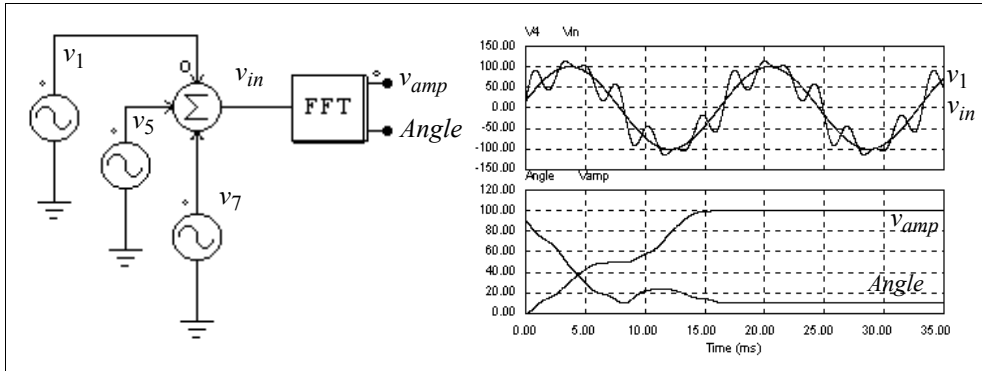
**Attributes:**

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  $V_m \sin(\omega t)$  will give a phase angle output of 0.

**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°.

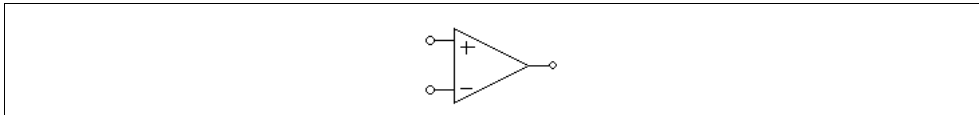


### 3.3 Other Function Blocks

#### 3.3.1 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:

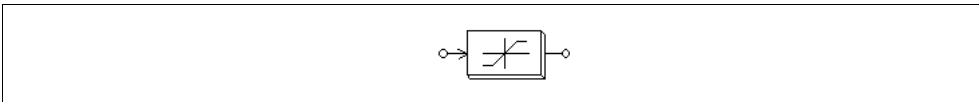


Note that the comparator image is similar to that of the op. amp. For the comparator, the noninverting input is at the upper left and the inverting input is at the lower left. For the op. amp., however, it is the opposite.

#### 3.3.2 Limiter

The output of a 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.

Image:



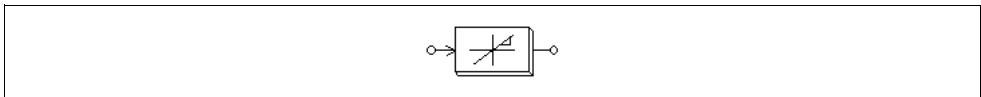
**Attributes:**

Parameters	Description
Lower Limit	Lower limit of the limiter
Upper Limit	Upper limit of the limiter

**3.3.3 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:**



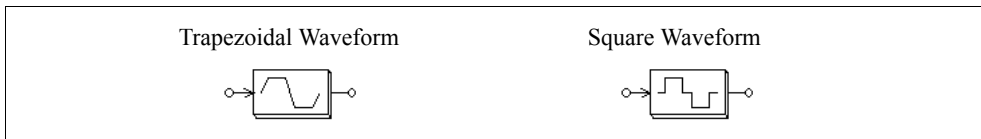
**Attribute:**

Parameter	Description
dv/dt Limit	Limit of the rate of change (dv/dt) of the input

**3.3.4 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:**



For the trapezoidal waveform block:

**Attributes:**

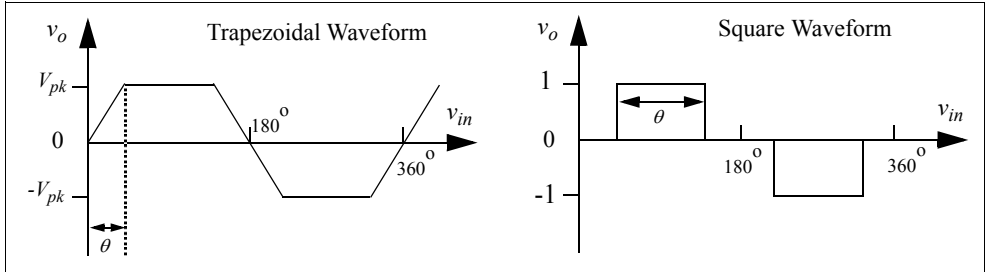
Parameters	Description
Rising Angle theta	Rising angle $\theta$ , in deg.
Peak Value	Peak value $V_{pk}$ of the waveform

For the square waveform block:

**Attribute:**

Parameter	Description
Pulse Width (deg.)	Pulse width $\theta$ in half cycle, in deg.

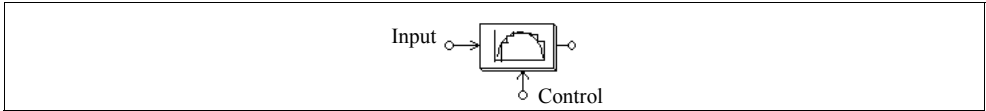
The waveforms of these two blocks are shown below. Note that the input  $v_{in}$  is in deg., and can be in the range of  $-360^\circ$  to  $360^\circ$ . Both waveforms are half-wave and quarter-wave symmetrical.



### 3.3.5 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:**

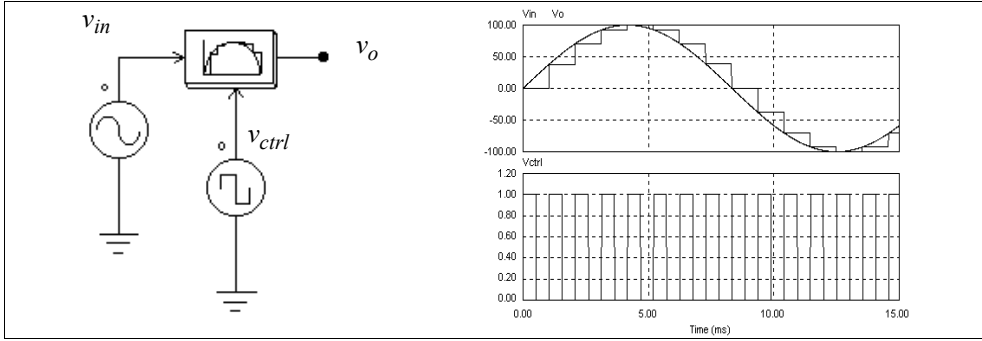


The difference between this block and the zero-order hold block (ZOH) in Digital Control Module is that this block is treated as a continuous element and sampling moments can be controlled externally; whereas the zero-order hold block is a discrete element and the sampling moments are fixed and of equal distance.

For a discrete system, the zero-order hold block should be used.

**Example:**

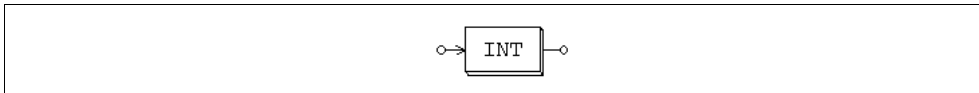
In this example, a sinusoidal input is sampled. The control signal is a square wave voltage source with an amplitude of 1.



### 3.3.6 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, the output  $V_{out} = 34$ .

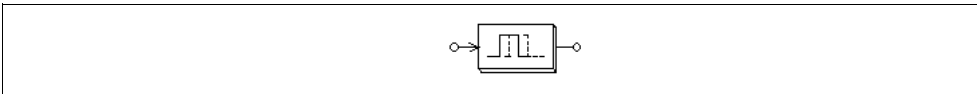
If  $V_{in} = 34.5678$ ;  $N = 1$ , truncation flag = 1, the output  $V_{out} = 34.5$ .

If  $V_{in} = 34.5678$ ;  $N = -1$ , truncation flag = 1, the output  $V_{out} = 30$ .

### 3.3.7 Time Delay Block

A time delay block delays the input waveform by a specified amount of time interval. It, for example, can be used to model the propagation delay of a logic element.

**Image:**



**Attribute:**

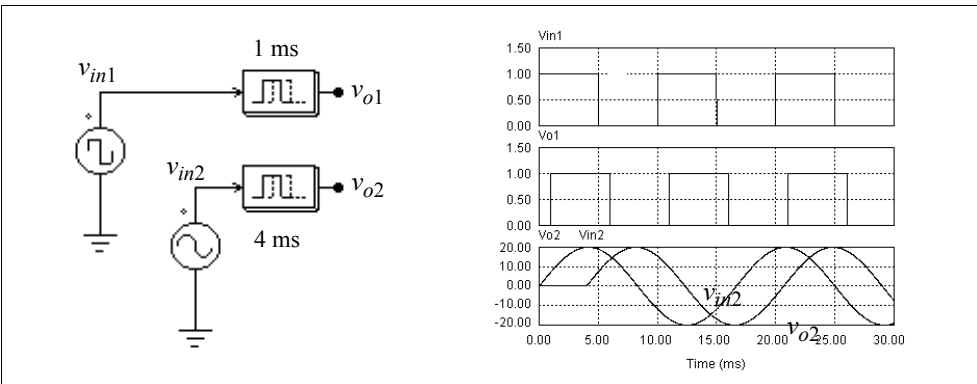
Parameter	Description
Time Delay	Time delay, in sec.

Note that the difference between this block and the unit delay block (UDELAY) in Digital Control Module is that this block is a continuous element and the delay time can be arbitrarily set; 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.

**Example:**

In this circuit, the first time delay block has a delay time of 1 ms, and the second block has a delay time of 4 ms. This example illustrates that the input of the time delay block can be either an analog or a digital signal.

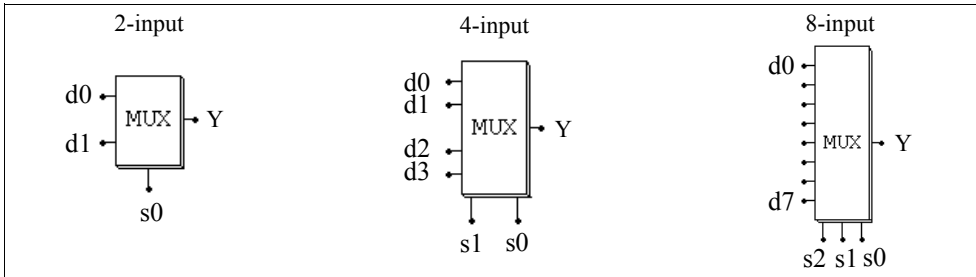




### 3.3.8 Multiplexer

The output of a multiplexer is equal to a selected input depending on the control signal. Three multiplexers are provided: multiplexers with 2 inputs; 4 inputs; and 8 inputs.

**Images:**



In the images, d0..d7 are the data inputs; and s0..s2 are the control signals. The truth tables of the multiplexers are as follows.

2-Input MUX	
s0	Y
0	d0
1	d1

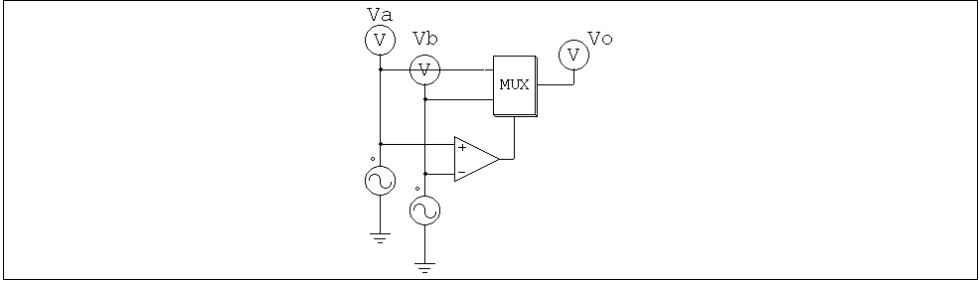
4-Input MUX		
s1	s0	Y
0	0	d0
0	1	d1
1	0	d2
1	1	d3

8-Input MUX			
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

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  $V_o = V_b$ .



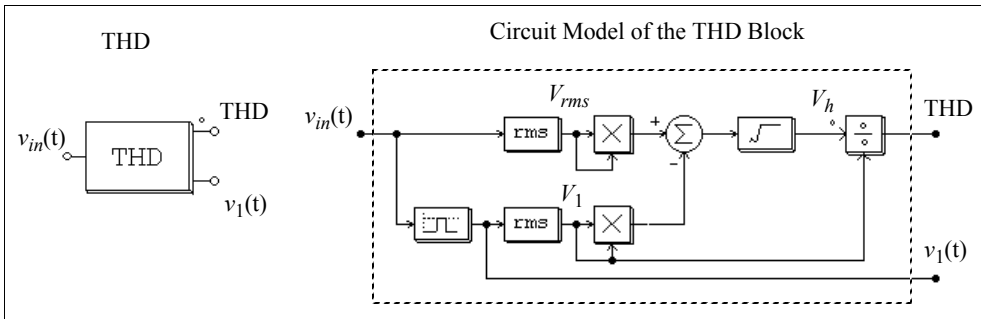
### 3.3.9 THD Block

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:**



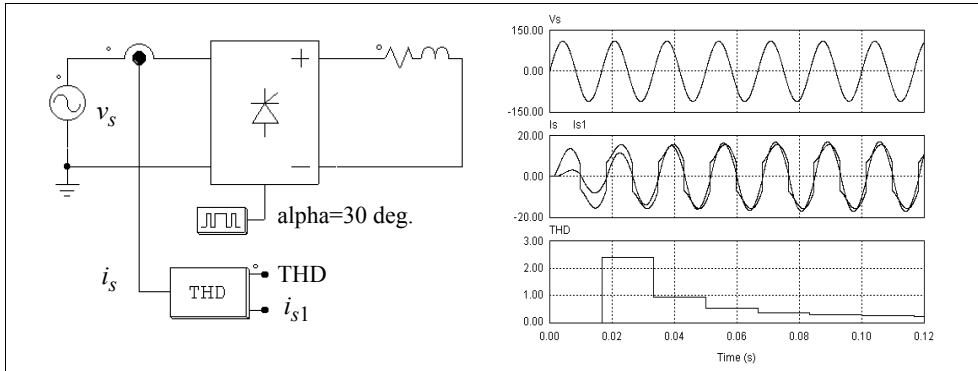
A second-order band-pass filter is used to extract the fundamental component. The center frequency and the passing band of the band-pass filter need to be specified.

**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  $30^\circ$ . 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.



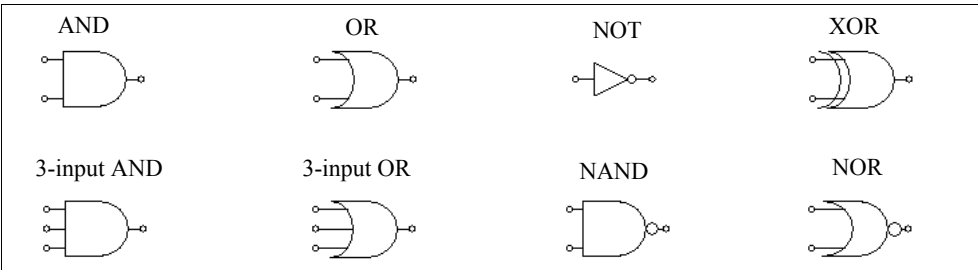
One of the THD block output is the input current fundamental component  $i_{s1}$ . By comparing the phase difference between the input voltage  $v_s$  and the current  $i_{s1}$ , one can calculate the input displacement power factor. This, together with the THD value, can be used to calculate the input power factor.

### 3.4 Logic Components

#### 3.4.1 Logic Gates

Basic logic gates are AND, OR, XORGATE (exclusive-OR), NOT, NAND, and NOR gates.

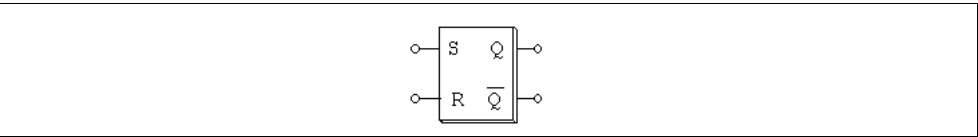
**Images:**



**3.4.2 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	Qn
0	0	no change	
0	↑	0	1
↑	0	1	0
↑	↑	not used	

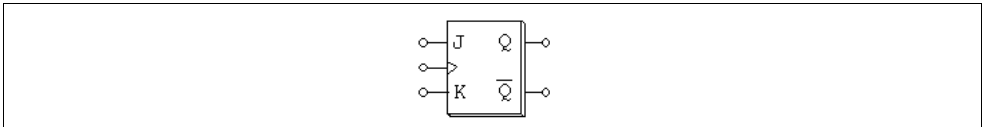
A level-triggered flip-flop, on the other hand, changes the states based on the input level. The truth table of a level-triggered set-reset flip-flop is:

S	R	Q	Qn
0	0	no change	
0	1	0	1
1	0	1	0
1	1	not used	

### 3.4.3 J-K Flip-Flop

A J-K flip-flop is positive edge-triggered.

Image:



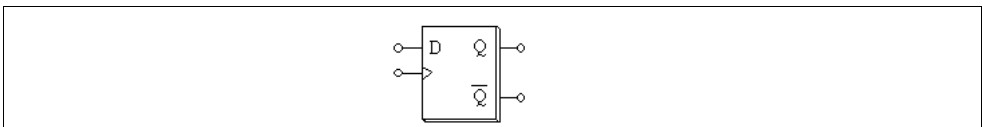
The truth table is:

J	K	D	Q	Qn
0	0	↑	no change	
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	Toggle	

### 3.4.4 D Flip-Flop

A D flip-flop is positive edge-triggered.

Image:



The truth table is:

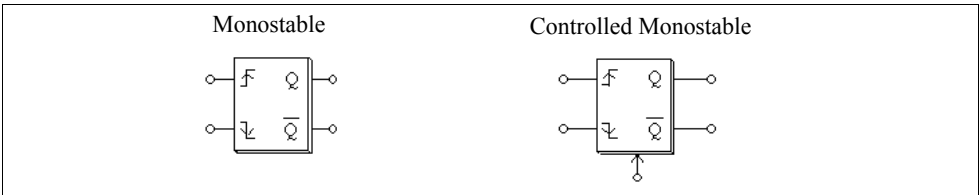
D	Clock	Q	Qn
0	↑	0	1
1	↑	1	0

### 3.4.5 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:**

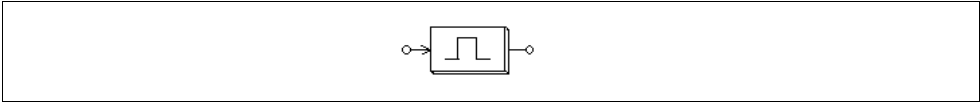
Parameter	Description
Pulse Width	On-time pulse width, in sec.

The input node at the bottom of the controlled monostable block is for the pulse width input.

### 3.4.6 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.

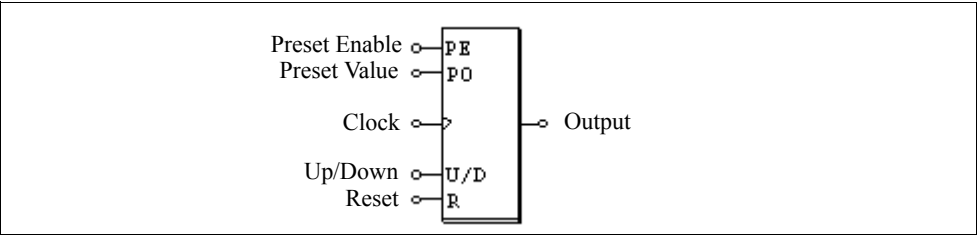
Image:



3.4.7 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:

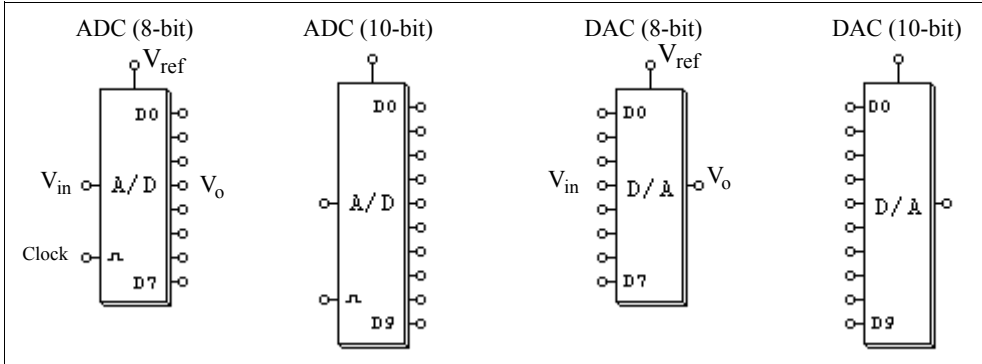
Up/Down	Preset Enable	Reset	Clock	Action
x	0	0	x	No count
1	0	0	↑	Count up
0	0	0	↑	Count down
x	1	0	x	Preset
x	x	1	x	Reset

x: Do not care

### 3.4.8 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:

$$V_o = \frac{2^N}{V_{ref}} \cdot V_{in}$$

For example, if  $V_{ref} = 5$  V,  $V_{in} = 3.2$  V,  $N = 8$  bits, then

$$V_o = 256/5 \cdot 3.2 = 163.84 = 10100011 \text{ (binary)}$$

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

$$V_o = 163/256 \cdot 5 = 3.1836$$



## 3.5 Digital Control Module

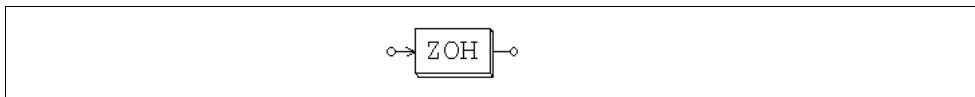
The Digital Control Module is an add-on module to the basic PSIM program. It provides discrete elements, such as zero-order hold, z-domain transfer function blocks, digital filters, etc., for digital control system simulation.

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.

### 3.5.1 Zero-Order Hold

A zero-order hold samples the input at the point of sampling. The output remains unchanged between two sampling points.

**Image:**



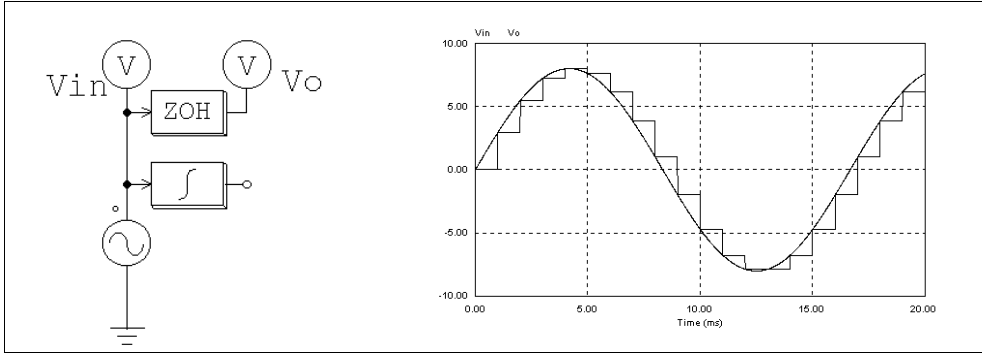
**Attribute:**

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.

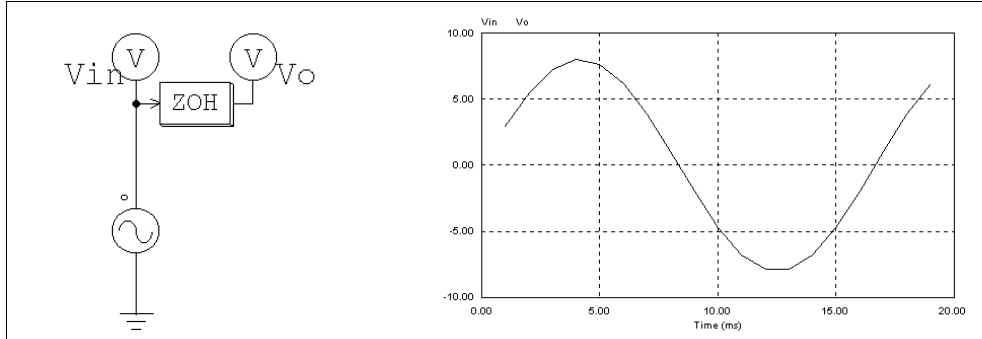
**Example:**

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.



### 3.5.2 z-Domain Transfer Function Block

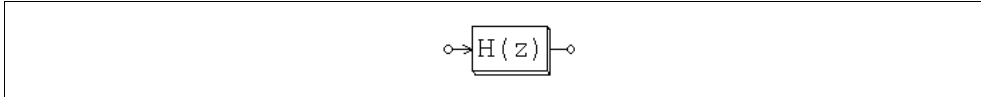
A z-domain transfer function block is expressed in polynomial form as:

$$H(z) = \frac{b_0 \cdot z^N + b_1 \cdot z^{N-1} + \dots + b_{N-1} \cdot z + b_N}{a_0 \cdot z^N + a_1 \cdot z^{N-1} + \dots + a_{N-1} \cdot z + a_N}$$

If  $a_0 = 1$ , the expression  $Y(z) = H(z) * U(z)$  can be expressed in difference equation as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \dots + b_N \cdot u(n-N) - [a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + \dots + a_N \cdot y(n-N)]$$

**Image:**



**Attributes:**

Parameters	Description
Order $N$	Order $N$ of the transfer function
Coeff. $b_0 \dots b_N$	Coefficients of the numerator (from $b_0$ to $b_N$ )
Coeff. $a_0 \dots a_N$	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:

$$H(z) = \frac{400 \cdot e^3}{z^2 + 1200 \cdot z + 400 \cdot e^3}$$

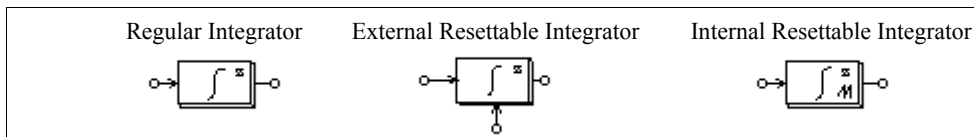
Assuming a sampling frequency of 3 kHz, the specification will be:

Order $N$	2
Coeff. $b_0 \dots b_N$	0. 0. 400.e3
Coeff. $a_0 \dots a_N$	1. 1200. 400.e3
Sampling Frequency	3000.

### 3.5.2.1 Integrator

There are three types of integrators: regular integrator, external resettable integrator, and internal resettable integrator.

**Images:**



**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 <i>external resettable integrator</i> only)
Lower Output Limit	Lower limit of the output (for <i>internal resettable integrator</i> only)
Upper Output Limit	Upper limit of the output (for <i>internal resettable integrator</i> 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,  $T$  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:

$$H(z) = \frac{T}{2} \cdot \frac{z+1}{z-1}$$

$$y(n) = y(n-1) + \frac{T}{2} \cdot (u(n) + u(n-1))$$

With backward Euler:

$$H(z) = T \cdot \frac{z}{z-1}$$

$$y(n) = y(n-1) + T \cdot u(n)$$

With forward Euler:

$$H(z) = T \cdot \frac{1}{z-1}$$

$$y(n) = y(n-1) + T \cdot u(n-1)$$

### 3.5.2.2 Differentiator

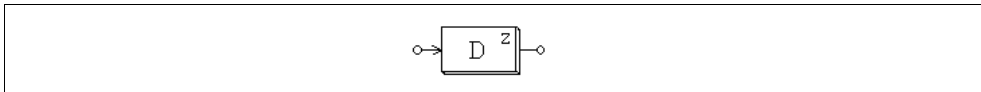
The transfer function of a discrete differentiator is:

$$H(z) = \frac{1}{T} \cdot \frac{z-1}{z}$$

where T is the sampling period. The input-output relationship can be expressed in difference equation as:

$$y(n) = \frac{1}{T} \cdot (u(n) - u(n-1))$$

**Image:**



**Attribute:**

Parameter	Description
Sampling Frequency	Sampling frequency, in Hz

**3.5.2.3 Digital Filters**

Two types of digital filters are provided: general digital filter and finite impulse response (FIR) filter. For both types, the filter coefficients can either be entered directly through the element property window, or be specified through a text file.

**Images:****Attributes:**

For elements that read 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, in Hz

For elements that read the coefficients from a text file:

Parameters	Description
File for Coefficients	Name of the file storing the filter coefficients
Sampling Frequency	Sampling frequency, in Hz

The transfer function of the general digital filter is expressed in polynomial form as:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + \dots + b_{N-1} \cdot z^{-(N-1)} + b_N \cdot z^{-N}}{a_0 + a_1 \cdot z^{-1} + \dots + a_{N-1} \cdot z^{-(N-1)} + a_N \cdot z^{-N}}$$

If  $a_0 = 1$ , the output  $y$  and input  $u$  can be expressed in difference equation form as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \dots + b_N \cdot u(n-N) - [a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + \dots + a_N \cdot y(n-N)]$$

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 the FIR filter is expressed in polynomial form as:

$$H(z) = b_0 + b_1 \cdot z^{-1} + \dots + b_{N-1} \cdot z^{-(N-1)} + b_N \cdot z^{-N}$$

If  $a_0 = 1$ , the output  $y$  and input  $u$  can be expressed in difference equation form as:

$$y(n) = b_0 \cdot u(n) + b_1 \cdot u(n-1) + \dots + b_N \cdot u(n-N)$$

The coefficient file for block FILTER\_D1 and FILTER\_FIR1 has the following format:

For Filter\_FIR1:

N  
 $b_0$   
 $b_1$   
 ... ..  
 $b_N$

For Filter\_D1, 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  $f_c = 1$  kHz, assuming the sampling frequency  $f_s = 10$  kHz, using MATLAB, we have:

$$\text{Nyquist frequency } f_n = f_s / 2 = 5 \text{ kHz}$$

$$\text{Normalized cut-off frequency } f_c^* = f_c / f_n = 1/5 = 0.2$$

$$[B,A] = \text{butter}(2, f_c^*)$$

which will give:

$$B = [0.0201 \quad 0.0402 \quad 0.0201] = [b_0 \quad b_1 \quad b_2]$$

$$A = [1 \quad -1.561 \quad 0.6414] = [a_0 \quad a_1 \quad a_2]$$

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. $b_0 \dots b_N$	0.0201    0.0402    0.0201
Coeff. $a_0 \dots a_N$	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:

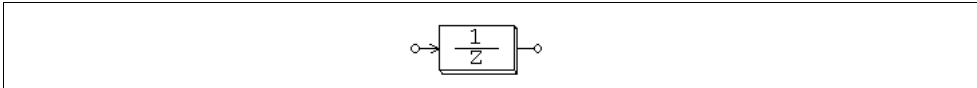


2  
0.0201, 1  
0.0402, -1.561  
0.0201, 0.6414

### 3.5.3 Unit Delay

A unit delay block provides one sampling period delay to the input.

**Image:**



**Attribute:**

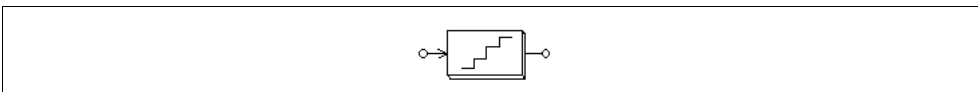
Parameter	Description
Sampling Frequency	Sampling frequency, in Hz

The difference between a unit delay block and a time delay block (TDELAY) is that the unit delay block is a discrete element and it delays the sampled points by one sampling period, whereas TDELAY is a continuous element and it delays the whole waveform by the delay time specified.

### 3.5.4 Quantization Block

A quantization block simulates the quantization error during an A/D conversion.

**Image:**



**Attributes:**

Parameters	Description
No. of Bits	Number of bits N
V <sub>in,min</sub>	Lower limit of the input value $V_{in,min}$
V <sub>in,max</sub>	Upper limit of the input value $V_{in,max}$

Vo_min	Lower limit of the output value $V_{o,min}$
Vo_max	Upper limit of the output value $V_{o,max}$
Sampling Frequency	Sampling frequency, in Hz

A quantization block performs two functions: scaling and quantization.

The input value  $V_{in}$ , sampled at the given sampling frequency, is first scaled based on the following:

$$V_{ox} = V_{in,min} + \frac{V_{in} - V_{in,min}}{V_{in,max} - V_{in,min}}(V_{o,max} - V_{o,min})$$

The number of bits determines the output resolution  $\Delta V$  which is defined as:

$$\Delta V = \frac{V_{o,max} - V_{o,min}}{2^N - 1}$$

The output  $V_o$  will be equal to the truncated value of  $V_{ox}$  based on the resolution  $\Delta V$ .

#### Example:

If  $N = 4$ ,  $V_{in,min} = 0$ ,  $V_{in,max} = 10$ ,  $V_{o,min} = -5$ ,  $V_{o,max} = 5$ , and  $V_{in} = 3.2$ , then:

$$V_{ox} = -5 + (3.2 - 0) * (5 - (-5)) / (10 - 0) = -1.8$$

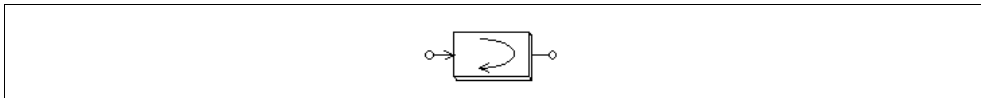
$$\Delta V = (5 - (-5)) / (2^4 - 1) = 0.66667$$

The value -1.8 is between -2.33332 and -1.66665. Therefore, the lower value is selected, that is,  $V_o = -1.66665$ .

### 3.5.5 Circular Buffer

A circular buffer is a memory location that can store an array of data.

Image:



**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 output of the circular buffer is a vector. To access to each memory location, use the memory read block MEMREAD.

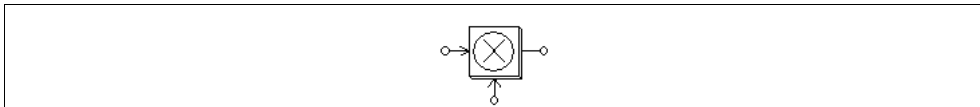
**Example:**

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			
		1	2	3	4
0	0.11	0.11	0	0	0
0.1	0.22	0.11	0.22	0	0
0.2	0.33	0.11	0.22	0.33	0
0.3	0.44	0.11	0.22	0.33	0.44
0.4	0.55	0.55	0.22	0.33	0.44
... ..					

**3.5.6 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:

$$A = [a_m \ a_{m-1} \ a_{m-2} \ \dots \ a_1]$$

$$B = [b_n \ b_{n-1} \ b_{n-2} \ \dots \ b_1]$$

We have the convolution of A and B as:

$$\begin{aligned} C &= A \otimes B \\ &= [c_{m+n-1} \ c_{m+n-2} \ \dots \ c_1] \end{aligned}$$

where

$$c_i = \sum [a_{k+1} * b_{j-k}], \quad k=0, \dots, m+n-1; j=0, \dots, m+n-1; i=1, \dots, m+n-1$$

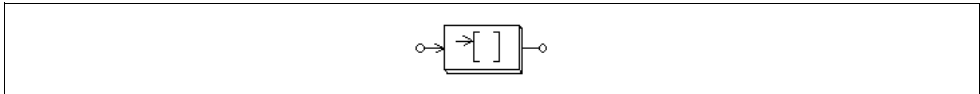
### Example:

If  $A = [1 \ 2 \ 3]$  and  $B = [4 \ 5]$ , we have  $m = 3$ ;  $n = 2$ ; and the convolution of  $A$  and  $B$  is:  $C = [4 \ 13 \ 22 \ 15]$ .

## 3.5.7 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.

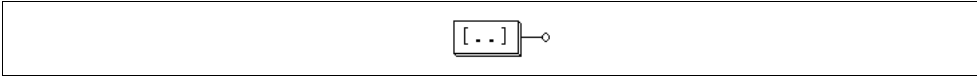
### Example:

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.

## 3.5.8 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 (I)* in the PSIM library).

Image:



Attributes:

Parameters	Description
Array Length	The length of the data array $N$ (for the element <i>Array</i> only)
Values	Values of the array (for the element <i>Array</i> only)
File for Coefficients	Name of the file storing the array (for the element <i>Array (I)</i> 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:

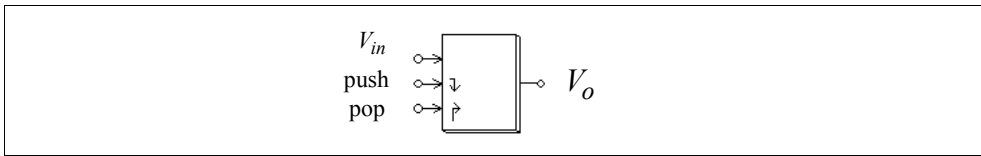
To define an array  $A = [2\ 4\ 6\ 8]$ , we will have: Array Length = 4; Values = 2 4 6 8. If the array is to be read from a file, the file will be:

4  
2.  
4.  
6.  
8.

### 3.5.9 Stack

A stack is a first-in-last-out register.

Image:



Attribute:

Parameter	Description
Stack Depth	The stack depth

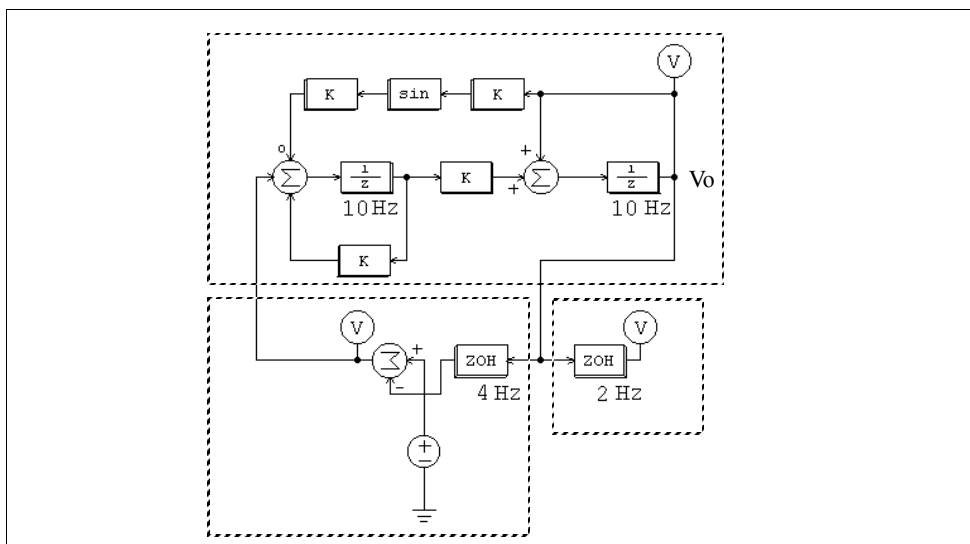
The rising edge triggers the push or pop action. When a pop action is performed and the stack is empty, the output remains unchanged. When a push action is performed and the stack is already full, the data at the bottom of the stack will be pushed out and will be lost.

### 3.5.10 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,  $V_o$ , 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.

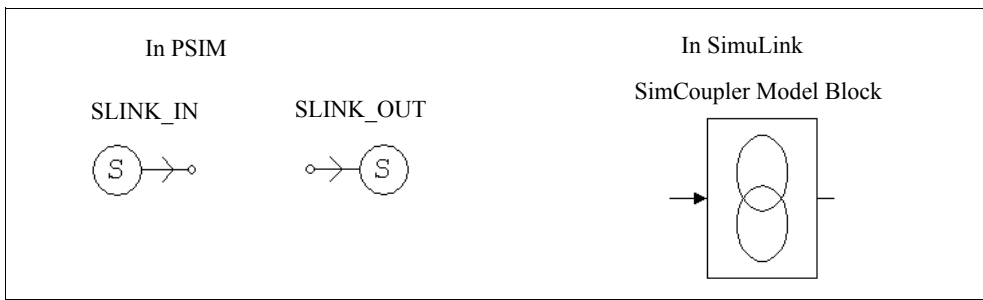


## 3.6 SimCoupler Module

The SimCoupler Module is an add-on module to the basic 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:**

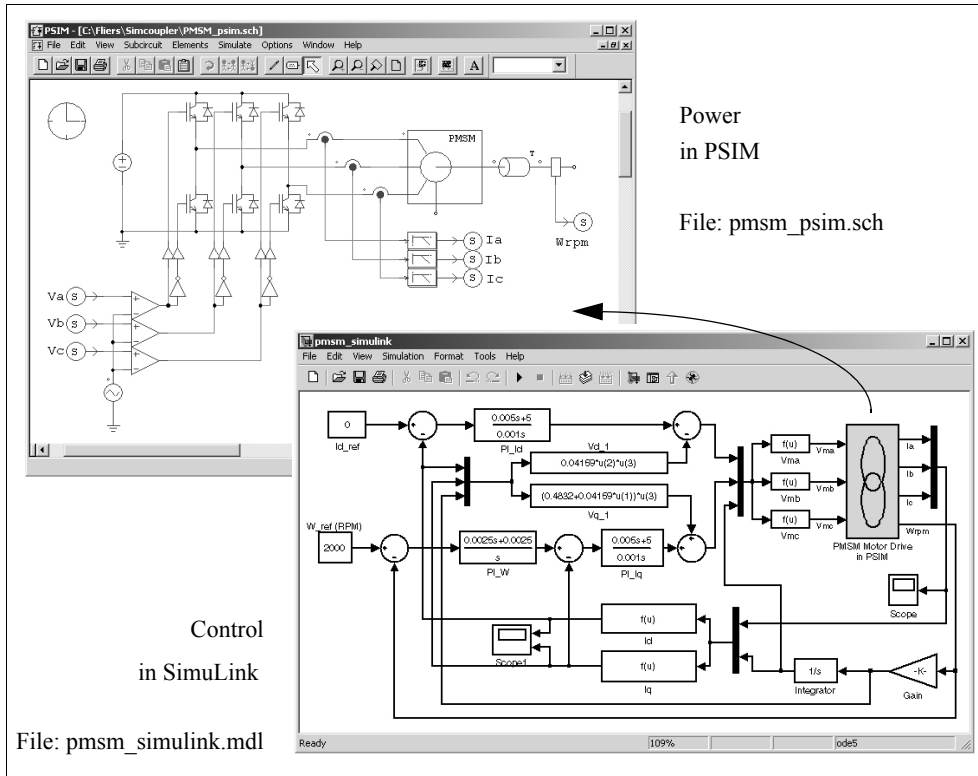


In PSIM, the SLINK\_IN nodes receive values from Simulink, and the SLINK\_OUT nodes send the values to Simulink. They are all control elements and can be used in the control circuit only. In Simulink, the SimCoupler model block is connected to the rest of the system through input/output ports.

### 3.6.1 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.





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 SLINK\_OUT nodes to the low-pass filters of Phase *A*, *B*, and *C* currents, and rename them as “Ia”, “Ib”, and “Ic”; and connect one SLINK\_OUT node to the speed sensor output and rename it as “Wrpm”.
- Connect three SLINK\_IN 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 SLINK\_IN nodes and SLINK\_OUT nodes 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 SLINK\_IN nodes, and “Ia”, “Ib”, “Ic”, and “Wrpm” for the SLINK\_OUT 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 14.

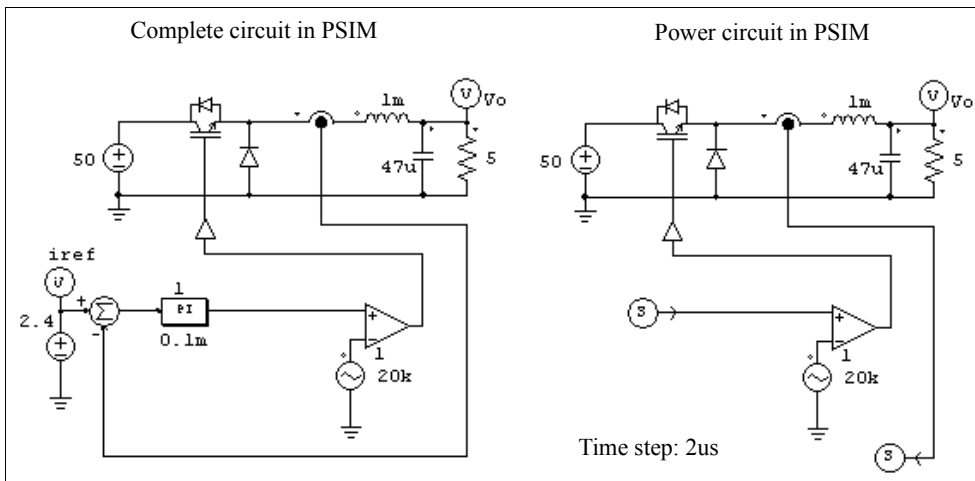
Please also note that when the SimCoupler model block is used in a feedback system in Simulink, the SimCoupler model block may be part of an algebraic loop (please refer to

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.

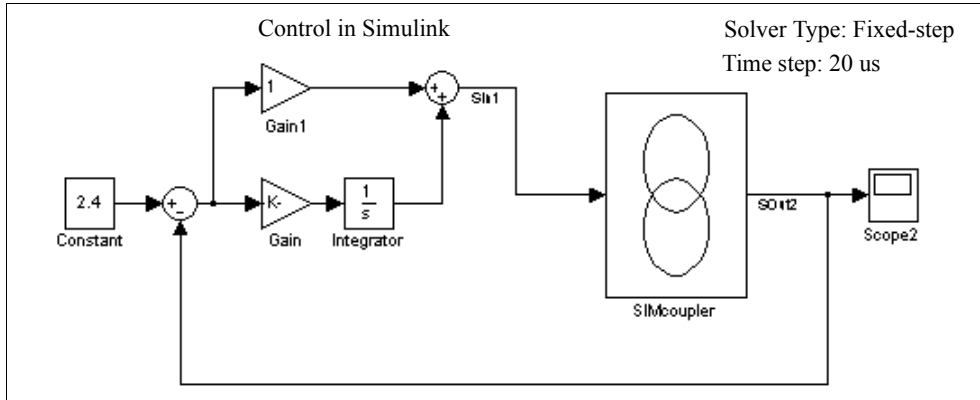
### 3.6.2 Solver Type and Time Step Selection in Simulink

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 simulation time step is 2  $\mu$ s.

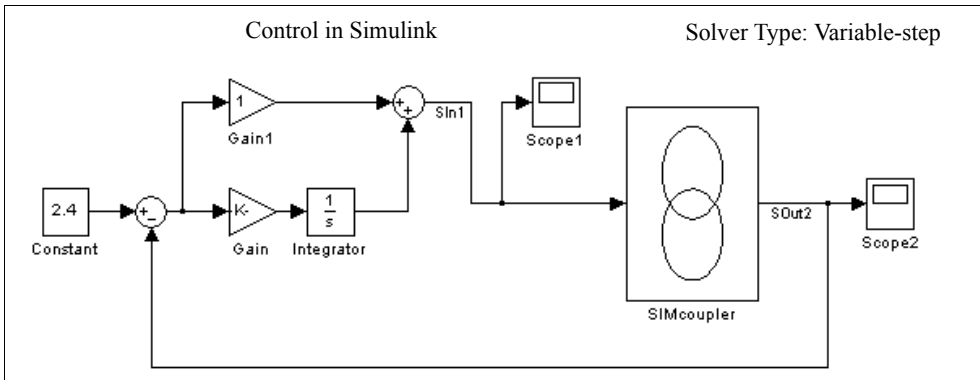


There are different ways of setting up Simulink to perform co-simulation. The recommend approach is to set the Solve Type to **Fixed-step** and define the **Fixed step size** to be the same or close to PSIM's time step. The figure below shows this option.

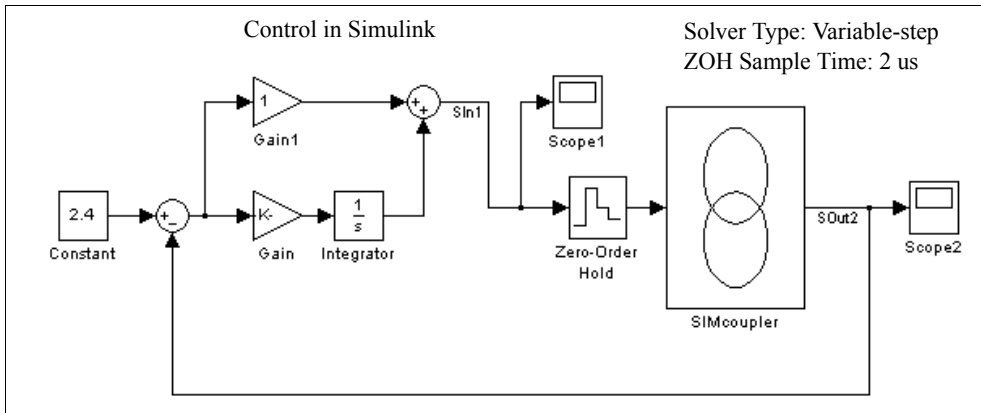


It is recommended that Simulink use the same time step as PSIM, although we have found that, even if the Simulink time step is slightly larger than PSIM time step, satisfactory results are obtained. In this case, for example, the time step is set to 20 us, 10 times larger than the PSIM time step.

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.



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

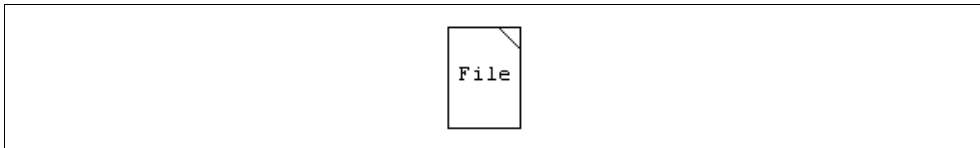


## Other Components

### 4.1 Parameter File

The parameter file element .FILE defines the name of the file that stores the component parameters and limit settings. For example, the resistance of a resistor can be specified as R1, and the value of R1 is defined in the parameter file.

**Image:**



The parameter file is a text file created by the user. The format is shown below:

`<name> = <value>`

`<name> <value>`

`LIMIT <name> <lower limit> <upper limit>`

`% A comment line`

The field `<value>` can be either a numerical number (e.g. “R1 = 12.3”) or a mathematical expression (e.g. “R3 = R1 + R2/2.”). The name and the value can be separated by either an equation sign (e.g. “R1 = 12.3”) or a space (e.g. “R1 12.3”). Text from the character “%” to the end of the line is treated as comments (e.g. “% R3 is the load resistance”).

For example, a parameter file may look like the following:

```
R1=12.3           [R1 is defined as 12.3]
R2 23.4Ohm       [Equation sign can be replaced by space]
% R3 is the load resistance    [This line is comments]
R3=R1+R2/2.      [Math expression is allowed]
L1=3m            [power-of-ten suffix is allowed. L1=0.003]
C1=100uF
```

LIMIT R3 5. 25. [R3 is limited between 5. and 25.]

## 4.2 Sources

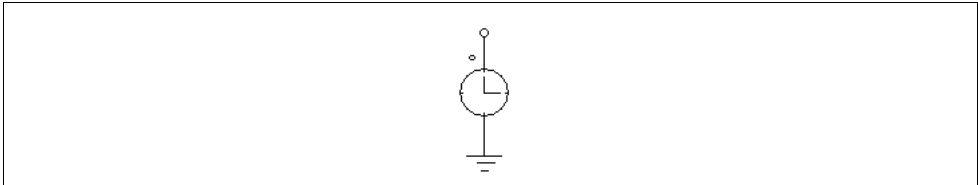
Several types of independent voltage/current sources are available in PSIM. The notation of a current source direction is: the current flows out of the higher-potential node, through the external circuit, and back into the lower-potential node of the source.

Note that current sources can be used in the power circuit only.

### 4.2.1 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.

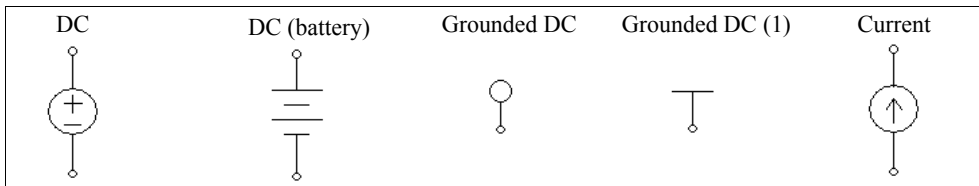
**Image:**



### 4.2.2 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

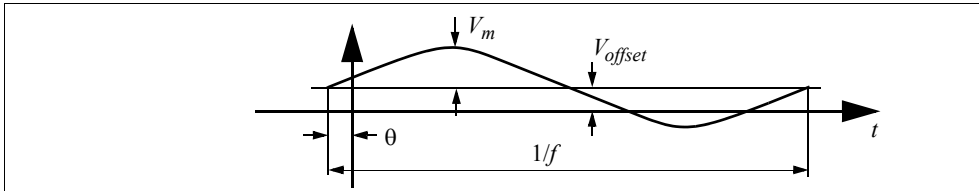


### 4.2.3 Sinusoidal Source

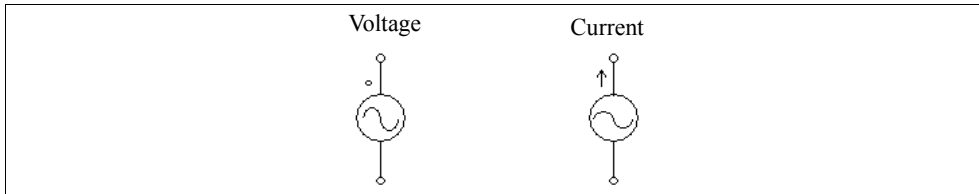
A sinusoidal source is defined as:

$$v_o = V_m \cdot \sin(2\pi \cdot f \cdot t + \theta) + V_{offset}$$

The specifications can be illustrated as follows.



**Images:**

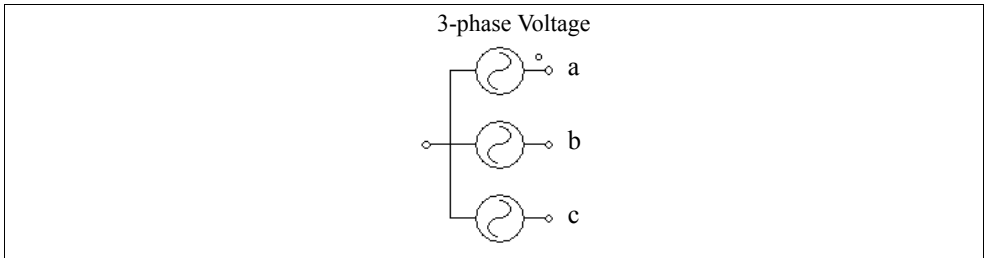


**Attributes:**

Parameters	Description
Peak Amplitude	Peak amplitude $V_m$
Frequency	Frequency $f$ , in Hz
Phase Angle	Initial phase angle $\theta$ , in deg.
DC Offset	DC offset $V_{offset}$
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 module is provided. The dotted phase of the module refers to Phase A.

Image:



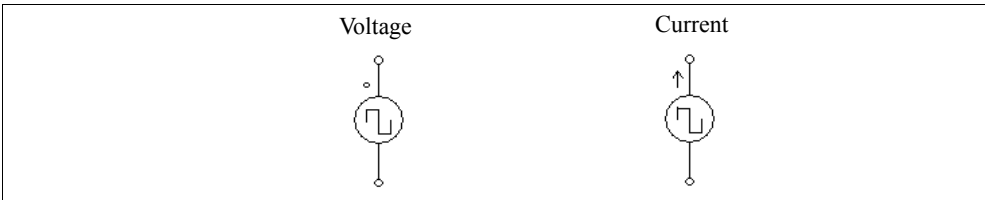
Attributes:

Parameters	Description
V (line-line-rms)	Line-to-line rms voltage amplitude
Frequency	Frequency $f$ , in Hz
Init. Angle (phase A)	Initial angle for Phase A

### 4.2.4 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.

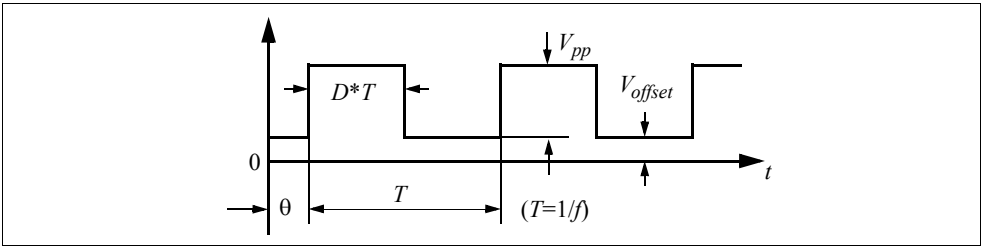
Images:



**Attributes:**

Parameters	Description
V <sub>peak-peak</sub>	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 $\theta$ of the waveform, in deg.

The specifications of a square wave source are illustrated as follows.

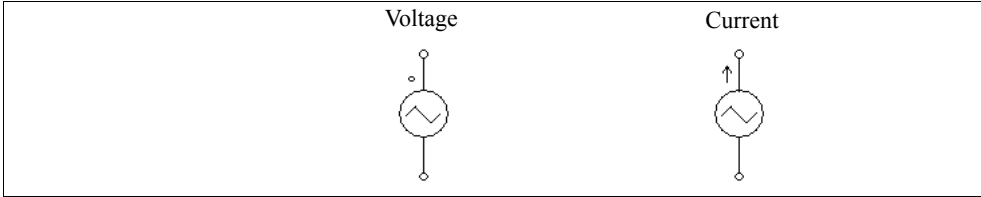


When the phase delay  $\theta$  is positive, the waveform is shifted to the right along the time axis.

### 4.2.5 Triangular Source

A triangular-wave voltage source (VTRI) or current source (ITRI) 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.

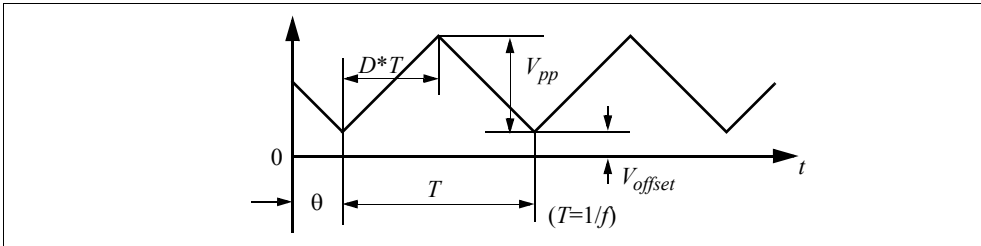
**Images:**



**Attributes:**

Parameters	Description
V <sub>peak-peak</sub>	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 $\theta$ of the waveform, in deg.

The specifications of a triangular wave source are illustrated as:

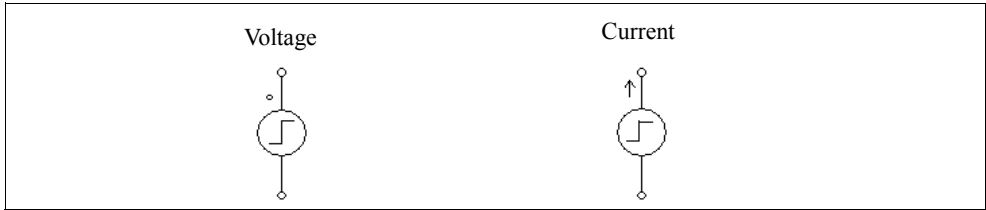


When the phase delay  $\theta$  is positive, the waveform is shifted to the right along the time axis.

**4.2.6 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 (1)* in the library).

**Images:**



**Attributes:**

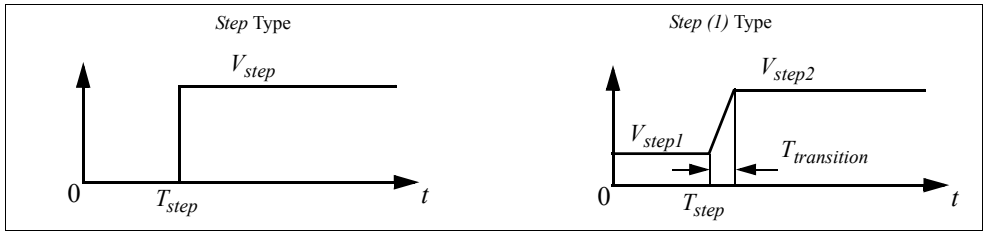
For the *Step* type source:

Parameters	Description
Vstep	Value $V_{step}$ after the step change
Tstep	Time $T_{step}$ at which the step change occurs

For the *Step (1)* 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
T_transition	Transition time $T_{transition}$ from $V_{step1}$ to $V_{step2}$

The specifications of the voltage step sources are illustrated as follows:



**4.2.7 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.

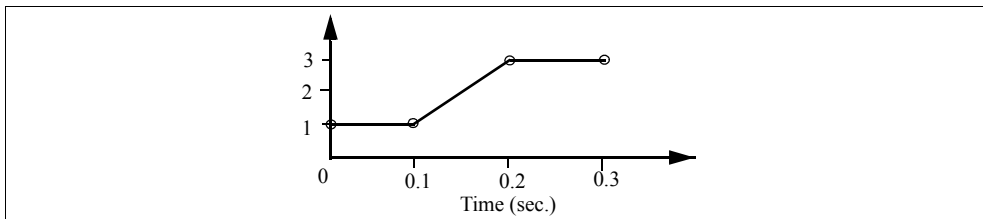
For the sources that define the values and times in pair:

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 (1)* voltage source will be:

Frequency	0.
Times, Values (t1,v1)...	(0., 1) (0.1, 1) (0.2, 3) (0.3, 3)

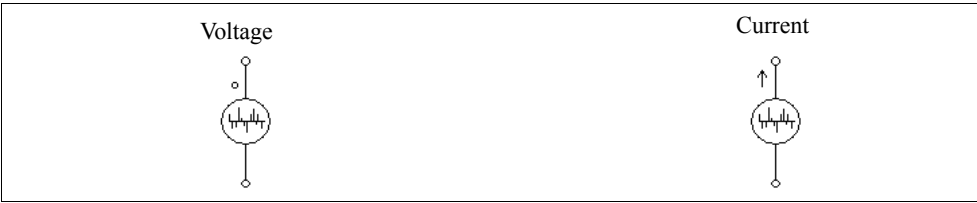
### 4.2.8 Random Source

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:**



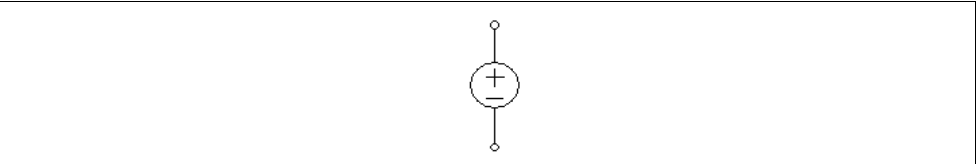
**Attributes:**

Parameters	Description
Peak-Peak Amplitude	Peak-to-peak amplitude of the source
DC Offset	DC offset

### 4.2.9 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:  $\sin(2*3.14159*60*t+2.09)$ .

**4.2.10 Voltage/Current-Controlled Sources**

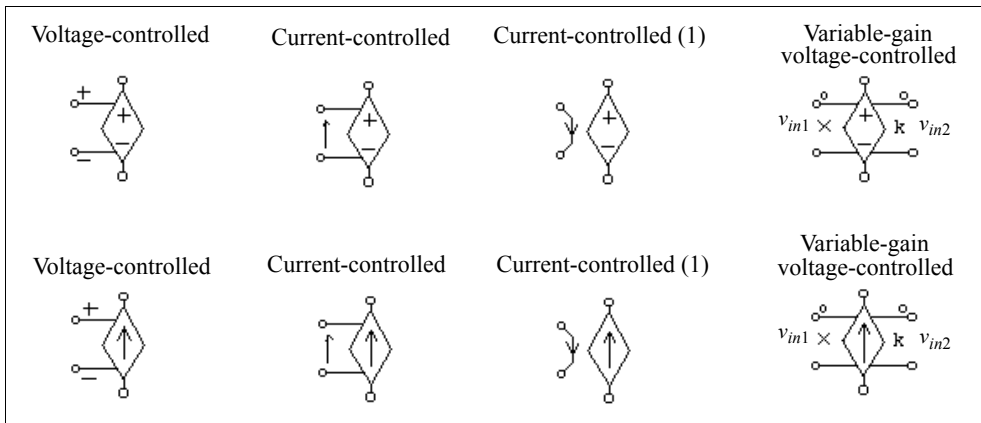
The following types of controlled sources are available:

- 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:**





**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  $k$ .

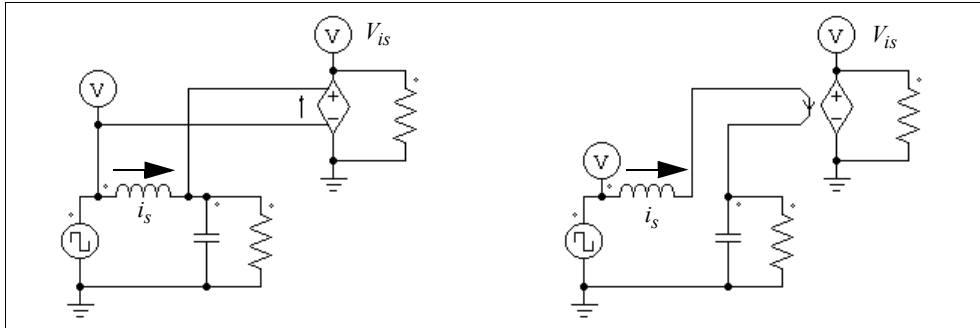
The difference between a variable-gain controlled source and a nonlinear source with multiplication is that, for the nonlinear source with multiplication, values of both  $v_{in1}$  and  $v_{in2}$  at the current time step are used to calculate the output and are updated in each iteration. But for the 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.

The circuit on the right is equivalent to that on the left, except that a different current controlled source is used instead.



#### 4.2.11 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  $i_o = k \cdot \frac{v_{in1}}{v_{in2}}$

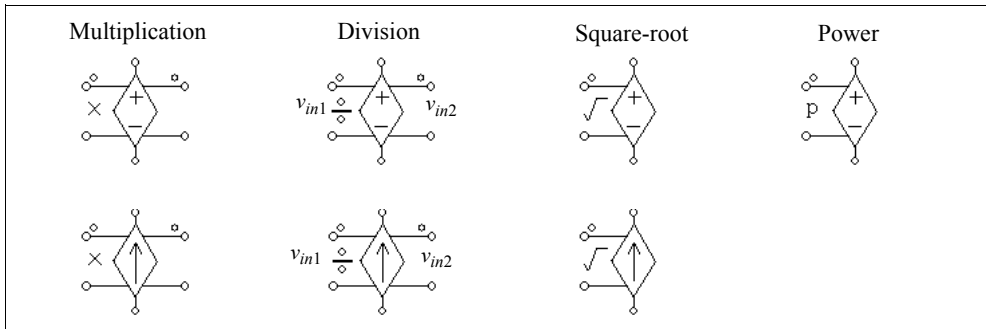
Nonlinear (square-root): Output  $v_o = k \cdot \sqrt{v_{in1}}$  or  $i_o = k \cdot \sqrt{v_{in1}}$

Nonlinear (power): Output  $v_o = \text{sign}(v_{in}) \cdot k \cdot (k_1 \cdot v_{in})^{k_2}$

In the nonlinear power source, the term  $\text{sign}(v_{in})$  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.

**4.3 Voltage/Current 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  $1\ \mu\Omega$ .

**Images:**



**Attribute:**

Parameter	Description
Gain	Gain of the sensor

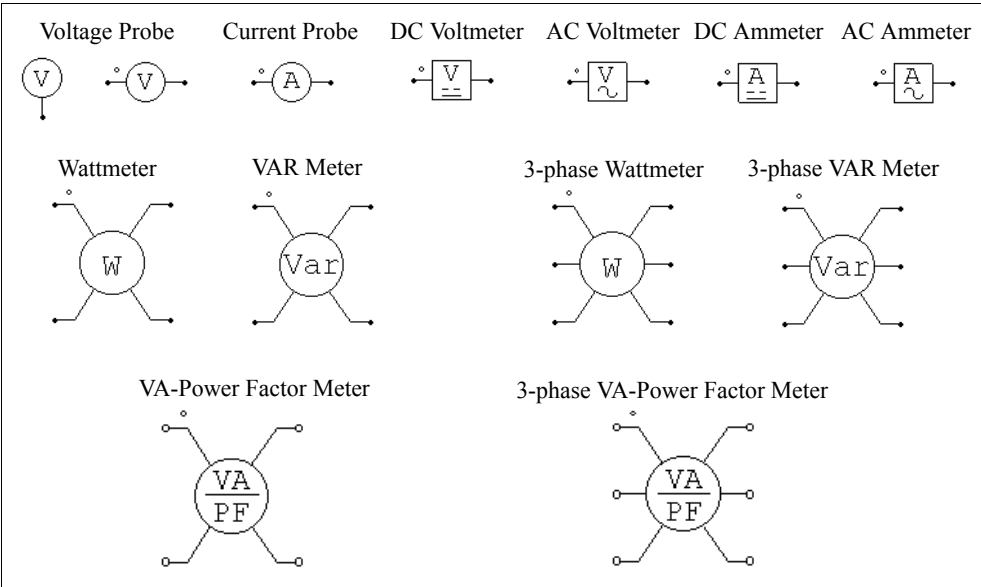
**4.4 Probes and Meters**

Probes and meters are used to measure voltages, currents, power, or other quantities. A voltage probe (VP) measures a node voltage with respect to ground. A two-terminal voltage probe (VP2) measures the voltage between two nodes. A current probe (IP) measures the current through the probe. Note that all the probes and meters, except the node-to-ground probe VP, are allowed in the power circuit only.

While probes measure a voltage or current quantity in its true form, meters can be used to measure the dc or ac voltage/current, or the real power and reactive power. These meters function in the same way as the actual meters.

A small resistor of 1  $\mu\Omega$  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
VA Display Flag	Display flag for apparent power (0: no display; 1: display)
PF Display Flag	Display flag for power factor (0: no display; 1: display)
DPF Display Flag	Display flag for displacement power factor (0: no display; 1: display)

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 and VAR 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.

For the single-phase VA-Power Factor meter, the 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.

$$v(t) = \sqrt{2}V_1 \sin(\omega_1 t + \phi_1) + \sqrt{2}V_2 \sin(\omega_2 t + \phi_2) + \dots$$

$$i(t) = \sqrt{2}I_1 \sin(\omega_1 t + \theta_1) + \sqrt{2}I_2 \sin(\omega_2 t + \theta_2) + \dots$$

where  $\omega_1$  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 apparent power is defined as:

$$S = V_{rms} \cdot I_{rms}$$

The real power (or average power) is defined as:

$$P = \frac{1}{T} \int_0^T (v(t) \cdot i(t)) dt$$

where  $T$  is the fundamental period. 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 three-phase circuit, the definitions are similar. Note that the meter VA\_PF3 is for the 3-phase 3-wire circuit, and the summation of the three phase voltages or currents must be equal to zero, that is:

$$v_a + v_b + v_c = 0$$

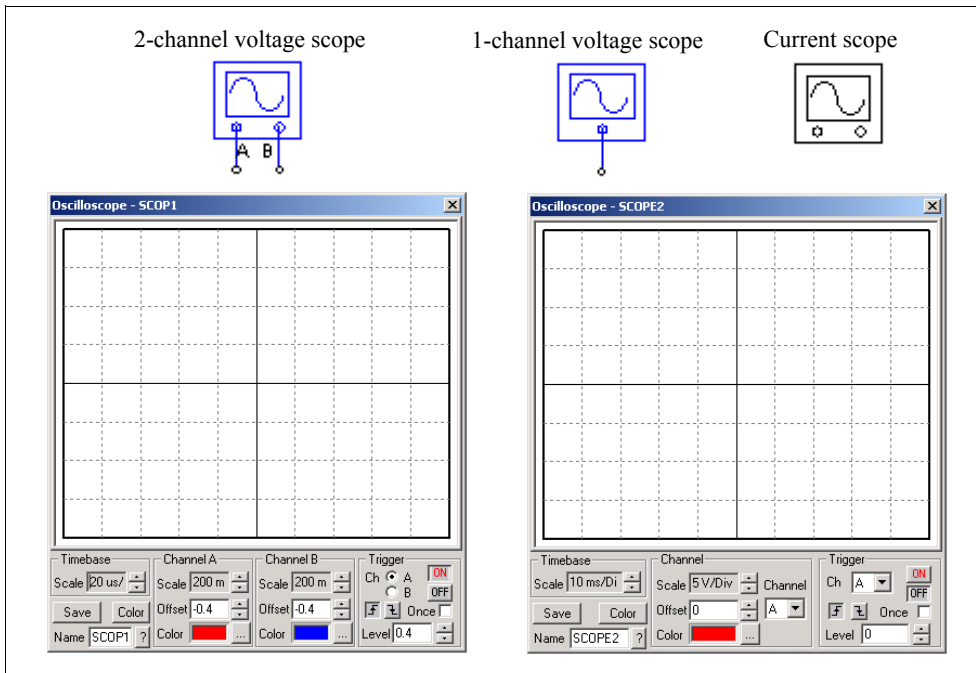
$$i_a + i_b + i_c = 0$$

## 4.5 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, and current scope.

Below are the images of the voltage and current scopes and their expanded view.



The 1-channel voltage scope and the current scope have the same interface.

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.

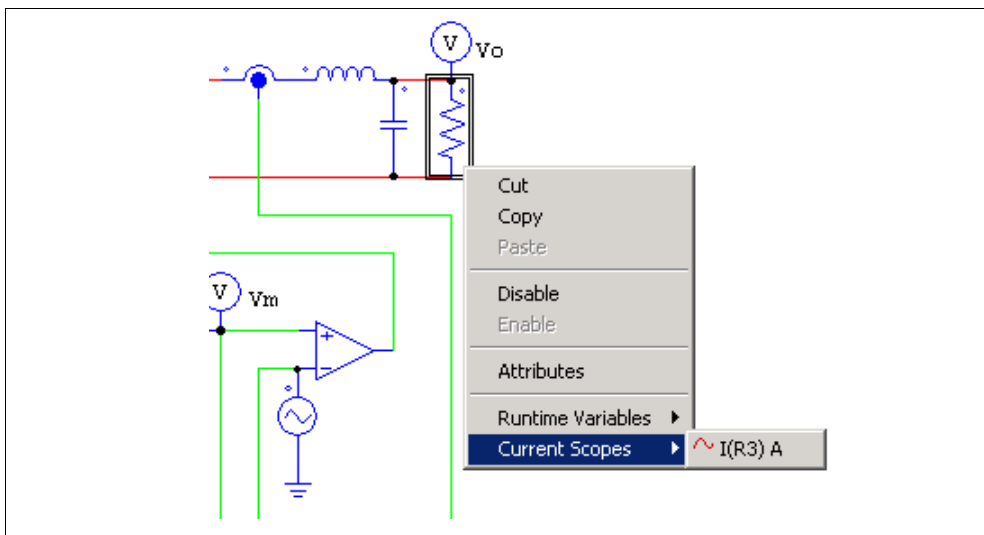
In the *Trigger* section, the trigger conditions are defined. The trigger can be set to either *ON* or *OFF*. When the trigger is off, the waveform is free-running, and the display of the waveform in the scope may vary from one frame to another. If the trigger is on, the waveform display will only start 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.

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  
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).

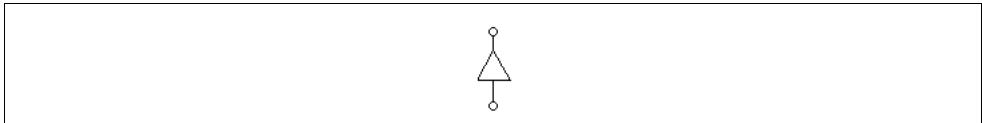
## 4.6 Switch Controllers

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.

### 4.6.1 On-Off Switch Controller

On-off switch controllers are used as the interface between control gating signals and power switches. 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.

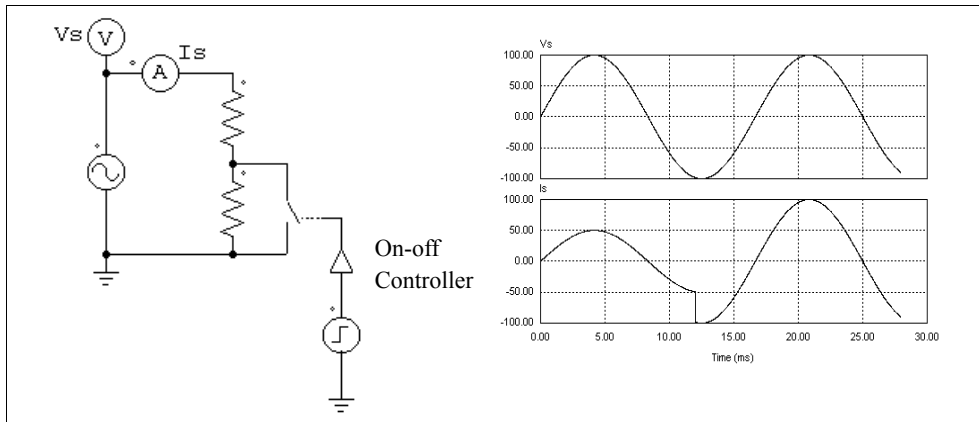
**Image:**



**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.

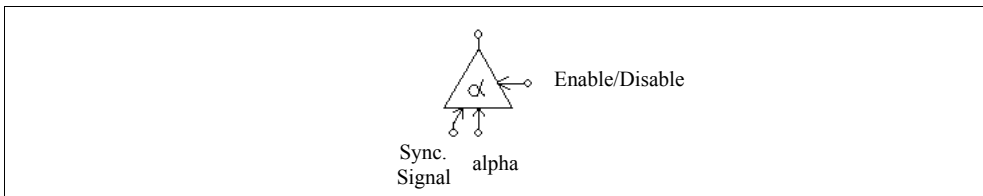




## 4.6.2 Alpha Controller

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:**



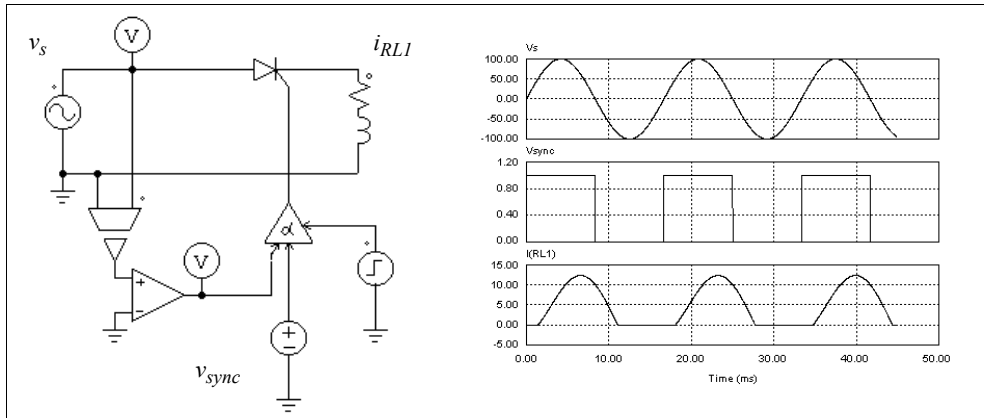
**Attributes:**

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:

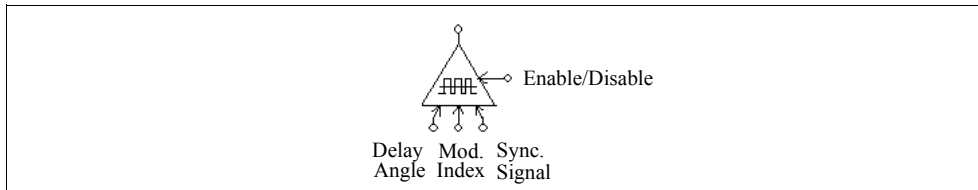
The figure below shows a thyristor circuit using delay angle control. In the circuit, the zero-crossing of  $v_s$ , which corresponds to the moment that the thyristor would start conducting naturally, is used to provide the synchronization. The delay angle is set at  $30^\circ$ . The gating signal is delayed from the rising edge of the synchronization signal by  $30^\circ$ .



### 4.6.3 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 $360^\circ$ , the gatings are updated at every cycle. If it is $60^\circ$ , the gatings are updated at every $60^\circ$ .
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:

$$\begin{aligned}
 &n, m_1, m_2, \dots, m_n \\
 &k_1 \\
 &G_{1,1}, G_{1,2}, \dots, G_{1,k_1} \\
 &\dots \dots \dots \\
 &k_n \\
 &G_{n,1}, G_{n,2}, \dots, G_{n,k_n}
 \end{aligned}$$

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 following table shows an example of a PWM pattern file with five modulation index levels and 14 switching points.

```

5, 0.901, 0.910253, 0.920214, 1.199442, 1.21
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.

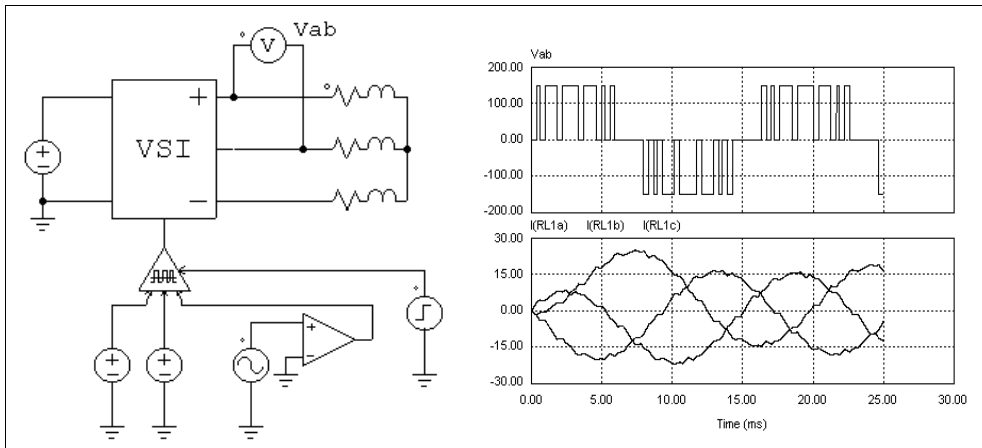
```

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.  
14  
10.189426 87.47009 88.97936 91.02065 92.52991 169.8106 180.  
190.1894 267.4701 268.9793 271.0207 272.5299 349.8106 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.

### Example:

This example shows a three-phase voltage source inverter (file: “vsi3pwm.sch”). The PWM for the converter uses the selected harmonic elimination. The gating patterns are described above and are pre-stored in File “vsi3pwm.tbl”. The gating pattern is selected based on the modulation index. The waveforms of the line-to-line voltage and the three-phase load currents are shown below.

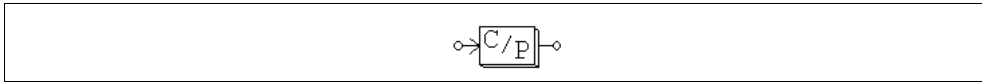


## 4.7 Function Blocks

### 4.7.1 Control-Power Interface Block

A control-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 constant 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.

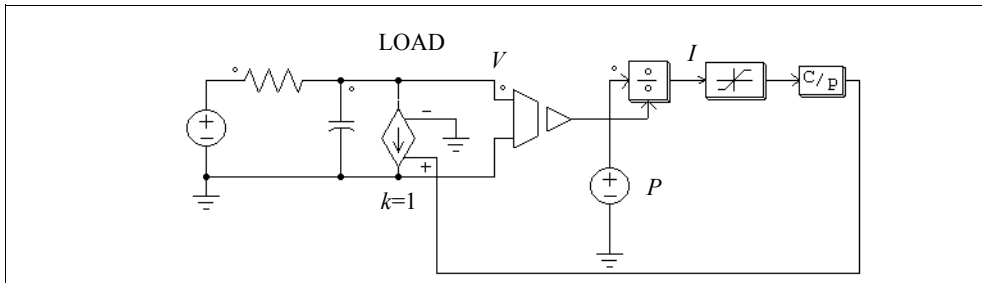
**Image:**



#### Example: A Constant-Power Load Model

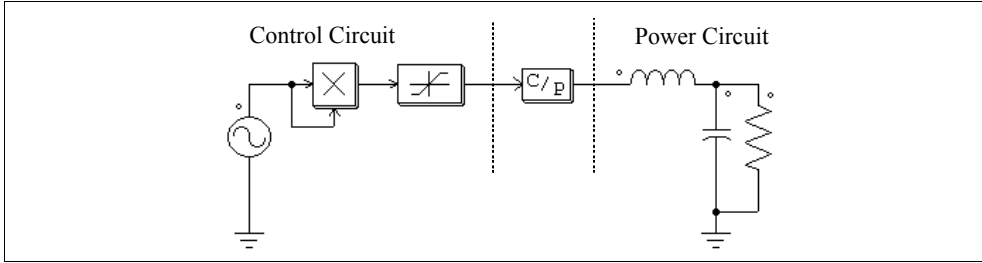
In a constant-power dc load, the voltage  $V$ , current  $I$ , and power  $P$  have the relationship as  $P=V*I$ . Given the voltage and the power, the current can be calculated as  $I=P/V$ . This can be implemented using the circuit as shown below.

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:

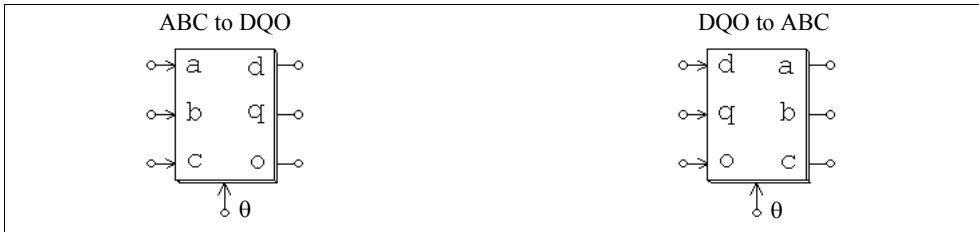
The following circuit illustrates how a control circuit signal can be passed to the power circuit. As seen from the power circuit, the CTOP block behaves as a grounded voltage source.



#### 4.7.2 ABC-DQO Transformation Block

The ABC-DQO function blocks perform the abc-dqo transformation. They convert three voltage quantities from one coordinate to another. These blocks can be used in either the power circuit or the control circuit.

**Images:**



It should be noted that, in the 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.

The transformation equations from abc to dqo are:

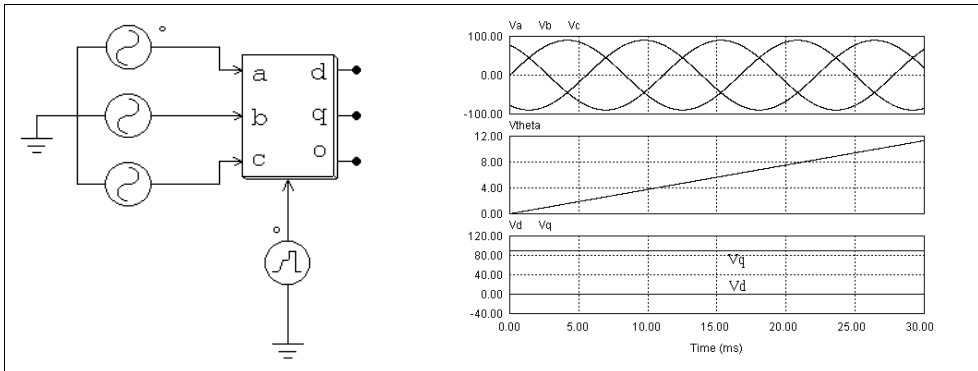
$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos \theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin \theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

The transformation equations from dqo to abc are:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix}$$

### Example:

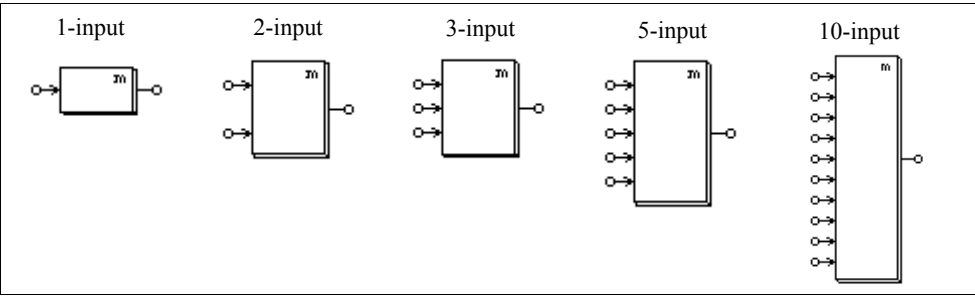
In this example, three symmetrical ac waveforms are transformed into dqo quantities. The angle  $\theta$  is defined as  $\theta = \omega t$  where  $\omega = 2\pi \cdot 60$ . Since the angle  $\theta$  changes linearly with time, a piecewise linear voltage which has a ramp waveform is used to represent  $\theta$ . The simulation waveforms show the three-phase ac (top), the angle  $\theta$  (middle), and the dqo output. In this example, the “q” component is constant, and both the “d” and the “o” components are zero.



### 4.7.3 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 $f(x_1, x_2, \dots, x_n)$	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.

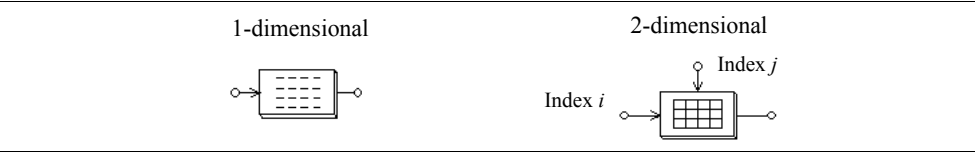
The variables that are allowed in the expression are:  $T$  or  $t$  for time, and  $x_i$  ( $i$  from 1 to  $n$ ) which represents the  $i_{th}$  input. For example, for the 3-input math function block, the allowed variables are:  $T$ ,  $t$ ,  $x_1$ ,  $x_2$ , and  $x_3$ . For the 1-input math function block, the variable  $x$ , which refers to the only input, is also allowed.

### 4.7.4 Lookup Tables

There are three types of lookup tables: one-dimensional lookup table, 2-dimensional lookup table with integer inputs, and 2-dimensional lookup table with floating-point inputs.

All three types of lookup tables can be used in both power circuit and control circuit.

**Images:**





**Attribute:**

Parameter	Description
File Name	Name of the file storing the lookup table

For the 2-dimensional lookup table block, the node at the left is for the row index input, and the node at the top is for the column index input.

The one-dimensional lookup table has one input and one output. Two data arrays, corresponding to the input and the output, are stored in the lookup table in a file. The format of the table is as follows.

$$\begin{array}{l}
 V_{in}(1), V_o(1) \\
 V_{in}(2), V_o(2) \\
 \dots \\
 V_{in}(n), V_o(n)
 \end{array}$$

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  $V_{in}(n)$ , the output will be clamped to  $V_o(1)$  or  $V_o(n)$ .

The 2-dimensional lookup table with integer input has two inputs. The output data is stored in a 2-dimensional matrix. The two input 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. The data for the lookup table are stored in a file and have the following format:

$$\begin{array}{l}
 m, n \\
 A(1,1), A(1,2), \dots, A(1,n) \\
 A(2,1), A(2,2), \dots, A(2,n) \\
 \dots \dots \dots \\
 A(m,1), A(m,2), \dots, A(m,n)
 \end{array}$$

where  $m$  and  $n$  are the number of rows and columns, respectively. Since the row or the column index must be an integer, the input value is automatically converted to an integer. If either the row or the 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 2-dimensional lookup table with floating-point inputs is similar to the 2-dimensional lookup table with integer inputs. The difference is that in this case, inputs are floating-point values, and interpolation is used to calculate the output.

The data for the lookup table are stored in a file and have the following format:

$m, n$   
 $V_{r,1} \ V_{r,2} \ \dots \ V_{r,m}$   
 $V_{c,1} \ V_{c,2} \ \dots \ V_{c,n}$   
 $A(1,1), A(1,2), \dots, A(1,n)$   
 $A(2,1), A(2,2), \dots, A(2,n)$   
 $\dots \dots \dots$   
 $A(m,1), A(m,2), \dots, A(m,n)$

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 the input falls 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 be the same as 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
Row	1.1	-2		4
	2	2.091	3.826	4.818
	2.2	3		5

#### 4.7.5 C Script Block

The C script block allows users to enter C code directly without compiling the code, unlike in the case of external DLL blocks where users need to compile the code into a DLL using a compiler. The code of the C script block will be interpreted and executed at runtime by a built-in C interpreter in PSIM.

This block makes it very easy to support custom C codes, and to define and modify the functionality of the block.

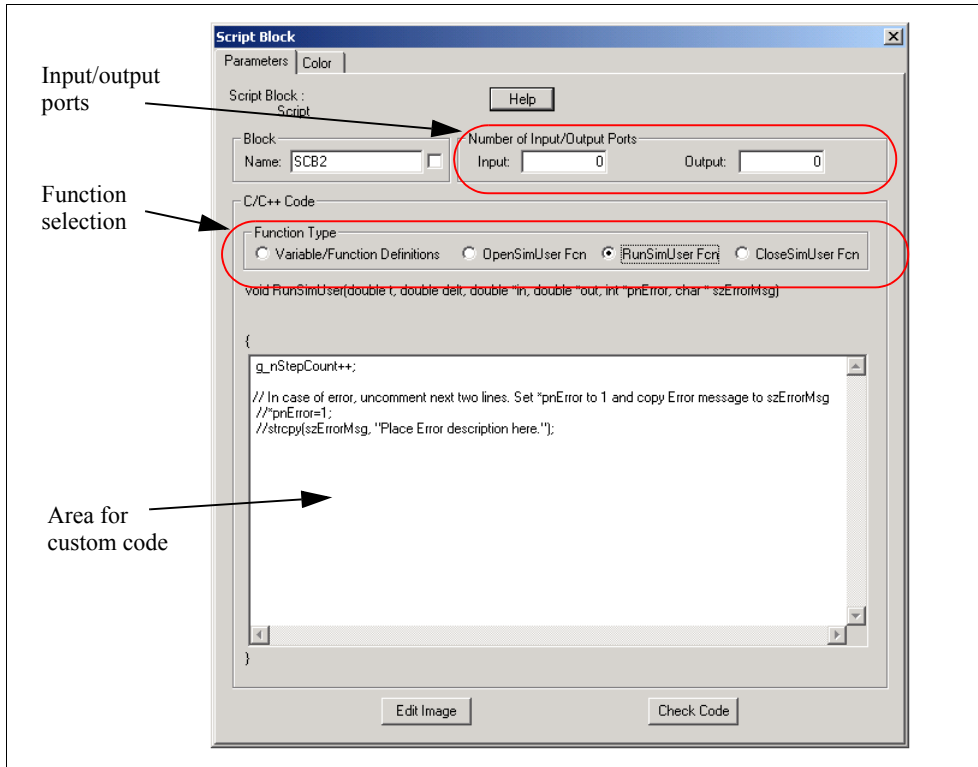
The interface of the C script block dialog window is shown below.

In the *Number of Input/Output Ports* section, the number of input and output ports of the ports is defined. After the number of ports is changed, the image of the block in the schematic will change accordingly.

In the *Function Type* section, there are four choices:

- Variable/Function Definitions:* For includes statements and global variable definition.
- OpenSimUser Fcn:* The function that is called only once at the beginning of the simulation for initialization.
- RunSimUser Fcn:* The function that is called at each simulation step.
- CloseSimUser Fcn:* The function that is called only once at the end of the simulation for termination.

When one of the choices is selected, the area underneath shows the corresponding code. The complete code consists the combined code of all the four parts.



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 script block.

The difference between the C script block and the external DLL block is that, even though the C script block is easier to use, it does have the disadvantage that the custom code in the C script 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.

#### 4.7.6 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

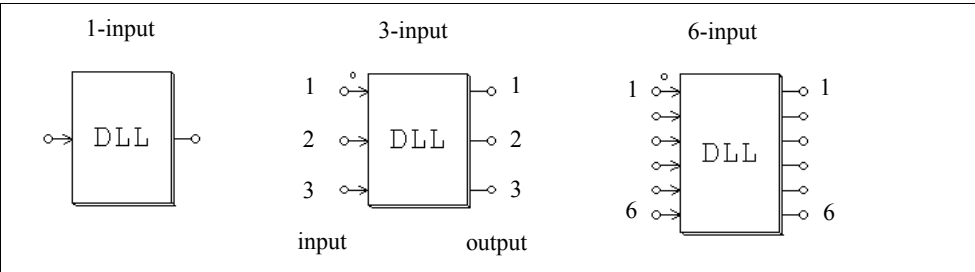
transfer function blocks, and digital filters), the DLL block is called only at the discrete sampling times.

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.

In general, the simple DLL block is easier to program and easier to use.

The images and parameters of the simple DLL blocks are shown below.

**Images:**



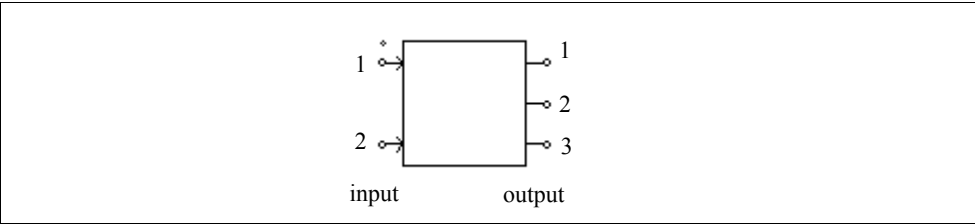
**Attribute:**

Parameter	Description
File Name	Name of the DLL file

The node with a dot is for the first input (in[0]). The sequence of the input/output nodes is from the top to the bottom.

The images and parameters of the simple DLL blocks 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 the DLL routine 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 the DLL routine (optional)
Parameter 2	Parameter to be passed from PSIM into the DLL routine (optional)
<b>Edit Image</b> (button)	Click this button to edit and customize the image of the DLL block.
<b>Display File</b> (button)	Click this button to display the content of the <i>Input Data File</i> (optional).
<b>Read File</b> (button)	If the <i>Input Data File</i> is modified, click this button to reload the data file (optional).

The node with a dot is for the first input (in[0]). The sequence of the input/output nodes 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 three places, in the order of precedence: in the PSIM directory, in the same directory as the schematic file that uses the DLL file, or in the directory as defined in the **Options** -> **Set Path** function in PSIM.

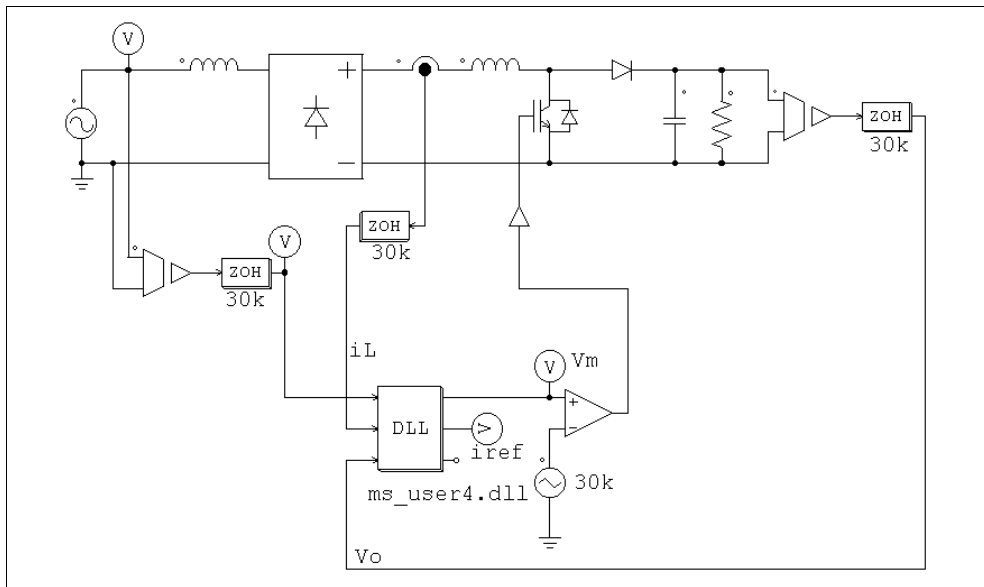
Sample DLL files are provided in PSIM, and users can use these files as the templates to

write their own. Procedures on how to compile the DLL routine and link with PSIM are provided in these files and in the on-line help.

### Example:

The following shows a power factor correction circuit with the inductor current and the load voltage feedback. The input voltage is used to generate the current reference. The control scheme is implemented in a digital environment, with a sampling rate of 30 kHz. The control scheme is implemented in an external C code and is interfaced to the power circuit through a simple DLL block.

The input of the DLL block are the sampled input voltage, inductor current, and output voltage. One of the DLL block outputs is the modulation wave  $V_m$ , which is compared with the carrier wave to generate the PWM gating signal for the switch. The other output is the inductor current reference for monitoring purpose.



Part of the source code, which is in the file “pfc\_vi\_dll.c”, is shown below. Both the inner current loop and the outer voltage loop use a PI controller. Trapezoidal rule is used to discretize the controllers. Discretization using Backward Euler is also implemented but the codes are commented out.

```

// This sample program implement the control of the circuit "pfc-vi-dll.sch" in a C routine.
// Input: in[0]=Vin; in[1]=iL; in[2]=Vo
// Output: Vm=out[0]; iref=out[1]
// You may change the variable names (say from "t" to "Time").
// But DO NOT change the function name, number of variables, variable type, and sequence.
// Variables:
//   t: Time, passed from PSIM by value
//   delt: Time step, passed from PSIM by value
//   in: input array, passed from PSIM by reference
//   out: output array, sent back to PSIM (Note: the values of out[*] can be modified in PSIM)
// The maximum length of the input and output array "in" and "out" is 20.
// Warning: Global variables above the function simuser (t,delt,in,out) are not allowed!!!
#include <math.h>
__declspec(dllexport) void simuser (t, delt, in, out)
// Note that all the variables must be defined as "double"
double t, delt;
double *in, *out;
{
    // Place your code here.....begin
    double Voref=10.5, Va, iref, iL, Vo, Vm, errv, erri, Ts=33.33e-6;
    static double yv=0., yi=0., uv=0., ui=0.;
    // Input
    Va=fabs(in[0]);
    iL=in[1];
    Vo=in[2];
    // Outer Loop
    errv=Voref-Vo;
    // Trapezoidal Rule
    yv=yv+(33.33*errv+uv)*Ts/2.;
    iref=(errv+yv)*Va;
    // Inner Loop
    erri=iref-iL;
    // Trapezoidal Rule
    yi=yi+(4761.9*erri+ui)*Ts/2.;
    Vm=yi+0.4*erri;
    // Store old values
    uv=33.33*errv;
    ui=4761.9*erri;
    // Output
    out[0]=Vm;
    out[1]=iref;
    // Place your code here.....end
}

```



### 4.7.7 Embedded Software Block

The 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

The Embedded Software Block has similar functionality as the 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*".



## Analysis Specification

### 5.1 Transient Analysis

The transient analysis is set up by selecting **Simulation Control** in the **Simulate** menu in PSIM, and defining the parameters as follows.

Time Step	Simulation time step, in sec.
Total Time	Total simulation time, in sec.
Free Run checkbox	<p>When the <i>Free Run</i> checkbox is not checked, the simulation will run up to the <i>Total Time</i> 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.</p> <p>In the free-run mode, voltage/current scopes can be used to monitor and display voltages and currents in the middle of the simulation.</p>
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 the print step is set to 1, every data point will be saved to the output file. If it is 10, for example, 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 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.

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.

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.

In PSIM, an interpolation technique is implemented which will calculate the exact switching instants. 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. If the selected time step is different from the one set by the user, it will be saved to the file “message.txt”.

## 5.2 AC Analysis

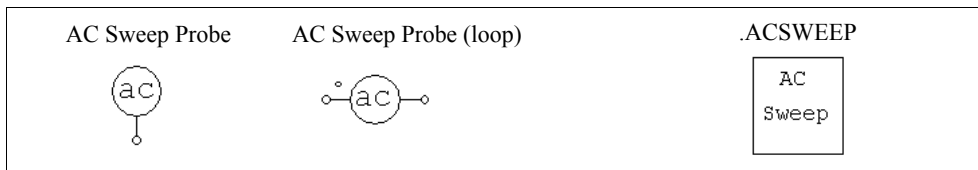
The frequency response of a circuit or a control loop can be obtained with the ac analysis. A key feature of the ac analysis in PSIM is that, a circuit can be in its original switchmode form, and no average model is required. Nevertheless, with the average model, the time it takes to perform the ac analysis will be shorter.

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 .ACSWEEP element 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 .ACSWEEP sweep element.

**Images:**



**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.

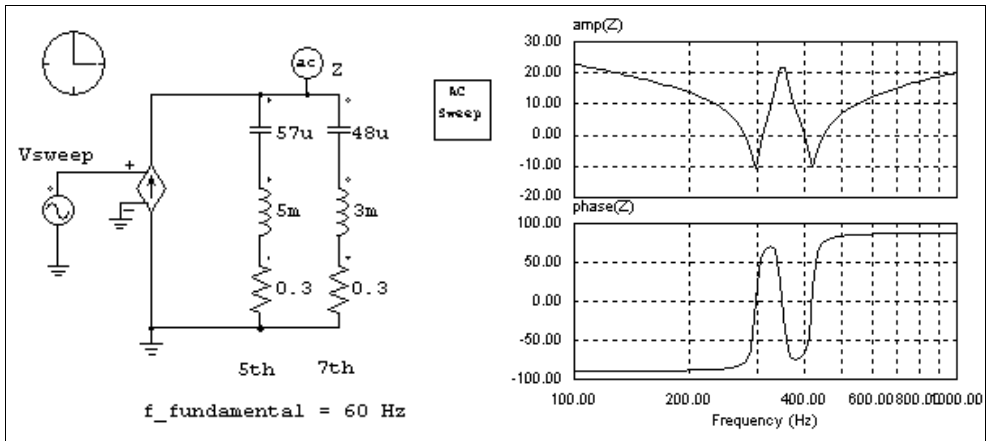
Sometimes, after ac analysis is complete, a warning message is displayed as follows:

**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.

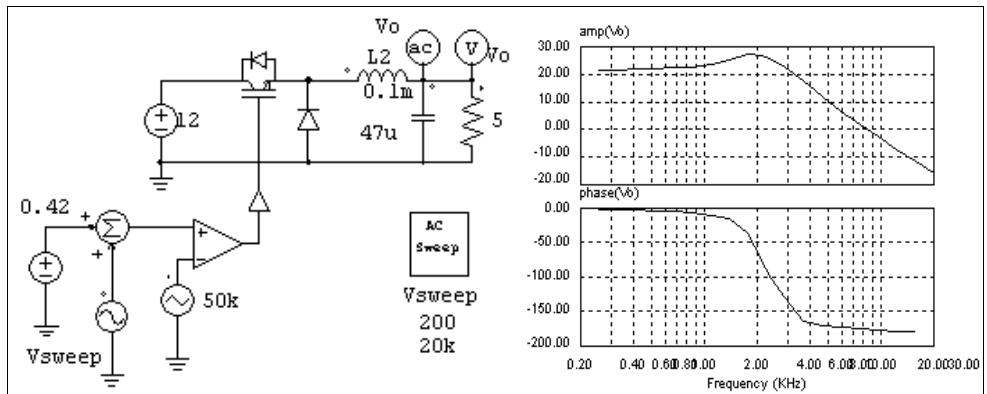
### Example: Impedance of Shunt Filters

The circuit below consists of two shunt filters tuned at the 5th and 7th harmonics (the fundamental frequency is 60 Hz). By injecting the excitation source as the current and measuring the voltage, we obtain the impedance characteristics of the filters. The ac analysis waveform on the right clearly shows two troughs at 300 Hz and 420 Hz.



### Example: Open-Loop Response of a Buck Converter

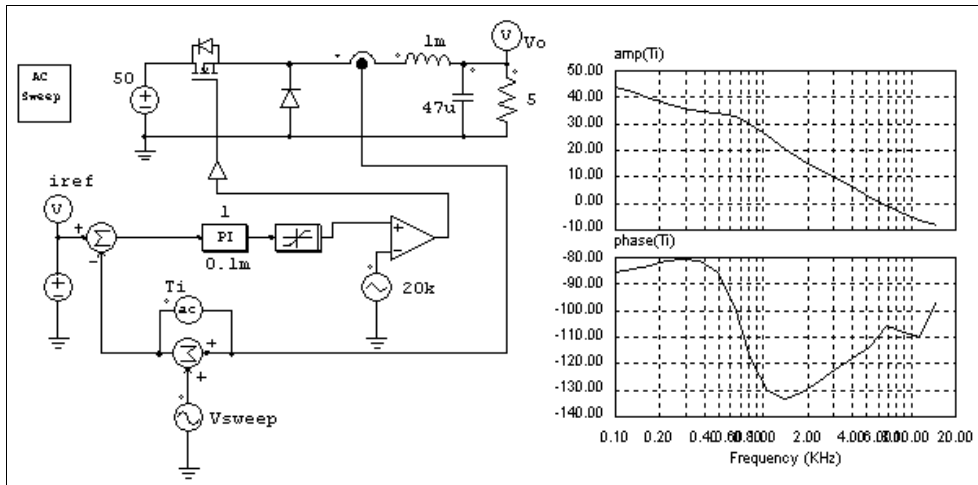
The circuit on the left is an one-quadrant buck converter. An excitation source is injected to the modulation signal, and the output voltage is measured. The result of the ac analysis, on the right, shows the open-loop response of the output voltage versus the modulation signal.



### Example: Loop Transfer Function of a Closed-Loop Circuit

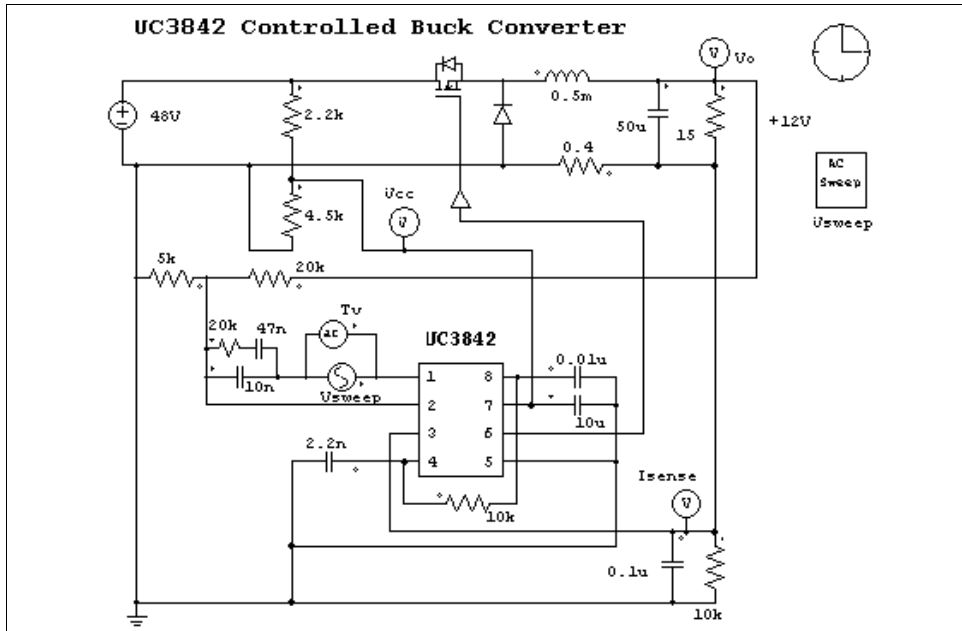
The ac analysis can be used to find out the loop response of a closed-loop system. The circuit below shows a buck converter with average current mode control. By injecting the excitation signal into the current feedback path, and using the node-to-node ac sweep probe, we can obtain the loop transfer 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.



### 5.3 Parameter Sweep

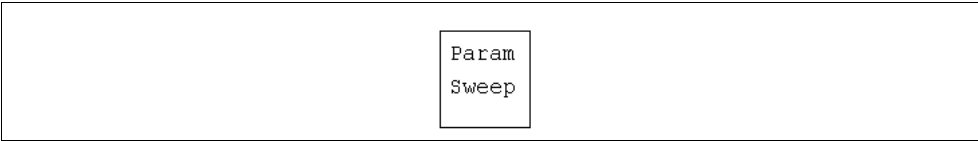
Parameter sweep can be performed for the following parameters:

- Resistance, inductance, and capacitance of RLC branches
- Gain of proportional blocks
- Time constant of integrators
- Gain and time constant of proportional-integral controllers
- Gain, cut-off frequency, and damping ratio of 2nd-order low-pass and high-pass filters
- Gain, center frequency, and passing and stopping band of 2nd-order band-pass and band-stop filters

The image and parameters of the parameter sweep element are shown below.



**Image:**



**Attributes:**

Parameters	Description
Start Value	Starting value of the parameter
End Value	End value of the parameter
Increment Step	Increment step
Parameter to be Swept	Parameter to be swept

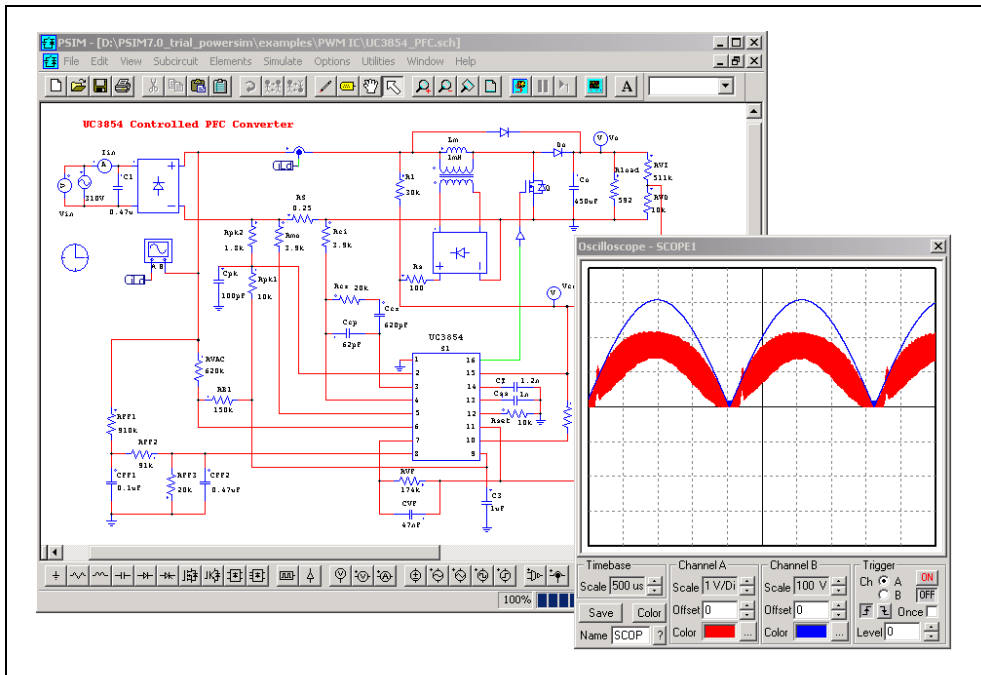
For example, let the resistance of a resistor be “Ro”. To sweep the resistance from 2 Ohm to 10 Ohm, with a step of 2 Ohm, the specification will be:

Start Value	2
End Value	10
Increment Step	2
Parameter to be Swept	Ro



## Circuit Schematic Design

PSIM's schematic program provides interactive and user-friendly interface for circuit schematic entry and editing. The following figure shows a boost power factor correction circuit in the PSIM environment.



In PSIM, all the elements are stored under the **Elements** menu. The elements are divided into four groups: *Power* (for power circuit element), *Control* (for control elements), *Other* (for switch controllers, sensors, probes, interface elements, and elements that are common to both power and control), and *Sources* (for voltage and current sources).

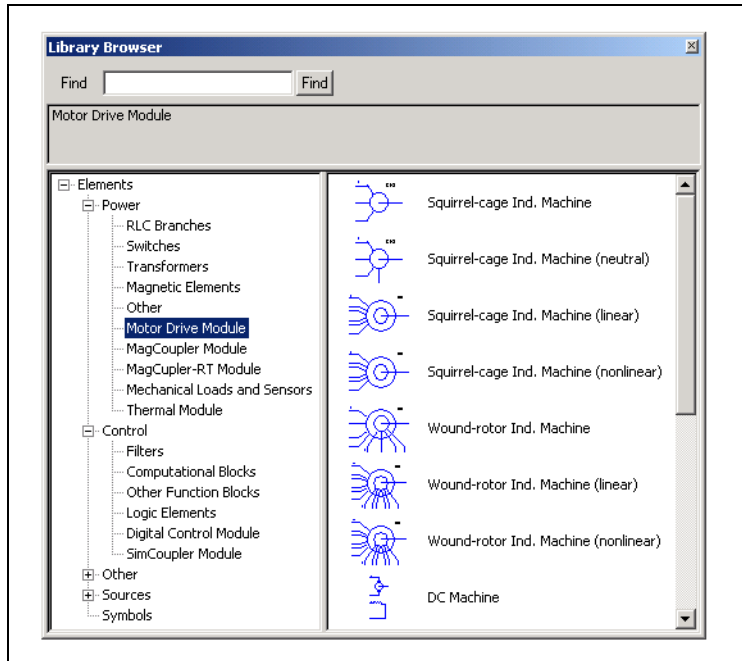
## 6.1 Creating a Circuit

The following functions are provided for circuit creation.

### Get

There are two 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.

Another is to use the Library Browser, as shown below.



The Library Browser provides a convenient way of navigating through the library. To display the Library Browser, go to **View -> Library Browser**.

Also, the most recent elements are listed in the pull-down button



on the toolbar.

### 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.

### Rotate

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

<b>Wire</b>	<p>To connect a wire between two nodes, select <b>Wire</b>. 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.</p> <p>For easy inspection, a floating node is displayed as a circle, and a junction node is displayed as a solid dot.</p>
<b>Label</b>	<p>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.</p> <p>The text of a label can be moved. To select the text, left click on the label, then press the <b>Tab</b> key.</p>
<b>Assign</b>	<p>To assign the parameters of an element, double click on the element. A dialog box will appear. Specify the values and hit the &lt;Return&gt; key or click on <b>OK</b>.</p>


## 6.2 Editing a Circuit

Double left click or single right click on top of an element will bring up the property dialog window. Also, to move the whole schematic, right click and drag the mouse.

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

<b>Select</b>	<p>To select an element, click on the element. A rectangle will appear around the element.</p> <p>To select a block of a circuit, keep the left button of a mouse pressed and drag the mouse until the rectangle covers the selected area.</p>
<b>Copy</b>	<p>To copy an element or a block of the circuit, select the element or the region, and choose <b>Copy</b>. Then choose <b>Paste</b> place the element or circuit.</p>
<b>Delete</b>	<p>To delete an element, a block of a circuit, or a wire, select the item, and choose <b>Cut</b>, or hit the &lt;Delete&gt; key. Note that if <b>Cut</b> is used, the last deleted item can be pasted back. This is equivalent to un-do.</p>
<b>Move</b>	<p>To move an element or a circuit block, select the element/circuit block and drag the mouse while keeping the left button pressed.</p>
<b>Text</b>	<p>To place text on the screen, choose <b>Text</b>. Enter the text in the</p>

dialog box, and click the left button of the mouse to place it.

<b>Disable</b>	To 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.
<b>Enable</b>	To enable a previously disabled element or circuit.
<b>Escape</b>	Quit from any of the above editing modes by choosing <b>Escape</b> .
<b>Copy to Clipboard</b>	To copy the schematic image to the clipboard which can then be pasted back in another software. One can choose either the <b>Color</b> option or the <b>Black &amp; White</b> option. The Black & White option will result in a smaller image file size.
<b>Drawing</b>	To draw images on the schematic for display purposes. The following images are provided: line, ellipse, rectangle, and half-circle.
<b>Add/Remove Current Scope</b>	To add or remove the current scope for elements that have current flags. After this is selected, a scope image will appear with the mouse. Then click on top of an element with the current flag, and select the branch current name. Select the branch current again to remove the current scope.
<b>Show/Hide Runtime Variables</b>	<p>To 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. 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.</p>

In addition, the following functions are provided in the **View** menu:

<b>Zoom</b>	Select <b>Zoom In</b> to zoom in the circuit, or <b>Zoom In Selected</b> to zoom in to a selected region. Choose <b>Zoom Out</b> to zoom out, or <b>Fit to Page</b> to zoom out to fit the entire circuit to the screen.
<b>Element List</b>	To generate the parts list of the circuit.
<b>Element Count</b>	To count the number of elements. Voltage/current probes and meters are not included in the element count.

**Custom Toolbars** To create customized toolbars.

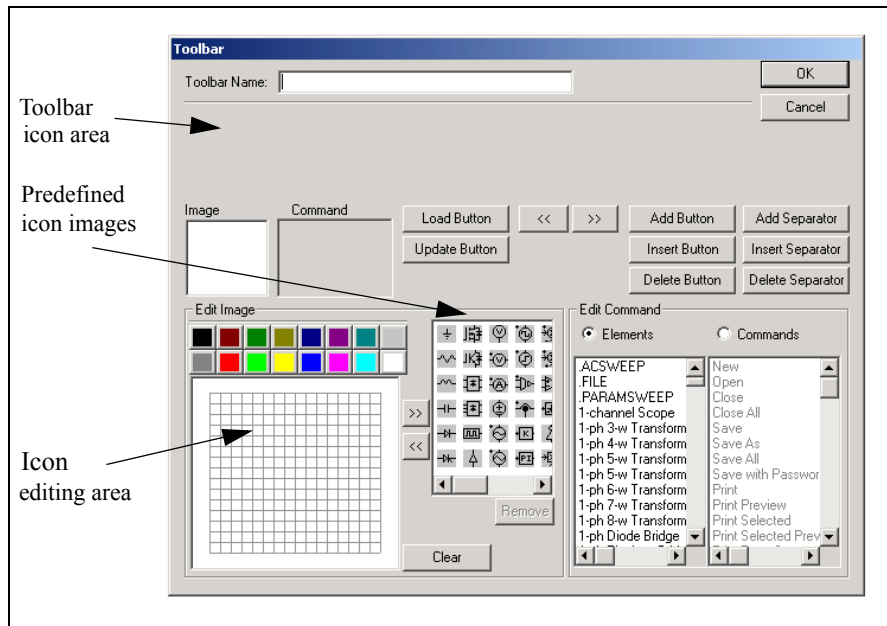
**Custom Keyboard** To customize keyboard. Functions can be assigned to the keyboard for easier circuit editing.

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

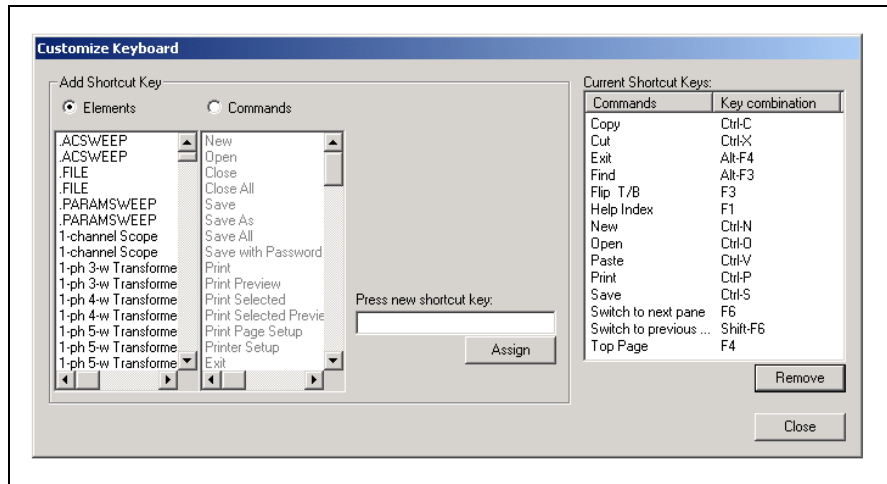
Custom Toolbars window.

- 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:

- 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.

## **6.3 Saving a File**

After a schematic is created, it can be saved to a binary file by choosing the **Save** or **Save As** function in the **File** menu.

By choosing **File -> Save with Password**, one can protect a file 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.

The function **Save in Package File** in the **File** menu allows users to save all the files that are associated with a simulation to one single 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 the single file out.

The function **Save as Version 6.2** in the **File** menu saves a file in the Version 6.2 format. Note that if the file uses elements that are unique in Version 7.0, these elements will be omitted. Also, Version 6.2 file format is very close to Version 6.1 format, and a file in Version 6.2 format can be opened by PSIM Version 6.1.

## 6.4 **Subcircuit**

The following functions are provided for subcircuit editing and manipulation.

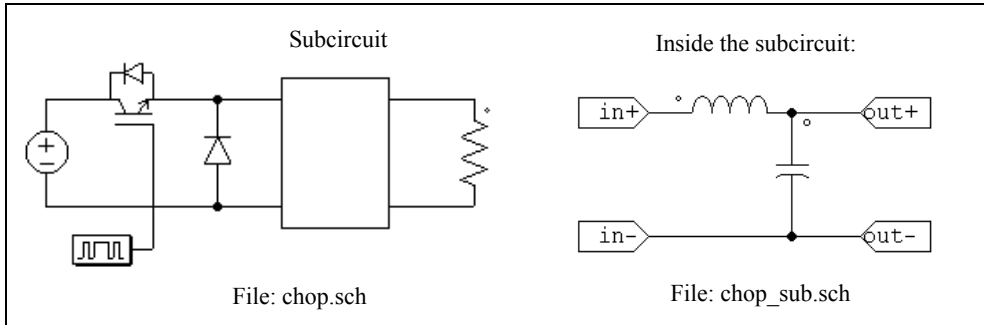
<b>New Subcircuit</b>	To create a new subcircuit
<b>Load Subcircuit</b>	To load an existing subcircuit. The subcircuit will appear on the screen as a block.
<b>Edit Subcircuit</b>	To edit the size and file name of the subcircuit
<b>Set Size</b>	To set the size of the subcircuit
<b>Place Port</b>	To place the connection port between the main circuit and the subcircuit
<b>Display Port</b>	To display the connection port of the subcircuit
<b>Edit Default Variable List</b>	To edit the default variable list of the subcircuit
<b>Edit Image</b>	To edit the subcircuit image
<b>Display Subcircuit Name</b>	To display the subcircuit name
<b>Show Subcircuit Ports</b>	To display the port names of the subcircuit in the main circuit
<b>Hide Subcircuit Ports</b>	To hide the port names of the subcircuit in the main circuit
<b>Subcircuit List</b>	To list the file names of the main circuit and the subcircuits
<b>One Page up</b>	To go back to the main circuit. The subcircuit is automatically

saved.

### Top Page

To jump from a lower-level subcircuit to the top-level main circuit. This is useful for circuits with multiple layers of subcircuits.

The one-quadrant chopper circuit below illustrates the use of the subcircuit.



## 6.4.1 Creating Subcircuit - In the Main Circuit

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

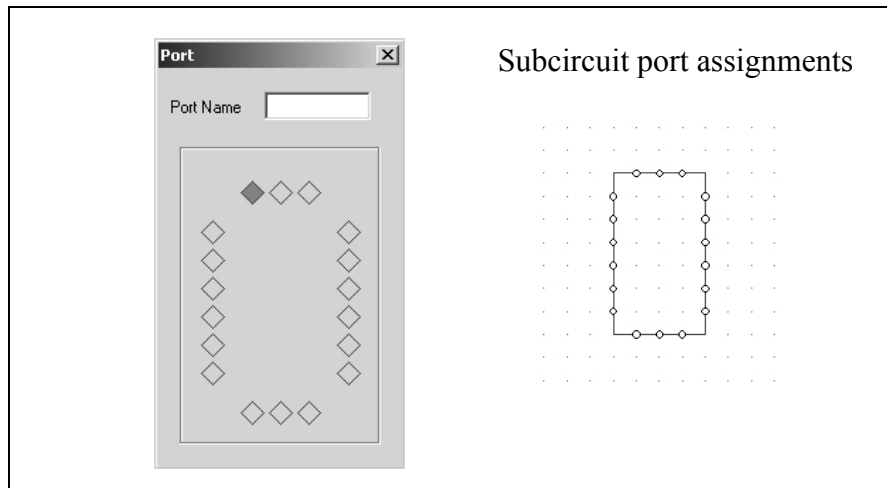
- Open or create the main circuit “chop.sch”.
- If the file “chop\_sub.sch” does not exist, go to the **Subcircuit** menu, and select **New Subcircuit**. If the file exists, select **Load Subcircuit** instead.
- A subcircuit block (rectangle) will appear on the screen. Place the subcircuit.

## 6.4.2 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.



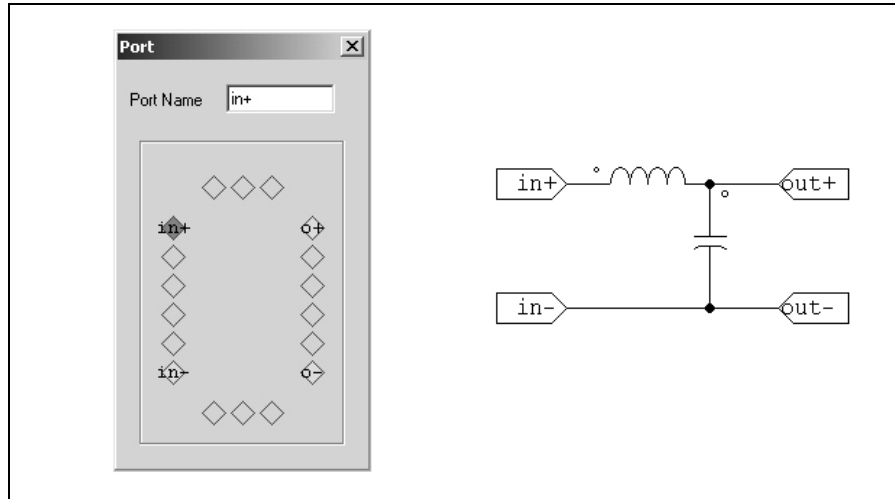
The diamonds on the four sides represent the connection nodes and the positions of the subcircuit. They correspond to the connection nodes of the subcircuit block on the right. There are no diamonds at the four corners since connections to the corners are not permitted.

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 in PSIM as shown below.



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

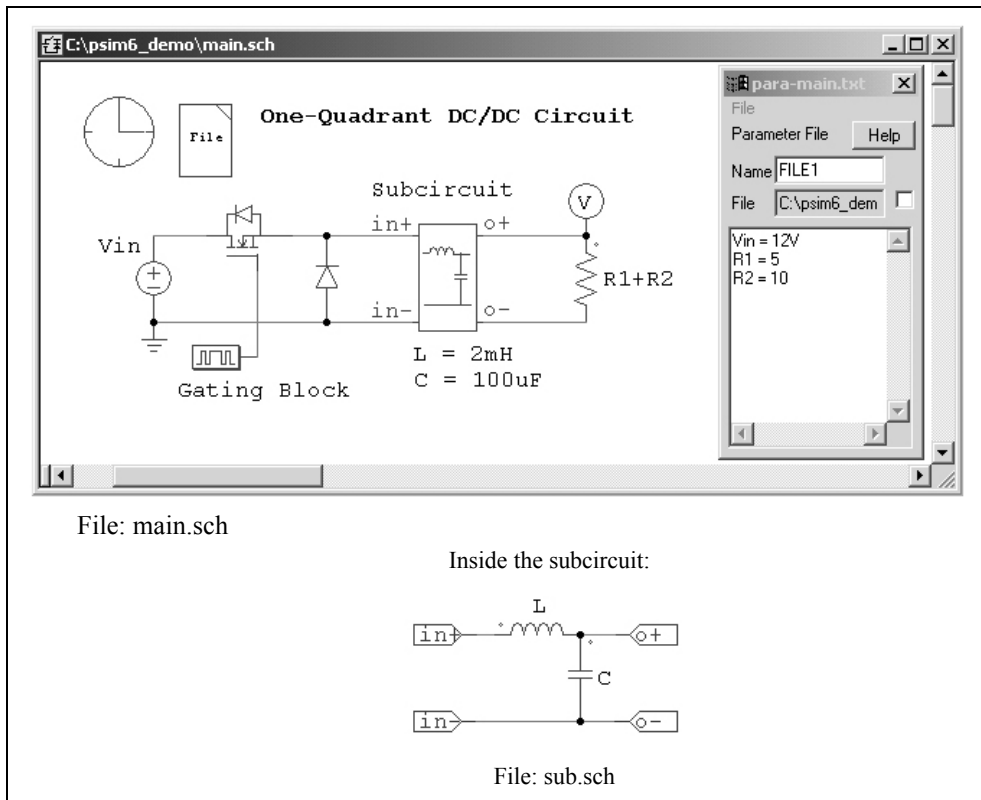
### 6.4.3 Connecting Subcircuit - In the Main Circuit

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.

### 6.4.4 Other Features of the Subcircuit

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



#### 6.4.4.1 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.

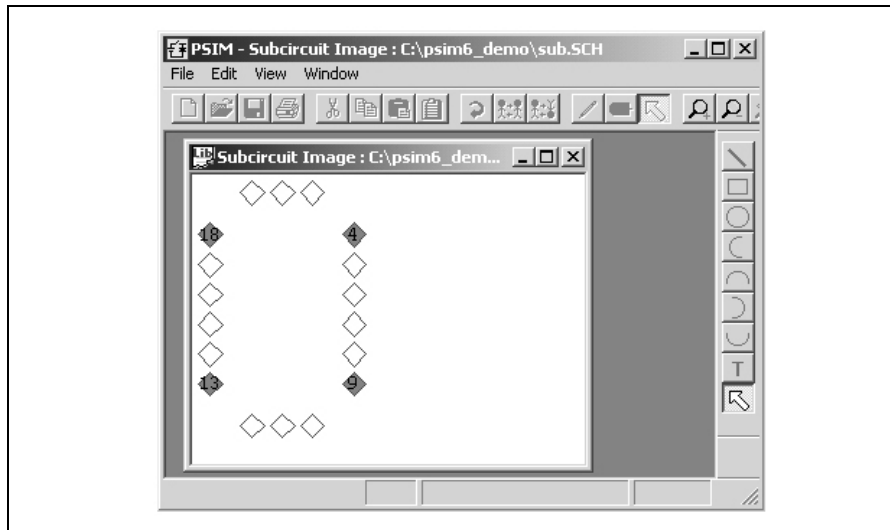
In the case where the same subcircuit is used several times in one main circuit, different parameters can be assigned to the same variable. For example, if the subcircuit “sub.sch” is used two times in above example, in one subcircuit L can be defined as 3mH, and in another subcircuit L can be defined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

#### 6.4.4.2 Customizing the Subcircuit Image

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

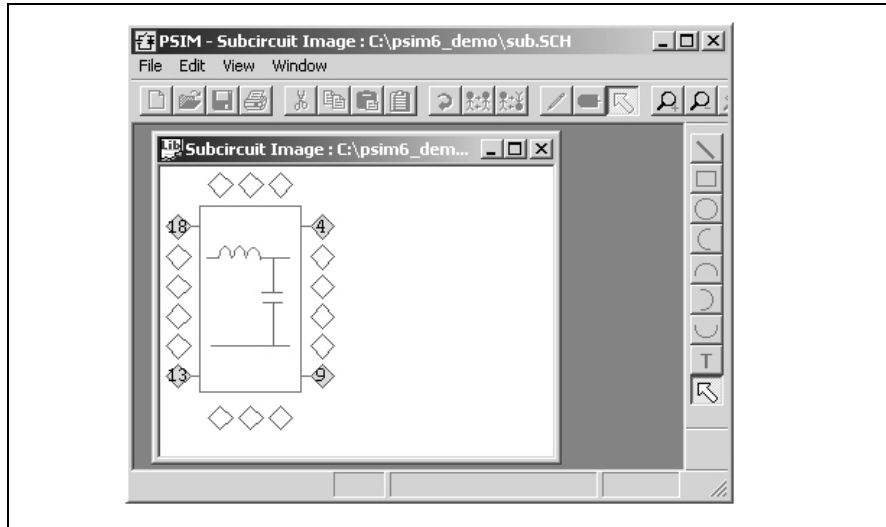
- In the subcircuit, select **Edit Image** in the **Subcircuit** menu. A window will pop-up, as shown below.



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.



- 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.

#### 6.4.4.3 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.

## 6.5 Running the Simulation

To run the simulation, choose **Run PSIM** from the **Simulate** menu. This will start the PSIM simulation.

To view the simulation results, choose **Run SIMVIEW** from the **Simulate** menu.

To view the simulation results in the middle of the simulation, one can either go to **Simulate -> Runtime Graphs** to select the waveforms, or use the voltage/current scopes (under **Elements -> Other -> Probes**) to view the waveforms.

The difference between the runtime graphs and the voltage/current scopes is that only waveforms that are saved for display in SIMVIEW (such as voltage probes, current probes, current flags, etc.) are available for the runtime graphs. In addition, 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.

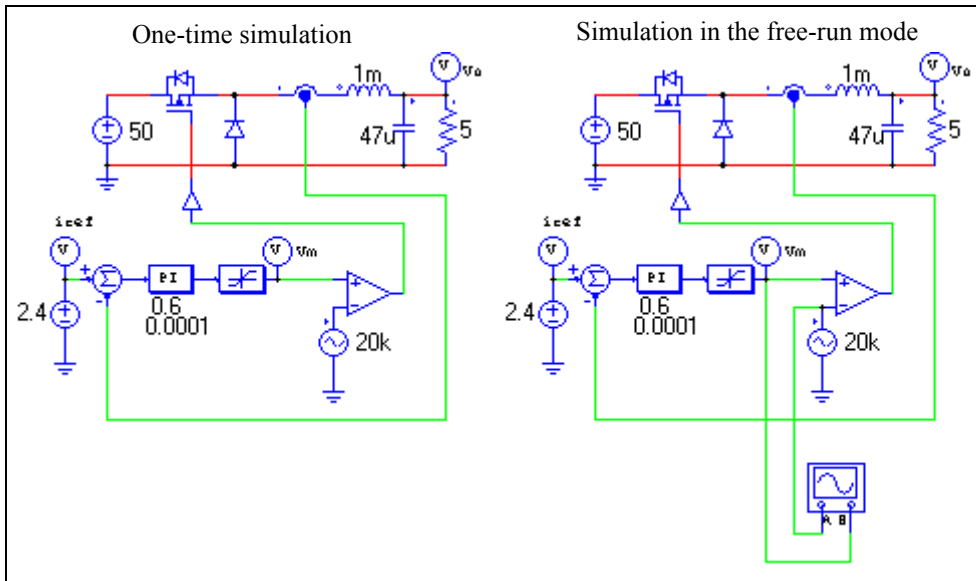
On the other hand, voltage/current scopes can be used in either the one-time simulation mode or in the free-run mode. Voltage scopes can be connected to any nodes, and will display the node-to-ground voltage waveforms. On the other hand, current scopes are available to elements that have current flags (such as R-L-C branches and switches).

Moreover, 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.

### **Running Simulation in the Free-Run Mode:**

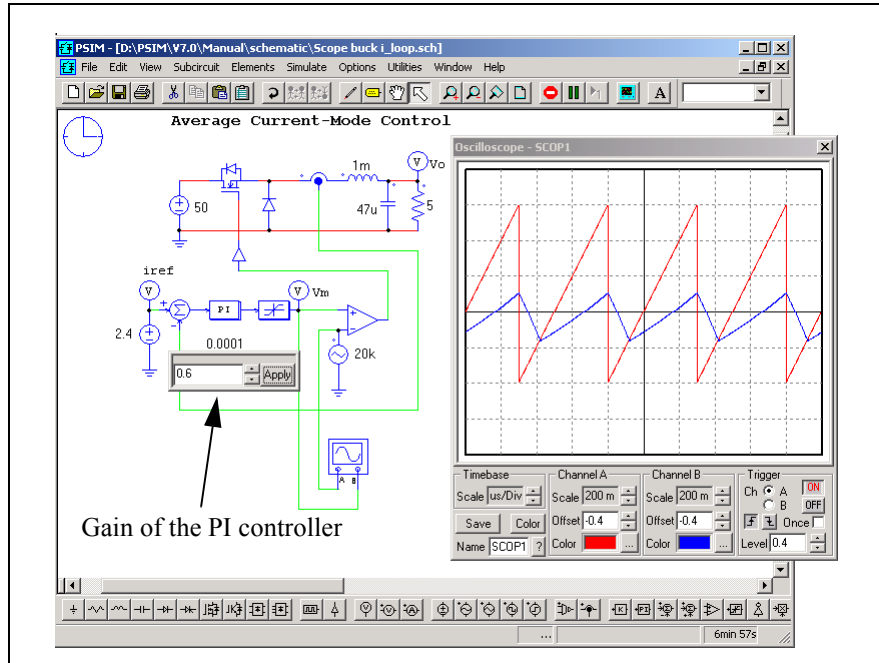
To illustrate how to run a simulation in the free-run mode, a buck converter circuit shown below is used as an example. The circuit on the left was originally set up for the one-time simulation, with the total simulation time set to a specific value.





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.



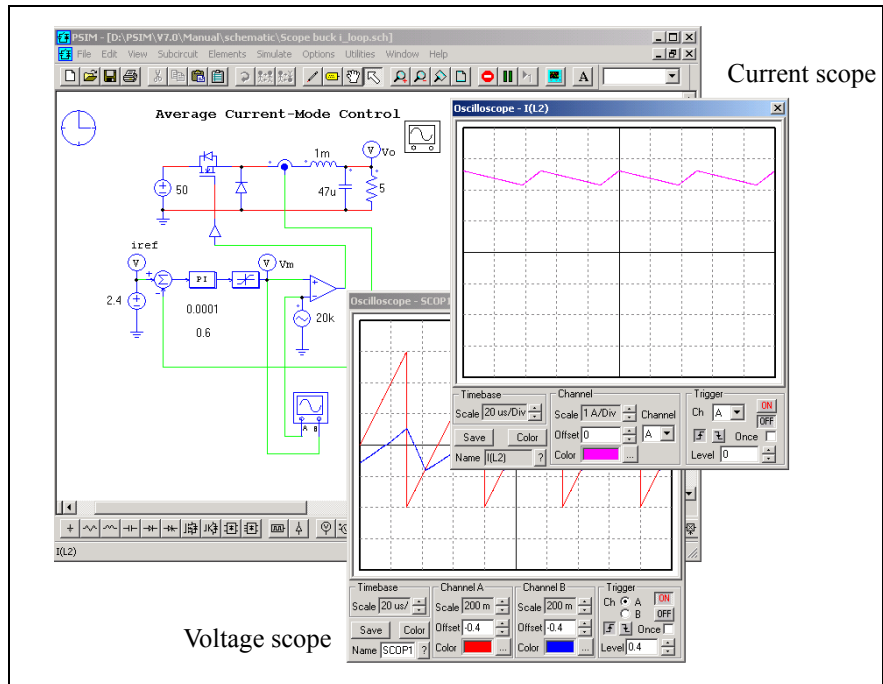
- Place the cursor inside the input field of the dialog window for the gain, and change the gain either by pressing on the upper/down arrow keys on the keyboard, or by entering a new value and then clicking on **Apply**. Watch how the waveforms change as the gain is changed.

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 with the Command-Line Option:**

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

```
PsimCmd -i "c:\psim\examples\chop.sch" -o "c:\psim\examples\chop.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

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

## 6.6 Managing the PSIM Library

The PSIM library consists of two parts: one is the image library (psimimage.lib) and the other is the netlist library (psim.lib). The netlist library can not be modified. But users can modify the image library, or create their own image library.

To create or modify the image library, go to **Edit -> Edit Library -> Edit Library Files**, and follow the instructions on the screen. Any image libraries in the PSIM directory will be automatically loaded into PSIM.

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.

Below are the procedures to add the custom models to the PSIM library.

### 6.6.1 Adding a New Subcircuit Element into the Library

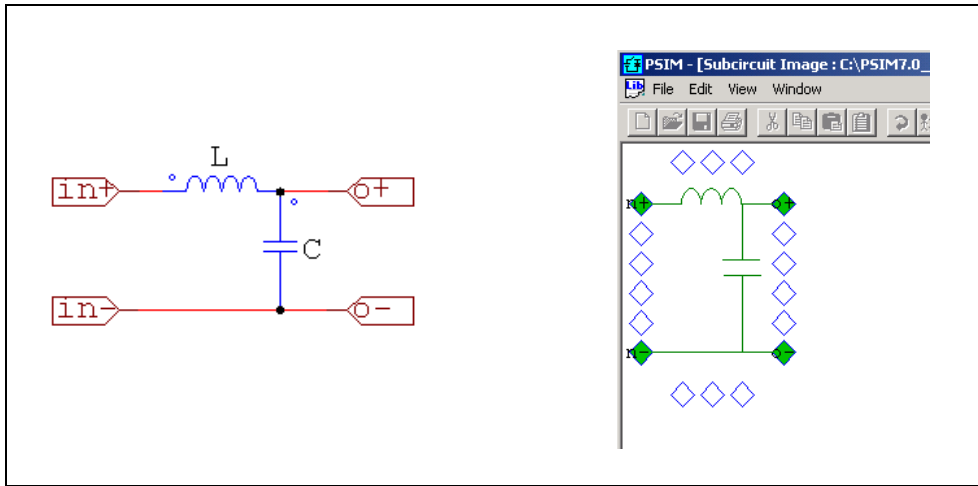
There are three main steps to add a new element, modeled in a subcircuit, into the PSIM library:

- 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.

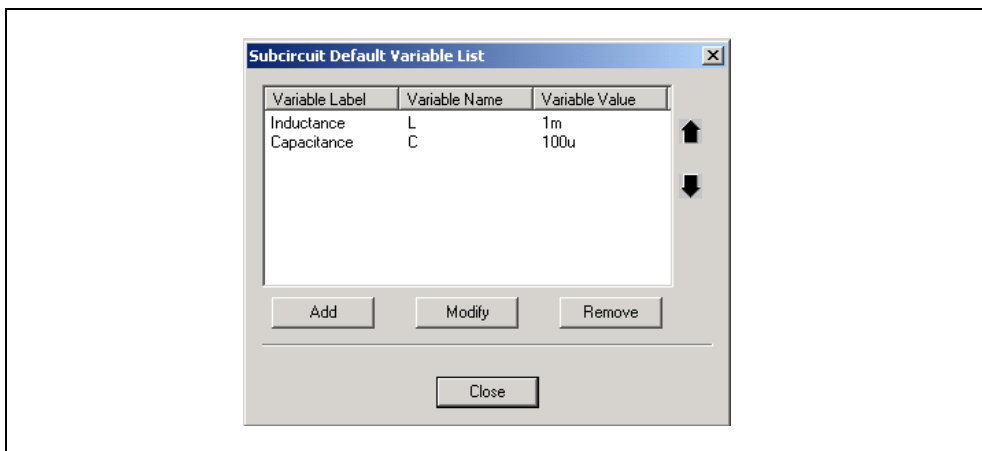
#### Creating the Subcircuit:

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  $C$ .

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 appear as follows.



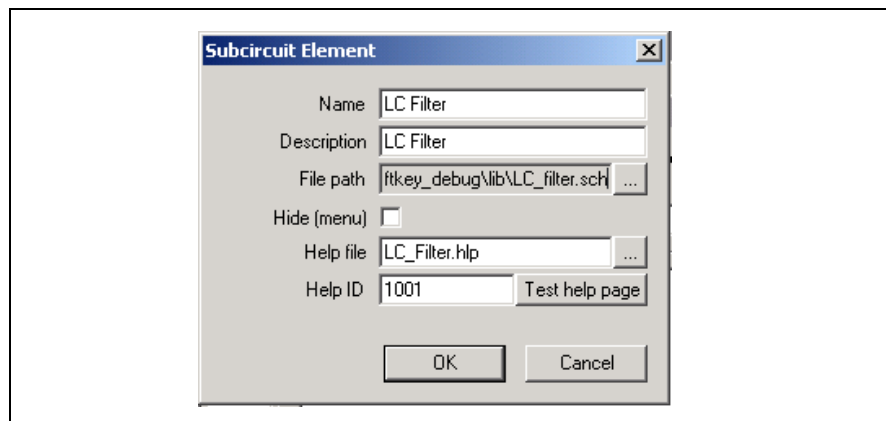
Here **Variable Label** is the text that describes the parameter, **Variable Name** is the variable that is used as the parameter value in the subcircuit, and **Variable Value** is the

default value of the parameter. For example, for the inductance  $L$ , the **Variable Label** is *Inductance*, the **Variable Name** is  $L$ , and the **Variable Value** is 1m.

### **Adding 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 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 Subcircuit**. Enter the information to the dialog window as shown below:



The explanation of each field is as follows:

- |              |   |
|--------------|---|
| 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 subcircuit file must be placed in the \lib sub-folder in the PSIM directory. |
| 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.   |
| Help ID:     | The ID used in the help map file to link the designated help page.  |

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

### **Creating the On-Line Help File for the New Element:**

An on-line help file can be created for this new element, so that when the **Help** button is clicked in the property dialog window, the corresponding help page will come up. The following is the procedure to create the help file using Microsoft Help Workshop.

- Prepare the help content in the .rtf file "LC\_Filter.rtf" using Microsoft Word. Note that the footnote of the file should look as follows:

```
#LC_FILTER
$LC_FILTER
kLC_FILTER; LC filter
```

Here the # label is used to link the help page with the LC filter element in PSIM. The k label will appear in the Index in the Help file, and the \$ label is not used.

- Prepare the help map file "HelpMap.txt". It is a text file, and has the following content:

```
LC_FILTER=1001
```

Here the text "LC\_FILTER" must be the same as the # label in the "LC\_Filter.rtf" file, and the number "1001" must be the same as the Help ID defined for the LC filter element when it was created.

- In the Help Workshop project file "LC\_Filter.hpj", have the following settings:
  - Click on the **Files** button, and add the file "LC\_Filter.rtf".
  - Click on the **Options** button, and in the **Files** tab, set *Help File* as *LC\_Filter.hlp*.
  - Click on the **Map** button, and then click on **Include** to include the map file *HelpMap.txt*.
- Click on **Save and Compile** to create the help file "LC\_Filter.hlp". Place this file in the PSIM directory.

## **6.6.2 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:**

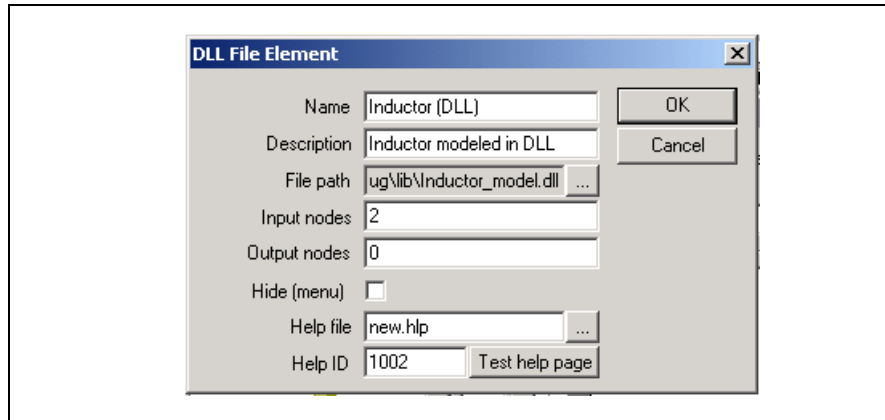
The first step is to create the inductance model in DLL. Please refer to the relevant section on how to create a custom DLL.

Here we assume that the DLL file, "inductor\_model.dll", has already been created using the Power Modeling Block. It has one parameter called "Inductance", and two connecting nodes.

### **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:



The explanation of each field is as follows:

- |              |  |
|--------------|--|
| 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. For the  |



Power Modeling Block, this value is the number of total terminals and is read from the DLL file automatically.

Output Nodes: Number of output terminals of the new element. For the Power Modeling Block, this value is 0.

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.

Help ID: The ID used in the help map file to link the designated help page.

- 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".

The process of creating the on-line help file is the same as described in the previous section.

## 6.7 Other Options

### 6.7.1 Generate and View the Netlist File

To generate the netlist, choose **Simulate -> Generate Netlist File**. This will create the netlist file with the ".cct" extension. The netlist file will be saved to the same directory as the schematic file.

To view the netlist file, choose **Simulate -> View Netlist File**.

### 6.7.2 Set Path

The **Set Path** function in the **Options** menu 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.

PSIM searches the DLL files in the following order:

- PSIM directory
- Schematic file directory

- Directories as defined in the **Set Path** function.

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

### 6.7.3 Settings

Grid display, rubber band feature, text fonts, simulation warning, and colors can be set in the **Settings...** in the **Option** menu.

Before a circuit is printed, its position on the paper can be viewed by selecting **Print Page Border** in the **Settings...** option. If a circuit is split into two pages, it can be moved into one single page. If the circuit is too big to fit in one page, one can zoom out and reduce the circuit size by clicking the **Zoom Out** button.

Print page legend, such as company name, circuit title, designer's name, date, etc., can be specified by choosing **Print Page Setup** in the **File** menu. It can be disabled in the **Settings...** option.

Also in the **Settings...** option, if **Disable simulation warning messages** is checked, warning messages generated during the simulation will be suppressed. Otherwise, warning messages will be shown before waveforms are displayed in SIMVIEW.

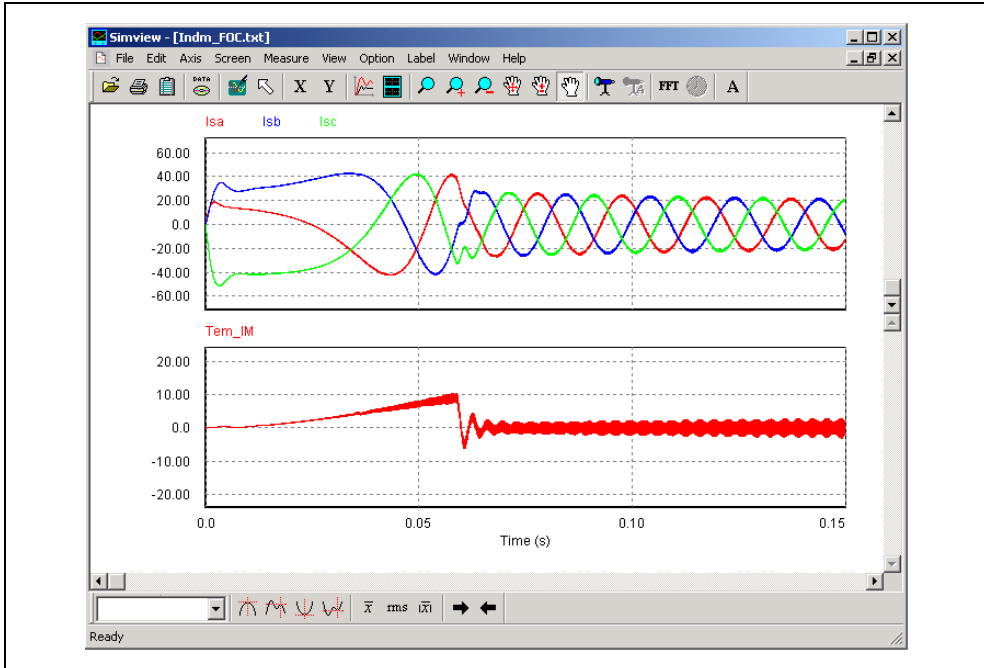
### 6.7.4 Printing the Circuit Schematic

The circuit schematic can be printed from a printer by choosing **Print** in the **File** menu. It is also possible to print the selected region of a circuit by choosing **Print Selected**.

The schematic can also be saved to the clipboard which can be imported into a word processor (such as Microsoft Word). By default, the schematic image is saved in monochrome in order to save memory space. One can save the image in color by selecting **Edit/Copy to Clipboard/Color**.

## Waveform Processing

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	Ilsa	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.

## 7.1 File Menu

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

<b>Open</b>	Load a data file in ASCII text format (with .txt extension) or SIMVIEW binary format (with .smv extension)
<b>Merge</b>	Merge another data file with the existing data file for display
<b>Re-Load Data</b>	Re-load data from the same text file
<b>Save As</b>	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.
<b>Print</b>	Print the waveforms
<b>Print Setup</b>	Set up the printer
<b>Print Page Setup</b>	Set up the hardcopy printout size
<b>Print Preview</b>	Preview the printout
<b>Exit</b>	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.

By using 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\_1” automatically.

## 7.2 Edit Menu

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

<b>Undo</b>	Go back to the previous X and Y axis settings
<b>Copy to Clipboard</b>	Copy the waveforms to the clipboard

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.

## 7.3 Axis Menu

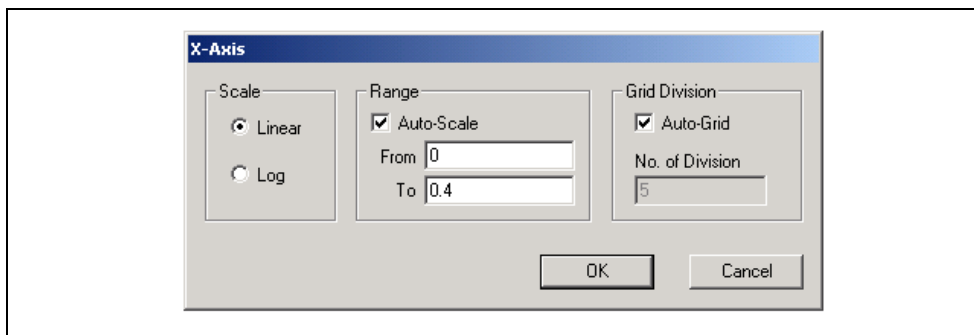
The **Axis Menu** has the following functions:

**X Axis**                      Change the settings of the X axis

**Y Axis**                      Change the settings of the Y axis

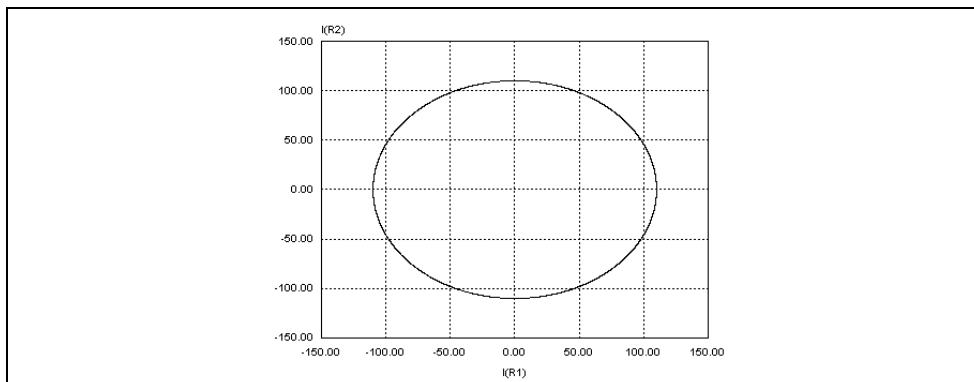
**Choose X-Axis Variable**    By default, the first column of the data is selected as the X axis. However, other columns can also be selected as the X axis through this function.

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



If the *Auto-Scale* and *Auto-Grid* boxes are checked, the maximum data range will be selected and the number of axis divisions will be automatically determined. Both the data range and grid division, however, can be manually set.

By default, the first column of the data, which is usually Time, is used as the X axis. However, any other column of the data can be used as the X axis. For example, the following figure shows a sine waveform as the X axis versus a cosine waveform in the Y axis.



## 7.4 Screen Menu

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

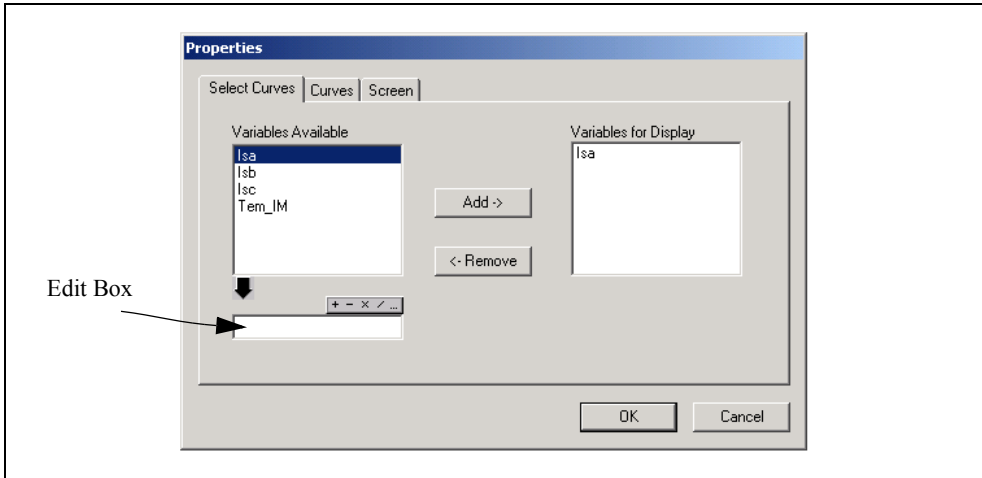
**Add/Delete Curves** Add or delete curves from the selected screen

**Add Screen** Add a new screen

**Delete Screen** Delete the selected screen

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 data 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, an mathematical expression 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	average function
INT	integration function

Type this expression in the Edit Box, and click on **Add** ->. Highlight the expression on the right, click on <- **Remove**, and the expression will be moved into the Edit Box for further editing.

Also in the property dialog window, in the **Curves** tab, the curve properties, such as color, line thickness, and marker symbol, can be defined.

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

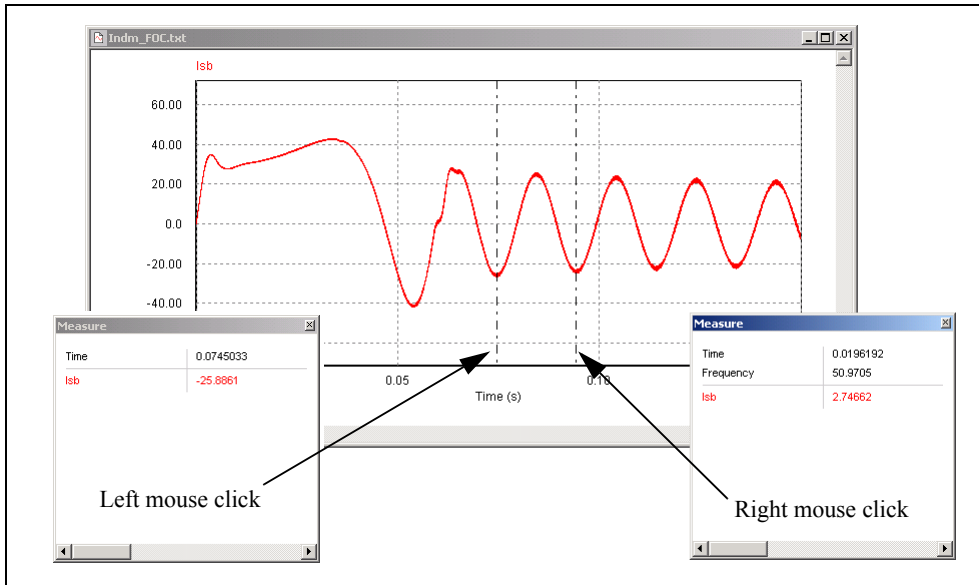
## 7.5 Measure Menu

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

<b>Measure</b>	Enter the measure mode.
<b>Max</b>	Find the global maximum of a selected curve
<b>Min</b>	Find the global minimum of a selected curve
<b>Next Max</b>	Find the next local maximum of a selected curve
<b>Next Min</b>	Find the next local minimum of a selected curve
<b>Avg</b>	Calculate the average of a selected curve within the selected time
<b>Avg( x )</b>	Calculate the average of the absolute value of a selected curve within the selected time
<b>rms</b>	Calculate the rms value of a selected curve within the selected time

A region is selected by pressing the left button of the mouse and, at the same time, drag the mouse.

The **Measure** function allows the measurement of waveforms. After **Measure** is selected, the measurement dialog box 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 measurement boxes in these two modes are shown below.



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.

## 7.6 View Menu

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

- |                |  |
|----------------|--|
| <b>Zoom</b>    | To zoom into a selected region               |
| <b>Re-Draw</b> | To re-draw the waveform using the auto-scale |
| <b>Escape</b>  | To escape from the Zoom or Measure mode      |



<b>Standard Toolbar</b>	To enable/disable standard toolbar
<b>Measure Toolbar</b>	To enable/disable measure toolbar
<b>Status Bar</b>	To enable/disable status bar

## 7.7 Option Menu

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

<b>FFT</b>	Perform the Fast Fourier Transform analysis
<b>Time</b>	Switch from the frequency spectrum display to time domain display
<b>Grid</b>	Enable or disable the grid display
<b>Color</b>	Set the curves to be either Color (default) or Black and White

By selecting **FFT**, the harmonic amplitudes of time domain waveforms can be calculated and displayed. Note that, in order to obtain correct FFT results, the simulation should reach the steady state, and the simulation data should be restricted (using the manual range setting in the **X Axis** function) to have the integer number of the fundamental period.

## 7.8 Label Menu

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

<b>Text</b>	Place text on the screen
<b>Line</b>	Draw a line
<b>Dotted Line</b>	Draw a dotted line
<b>Arrow</b>	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.

## 7.9 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**.

## Error/Warning Messages and Other Simulation Issues

### 8.1 Simulation Issues

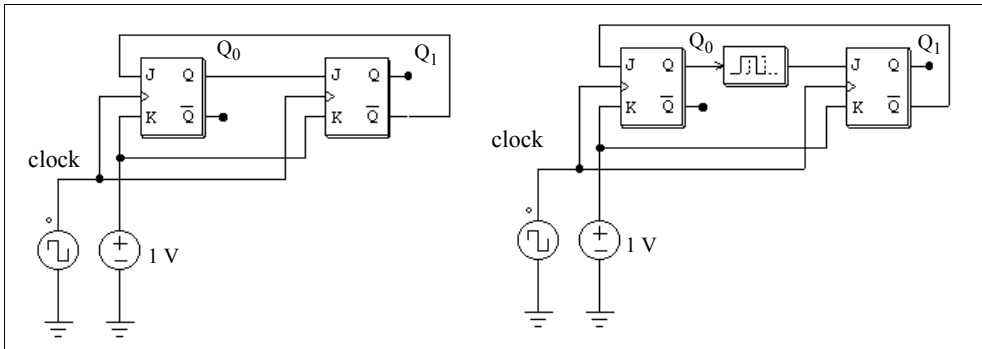
#### 8.1.1 Time Step Selection

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.

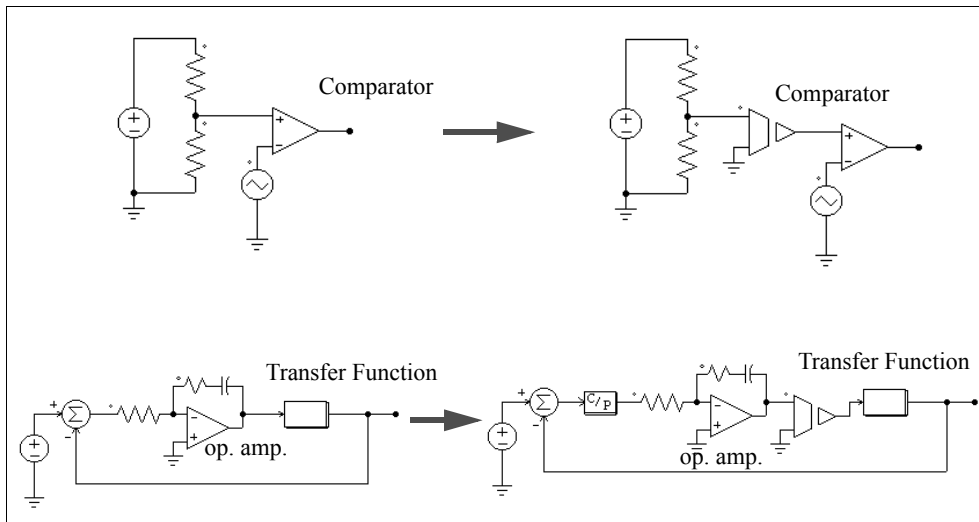
To prevent this, a time delay element with the delay period of one time step needs to be

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,  $\Delta t$ , and the data length  $T_{\text{length}}$ . The data point interval  $\Delta t$  is equal to the simulation time step multiplied by the print step. In the FFT results, the frequency incremental step will be  $1/T_{\text{length}}$ , and the maximum frequency will be  $1/(2*\Delta t)$ .

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  $\mu\text{s}$ , that is,  $T_{\text{length}} = 1 \text{ ms}$ , and  $\Delta t = 10 \text{ us}$ , the frequency incremental step will be:  $\Delta f = 1/T_{\text{length}} = 1 \text{ kHz}$ . The maximum frequency will be:  $f_{\text{max}} = 1/(2*\Delta t) = 50 \text{ kHz}$ .

## 8.2 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.

## 8.3 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.

