PSIM

User’s Guide

Powersim Inc.

PSIM User's Guide

Version 6.0

June 2003

Disclaimer

Powersim Inc.

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

Contents

7 Waveform Processing

8 Error/Warning Messages and Other Simulation Issues

1

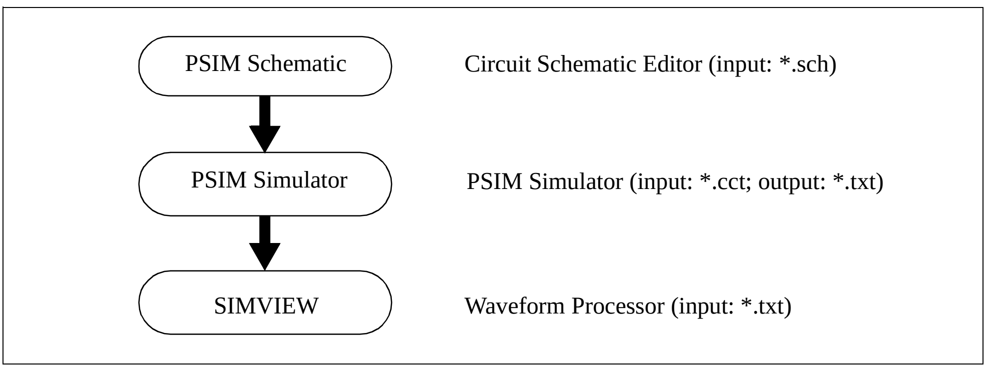
General Information

1.1 Introduction

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

This manual covers both PSIM[1] and its three add-on Modules: Motor Drive Module, Digital Control Module, and SimCoupler Module. The Motor Drive Module has built-in machine models and mechanical load models for 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 analysis. The SimCoupler Module provides interface between PSIM and Matlab/Simulink[2] for co-simulation.

The PSIM simulation package consists of three programs: circuit schematic program PSIM, PSIM simulator, and waveform processing program SIMVIEW[1]. The simulation environment is illustrated as follows.



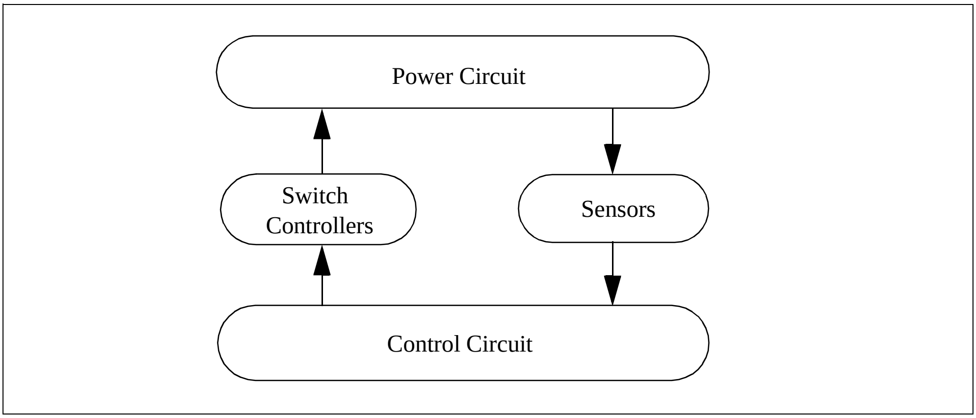
Chapter 1 of this manual describes the circuit structure, software/hardware requirement, and parameter specification format. Chapter 2 through 4 describe the power and control

1. PSIM and SIMVIEW are copyright by Powersim Inc., 2001-2003
2. Matlab and Simulink are registered trademarks of the MathWorks, Inc.

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 measure power circuit voltages and currents and pass the values 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 Requiremen

PSIM runs in Microsoft Windows environment 98/NT/2000/XP on personal computers. The minimum RAM memory requirement is 32 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.dll	PSIM simulator
psim.exe	PSIM circuit schematic editor
simview.exe	Waveform processor SIMVIEW
psim.lib, psimimage.lib	PSIM libraries
*.hlp	Help files
*.sch	Schematic files

File extensions used in PSIM are:

*.sch	PSIM schematic file (binary
*.cct	PSIM netlist file (text)
*.txt	PSIM simulation output file (text)
*.fra	PSIM ac analysis output file (text)
*.smv	SIMVIEW waveform file (binary)

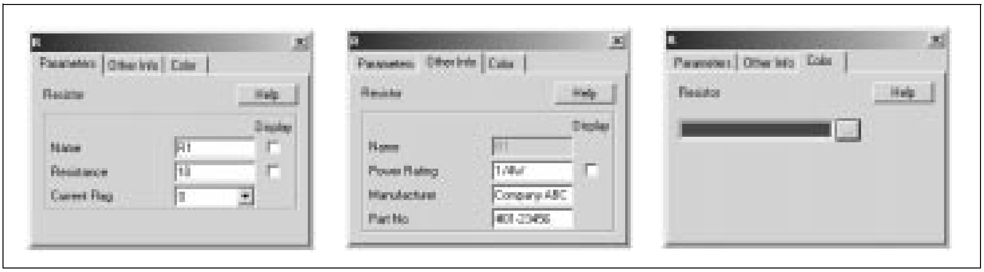
1.5 Simulating a Circuit

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

- Start PSIM. Choose **Open** from the **File** menu to load the file “chop.sch”.
- From the **Simulate** menu, choose **Run PSIM** to start the simulation. The simulation results will be saved to File “chop.txt”. Any warning messages occurred in the simulation will be saved to File “message.doc”.
- 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 **Auto-run SIMVIEW** 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.Oh
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
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]

2

Power Circuit Components

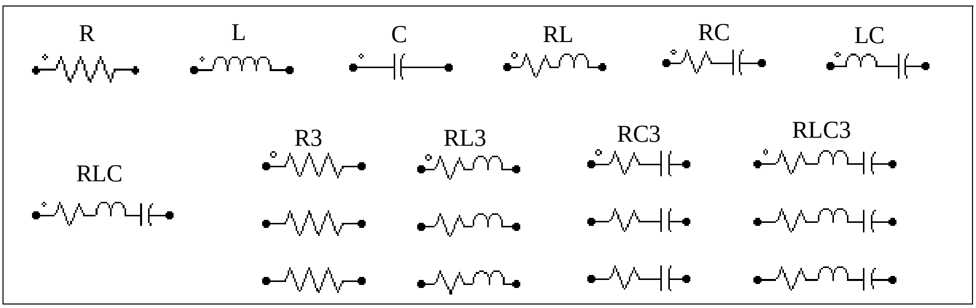
2.1 Resistor-Inductor-Capacitor Branches

2.1.1 Resistors, Inductors, and Capacitors

Both individual resistor, inductor, capacitor branches 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:



The names above the element images are the netlist names of the elements. For example, a resistor appears as “Resistor” in the library menu, and the netlist name is “R”.

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

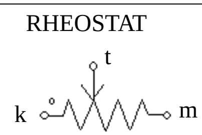
Parameters	Description
Initial Current	Initial inductor current, in A
Initial Cap.Voltage	Initial capacitor voltage, in
Current Flag	Flag for branch current output. If the flag is zero, there is no current output. If the flag is 1, the current will be saved to the output file for display in SIMVIEW. The current is positive when it flows into the dotted terminal of the branch.
Current Flag_A; Current Flag_B; Current Flag_C	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:

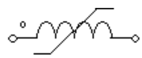
Parameters	Description
Total Resistance	Total resistance of the rheostat *R* (between Node k and m), in Ohm
Tap Position (0 to 1)	The tap position *Tap*. The resistance between Node k and t is: *R***Tap*.
Current Flag	Flag for the current that flows into Node k.

2.1.3 Saturable Inductor

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

Image:

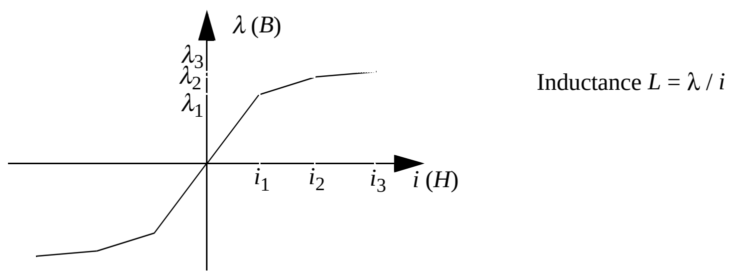
L_SAT



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 λ and the magnetizing force *H* is proportional to the current *i*, the *B*-*H* curve can be represented by the λ-*i* curve instead, as shown below.



The inductance is defined as: $L = \lambda / i$, which is the slope of the λ-*i* curve at different points. The saturation characteristics can then be expressed by pairs of data points as: (i_1, L_1), (i_2, L_2), (i_3, L_3), etc.

2.1.4 Nonlinear Elements

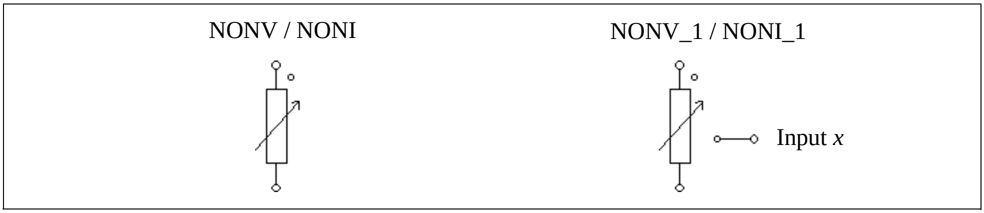
Four elements with nonlinear voltage-current relationship are provided:

- Resistance-type (NONV) [$v = f(i)$]
- Resistance-type with additional input *x* (NONV_1) [$v = f(i,x)$]
- Conductance-type (NONI $i = f(v)$]

- Conductance-type with additional input *x* (NONI_1) [$i = f(v,x)$]

The additional input *x* must be a voltage signal.

Images:



Attributes:

For resistance-type elements:

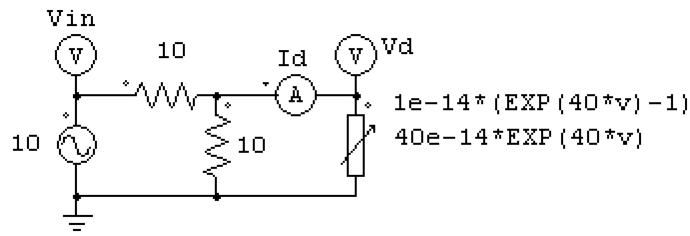
Parameters	Description
Expression $f(i)$ or $f(i,x)$	Expression $v = f(i)$ for NONV and $v = f(i,x)$ for NONV_1
Expression df/di	The derivative of the voltage *v* versus current *i*, i.e. $df(i)/di$
InitialValue i_o	The initial value of the current *i*
Lower Limit of *i*	The lower limit of the current *i*
Upper Limit of *i*	The upper limit of the current *i*

For conductance-type elements:

Parameters	Description
Expression $f(v)$ or $f(v,x)$	Expression $i = f(v)$ for NONI and $i = f(v,x)$ for NONI_1
Expression df/dv	The derivative of the current *i* versus voltage *v*, i.e. $df(v)/dv$
InitialValue v_o	The initial value of the voltage *v*
Lower Limit of *v*	The lower limit of the voltage *v*
Upper Limit of *v*	The upper limit of the voltage *v*

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

Example: Nonlinear Diode



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

Expression *f*(*v*)	1e-14*(EXP(40*v)-1)
Expression *df*/*dv*	40e-14*EXP(40*v)
InitialValue v_o	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 switchmode. It operates either in the cut-off region (off state) or saturation region (on state). The other is linear. It can operates in either cut-off, linear, or saturation region.

Switches in switchmode include the following:

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

- Bi-directional switch (SSWI)

The names inside the bracket are the netlist names used in PSIM.

Switch models in PSIM are ideal. That is, both turn-on and turn-off transients are neglected. A switch has an on-resistance of 10 μΩ and an off-resistance of 1M Ω. Snubber circuits are not required for switches.

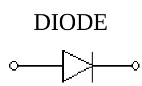
Linear switches include the following:

- npn bipolar junction transistor (NPN_1)
- pnp bipolar junction transistor (PNP_1)

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:

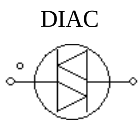


Attributes:

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 open. If it is 1, the diode is closed.
Current Flag	Flag for the diode current output. If the flag is 0, there is no current output. If the flag is 1, the diode current will be saved to the output file for display in SIMVIEW.

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:

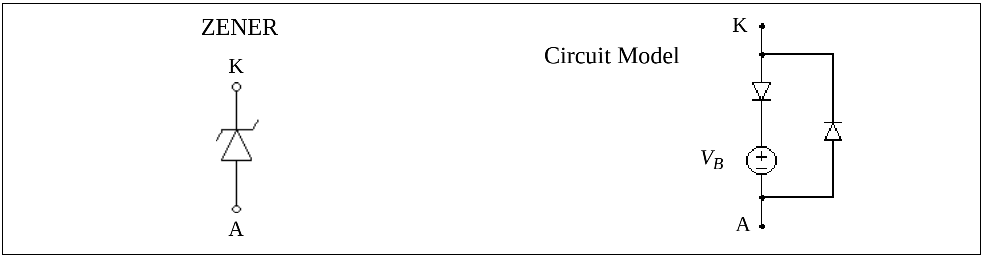


Attributes:

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)
Current Flag	Flag for zener current output (from anode to cathode)

If the zener diode is positively biased, it behaviors as a regular diode. When it is reverse biased, it will block the conduction as long as the cathode-anode voltage V_{KA} is less than the breakdown voltage V_B. When V_{KA} exceeds V_B, the voltage V_{KA} will be clamped to V_B. [Note: when the zener is clamped, since the diode is modelled with an on-resistance of 10μΩ, 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 behaviors in the same way as two thyristors in the opposite direction connected in parallel.

Images:

Attributes:

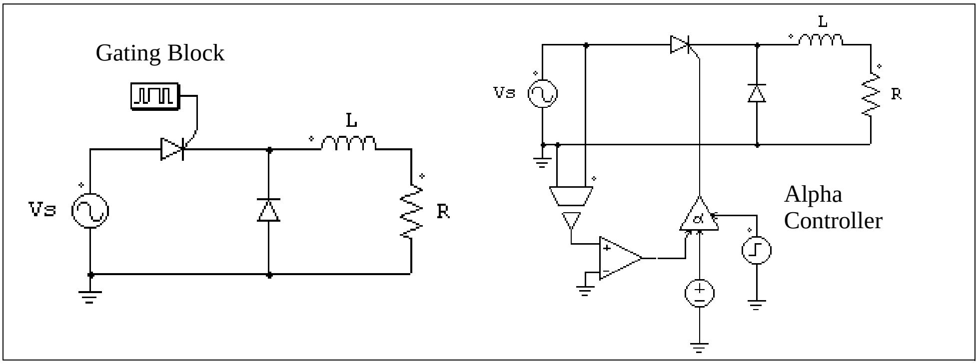
Parameters	Description
Voltage Drop	Thyristor conduction voltage drop, in
Holding Current	Minimum conduction current below which the device stops conducting and returns to the OFF state (for THY only)
Latching Current	Minimum ON state current required to keep the device in the ON state after the triggering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

TRIAC 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 (GATING), and the other is to use a switch controller. The gate node of a thyristor or TRIAC, therefore, 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 (see Section 2.2.5). The switching gating pattern and the frequency are pre-defined, and will remain unchanged throughout the simulation. The circuit on the right uses an alpha controller (see Section 4.5.2). The delay angle alpha, in deg., is specified through the dc source in the circuit.

2.2.3 GTO, Transistors, and Bi-Directional Switch

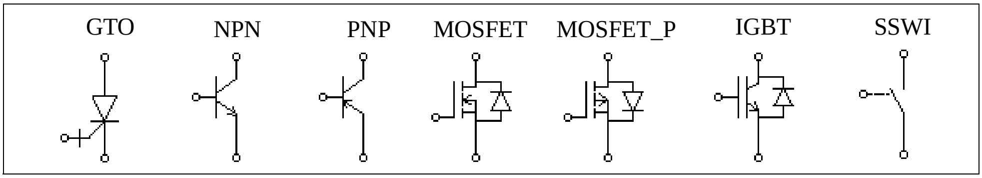
Self-commutated switches in the switchmode, except pnp bipolar junction transistor (BJT) and p-channel MOSFET, are turned on when the gating is high (when a voltage of 1V or higher is applied to the gate node) and the switch is positively biased (collector-emitter or drain-source voltage is positive). It is turned off whenever the gating is low or the current drops to zero. For pnp BJT and p-channel MOSFET, switches are turned on when the gating 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 (SSWI) conducts currents in both directions. It is on when the gating is high and is off when the gating 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 can 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:

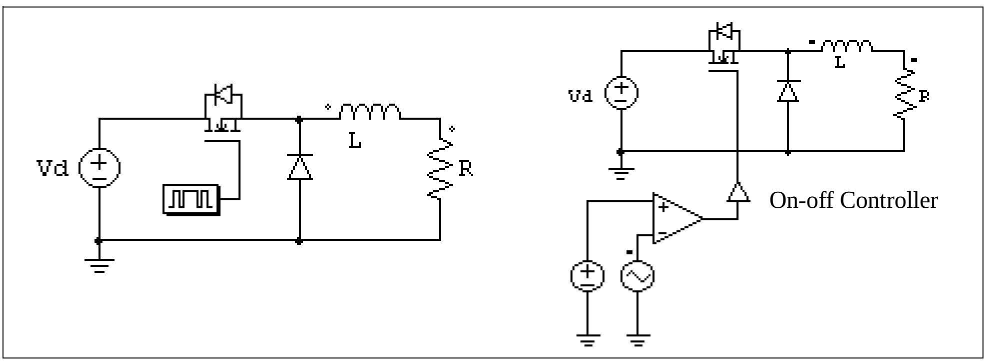


Attributes:

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 (GATING) 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.

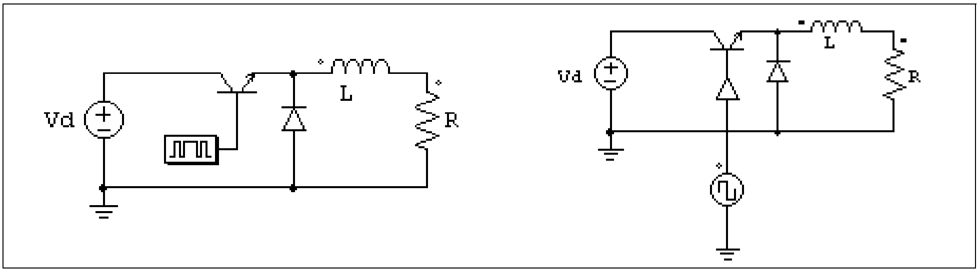
Examples: Control of a MOSFET Switch



The circuit on the left uses a gating block, and the one on the right uses an on-off switch controller (see Section 4.5.1). The gating signal is determined by the comparator output.

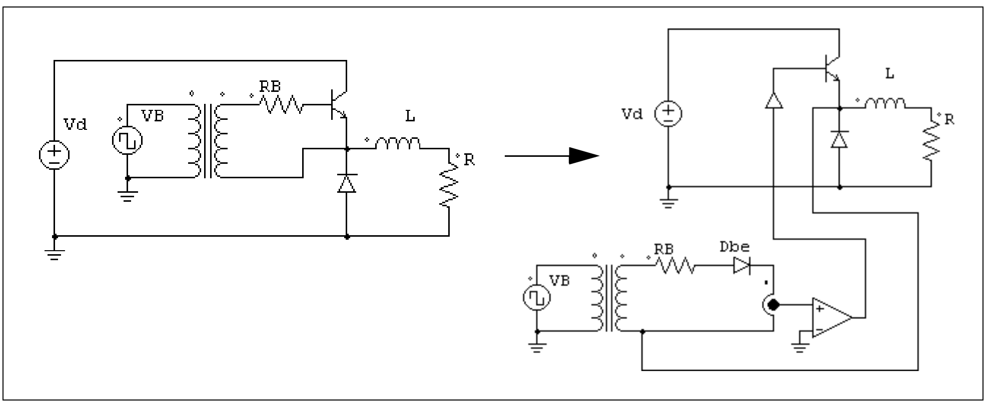
Example: Control of a npn Bipolar Junction Transistor

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 *VB* is applied to the transistor base drive circuit through a transformer, and the base current determines the conduction state of the transistor.

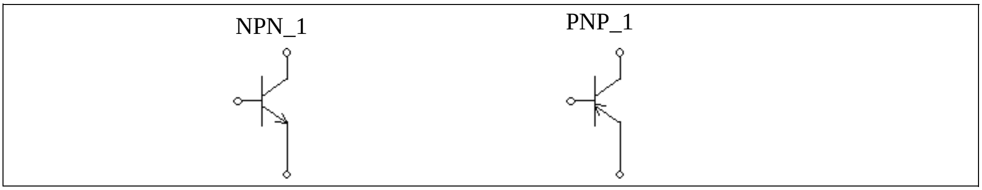
This circuit can be modelled and implemented in PSIM as shown on the right. A diode, D_{be}, with a conduction voltage drop of 0.7V, is used to model the pn junction between the base and the emitter. When the base current exceeds 0 (or a certain threshold value, in which case the base current will be 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 (NPN_1) and pnp bipolar junction transistor (PNP_1). They can operate in either cut-off, linear, or saturation region.

Images:



Attributes:

Parameters	Description
Current Gain beta	Transistor current gain β, defined as: $\beta = I_c/I_b$
Bias Voltage $_r$	Forward bias voltage between base and emitter for NPN_1, or between emitter and base for PNP_1
$V_{ce,sat}$ [or $V_{ec,sat}$ for PNP_1]	Saturation voltage between collector and emitter for NPN_1, and between emitter and collector for PNP_1

A linear BJT switch is controlled by the base current I_b. It can operate in either one of the three regions: cut-off (off state), linear, and saturation region (on state). The properties of these regions for NPN_1 are:

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

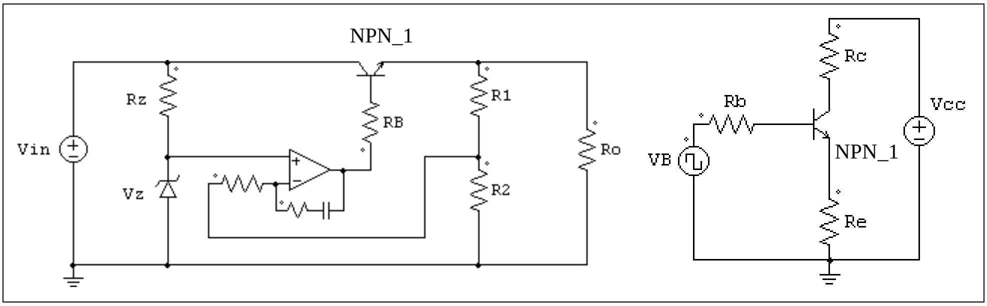
where V_{be} is the base-emitter voltage, $_{ce}$ is the collector-emitter voltage, and $_c$ is the collector current.

Note that for NPN_1 and PNP_1, 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_1 and PNP_1 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 the linear switch. 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.

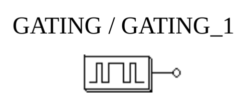


2.2.5 Switch Gating Block

A switch gating block defines the gating pattern of a switch or a switch module. The gating pattern can be specified either directly (with the gating block GATING) or in a text file (with the gating block GATING_1).

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 GATING only)
Switching Points	Switching points, in deg. If the frequency is zero, the switching points is in second. (for GATING only)
File for Gating Table	Name of the file that stores the gating table (for GATING_1 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.

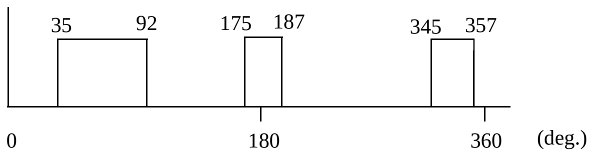
For GATING_1, the file for the gating table must be in the same directory as the schematic file. The gating table file has the following format:

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 GATING for this switch will be:

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

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

If the gating block GATING_1 is used instead, the specification will be:

Frequency	2000.
File for Gating Table	test.tbl

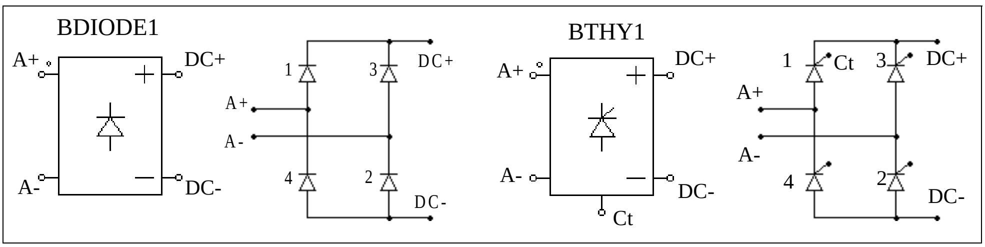
The file “test.tbl” will contain the following:

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

2.2.6 Single-Phase Switch Modules

Built-in single-phase diode bridge module (BDIODE1) and thyristor bridge module (BTHY1) are provided in PSIM. The images and internal connections of the modules are shown below.

Images:



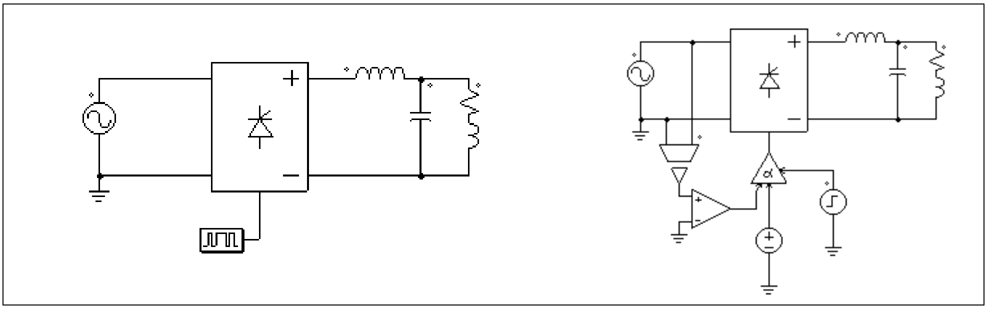
Attributes:

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

Node Ct at the bottom of the thyristor module BTHY1 is the gating control node for Switch 1. For the thyristor module, only the gatings for Switch 1 need to be specified. The gatings for other switches will be derived internally in PSIM.

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

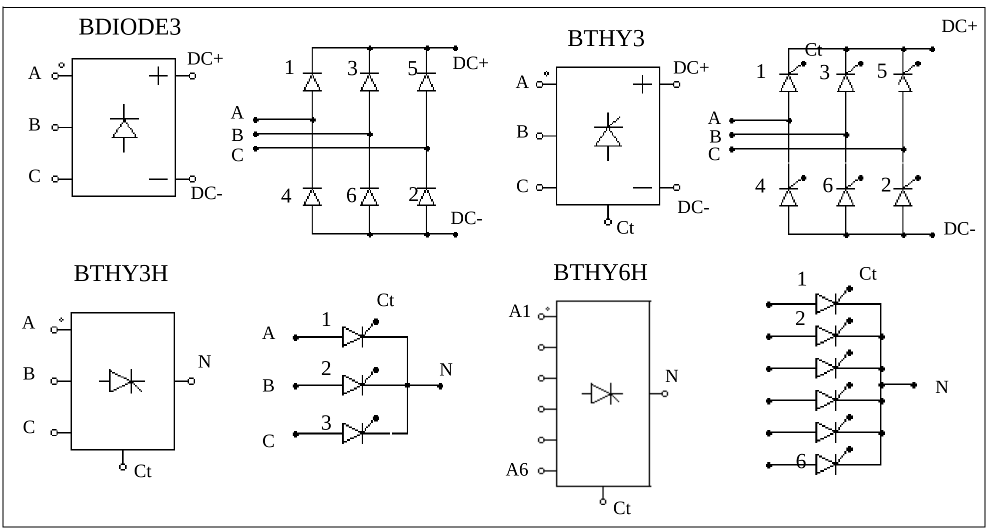


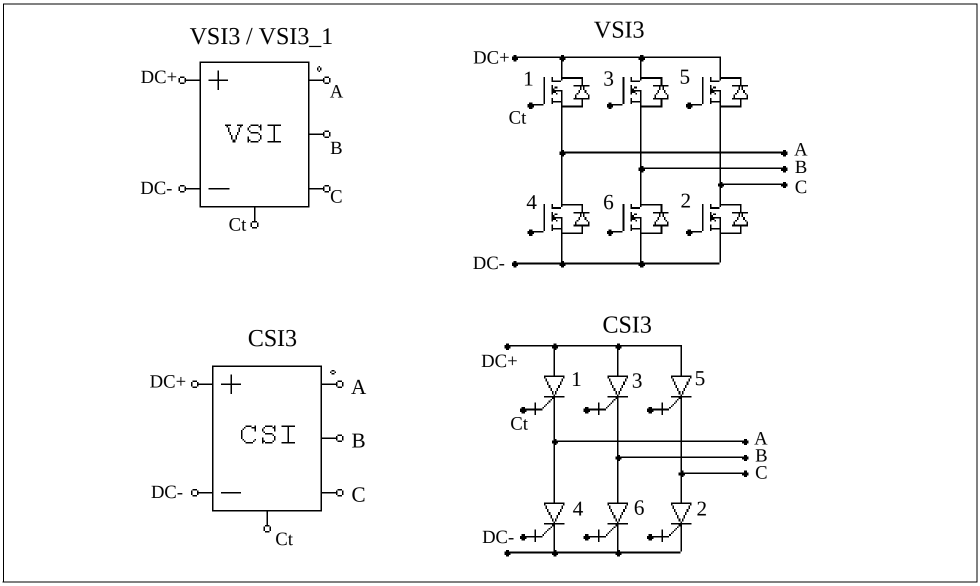
The gatings for the circuit on the left are specified through a gating block, and on the right are controlled through an alpha controller. A major advantage of the alpha controller is that the delay angle alpha of the thyristor bridge, in deg., can be directly controlled.

2.2.7 Three-Phase Switch Modules

The following figure shows three-phase switch modules and the internal circuit connections. The three-phase voltage source inverter module VSI3 consists of MOSFET-type switches, and the module VSI3_1 consists of IGBT-type switches. The current source inverter module CSI3 consists of GTO-type switches, or equivalently IGBT in series with diodes.

Images:





Attributes:

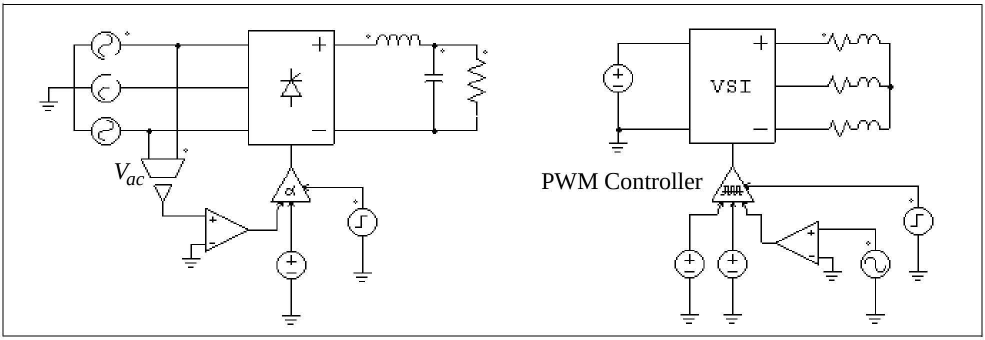
Parameters	Description
On-Resistance	On resistance of the MOSFET switch during the on state, in Ohm (for VSI3 only)
Saturation Voltage	Conduction voltage drop of the IGBT switch, in V (for VSI3_1 only)
Voltage Drop	Conduction voltage drop of the switch, in V (for CSI3 only)
Diode Voltage Drop	Conduction voltage drop of the anti-parallel diode, in V (for VSI3 and VSI3_1 only)
Init. Position_*i*	Initial position for Switch *i*
Current Flag_*i*	Current flag for Switch *i*

Similar to single-phase modules, only the gatings for Switch 1 need to be specified for three-phase modules. Gatings for other switches will be automatically derived. For the half-wave thyristor bridge (BTHY3H), the phase shift between two consecutive switches is 120^o. For all other bridges, the phase shift is 60^o.

Thyristor bridges (BTHY3 / BTHY3H / BTHY6H) can be controlled by an alpha controller. Similarly, voltage/current source inverters can be controlled by a PWM lookup table controller (PATTCTRL).

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

Example: Control of Three-Phase Thyristor and VSI Modules

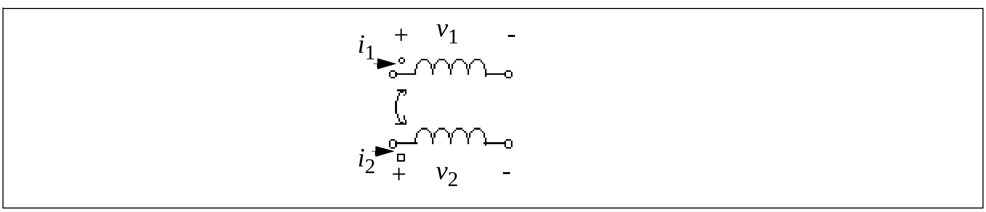


The thyristor circuit on the left uses an alpha controller. For a three-phase circuit, the zero-crossing of the voltage V_{ac} corresponds to the moment when the delay angle alpha is equal to zero. This signal is, therefore, used to provide synchronization to the controller

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 input of the PWM lookup table controller includes the delay angle, the synchronization, and the enable/disable signal. A detailed description of the PWM lookup table controller is given in Section 4.5.3.

2.3 Coupled Inductors

Coupled inductors with two, three, and four branches are provided. The following shows coupled inductors with two branches.

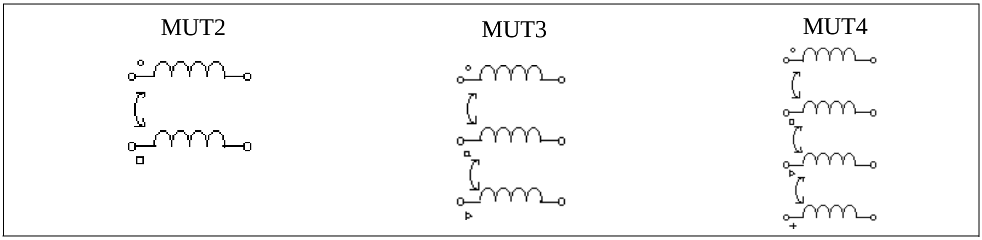


Let L11 and L22 be the self-inductances of Branch 1 and 2, and L12 and L21 the mutual inductances, the branch voltages and currents have the following relationship:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} L11 & L12 \\ L21 & L22 \end{bmatrix} \cdot \frac{d}{dt} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

The mutual inductances between two windings are assumed to be always equal, i.e., L12=L21.

Images:



Attributes:

Parameters	Description
L*ii* (self)	Self inductance of the inductor *i*, in H
L*ij* (mutual)	Mutual inductance between Inducto *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 refer to Inductor 1, 2, 3, and 4, respectively.

Example:

Two mutually coupled inductors have the following self inductances and mutual inductance: L11 = 1 mH, L22 = 1.1 mH, and L12 = L21 = 0.9 mH. The specification of the element MUT2 will be:

L11 (self)	1m
L12 (mutual)	0.9m
L22 (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
Np (primary)	No. of turns of the primary winding
Ns (secondary)	No. of turns of the secondary winding

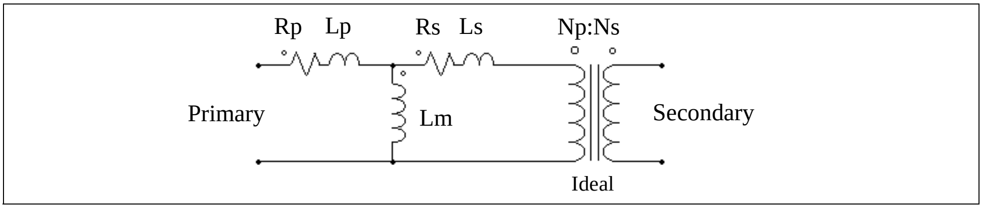
Since the turns ratio is equal to the ratio of the rated voltages, the number of turns can be replaced by the rated voltage at each side.

2.4.2 Single-Phase Transformers

The following single-phase transformer modules are provided in PSIM:

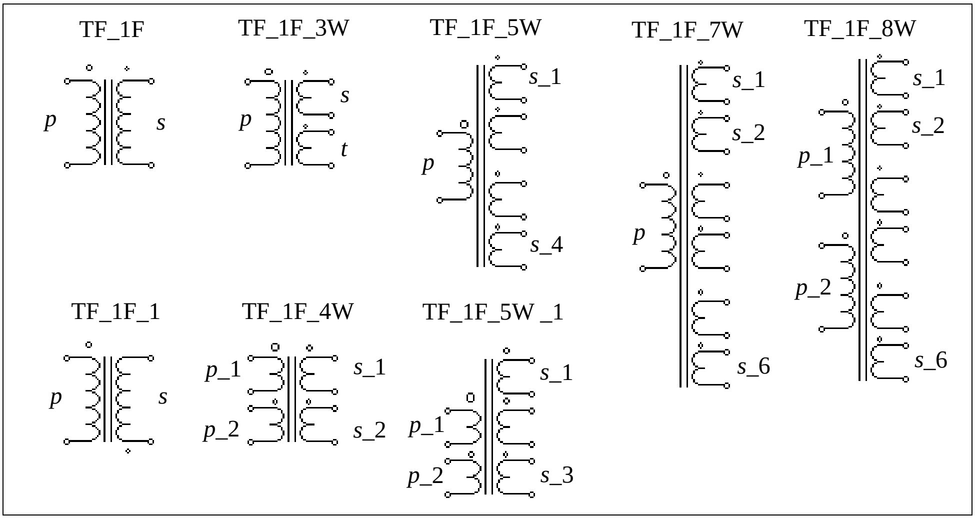
- Transformer with 1 primary and 1 secondary windings (TF_1F / TF_1F_1)
- Transformer with 1 primary and 2 secondary windings (TF_1F_3W)
- Transformer with 2 primary and 2 secondary windings (TF_1F_4W)
- Transformer with 1 primary and 4 secondary windings (TF_1F_5W / TF_1F_5W_1)
- Transformer with 1 primary and 6 secondary windings (TF_1F_7W)
- Transformer with 2 primary and 6 secondary windings (TF_1F_8W)

A single-phase two-winding transformer is modelled as:



where Rp and Rs are the primary and secondary winding resistances; Lp and Ls are the primary and secondary winding leakage inductances; and Lm is the magnetizing inductance. All the values are referred to the primary winding side. If there are multiple primary windings, all the values are referred to the first primary winding.

Images:



In the images, *p* refers to primary, *s* refers to secondary, and *t* refers to tertiar .

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
Rp (primary); Rs (secondary); Rt (tertiary)	Resistance of the primary/secondary/tertiary winding, in Ohm
Lp (pri. leakage); Ls (sec. leakage); Lt (ter. leakage)	Leakage inductance of the primary/secondary/tertiary winding, in H (seen from the primary)
Lm (magnetizing)	Magnetizing inductance, in H
Np (primary); Ns (secondary); Nt (tertiary)	No. of turns of the primary/secondary/tertiary winding

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

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

Attributes:

Parameters	Description
Rp_*i* (primary *i*); Rs_*i* (secondary *i*)	Resistance of the i_{th} primary/secondary/tertiary winding, in Ohm
Lp_*i* (pri. *i* leakage); Ls_*i* (sec. *i* leakage)	Leakage inductance of the i_{th} primary/secondary/tertiary winding, in H (referred to the first primary winding)
Lm (magnetizing)	Magnetizing inductance, in H (seen from the first primary winding)
Np_*i* (primary *i*); Ns_*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.

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 Np:Ns = 220:440. In PSIM, the transformer will be TF_1F with the specifications as:

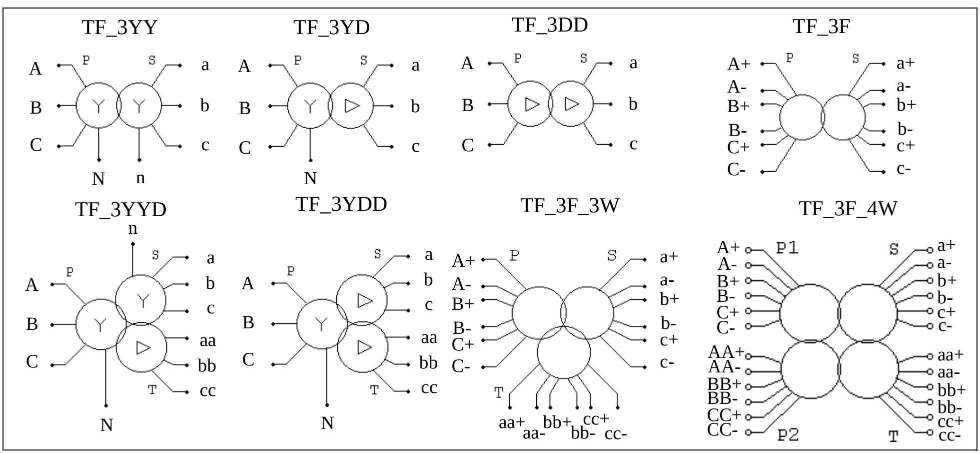
Rp (primary)	2m
Rs (secondary)	2m
Lp (primary)	1m
Ls (secondary)	1m
Lm (magnetizing)	100m
Np (primary)	220
Ns (secondary)	440

2.4.3 Three-Phase Transformers

PSIM provides two-winding and three-winding transformer modules as shown below. They all have 3-leg cores.

- 3-phase transformer (windings unconnected) (TF_3F)
- 3-phase Y/Y and Y/Δ connected transformer (TF_3YY / TF_3YD)
- 3-phase 3-winding transformer (windings unconnected) (TF_3F_3W)
- 3-phase 3-winding Y/Y/Δ and Y/Δ/Δ connected transformer (TF_3YYD / TF_3YDD)
- 3-phase 4-winding transformer (windings unconnected) (TF_3F_4W)

Images:



Attributes:

Parameters	Description
Rp (primary); Rs (secondary); Rt (tertiary)	Resistance of the primary/secondary/tertiary winding, in Ohm
Lp (pri. leakage); Ls (sec. leakage); Lt (ter. leakage)	Leakage inductance of the primary/secondary/tertiary winding, in H
Lm (magnetizing)	Magnetizing inductance, in H (seen from the primary side)
Np (primary); Ns (secondary); Nt (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.

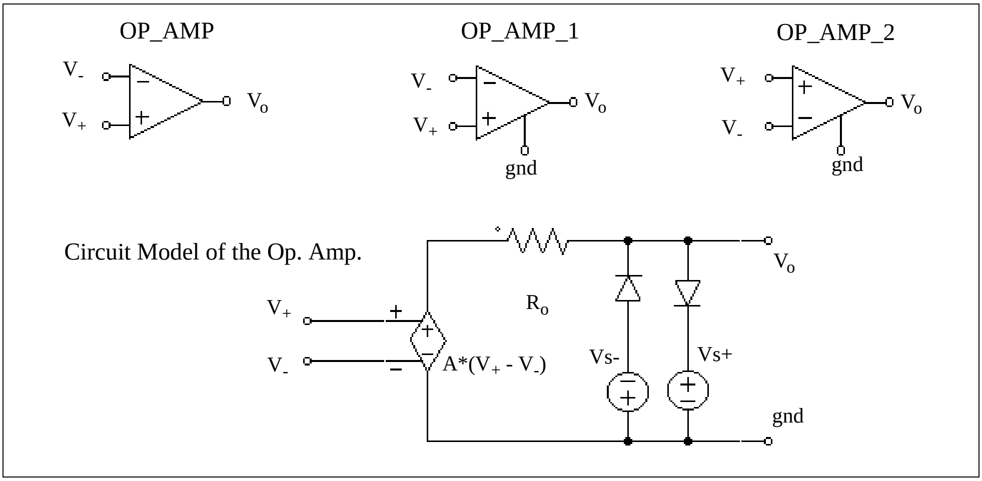
Three-phase transformers are modelled in the same way as single-phase transformers.

2.5 Other Elements

2.5.1 Operational Amplifier

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:

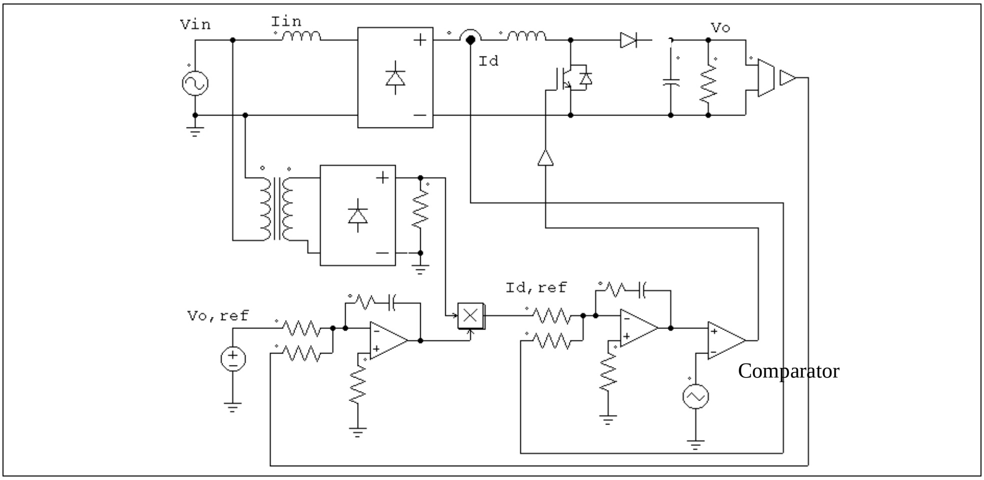
Parameters	Description
Voltage Vs+	Upper voltage source level of the op. amp.
Voltage Vs-	Lower voltage source levels of the op. amp.

The difference between OP_AMP and OP_AMP_1 or OP_AMP_2 is that, for OP_AMP, the reference ground node of the op. amp. model is connected to the power ground, whereas in OP_AMP_1 and OP_AMP_2, the reference ground node of the model is accessible and can be floating.

Note that the image of an op. amp. 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. It has the inner current loop and the outer voltage loop. The PI regulators of both loops are implemented using op. amp.



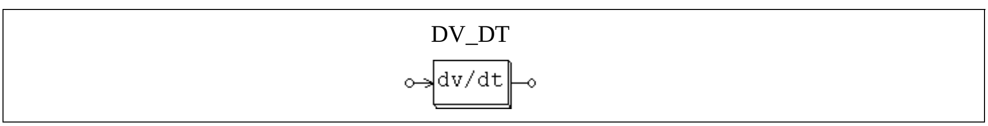
2.5.2 dv/dt Block

A dv/dt block has the same function as the differentiator in the control circuit, except that it can be used in the power circuit. The output of the dv/dt block is equal to the derivative of the input voltage versus time. It is calculated as:

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

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

Image:



2.6 Motor Drive Module

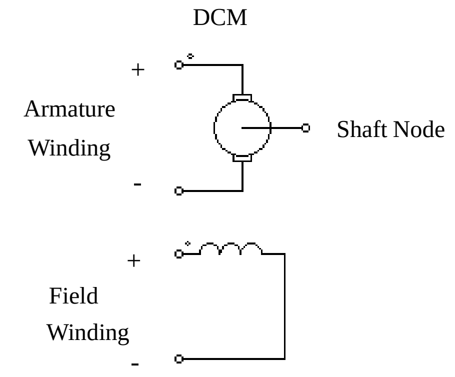
The Motor Drive Module is an add-on module to the basic PSIM program. It provides machine models and mechanical load models for motor drive system studies.

2.6.1 Electric Machines

2.6.1.1 DC Machine

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

Image:



Attributes:

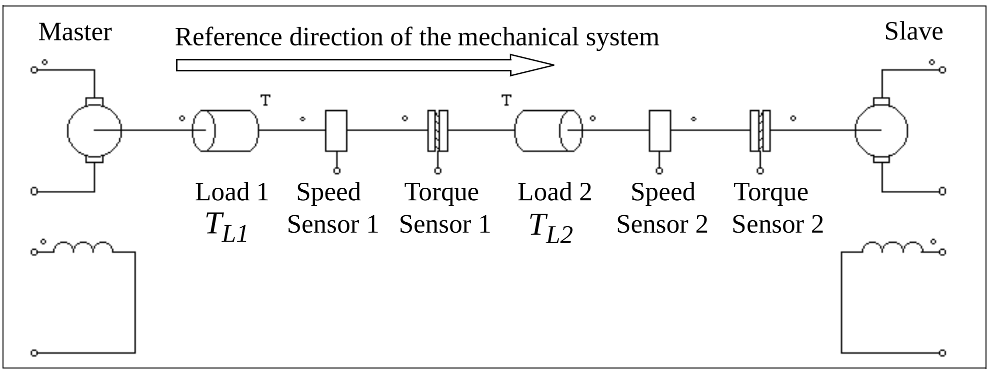
Parameters	Description
R_a (armature)	Armature winding resistance, in Ohm
L_a (armature)	Armature winding inductance, in H
R_f (field)	Field winding resistance, in Ohm
L_f (field)	Field winding inductance, in H
Moment of Inertia	Moment of inertia of the machine, in kg*m^2
V_t (rated)	Rated armature terminal voltage, in V
I_a (rated)	Rated armature current, in A

Parameters	Description
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	Flag for the master/slave mode (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.

A machine is set to either the master or slave mode. When there is only one machine in a mechanical system, this machine must be set to the master mode. When there are two or more machines in a system, only one must be set to the master mode and the rest to the slave mode. The same applies to a mechanical-electrical interface block, as explained later

The machine in the master mode is referred to as the master machine, and it defines the reference direction of the mechanical system. The reference direction is defined as the direction from the shaft node of the master machine along the shaft to the rest of the mechanical system, as illustrated below:



In this mechanical system, the machine on the left is the master and the one on the right is the slave. The reference direction of the mechanical system is, therefore, from left to the right along the mechanical shaft. Furthermore, if the reference direction enters an element at the dotted side, this element is along the reference direction. Otherwise it is opposite to the reference direction. For example, Load 1, Speed Sensor 1, and Torque Sensor 1, are along the reference direction, and Load 2, Speed Sensor 2, and Torque Sensor 2 are opposite to the reference direction.

It is further assumed the mechanical speed is positive when both the armature and the field currents of the master machine are positive.

Based on this notation, if the speed sensor is along the reference direction of the mechanical system, a positive speed produced by the master machine will give a positive speed sensor output. Otherwise, the speed sensor output will be negative. For example, if the speed of the master machine in the example above is positive, Speed Sensor 1 reading will be positive, and Speed Sensor 2 reading will be negative.

The reference direction also determines how a mechanical load interacts with the machine. In this system, there are two constant-torque mechanical loads with the amplitudes of T_{L1} and T_{L2}, respectively. Load 1 is along the reference direction, and Load 2 is opposite to the reference direction. Therefore, the loading torque of Load 1 to the master machine is T_{L1}, whereas the loading torque of Load 2 to the master machine is $-T_{L2}$.

The operation of a dc machine is described by the following equations:

$$v_t = E_a + i_a \cdot R_a + L_a \frac{di_a}{dt}$$

$$v_f = i_f \cdot R_f + L_f \frac{di_f}{dt}$$

$$E_a = k \cdot \phi \cdot \omega_m$$

$$T_{em} = k \cdot \phi \cdot i_a$$

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

where v_t, v_f, i_a, and i_f are the armature and field winding voltage and current, respectively; E_a is the back emf, ω_m is the mechanical speed in rad./sec., T_{em} is the internal developed torque, and T_L is the load torque. The back emf and the internal torque can also be expressed as:

$$E_a = L_{af} \cdot i_f \cdot \omega_m$$

$$T_{em} = L_{af} \cdot i_f \cdot i_a$$

where L_{af} is the mutual inductance between the armature and the field windings. It can be calculated based on the rated operating conditions as:

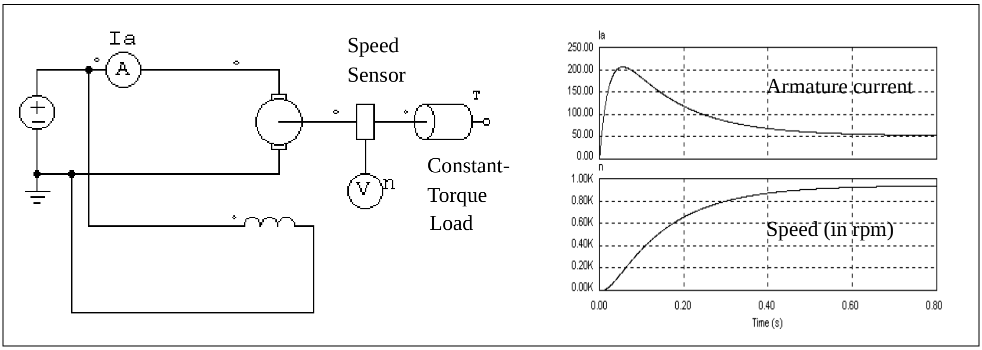
$$L_{af} = \frac{(V_t - I_a \cdot R_a)}{I_f \cdot \omega_m}$$

Note that the dc machine model assumes magnetic linearity. Saturation is not considered.

Example: A DC Motor with a Constant-Torque Load

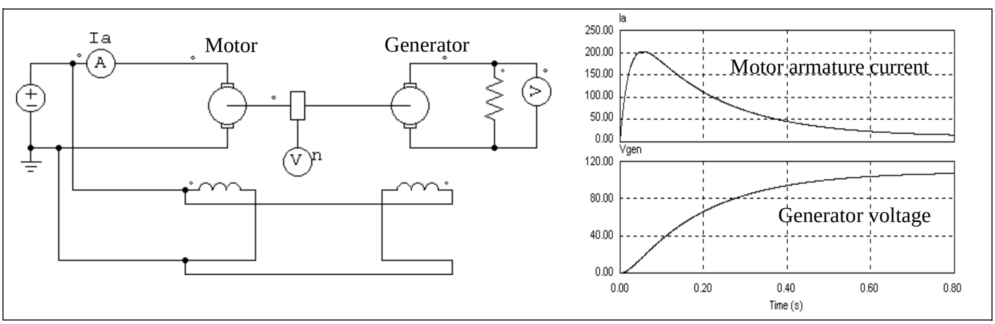
The circuit below shows a shunt-excited dc motor with a constant-torque load T_L. Since the load is along the reference direction of the mechanical system, the loading torque to the machine is T_L. Also, the speed sensor is along the reference direction. It will give a positive output for a positive speed.

The simulation waveforms of the armature current and the speed are shown on the right.



Example: A DC Motor-Generator Set

The circuit below shows a dc motor-generator set. The motor on the left is set to the master mode and the generator on the right is set to the slave mode. The simulation waveforms of the motor armature current and the generator voltage show the start-up transient.



2.6.1.2 Induction Machine

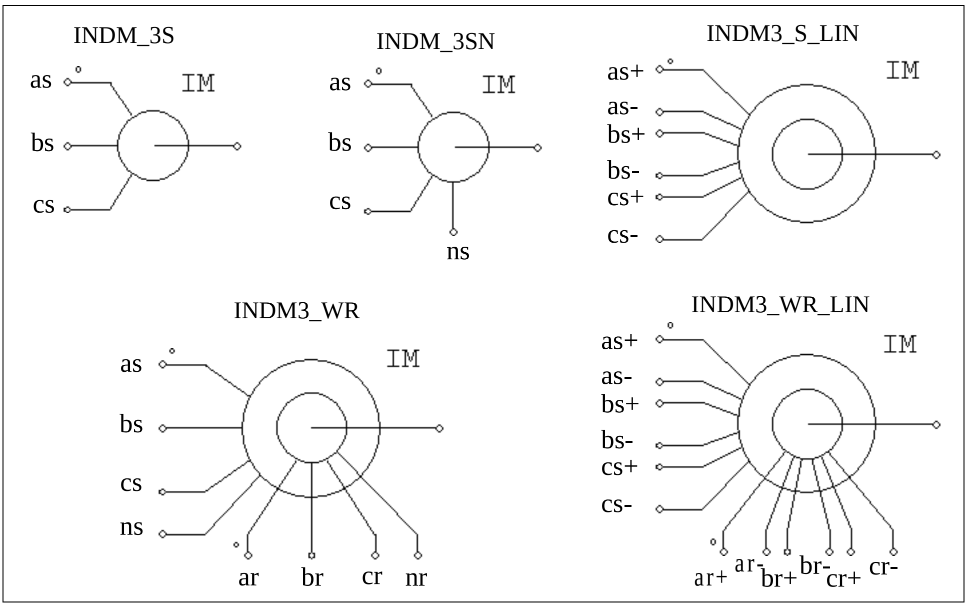
Two types of models are provided for both squirrel-cage and wound-rotor induction machines: linear and nonlinear model. The linear model is further divided into general type and symmetrical type. This section describes the linear models.

Four linear models are provided:

- Symmetrical 3-phase squirrel-cage induction machine (INDM_3S / INDM_3SN)
- General 3-phase squirrel-cage induction machine (INDM3_S_LIN)
- Symmetrical 3-phase wound-rotor induction machine (INDM3_WR)
- General 3-phase wound-rotor induction machine (INDM3_WR_LIN)

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^2
Torque Flag	Flag for internal torque (T_{em}) output. When the flag is set to 1, the output of the internal torque is requested.
Master/Slave Flag	Flag for the master/slave mode (1: master; 0: slave)

All the parameters are referred to the stator side.

Again, the master/slave flag defines the mode of operation for the machine. Refer to Section 2.6.1.1 for detailed explanation. It is assumed the mechanical speed is positive when the input source sequence is positive.

The model INDM_3SN is the same as INDM_3S, except that the stator neutral point is accessible.

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

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

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

where

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

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

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

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

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

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

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

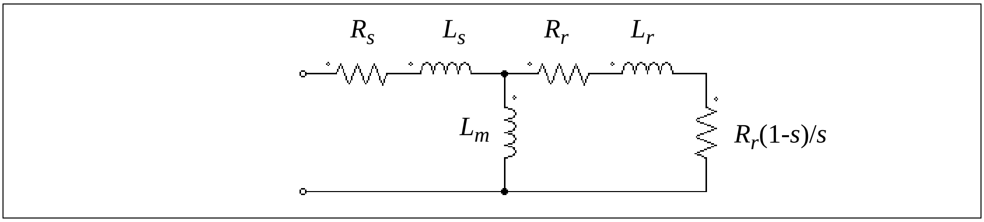
The mechanical equation is expressed as:

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

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

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

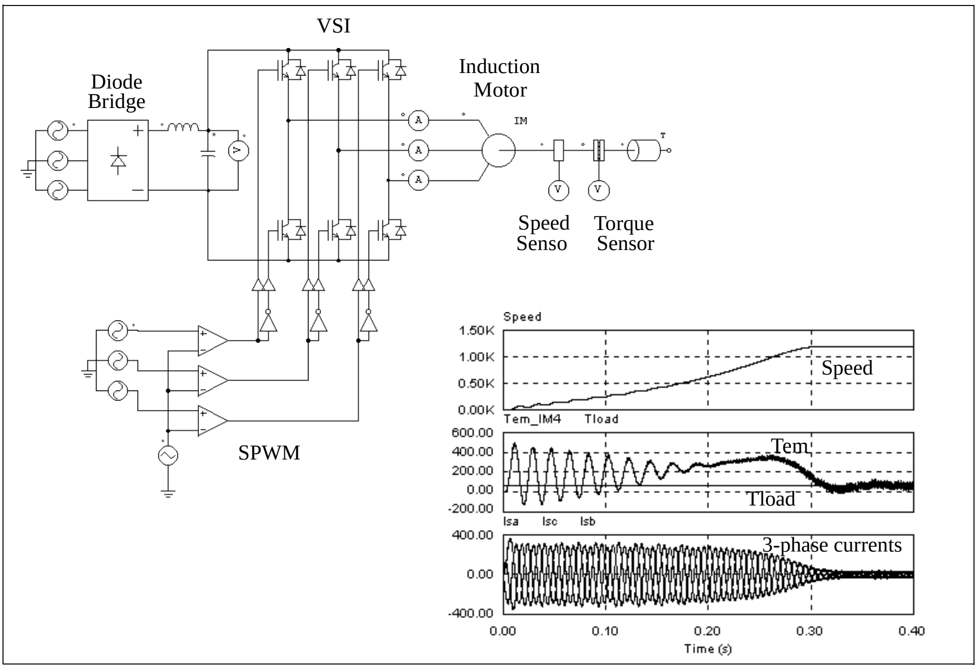
For a symmetrical squirrel-cage induction machine, the steady state equivalent circuit is shown below. In the figure, *s* is the slip.



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.

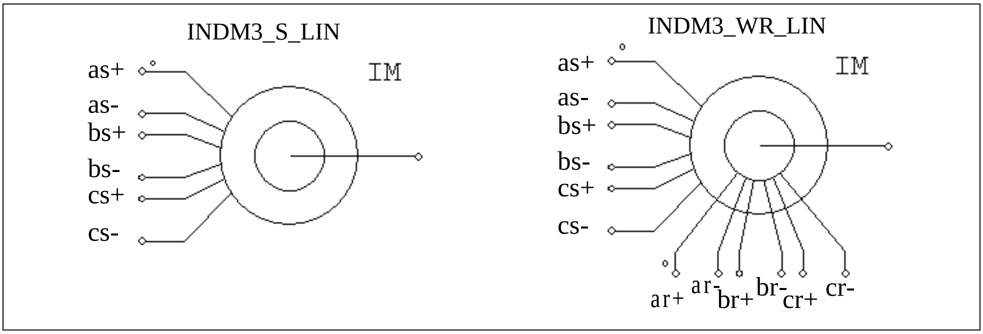


2.6.1.3 Induction Machine with Saturation

Two models of induction machines with saturation are provided:

- 3-phase squirrel-cage induction machine (INDM3_S_NON)
- 3-phase wound-rotor induction machine (INDM3_WR_NON)

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^2
Torque Flag	Flag for internal torque (T_{em}) output. When the flag is set to 1, the output of the internal torque is requested.
Master/Slave Flag	Flag for the master/slave mode (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.

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

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

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

where

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

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

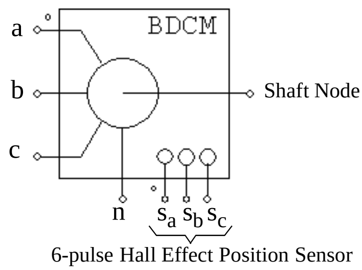
In this case, the inductance M_{sr} is no longer constant, but a function of the magnetizing current I_m.

2.6.1.4 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.)	Stator mutual inductance *M*, in H The mutual inductance *M* is a negative value. Depending on the winding structure, the ratio between *M* and the stator self inductance *L* is normally between -1/3 and -1/2. If *M* is unknown, a reasonable value o *M* equal to -0.4**L* can be used as the default value.
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed)
Vrms / krpm	RMS line-to-line back emf constant, in V/krpm (mechanical speed). The values of Vpk/krpm and Vrms/krpm should be available from the machine data sheet. If these values are not available, they can be obtained through experiments by operating the machine as a generator at 1000 rpm and measuring the peak and rms values of the line-to-line voltage.
No. of Poles *P*	Number of poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Mech. Time Constant	Mechanical time constant τ_{mech}

Parameters	Description
theta_0 (deg.)	Initial rotor angle θ_r, in electrical deg. The initial rotor angle is the rotor angle at t=0. The zero rotor angle position is defined as the position where Phase A back emf crosses zero (from negative to positive) under a positive rotation speed.
theta_advance (deg.)	Position sensor advance angle $\theta_{advance}$, in electrical deg. The advance angle is defined as the angle difference between the turn-on angle of Phase *A* upper switch and 30^o in an 120^o conduction mode. For example, if Phase *A* is turned on at 25^o, the advance angle will be 5^o (i.e. 30 - 25 = 5).
Conduction Pulse Width	Position sensor conduction pulse width, in electrical deg. Positive conduction pulse can turn on the upper switch and negative pulse can turn on the lower switch in a full bridge inverter. The conduction pulse width is 120 electrical deg. for 120^o conduction mode.
Torque Flag	Output flag for internal developed torque T_{em} (1: output; 0: no output)
Master/Slave Flag	Flag for the master/slave mode (1: master; 0: slave). The flag defines the mode of operation for the machine. Refer to Section 2.6.1.1 for detailed explanation.

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.

The equations of the 3-phase brushless dc machine are:

$$v_a = R \cdot i_a + (L - M) \cdot \frac{di_a}{dt} + E_a$$

$$v_b = R \cdot i_b + (L - M) \cdot \frac{di_b}{dt} + E_b$$

$$v_c = R \cdot i_c + (L - M) \cdot \frac{di_c}{dt} + E_c$$

where v_a, v_b, and v_c are the phase voltages, i_a, i_b, and i_c are the phase currents, R, L, and M are the stator phase resistance, self inductance, and mutual inductance, and E_a, E_b, and E_c are the back emf of Phase *A*, *B*, and *C*, respectively.

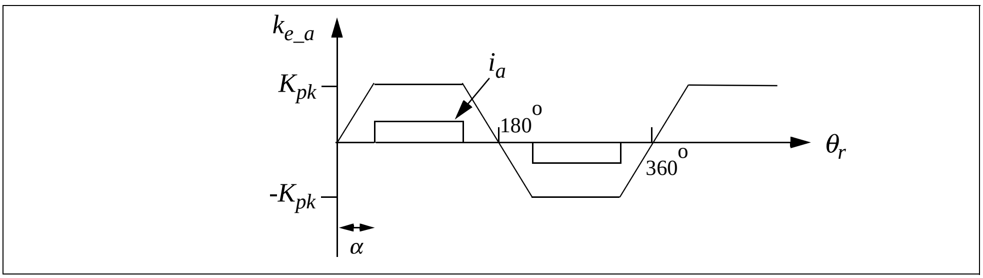
The back emf voltages are a function of the rotor mechanical speed ω_m and the rotor electrical angle θ_r, that is:

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

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

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

The coefficients k_{e_a}, k_{e_b}, and k_{e_c} are dependent on the rotor angle θ_r. In this model, an ideal trapezoidal waveform profile is assumed, as shown below for Phase *A*. Also shown is the Phase *A* current.



where K_{pk} is the peak trapezoidal value, in V/(rad./sec.), which is defined as: $K_{pk} = \frac{V_{pk}/krpm}{2} \cdot \frac{1}{1000 \cdot \pi/60}$. Given the values of Vpk/krpm and Vrms/krpm, the angle α 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 τ_{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.

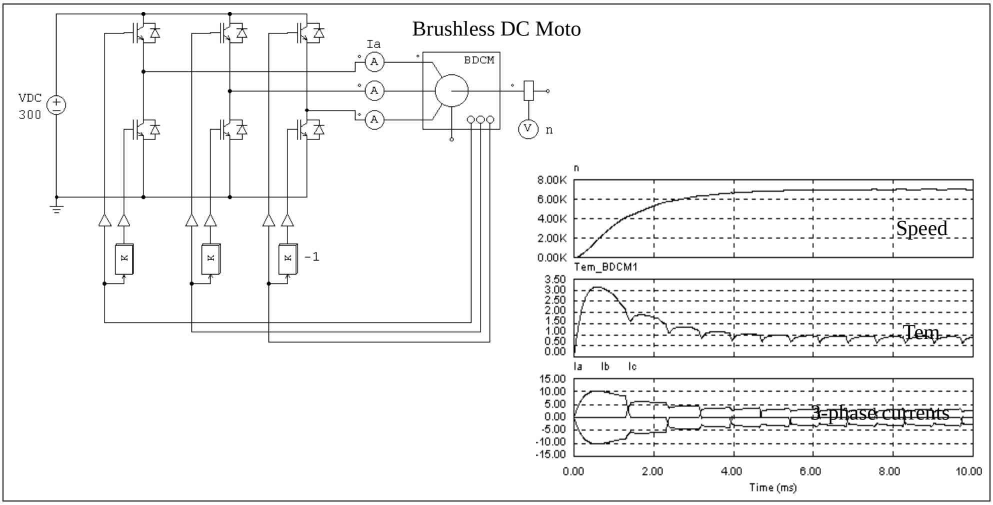
The hall switch is a semiconductor switch (e.g. MOSFET or BJT) that opens or closes when the magnetic field is higher or lower than a certain threshold value. It is based on the hall effect, which generates an emf proportional to the flux-density when the switch is carrying a current supplied by an external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

The figure below shows an open-loop brushless dc motor drive system. The motor is fed by a 3-phase voltage source inverter. The outputs of the motor hall effect position sensors are used as the gatings signals for the inverter, resulting a 6-pulse operation.

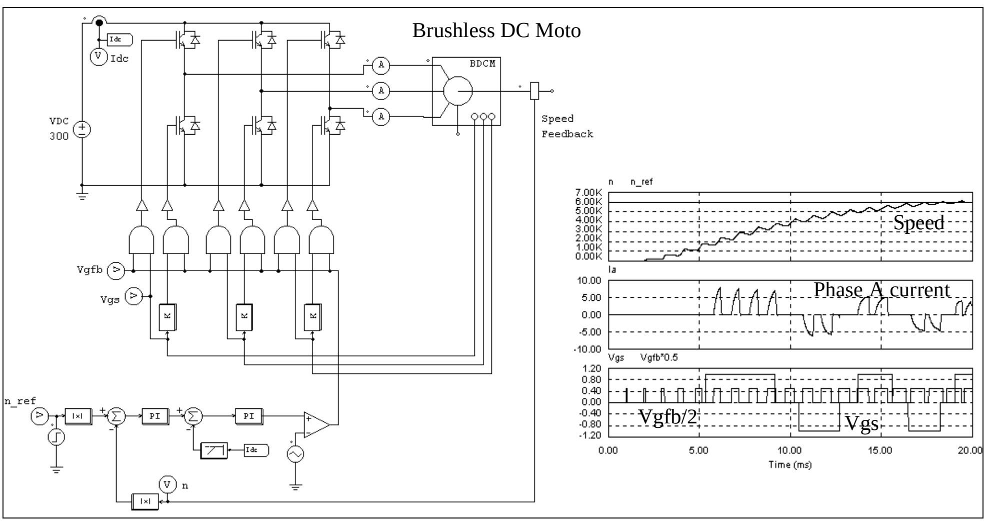
The simulation waveforms show the start-up transient of the mechanical speed (in rpm), developed torque T_{em}, and 3-phase input currents.



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 (Vgs for Phase *A* in this case) with another high-frequency pulses (Vgfb 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 Vgs and Vgfb. Note that Vgfb is divided by half for display purpose.



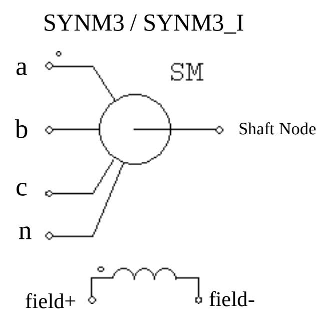
2.6.1.5 Synchronous Machine with External Excitation

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 (model SYNM3), and the other is the current-type interface (model SYNM3_I). 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
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

Parameters	Description
R_{dr} (damping cage)	Rotor damping cage d-axis resistance, in Ohm
L_{drl} (damping cage)	Rotor damping cage d-axis leakage inductance, in
R_{qr} (damping cage)	Rotor damping cage q-axis resistance, in Ohm
L_{qrl} (damping cage)	Rotor damping cage q-axis leakage inductance, in
Ns/Nf (effective)	Stator-field winding effective turns ratio
Number of Poles P	Number of Poles P
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Torque Flag	Output flag for internal developed torque T_{em}
Master/Slave Flag	Flag for the master/slave mode (1: master; 0: slave).

All the parameters are referred to the stator side.

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

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

where

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

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

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 θ_r is the rotor angle.

The developed torque can be expressed as:

$$T = \frac{P}{2} \cdot [I] \cdot \frac{d}{d\theta_r}[L] \cdot [I]$$

The mechanical equations are:

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

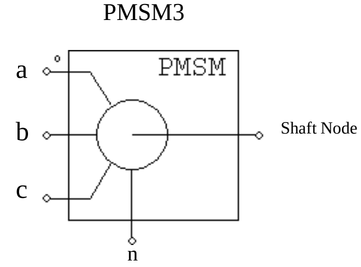
$$\frac{d\theta_r}{dt} = \frac{P}{2} \cdot \omega_m$$

2.6.1.6 Permanent Magnet Synchronous Machine

A 3-phase permanent magnet synchronous machine has 3-phase windings on the stator, and permanent magnet on the rotor. The difference between this machine and the brushless dc machine is that the machine back emf is sinusoidal.

The image and parameters of the machine are shown as follows.

Image:



Attributes:

Parameters	Description
R_s (stator resistance)	Stator winding resistance, in Ohm
L_d (d-axis ind.)	Stator d-axis inductance, in H
L_q (q-axis ind.)	Stator q-axis inductance, in H. The d-q coordinate is defined such that the d-axis passes through the center of the magnet, and the q-axis is in the middle between two magnets. The q-axis is leading the d-axis.
Vpk / krpm	Peak line-to-line back emf constant, in V/krpm (mechanical speed). The value of Vpk/krpm should be available from the machine data sheet. If this data is not available, it can be obtained through an experiment by operating the machine as a generator at 1000 rpm and measuring the peak line-to-line voltage.
No. of Poles *P*	Number of poles *P*
Moment of Inertia	Moment of inertia *J* of the machine, in kg*m^2
Mech. Time Constant	Mechanical time constant τ_{mech}
Torque Flag	Output flag for internal developed torque T_{em} (1: output; 0: no output)
Master/Slave Flag	Flag for the master/slave mode (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.

The equations of the permanent-magnet synchronous machine are:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}$$

where v_a, v_b, v_c, and i_a, i_b, and i_c, and λ_a, λ_b, λ_c are the stator phase voltages, currents, and flux linkages, respectively, and R_s is the stator phase resistance. The flux linkages are further defined as:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_{pm} \cdot \begin{bmatrix} \cos(\theta_r) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix}$$

where θ_r is the rotor electrical angle, and λ_{pm} is 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:

$$L_{aa} = L_{sl} + L_o + L_2 \cdot \cos(2\theta_r)$$

$$L_{bb} = L_{sl} + L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right)$$

$$L_{cc} = L_{sl} + L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right)$$

$$L_{ab} = L_{ba} = -L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right)$$

$$L_{ac} = L_{ca} = -L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right)$$

$$L_{bc} = L_{cb} = -L_o + L_2 \cdot \cos(2\theta_r)$$

where L_{sl} is the stator leakage inductance. The d-axis and q-axis inductances are associated with the above inductances as follows:

$$L_d = L_{sl} + \frac{3}{2}L_o + \frac{3}{2}L_2$$

$$L_q = L_{sl} + \frac{3}{2}L_o - \frac{3}{2}L_2$$

The developed torque can be expressed as:

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

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

The mechanical equations are:

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

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

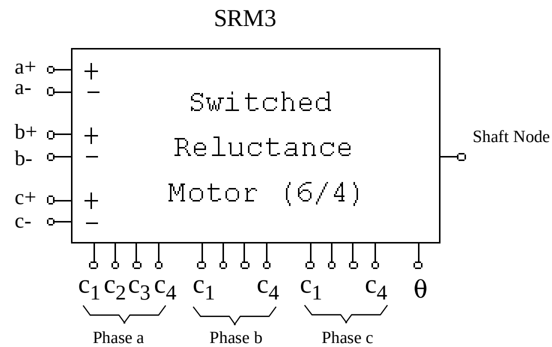
where B is a coefficient, T_{load} is the load torque, and P is the no. of poles. The coefficient B is calculated from the moment of inertia J and the mechanical time constant τ_{mech} as below:

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

2.6.1.7 Switched Reluctance Machine

PSIM provides the model of a 3-phase switched reluctance machine with 6 stator teeth and 4 rotor teeth. 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
θ_r	Duration of the interval where the inductance increases, in deg.
Moment of Inertia	Moment of inertia J of the machine, in kg*m^2
Torque Flag	Output flag for internal torque T_{em}. When the flag is set to 1, the output of the internal torque is requested.
Master/Slave Flag	Flag for the master/slave mode (1: master; 0: slave)

The master/slave flag defines the mode of operation for the machine. See Section 2.6.1.1 for detailed explanation on how to set the master/slave flag.

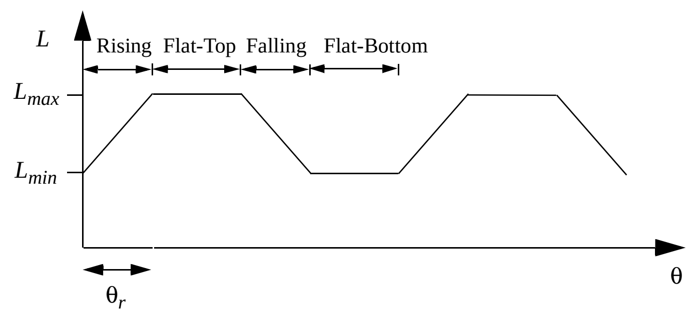
The node assignments are: Nodes a+, a-, b+, b-, and c+, c- are the stator winding terminals for Phase *a*, *b*, and *c*, respectively. The shaft node is the connecting terminal for the mechanical shaft. They are all power nodes and should be connected to the power circuit.

Node c_1, c_2, c_3, and c_4 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 θ 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 θ, 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, θ = 0. The value of the inductance can be in either rising stage, flat-top stage, falling stage, or flat-bottom stage.

If we define the constant *k* as:

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

we can express the inductance *L* as a function of the rotor angle θ:

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

$L = L_{max}$ [flat-top stage. Control signal c_2=1)

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

$L = L_{min}$ [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:

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

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

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

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

Note that saturation is not considered in this model.

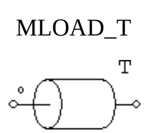
2.6.2 Mechanical Loads

Several mechanical load models are provided in PSIM: constant-torque, constant-power, constant-speed, and general-type load.

2.6.2.1 Constant-Torque Load

The image of a constant-torque load is:

Image:



Attributes:

Parameters	Description
Constant Torque	Torque constant T_{const}, in N*m
Moment of Inertia	Moment of inertia of the load, in kg*m^2

If the reference direction of a mechanical system enters the dotted terminal, the load is along the reference direction, and the loading torque to the master machine is T_{const}. Otherwise the loading torque will be $-T_{const}$. See Section 2.6.1.1 for more detailed explanation on the reference direction.

A constant-torque load is expressed as:

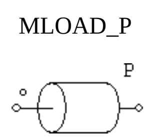
$$T_L = T_{const}$$

The torque does not depend on the speed direction.

2.6.2.2 Constant-Power Load

The image of a constant-power load is:

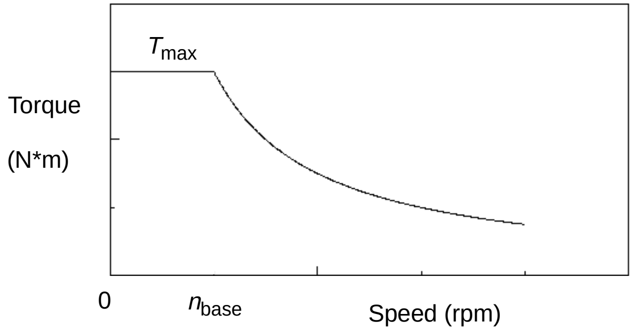
Image:



Attributes:

Parameters	Description
Maximum Torque	Maximum torque T_{max} of the load, in N*m
Base Speed	Base speed n_{base} of the load, in rpm
Moment of Inertia	Moment of inertia of the load, in kg*m^2

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



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

$$T_L = T_{max}$$

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

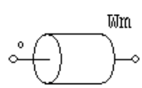
$$T_L = \frac{P}{|\omega_m|}$$

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

2.6.2.3 Constant-Speed Load

The image of a constant-torque load is:

Image:



Attributes:

Parameters	Description
Constant Speed (rpm	Speed constant, in rp
Moment of Inertia	Moment of inertia of the load, in $kg*m^2$

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

2.6.2.4 General-Type Load

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

Image:

MLOAD

Attributes:

Parameters	**Description**
Tc	Constant torque term
k_1 (coefficient)	Coefficient for the linear term
k_2 (coefficient)	Coefficient for the quadratic term
k_3 (coefficient)	Coefficient for the cubic term
Moment of Inertia	Moment of inertia of the load, in kg*m^2

A general-type load is expressed as:

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

where ω_m is the mechanical speed in rad./sec.

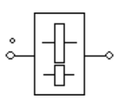
Note that the torque of the general-type load is dependent on the speed direction.

2.6.3 Gear Box

The image is a gear box is shown below.

Image:

GEARBOX



Attribute:

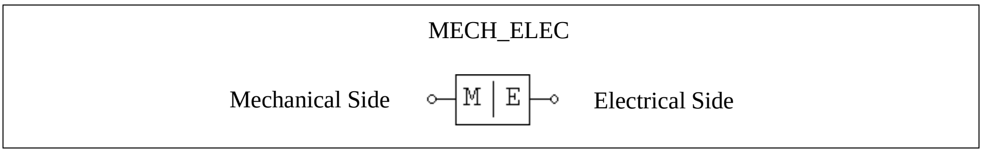
Parameter	Description
Gear Ratio	The gear ratio a

If the numbers of teeth of the first gear and the second gear are n_1 and n_2, respectively, the gear ratio a is defined as: $a = n_1 / n_2$. Let the radius, torque, and speed of these two gears be: r_1, r_2, T_1, T_2, ω_1, and ω_2, we have: $T_1 / T_2 = r_1 / r_2 = \omega_2 / \omega_1 = a$.

2.6.4 Mechanical-Electrical Interface Block

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

Image:



Attribute:

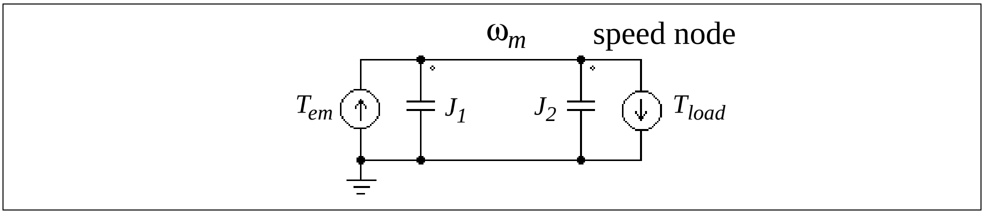
Parameter	Description
Master/Slave Flag	Flag for the master/slave mode (1: master, 0: slave)

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. In a mechanical system, only one and at least one machine/interface block must be set to the master mode. Refer to Section 2.6.1.1 for more explanation on the master/slave flag.

Let's assume that a drive system consists of a motor (with a developed torque of T_{em} and a moment of inertia of J_1) and a mechanical load (with a load torque of T_{load} and a moment of inertia of J_2). The equation that describes the mechanical system is:

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

where ω_m is the shaft mechanical speed. In PSIM, this equation is modelled by an equivalent circuit as shown below.



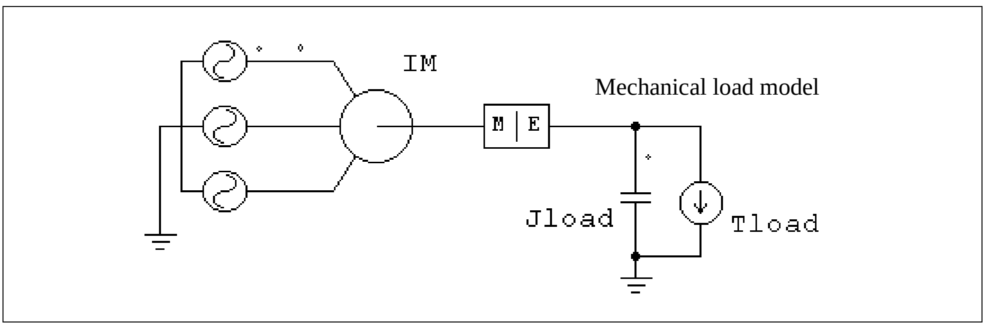
In this circuit, the two current sources have the values of T_{em} and T_{load}, and the capacitors have the values of J_1 and J_2. The node-to-ground voltage (speed node voltage) represents the mechanical speed ω_m. This is analogous to C*dV/dt = i for a capacitor where $C = J_1+J_2$, $V = \omega_m$,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

In PSIM, mechanical equivalent circuits for motors and mechanical loads all use the capacitor-based circuit model. The mechanical-electrical interface block provides the access to the internal mechanical equivalent circuit. If the mechanical side of an interface block (with the letters “MECH”) is connected to a mechanical shaft, the electrical side (with the letters “ELEC”) 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.

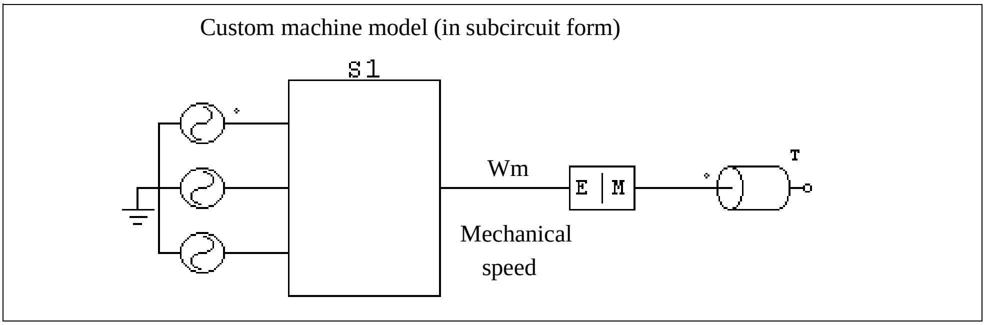
Example: An induction machine with a custom mechanical load model

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

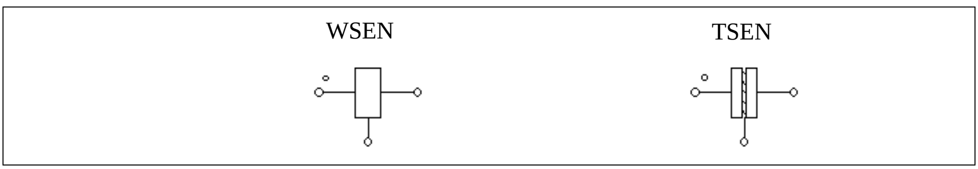
Similarly, one can build a custom machine model and connect it to the mechanical load in PSIM. The figure below shows such a circuit. The custom machine model must use the capacitor analogy to model the mechanical equation. The node representing the mechanical speed is then made available and is connected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



2.6.5 Speed/Torque Sensors

A speed sensor (WSEN) or torque sensor (TSEN) is used to measure the mechanical speed or torque.

Images:

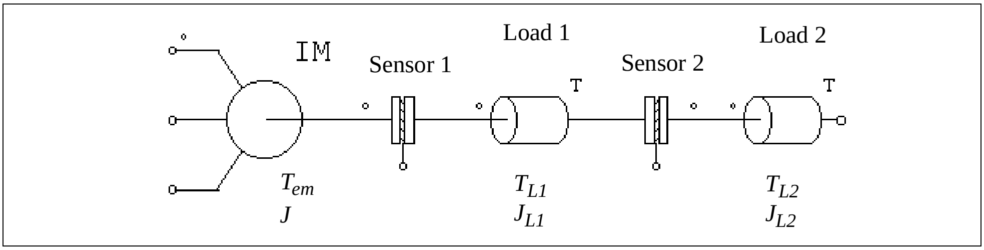


Attribute:

Parameter	Description
Gain	Gain of the sensor

If the reference direction of a mechanical system enters the dotted side of the sensor, the sensor is along the reference direction. Refer to Section 2.6.1.1 for more details on the reference direction. Note that the output of the speed sensor is in rpm.

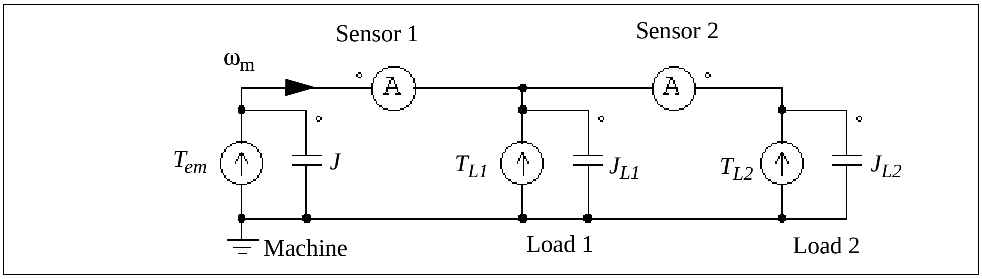
The torque sensor measures the torque transferred from the dotted side of the sensor to the other side alone the positive speed direction. To illustrate this, the following mechanical system is taken as an example:



The system consists of one machine, 2 torque sensors, and 2 mechanical loads. The torques and moment of inertia for the machine and the loads are as labelled in the diagram. The reference direction of this mechanical system is from left to right. The equation for this system can be written as:

$$(J + J_{L1} + J_{L2}) \cdot \frac{d\omega_m}{dt} = T_{em} - T_{L1} - T_{L2}$$

The equivalent electrical circuit of the equation is shown below:



The node voltage in the circuit represents the mechanical speed ω_m. The current probe on the left represents the reading of the torque sensor No. 1. Similarly, the current probe on the right represents the reading of the torque sensor No. 2. Note that the second current probe is from right to left since Sensor 2 is opposite to the reference direction of the mechanical system.

The equivalent circuit also illustrates how mechanical power is transferred. The multiplication of the current to the voltage, which is the same as the torque times the mechanical speed, represents the mechanical power. If the power is positive, it is transferred in the direction of the speed ω_m.

3

Control Circuit Components

3.1 Transfer Function Blocks

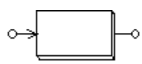
A transfer function block is expressed in polynomial form as:

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

Two types of transfer function blocks are provided: one with zero initial values (TFCTN) and the other with initial values as input parameters (TFCTN1).

Images:

TFCTN / TFCTN1



Attributes:

Parameters	Description
Order n	Order *n* of the transfer function
Gain	Gain *k* of the transfer function
Coeff. $B_n...B_o$	Coefficients of the nominator (from B_n to B_o)
Coeff. $A_n...A_o$	Coefficients of the denominator (from A_n to A_o)
InitialValues $x_n..x_1$	Initial values of the state variables x_n to x_1 (for TFCTN1 only)

Let $Y(s) = G(s)*U(s)$ where $Y(s)$ is the output and $U(s)$ is the input, we can convert the s-domain expression into the differential equation form as follows:

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

The output equation in the time domain can be expressed as:

$$y = x_n + k \cdot \frac{B_n}{A_n} \cdot u$$

The initial values of the state variables x_n to x_1 can be specified at the input in the element TFCTN1.

Example:

The following is a second-order transfer function:

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

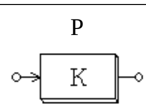
In PSIM, the specification will be:

Order n	2
Gain	1.5
Coeff. B_n...B_o	0. 0. 400.e3
Coeff. A_n...A_o	1. 1200. 400.e3

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

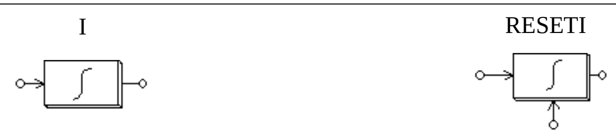
3.1.2 Integrator

The transfer function of an integrator is:

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

There are two types of integrators. One is the regular integrator (I). The other is the resettable integrator (RESETI).

Images:



Attributes:

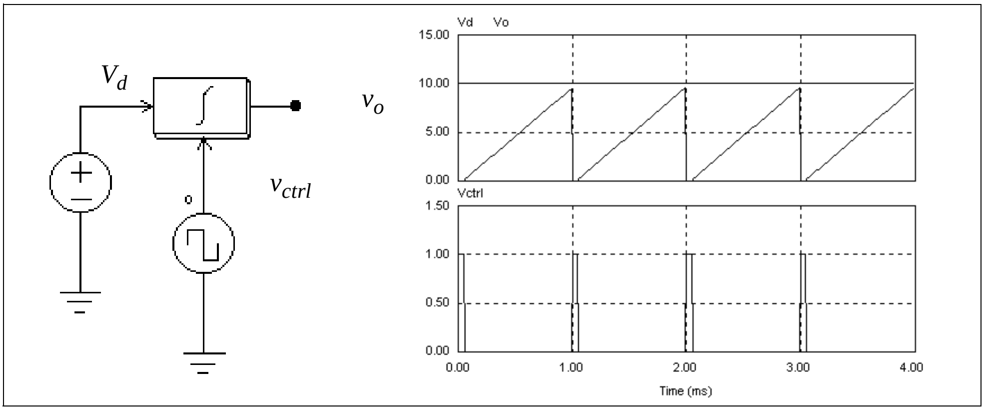
Parameters	Description
Time Constant	Time constant *T* of the integrator, in sec.
Initial Output Value	Initial value of the output
Reset Flag	Reset flag (0: edge reset; 1: level reset) (for RESETI only)

The output of the 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).

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:

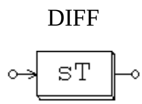
$$G(s) = sT$$

A differentiator is calculated as follows:

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

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

Image:



Attribute:

Parameter	Description
Time Constant	Time constant *T* of the differentiator, in sec.

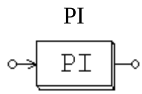
Since sudden changes of the input will generate spikes at the output, it is recommended that a low-pass filter be placed at the input of the differentiator.

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. 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:

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

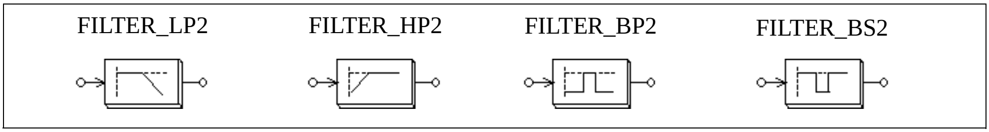
For a second-order band-pass filter:

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

For a second-order band-stop filter:

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

Images:



Attributes:

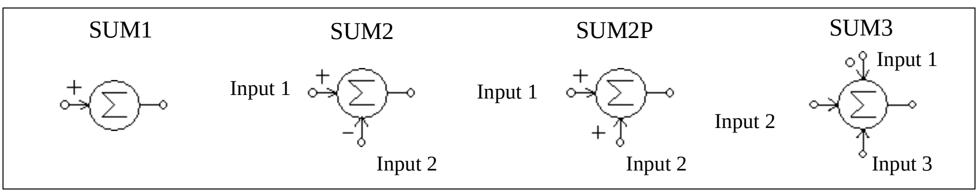
Parameters	Description
Gain	Gain k
Damping Ratio	Damping ratio ξ
Cut-off Frequency	Cut-off frequency f_c ($f_c = \frac{\omega_c}{2\pi}$) for low-pass and high-pass filters, in Hz
Center Frequency	Center frequency f_o ($f_o = \frac{\omega_o}{2\pi}$) for band-pass and band-stop filter, in Hz
Passing Band; Stopping Band	Frequency width f_b ($f_b = \frac{B}{2\pi}$) of the passing/stopping band for band-pass/band-stop filters, in Hz

3.2 Computational Function Blocks

3.2.1 Summer

The input of a one-input summer (SIM1) or two-input summer (SUM2 / SUM2P) can be either a scalar or a vector. The input of a three-input summer (SUM3) can only be a scalar.

Images:



Attribute:

Parameter	Description
Gain_*i*	Gain k_i for the i_{th} input

For the three-input summer SUM3, 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:

$$V_o = k_1 V_1 + k_2 V_2 + ... + k_n V_n$$

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

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

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

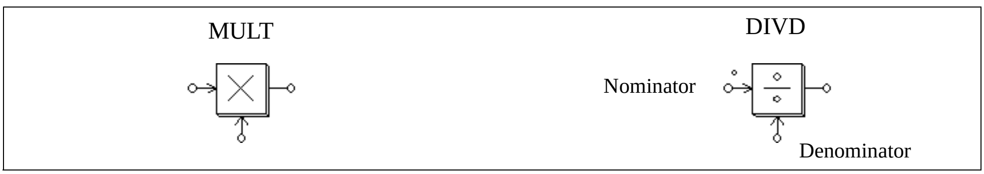
$$V_o = V_1 + V_2 = [a_1+b_1 \ a_2+b_2 \ ... \ a_n+b_n]$$

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

3.2.2 Multiplier and Divider

The output of a multipliers (MULT) or dividers (DIVD) is equal to the multiplication or division of two inputs.

Images:



For the divider, the dotted node is for the nominator 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:

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

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

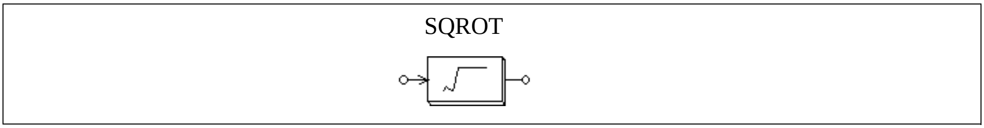
The output, which is a scalar, will be:

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

3.2.3 Square-Root Block

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

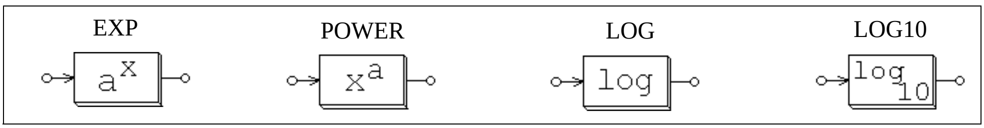
Image:



3.2.4 Exponential/Power/Logarithmic Function Blocks

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

Images:



Attributes (for EXP and POWER):

Parameters	**Description**
Coefficient k_1	Coefficient k_1
Coefficient k_2	Coefficient k_2

The output of an exponential function block EXP 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 POWER 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.

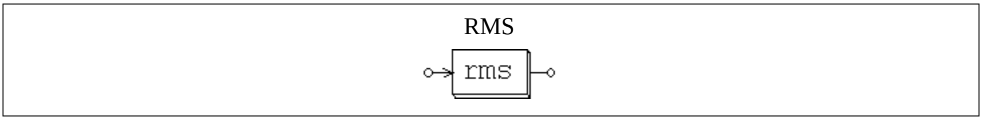
3.2.5 Root-Mean-Square Block

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:



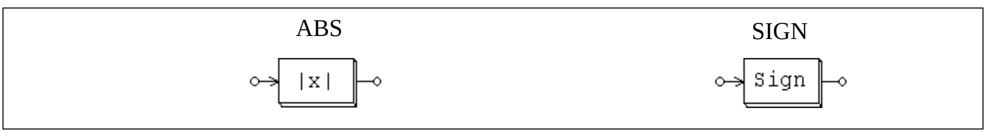
Attribute:

Parameter	Description
Base frequency	Base frequency f_b, in Hz

3.2.6 Absolute and Sign Function Blocks

An absolute value function block (ABS) 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.

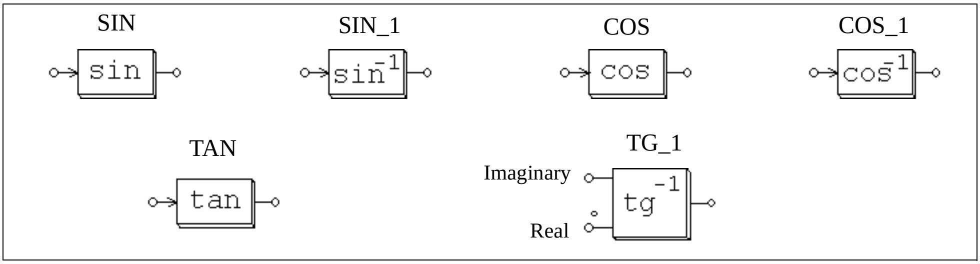
Images:



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 (TG_1). The output is equal to the corresponding trigonometric function of the input. For Blocks SIN, COS, and TAN, the input is in deg., and for Blocks SIN_1, COS_1, and TG_1, 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. $\theta = tg^{-1}\left(\frac{V_{imaginary}}{V_{real}}\right)$

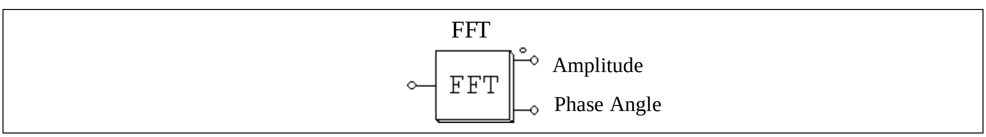
3.2.8 Fast Fourier Transform Block

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:

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

Image:



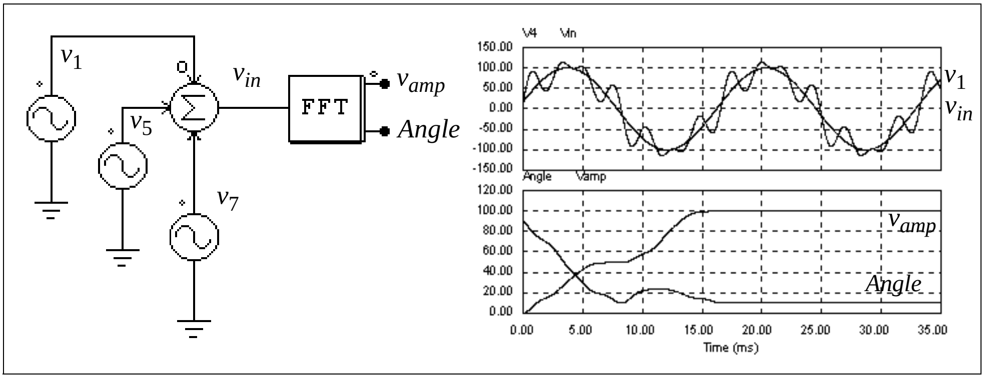
Attributes:

Parameters	Description
No. of Sampling Points	No. of sampling points N
Fundamental Frequency	Fundamental frequency f_b, in Hz.

The dotted node of the block refers to the output of the amplitude. Note that the phase angle output has been internally adjusted such that a sine function V_m*sin(ωt) will give a phase angle output of 0.

Example:

In the circuit below, the voltage v_{in} contains a fundamental component v_1 (100 V at 60 Hz), a 5th harmonic voltage v_5 (25 V at 300 Hz), and a 7th harmonic v_7 (25 V at 420 Hz). After one cycle, the FFT block output reaches the steady state with the amplitude of 100 V and the phase angle of 0^o.

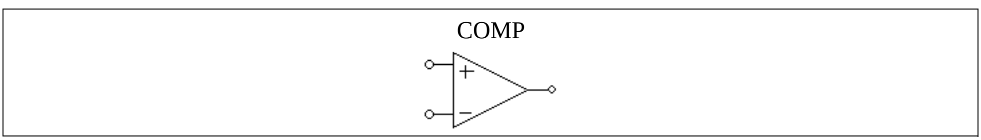


3.3 Other Function Blocks

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:



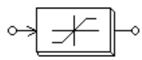
Note that the comparator image is similar to that of the op. amp. For the comparator, the noninverting input is at the upper left and the inverting input is at the lower left. For the op. amp., however, it is the opposite.

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:

LIM



Attributes:

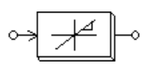
Parameters	Description
Lower Limit	Lower limit of the limiter
Upper Limit	Upper limit of the limiter

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:

LIMIT_DVDT



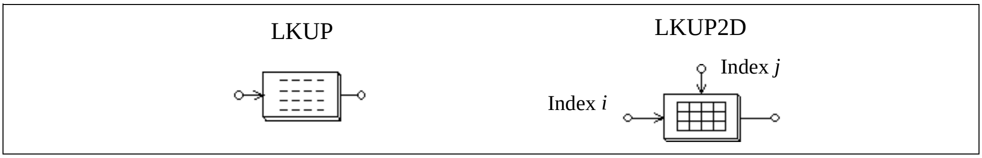
Attribute:

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

3.3.4 Look-up Table

There are two types of lookup tables: one-dimensional lookup table (LKUP), and 2-dimensional lookup table (LKUP2D).

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.

Please note that the one-dimensional lookup table (LKUP) can also be used in the power circuit.

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.

$V_{in}(1)$, $V_o(1)$

$V_{in}(2)$, $V_o(2)$

...

$V_{in}(n)$, $V_o(n)$

The input array V_{in} must be monotonically increasing. Between two points, linear interpolation is used to obtain the output. When the value of the input is less than $V_{in}(1)$ or greater than $V_{in}(n)$, the output will be clamped to $V_o(1)$ or $V_o(n)$.

The 2-dimensional lookup table has two input and one output. 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:

m, n
$A(1,1)$, $A(1,2)$, ..., $A(1,n)$
$A(2,1)$, $A(2,2)$, ..., $A(2,n)$
... ...
$A(m,1)$, $A(m,2)$, ..., $A(m,n)$

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.

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:

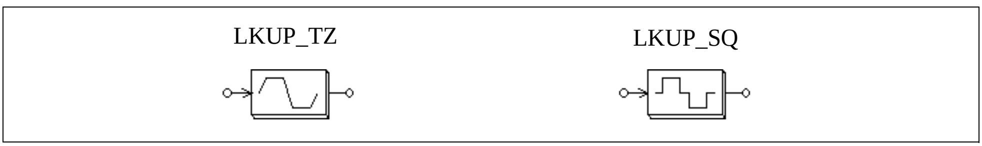
3, 4
1., -2., 4., 1.
2., 3., 5., 8.
3., 8., -2., 9.

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.

3.3.5 Trapezoidal and Square Blocks

Trapezoidal waveform blocks (LKUP_TZ) and square waveform blocks (LKUP_SQ) 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:

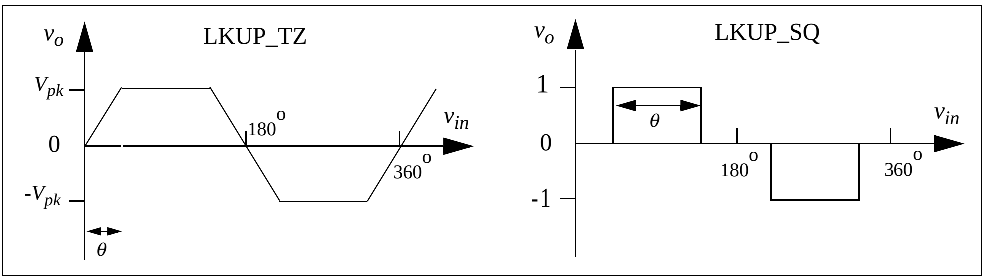
Parameters	Description
Rising Angle theta	Rising angle θ, in deg.
Peak Value	Peak value V_{pk} of the waveform

For the square waveform block:

Attribute:

Parameter	Description
Pulse Width (deg.)	Pulse width θ in half cycle, in deg.

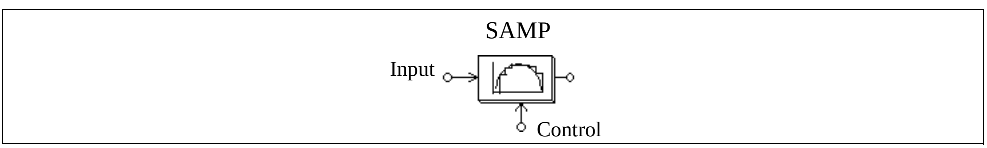
The waveforms of these two blocks are shown below. Note that the input v_{in} is in deg., and can be in the range of -360^o to 360^o. Both waveforms are half-wave and quarter-wave symmetrical.



3.3.6 Sampling/Hold Block

A sampling/hold block samples the input when the control signal changes from low to high (from 0 to 1), and holds this value until the next point is sampled.

Image:

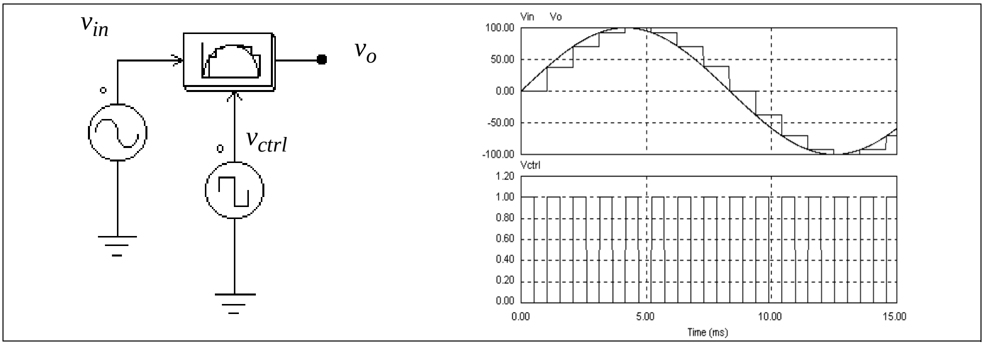


The difference between this block and the zero-order hold block (ZOH) in Digital Control Module is that this block is treated as a continuous element and sampling moments can be controlled externally; whereas the zero-order hold block is a discrete element and the sampling moments are fixed and of equal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

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

Example:

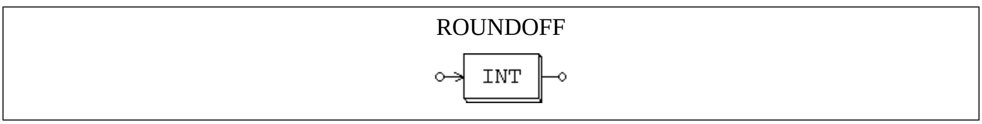
In this example, a sinusoidal input is sampled. The control signal is a square wave voltage source with an amplitude of 1.



3.3.7 Round-Off Block

The image of a round-off block is shown below:

Image:



Attributes:

Parameters	Description
No. of Digits	No. of digits *N* after the decimal point
Truncation Flag	Truncation flag (1: truncation; 0: round-off)

Let the input of the round-off block be V_{in}. The input is first scaled based on the following expression:

$$V_{in,\ new} = V_{in} \cdot 10^N$$

If the truncation flag is 1, the output will be equal to $V_{in,new}$ truncated, and then divided by 10^N. Otherwise, the output will be equal to $V_{in,new}$ rounded off to the nearest integer, and then divided by 10^N.

Examples:

If V_{in} = 34.5678; N = 0, truncation flag = 0, then we have the output V_{out} = 35.

Similarly, if V_{in} = 34.5678; N = 0, truncation flag = 1, the output V_{out} = 34.

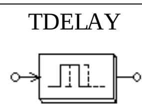
If V_{in} = 34.5678; N = 1, truncation flag = 1, the output V_{out} = 34.5.

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

3.3.8 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:

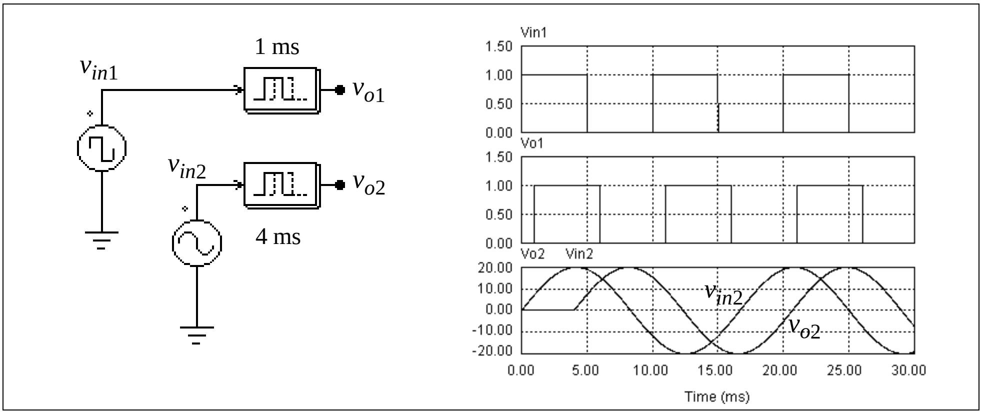
Parameter	Description
Time Delay	Time delay, in sec.

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:

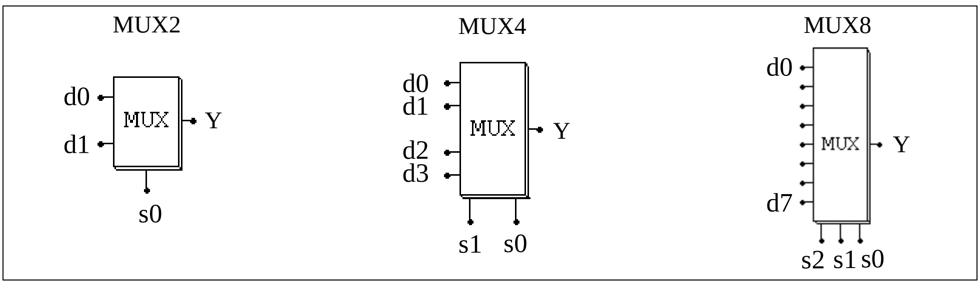
In this circuit, the first time delay block has a delay time of 1 ms, and the second block has a delay time of 4 ms. This example illustrates that the input of the time delay block can be either an analog or a digital signal.



3.3.9 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

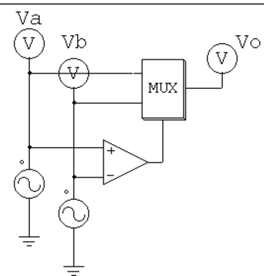
4-Input MUX		
s1	s0	Y
0	0	d0
0	1	d1
1	0	d2
1	1	d3

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 $V_o = V_b$.



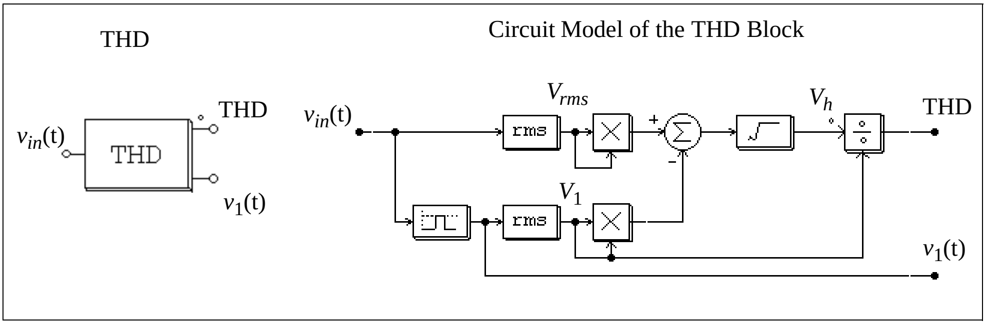
3.3.10 THD Block

The total harmonic distortion (THD) of an ac waveform that contains both the fundamental and harmonic components is defined as:

$$THD = \frac{V_h}{V_1} = \frac{\sqrt{V_{rms}^2 - V_1^2}}{V_1}$$

where V_1 is the fundamental component (rms), V_h is the harmonic rms value, and V_{rms} is the overall rms value of the waveform. The THD block is modelled as shown below.

Image:



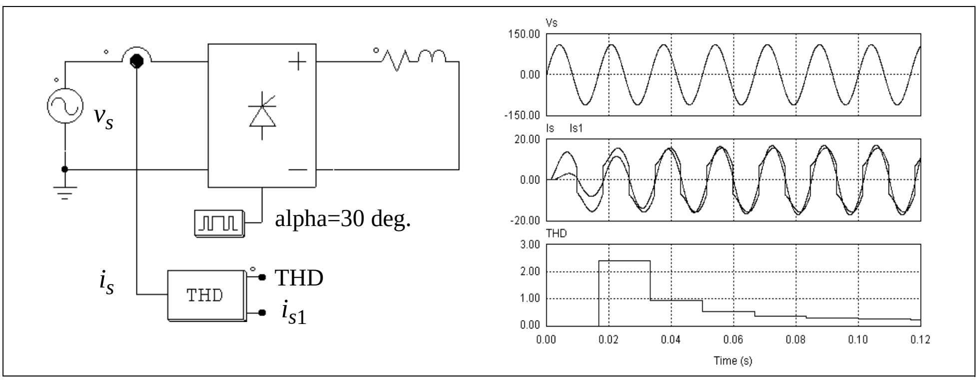
A second-order band-pass filter is used to extract the fundamental component. The center frequency and the passing band of the band-pass filter need to be specified.

Attributes:

Parameters	**Description**
Fundamental Frequency	Fundamental frequency of the input, in Hz
Passing Band	Passing band of the band-pass filter, in Hz

Example:

In the single-phase thyristor circuit below, a THD block is used to measure the THD of the input current. The delay angle of the thyristor bridge is chosen as 30^o. For the THD block, the fundamental frequency is set at 60 Hz and the passing band of the filter is set at 20 Hz. The simulation results are shown on the right.



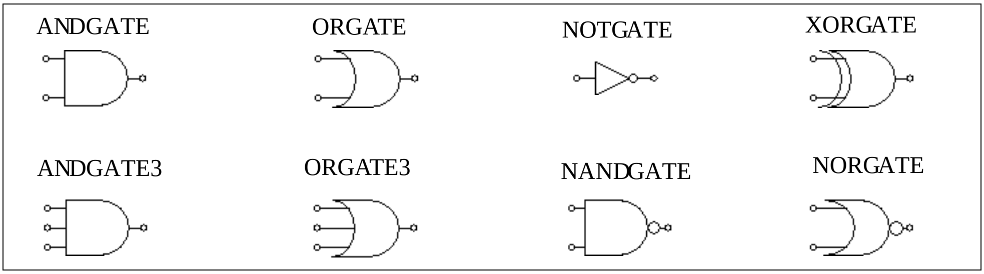
One of the THD block output is the input current fundamental component i_{s1}. By comparing the phase difference between the input voltage v_s and the current i_{s1}, one can calculate the input displacement power factor. This, together with the THD value, can be used to calculate the input power factor.

3.4 Logic Components

3.4.1 Logic Gates

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

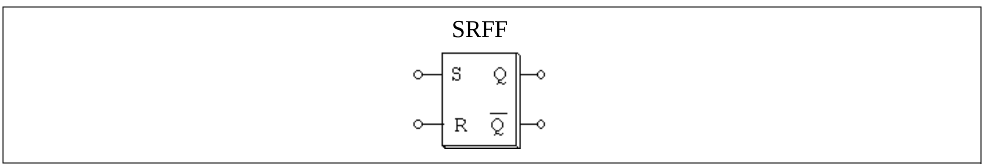
Images:



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	

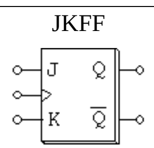
A level-triggered flip-flop, on the other hand, changes the states based on the input level. The truth table of a level-triggered set-reset flip-flop is:

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:



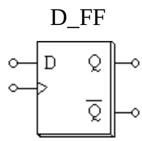
The truth table is:

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 (MONOC). Its on-time pulse width, in second, is determined by the control input.

Images:

Attribute:

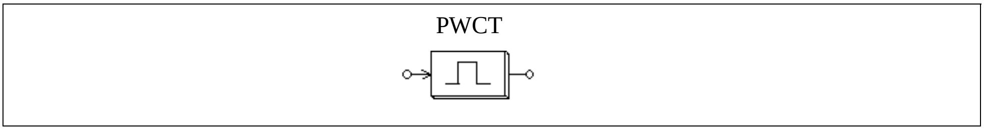
Parameter	Description
Pulse Width	On-time pulse width, in sec.

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.

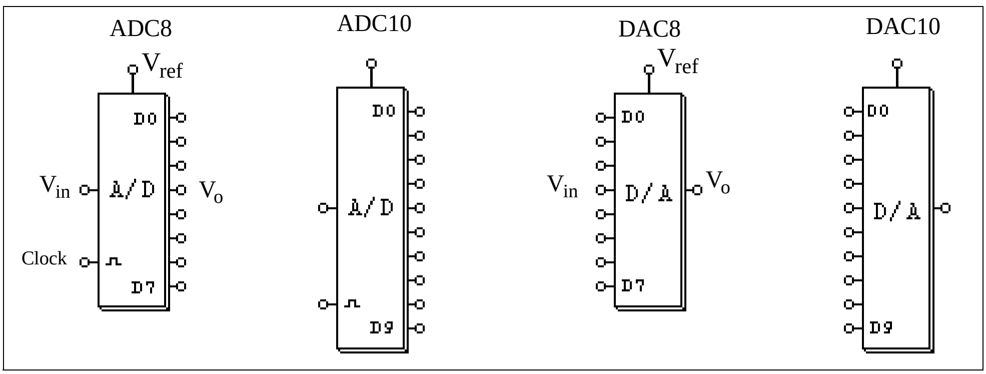
Image:



3.4.7 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*3.2 = 163.84 = 10100011 (binary)

The output of the D/A converter is calculated as:

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

For example, if V_{ref} = 5 V, V_{in} = 10100011 (binary) = 163, N = 8 bits, then

$$V_o = 163/256*5 = 3.1836$$

3.5 Digital Control Module

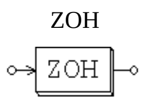
The Digital Control Module is an add-on module to the basic PSIM program. It provides discrete elements, such as zero-order hold, z-domain transfer function blocks, digital filters, etc., for digital control system simulation.

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:



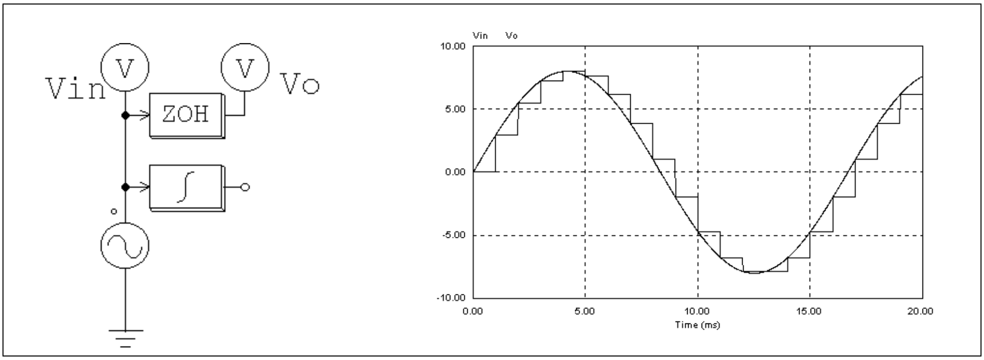
Attribute:

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.

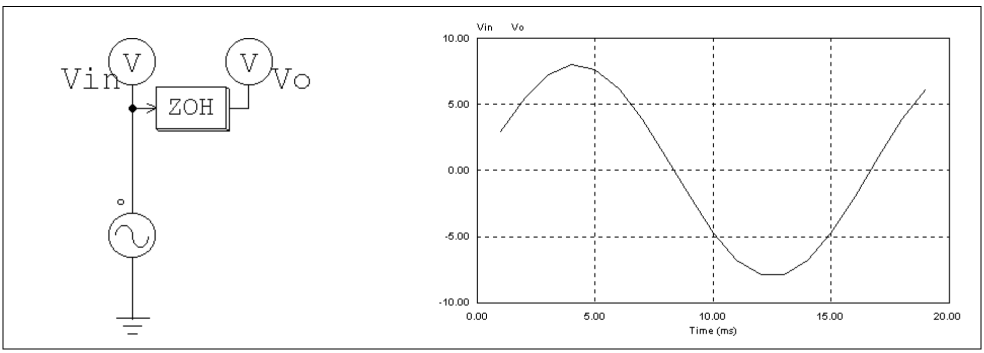
Example:

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:

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

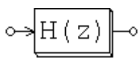
If $a_0 = 1$, the expression $Y(z) = H(z) * U(z)$ can be expressed in difference equation as:

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

$$[a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + \ldots + a_N \cdot y(n-N)]$$

Image:

TFCTN_D



Attributes:

Parameters	**Description**
Order N	Order N of the transfer function
Coeff. $b_0 \ldots b_N$	Coefficients of the nominator (from b_0 to b_N)
Coeff. $a_0 \ldots a_N$	Coefficients of the nominator (from a_0 to a_N)
Sampling Frequency	Sampling frequency, in Hz

Example:

The following is a second-order transfer function:

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

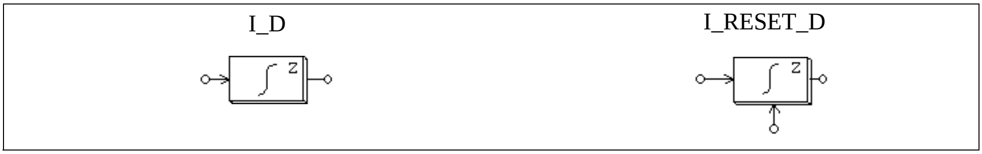
Assuming a sampling frequency of 3 kHz, the specification will be:

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

3.5.2.1 Integrator

There are two types of integrators. One is the regular integrator (I_D). The other is the resettable integrator (I_RESET_D).

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)
Sampling Frequency	Sampling frequency, in Hz

The output of a 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).

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

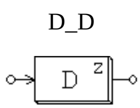
The transfer function of a discrete differentiator is:

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

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:

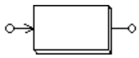
Parameter	Description
Sampling Frequency	Sampling frequency, in Hz

3.5.2.3 Digital Filters

Two types of digital filters are provided: general digital filter (FILTER_D / FILTER_D1) and finite impulse response (FIR) filter (FILTER_FIR / FILTER_FIR1). For blocks FILTER_D1 and FILTER_FIR1, filter coefficients are specified through a file.

Images:

FILTER_D / FILTER_D1



FILTER_FIR / FILTER_FIR1

Attributes:

For Filter_D and FILTER_FIR:

Parameters	Description
Order N	Order N of the transfer function
Coeff. $b_0...b_N$	Coefficients of the nominator (from b_0 to b_N)
Coeff. $a_0...a_N$	Coefficients of the nominator (from a_0 to a_N)
Sampling Frequency	Sampling frequency, in Hz

For Filter_D1 and FILTER_FIR1:

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:

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

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

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

$$[a_1 \cdot y(n-1) + a_2 \cdot y(n-2) + ... + a_N \cdot y(n-N)]$$

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:

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

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

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

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 fc = 1 kHz, assuming the sampling frequency fs = 10 kHz, using MATLAB, we have:

Nyquist frequency fn = fs / 2 = 5 kHz

Normalized cut-off frequency fc* = fc/fn = 1/5 = 0.2

[B,A] = butter (2, fc*)

which will give:

B = [0.0201 0.0402 0.0201] = [b_0 b_1 b_2]

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

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. $b_0 ... b_N$	0.0201 0.0402 0.0201
Coeff. $a_0 ... a_N$	1. -1.561 0.6414
Sampling Frequency	10000.

If the coefficients are stored in a file, the file content will be:

```
2
0.0201
0.0402
0.0201
1.
-1.561
0.6414
```

Or the file can also have the content as follows:

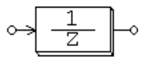
```
2
0.0201, 1
0.0402, -1.561
0.0201, 0.6414
```

3.5.3 Unit Delay

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

Image:

UDELAY



Attribute:

Parameter	Description
Sampling Frequency	Sampling frequency, in Hz

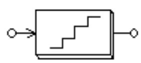
The difference between a unit delay block and a time delay block (TDELAY) is that the unit delay block is a discrete element and it delays the sampled points by one sampling period, whereas TDELAY is a continuous element and it delays the whole waveform by the delay time specified.

3.5.4 Quantization Block

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

Image:

DIGIT



Attributes:

Parameters	Description
No. of Bits	Number of bits N
Vin_min	Lower limit of the input value $V_{in,min}$
Vin_max	Upper limit of the input value $V_{in,max}$
Vo_min	Lower limit of the output value $V_{o,min}$
Vo_max	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 ΔV 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 ΔV.

Example:

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

$$V_{ox} = -5 + (3.2 - 0) * (5 - (05)) / (10 - 0) = -1.8$$

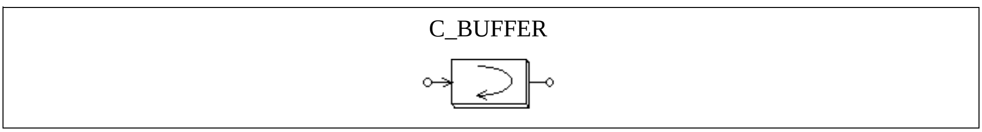
$$\Delta V = (5 - (-5)) / (2^4 - 1) = 0.66667$$

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:

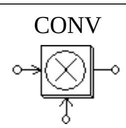
If a circular buffer has a buffer length of 4 and a sampling frequency of 10 Hz, we have the buffer storage at different time as follows:

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:

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

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

We have the convolution of A and B as:

$$C = A \otimes B$$

$$= [c_{m+n-1}\ c_{m+n-2}\ ...\ c_1]$$

where

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

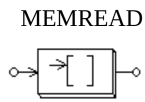
Example:

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

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.

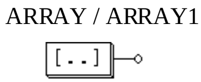
Example:

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 (in ARRAY) or specified in a file (in ARRAY1).

Image:



Attributes:

Parameters	**Description**
Array Length	The length of the data array *N* (for ARRAY only)
Values	Values of the array (for ARRAY only)
File for Coefficients	Name of the file storing the array (for ARRAY1 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:

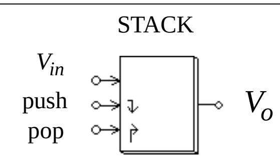
To define an array A = [2 4 6 8], we will have: Array Length =4; Values = 2 4 6 8. If the array is to be read from a file, the file will be:

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

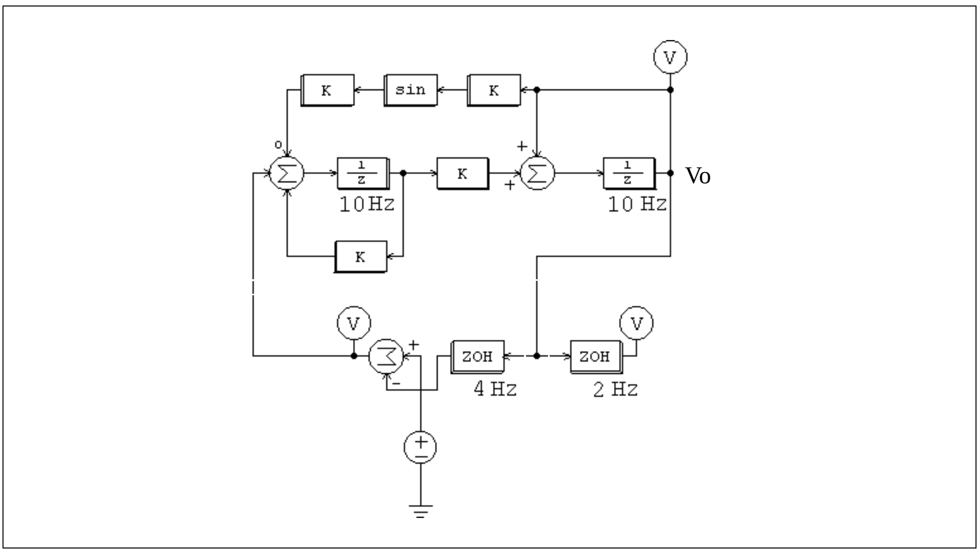
The rising edge triggers the push or pop action. When a pop action is performed and the stack is empty, the output remains unchanged. When a push action is performed and the stack is already full, the data at the bottom of the stack will be pushed out and will be lost.

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, Vo,fed back to the system and is sampled at 4 Hz in the second section. In the third section, the output is displayed at a sampling rate of 2 Hz.

It should be noted that a zero-order hold must be used between two elements with different sampling rates.

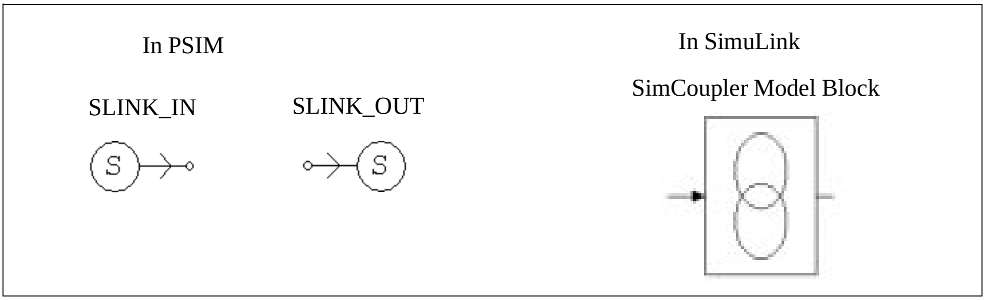


3.6 SimCoupler Module

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

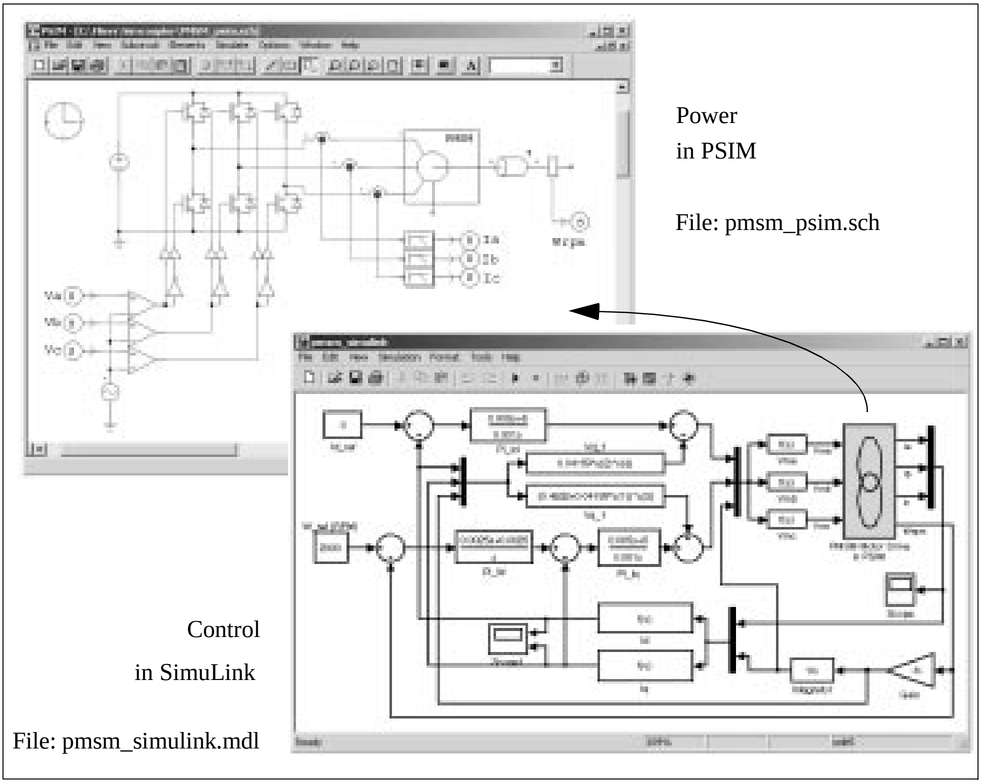
Images:



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.

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.

- Go to the **Simulate** menu, and select **Generate Netlist File**. A netlist file with the .cct extension will be generated and saved under the same directory as the schematic file. In this example, we assume that the netlist is located in the directory “C:\PSIM6.0”. The netlist file name and path will be “C:\PSIM6.0\pmsm_psim.cct”.

In Simulink

- Copy the version of the SimCoupler DLL file to “SimCoupler.dll”. For example, for Release 13, copy “SimCoupler_R13.dll” to “SimCoupler.dll”. Note: the default “SimCoupler.dll” file is for Release 11. It is found that it also works for higher releases.

- Start Matlab. Change the working directory to the PSIM directory. If PSIM is installed in the directory “C:\PSIM6.0”, change the directory to “C:\PSIM6.0”. Then launch Simulink and open the existing file or create a new file.

- After the rest of the system is created, open the Simulink file “SimCoupler_Block_R11.mdl” (created in Matlab/Simulink Release 11) that store the SimCoupler model block. Copy and paste the SimCoupler model block into the PMSM example file.

- In the PMSM example file, double click on the SimCoupler block, and enter the name and the location of the PSIM netlist name, and click on **Apply**. In this example, it will be “C:\PSIM6.0\pmsm_psim.cct”. The number of input and output ports for 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. If the number of link nodes in the netlist is changed later, go to the **Edit** menu and choose **Update Diagram**. This will update the model block 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 11, 12.0, 12.1, and 13.

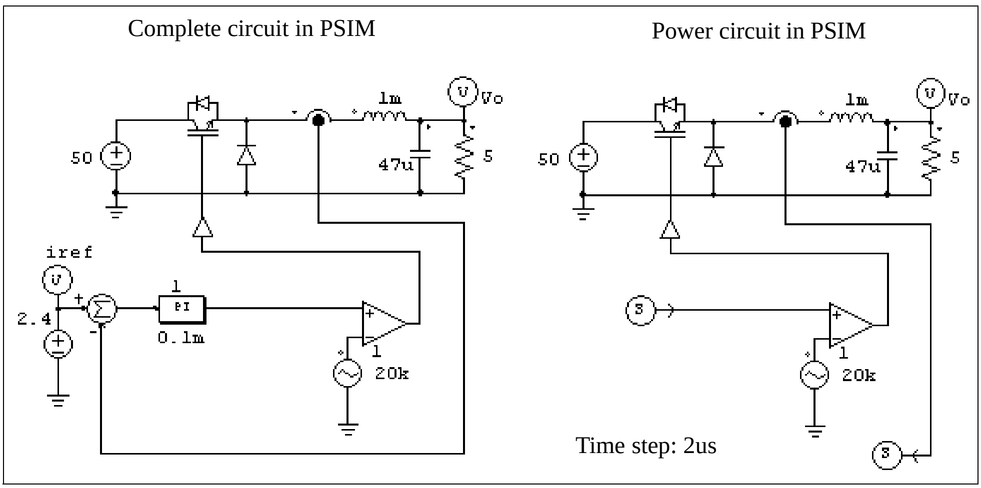
Please note that the SimCoupler file “SimCoupler.dll” is created in Matlab/Simulink Release 11. It is found that this file also works with higher releases of Matlab/Simulink. We also compiled and provided this file for other Matlab/Simulink releases. They are stored in “SimCoupler_R*xx*.dll” where *xx* is the release version number. For example, to use “SimCoupler.dll” compiled for Release 13, first delete “SimCoupler.dll”, then copy “SimCoupler_R13.dll” to another file, and rename that file to “SimCoupler.dll”.

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.

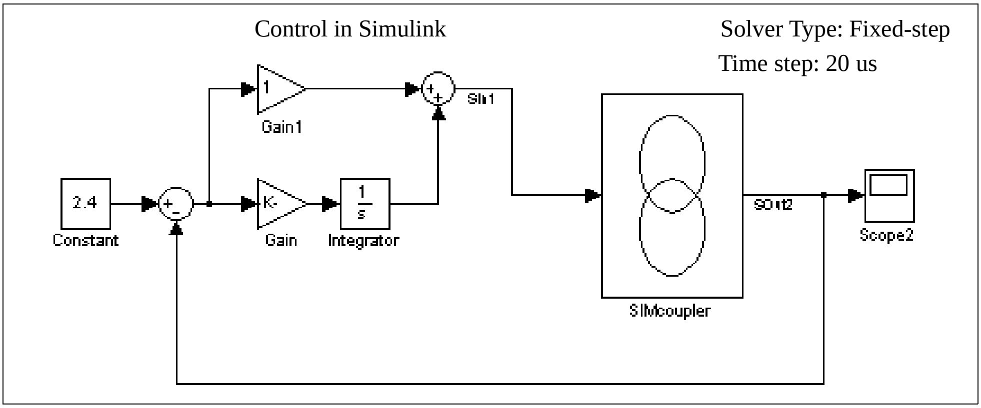
3.6.2 Solver Type and Time Step Selection in Simulink

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 us.

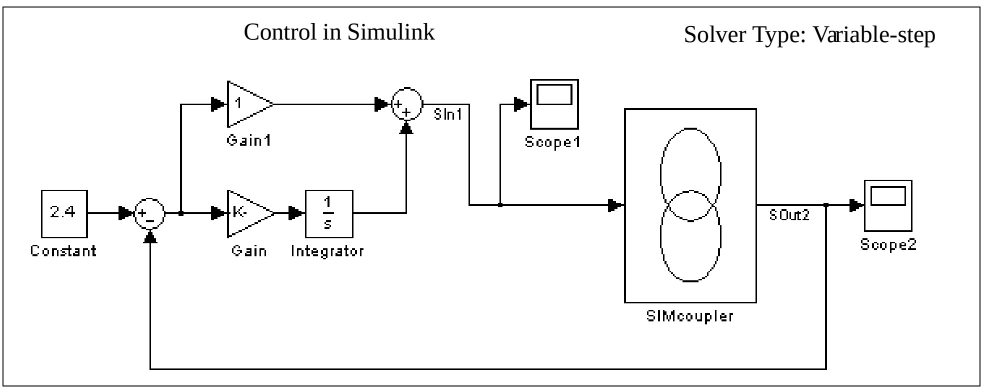


There are different ways of setting up Simulink to perform co-simulation. The recommend approach is to set the Solve Type to **Fixed-step** and define the **Fixed step size** to be the same or close to PSIM’s time step. The figure below shows this option.

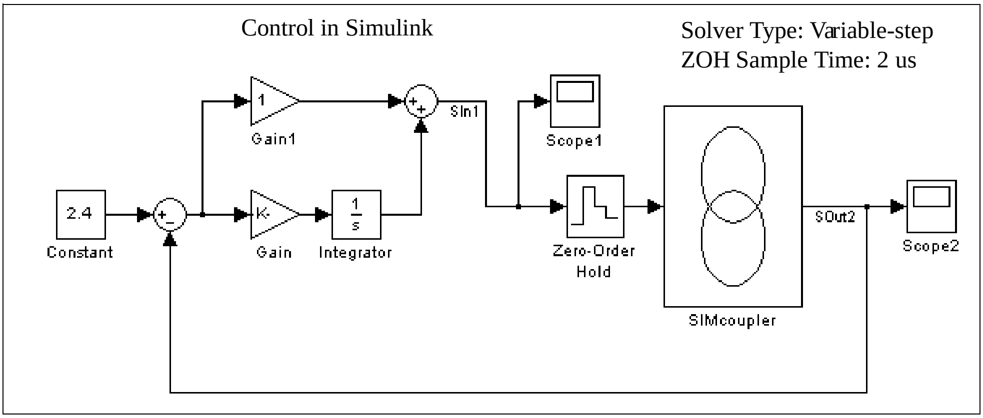


It is recommended that Simulink use the same time step as PSIM, although we have found that, even if the Simulink time step is slightly larger than PSIM time step, satisfactory results are obtained. In this case, for example, the time step is set to 20 us, 10 times larger than the PSIM time step.

If the Simulink Solver type is instead set to **Variable-step**, the simulation results will not be correct. The figure below shows this option.



When the Simulink Solver type is set to **Variable-step**, in order to obtain correct results, a zero-order-hold must be placed at the input of the SimCoupler model block. Moreover, the zero-order-hold sample time must be the same or close to PSIM time step. The figure below shows the configuration.



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

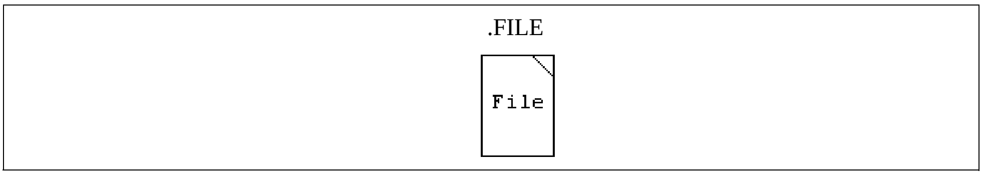
4

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

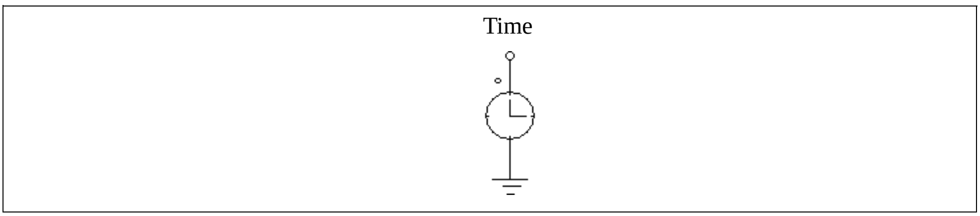
Several types of independent voltage/current sources are available in PSIM. The notation of a current source direction is: the current flows out of the higher-potential node, through the external circuit, and back into the lower-potential node of the source.

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.

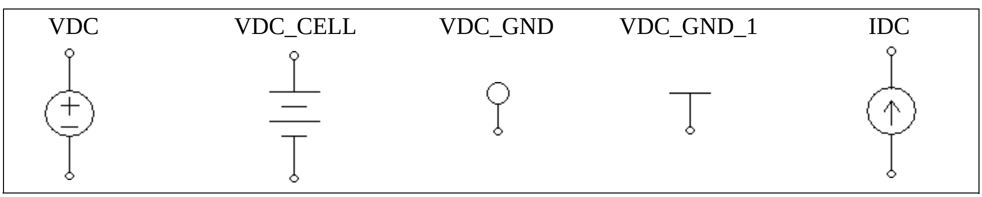
Image:



4.2.2 DC Source

A dc source has a constant amplitude. The reference of the dc voltage sources VDC_GND and VDC_GND_1 is the ground.

Images:



Attribute:

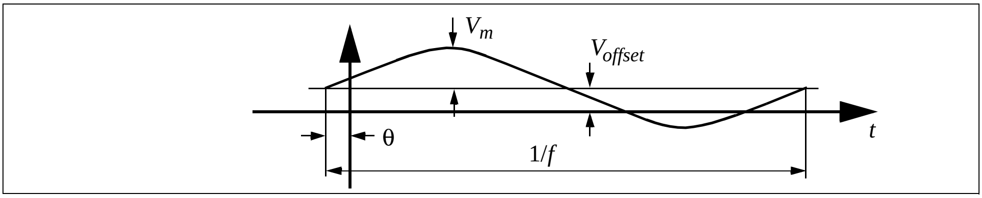
Parameter	Description
Amplitude	Amplitude of the source

4.2.3 Sinusoidal Source

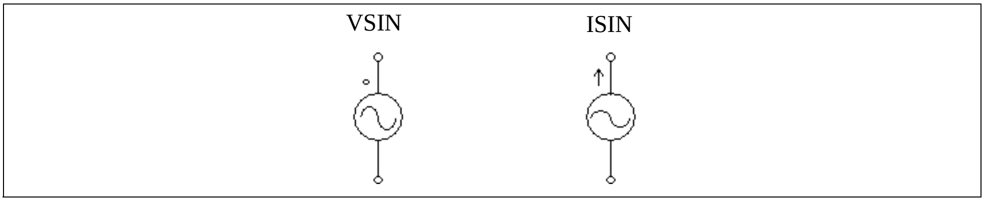
A sinusoidal source is defined as:

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

The specifications can be illustrated as follows.



Images:

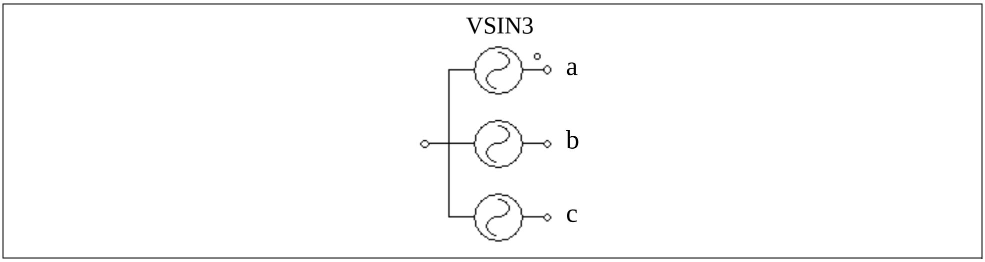


Attributes:

Parameters	Description
Peak Amplitude	Peak amplitude V_m
Frequency	Frequency f, in Hz
Phase Angle	Initial phase angle θ, 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 (VSIN3) is provided. The dotted phase of the module refers to Phase A.

Image:



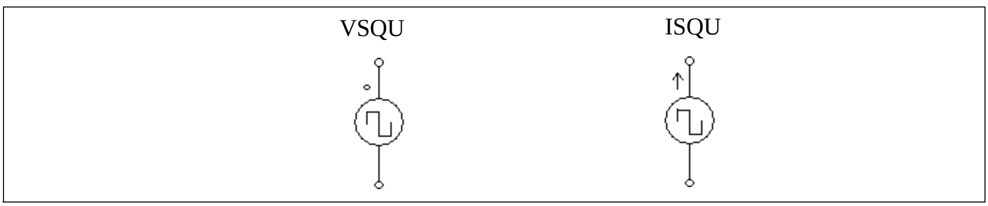
Attributes:

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 (VSQU) or current source (ISQU) is defined by peak-to-peak amplitude, frequency, duty-cycle, and DC offset. The duty cycle is defined as the ratio between the high-potential interval versus the period.

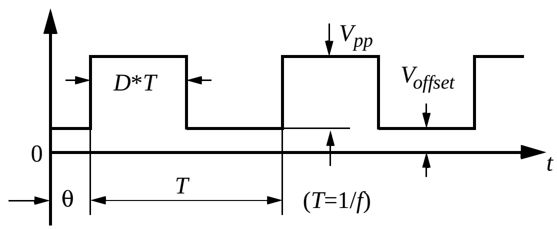
Images:



Attributes:

Parameters	Description
Vpeak-peak	Peak-to-peak amplitude V_{pp}
Frequency	Frequency, in Hz
Duty Cycle	Duty cycle D of the high-potential interval
DC Offset	DC offset V_{offset}
Phase Delay	Phase delay θ of the waveform, in deg.

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



When the phase delay θ is positive, the waveform is shifted to the right along the time axis.

4.2.5 Triangular Source

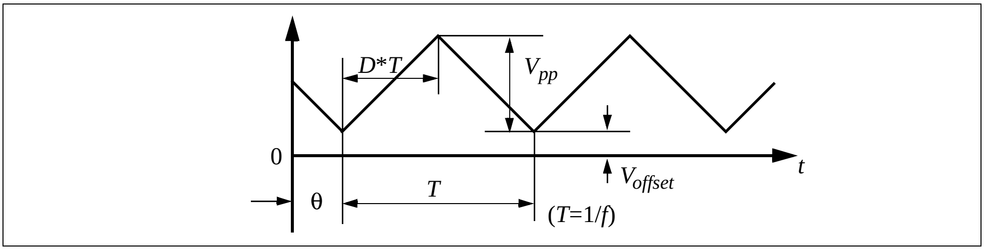
A triangular-wave voltage source (VTRI) or current source (ITRI) is defined by peak-to-peak amplitude, frequency, duty-cycle, and DC offset. The duty cycle is defined as the ratio between the rising-slope interval versus the period.

Images:

Attributes:

Parameters	Description
Vpeak-peak	Peak-to-peak amplitude V_{pp}
Frequency	Frequency, in Hz
Duty Cycle	Duty cycle D of the rising slope interval
DC Offset	DC offset V_{offset}
Phase Delay	Phase delay θ of the waveform, in deg.

The specifications of a triangular wave source are illustrated as:

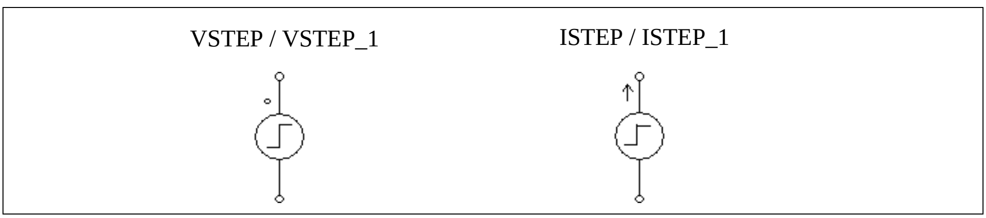


When the phase delay θ is positive, the waveform is shifted to the right along the time axis.

4.2.6 Step Sources

A step voltage/current source changes from one level to another at a given time.

Images:



Attributes:

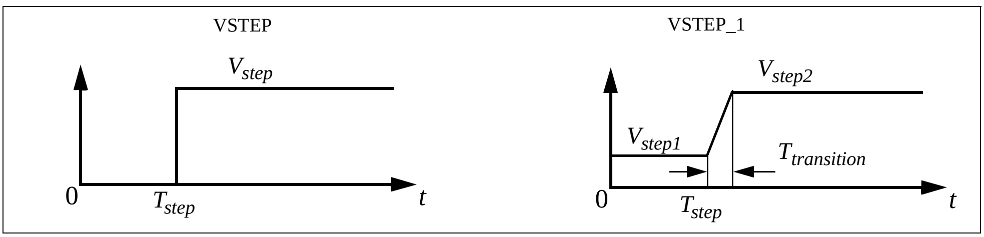
For VSTEP and ISTEP:

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

For VSTEP_1 and ISTEP_1:

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}

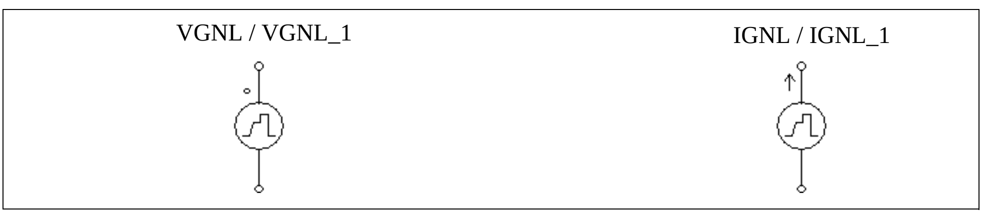
The specifications of the voltage step sources are illustrated as follows:



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.).

Images:



Attributes:

For VGNL and IGNL:

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.

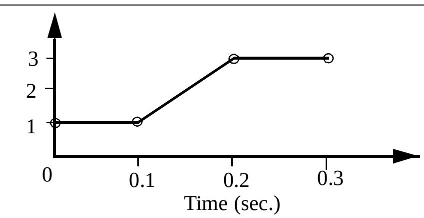
For VGNL_1 and IGNL_1:

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 VGNL 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 VGNL_1 will be:

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

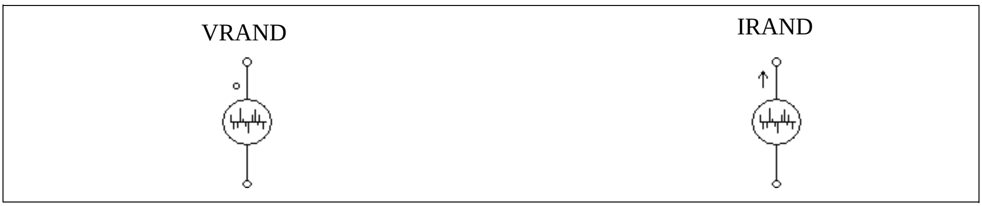
4.2.8 Random Source

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:



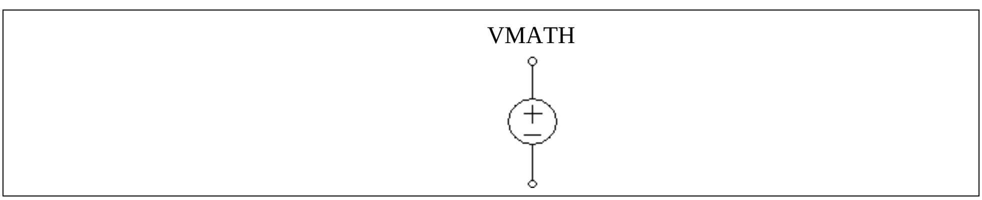
Attributes:

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

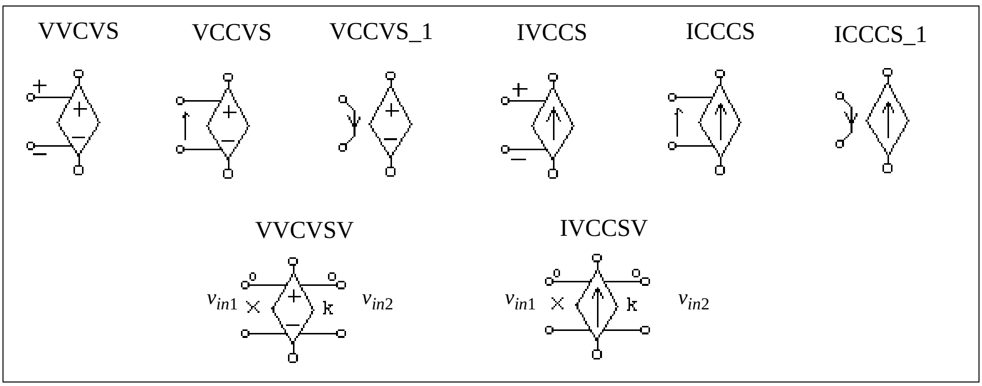
Four types of controlled sources are available:

- Voltage controlled voltage source (VVCVS)
- Current controlled voltage source (VCCVS / VCCVS_1)
- Voltage controlled current source (IVCCS)
- Current controlled current source (ICCCS / ICCCS_1)
- Variable-gain voltage controlled voltage source (VVCVSV)
- Variable-gain voltage controlled current source (IVCCSV)

The controlling current of a current controlled source (VCCVS / ICCCS) 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 VVCVS and IVCCS, the controlling voltage is from the positive node (+) to the negative node (-).

For current-controlled sources VCCVS and ICCCS, 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 VCCVS_1 and ICCCS_1, the controlling current flows into one control node and out of the other. 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 VVCVSV and IVCCSV, 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 variable-gain controlled sources and nonlinear sources VNONM and INONM described in the following section is that for VNONM and INONM, 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 variable-gain controlled sources, 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 o v_{in2}. The variable-gain controlled sources can be used in circuits which may otherwise have convergence problem with nonlinear sources VNONM and INONM.

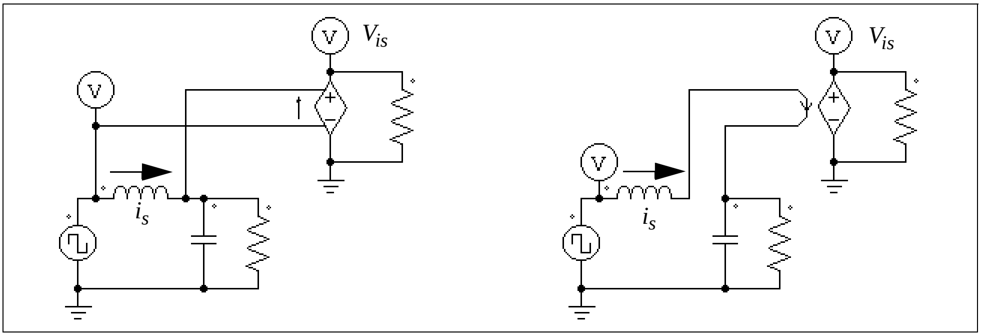
Example:

The circuits below illustrates the use of current controlled voltage sources VCCVS and VCCVS_1.

In the circuit on the left, the voltage source VCCVS 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 the source VCCVS_1 is used instead.



4.2.11 Nonlinear Voltage-Controlled Sources

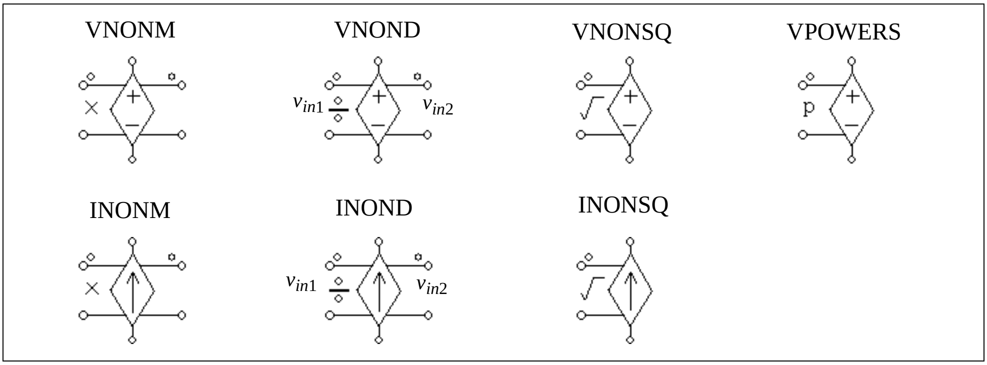
The output of a nonlinear voltage-controlled source is either the multiplication, division, or square-root of the inputs. They are defined as:

VNONM	- Voltage source where $v_o = k \cdot v_{in1} \cdot v_{in2}$
INONM	- Current source where $i_o = k \cdot v_{in1} \cdot v_{in2}$
VNOND	- Voltage source where $v_o = k \cdot \frac{v_{in1}}{v_{in2}}$
INOND	- Current source where $i_o = k \cdot \frac{v_{in1}}{v_{in2}}$
VNONSQ	- Voltage source where $v_o = k \cdot \sqrt{v_{in1}}$
INONSQ	- Current source where $i_o = k \cdot \sqrt{v_{in1}}$
VPOWERS	- Voltage source where $v_o = sign(v_{in}) \cdot k \cdot (k_1 \cdot v_{in})^{k_2}$

In VPOWERS, the term sign(v_{in}) is 1 if v_{in} is positive, and it is -1 if v_{in} is negative.

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

Images:



Attributes:

For all the sources except VPOWERS:

Parameter	**Description**
Gain	Gain *k* of the source

For VPOWERS:

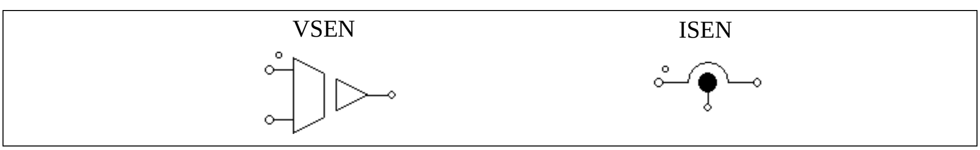
Parameters	**Description**
Gain	Gain *k* of the source
Coefficient k_1	Coefficient k_1
Coefficient k_2	Coefficient k_2

For VNOND and INOND, 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 μΩ.

Images:



Attribute:

Parameter	Description
Gain	Gain of the sensor

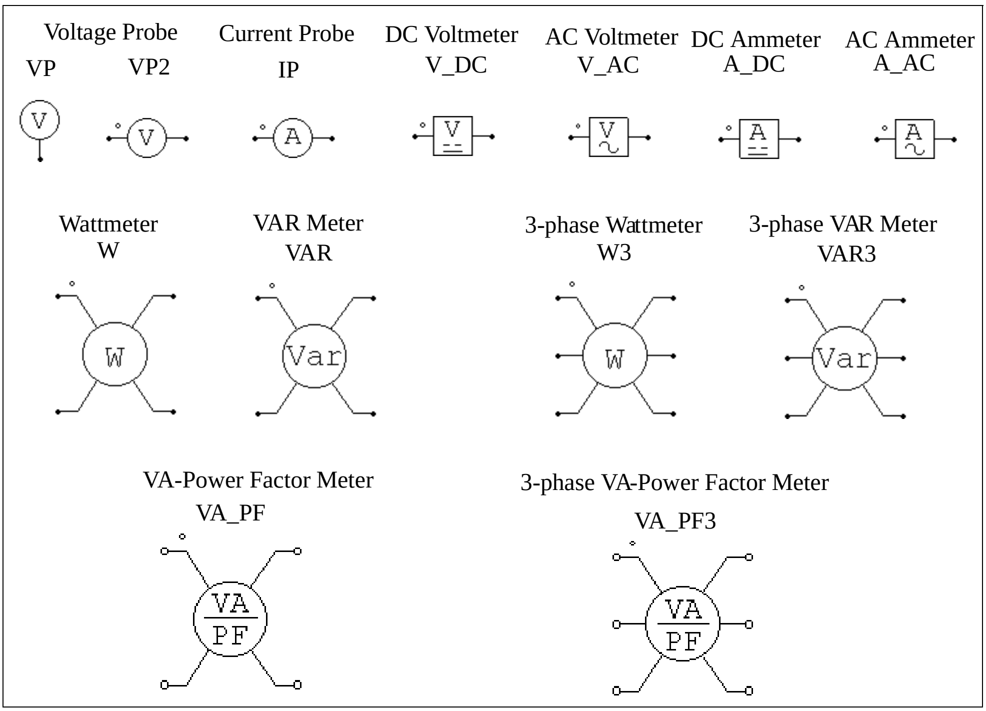
4.4 Probes and Meters

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 μΩ is used in the current probe internally to measure the current.

Images:



Attributes:

Parameters	**Description**
Operating Frequency	Operating frequency or fundamental frequency of the ac meter, in Hz
Cut-off Frequency	Cut-off frequency of the low-pass/high-pass filter, in Hz
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 (VP / VP2 / IP), 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.

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

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

where ω_1 is the fundamental frequency and all others are harmonic frequencies. We have the rms values of the voltage and current as:

$$V_{rms} = \sqrt{V_1^2 + V_2^2 + ..}$$

$$I_{rms} = \sqrt{I_1^2 + I_2^2 + ..}$$

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:

$$v_a + v_b + v_c = 0$$

$$i_a + i_b + i_c = 0$$

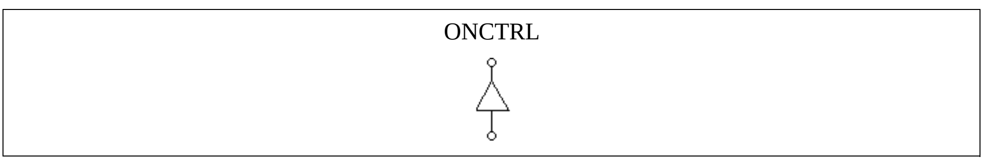
4.5 Switch Controllers

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.

4.5.1 On-Off Switch Controller

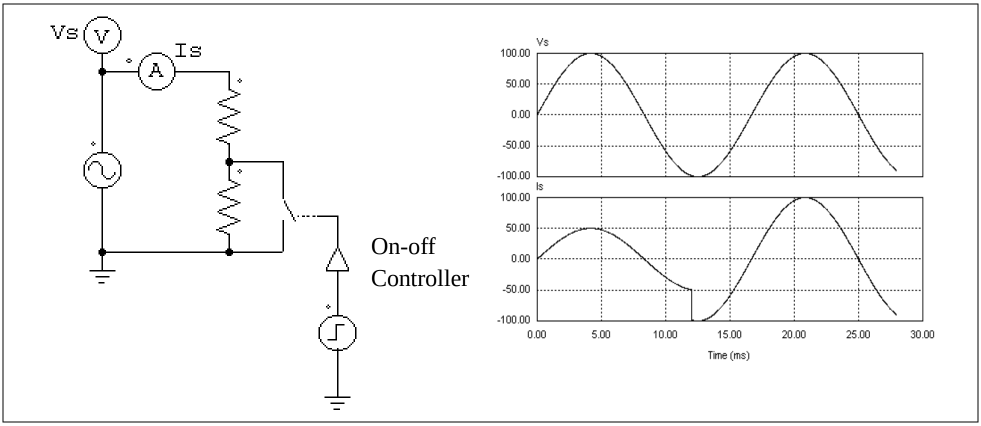
On-off switch controllers are used as the interface between control gating signals and power switches. The input, which is a logic signal (either 0 or 1) from the control circuit, is passed to the power circuit as the gating signal.

Image:



Example:

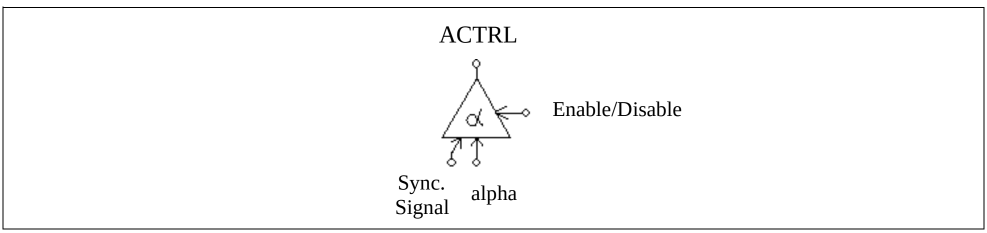
The circuit below implements the step change of a load. In the circuit, the on-off switch controller is used to control the bi-directional switch. The step voltage source, which is connected to the controller input, changes from 0 to 1 at the time of 12 ms. The closure of the switch results in the short-circuit of the resistor across the switch and the increase of the current.



4.5.2 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:



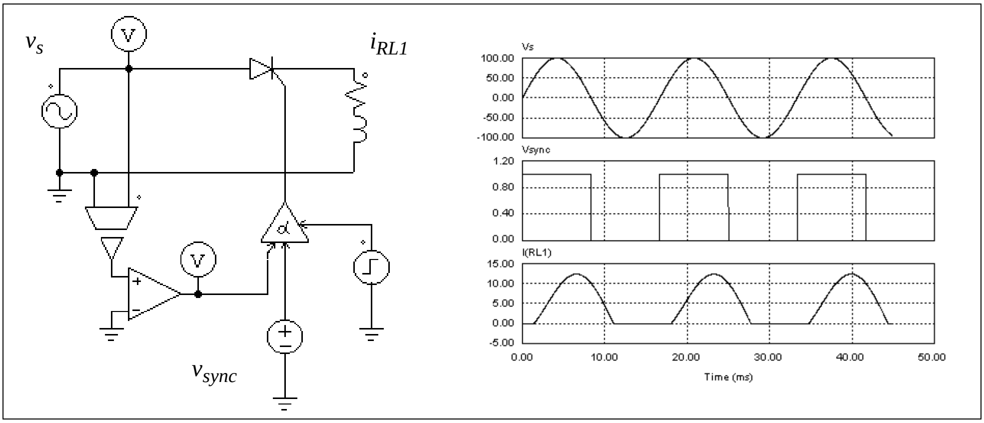
Attributes:

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:

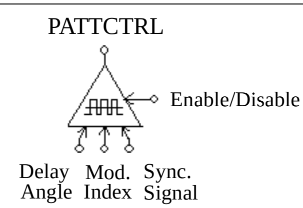
The figure below shows a thyristor circuit using delay angle control. In the circuit, the zero-crossing of v_s, which corresponds to the moment that the thyristor would start conducting naturally, is used to provide the synchronization. The delay angle is set at 30^o. The gating signal is delayed from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



4.5.3 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^o, the gatings are updated at every cycle. If it is 60^o, the gatings are updated at every 60^o.
File Name	Name of the file storing the PWM gating pattern

A lookup table, which is stored in a file, contains the gating patterns. It has the following format:

n, m_1, m_2, ..., m_n
k_1
$G_{1,1}$, $G_{1,2}$, ..., $G_{1,k1}$
... ...
k_n
$G_{n,1}$, $G_{n,2}$, ..., $G_{n,kn}$

where n is the number of gating patterns; m_i is the modulation index correspondent to Pattern i; and k_i is the number of switching points in Pattern i. The modulation index array m_1 to m_n should be monotonically increasing. The output will select the i_{th} pattern if the input is smaller than or equal to m_i. If the input exceeds m_n, the last pattern will be selected.

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

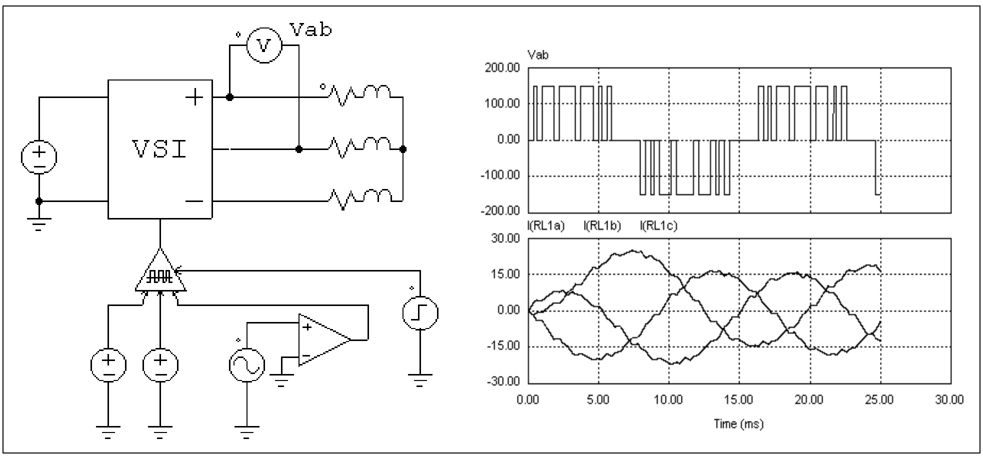
```
5, 0.901, 0.910253, 0.920214, 1.199442, 1.21
14
7.736627   72.10303   80.79825   99.20176   107.8970   172.2634   180.
187.7366   252.1030   260.7982   279.2018   287.8970   352.2634   360.
14
7.821098   72.27710   80.72750   99.27251   107.7229   172.1789   180.
187.8211   252.2771   260.7275   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.

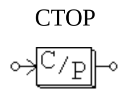


4.6 Function Blocks

4.6.1 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.

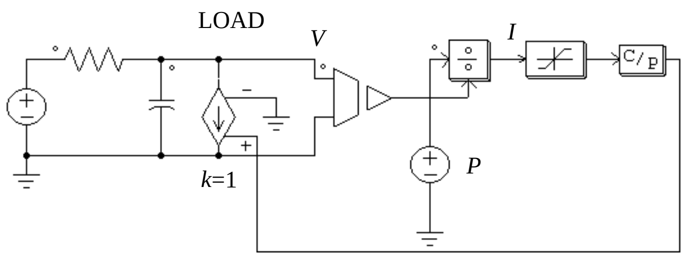
Image:



Example: A Constant-Power Load Model

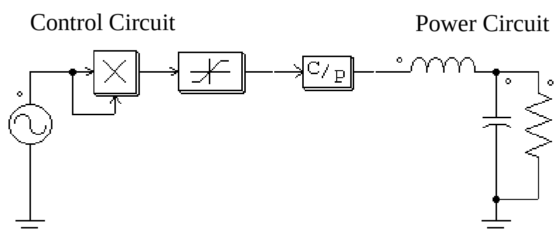
In a constant-power dc load, the voltage *V*, current *I*, and power *P* have the relationship as *P*=*V***I*. Given the voltage and the power, the current can be calculated as *I*=*P*/*V*. This can be implemented using the circuit as shown below.

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 behaviors as a grounded voltage source.



4.6.2 ABC-DQO Transformation Block

Function blocks ABC2DQO and DQO2ABC 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.

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.

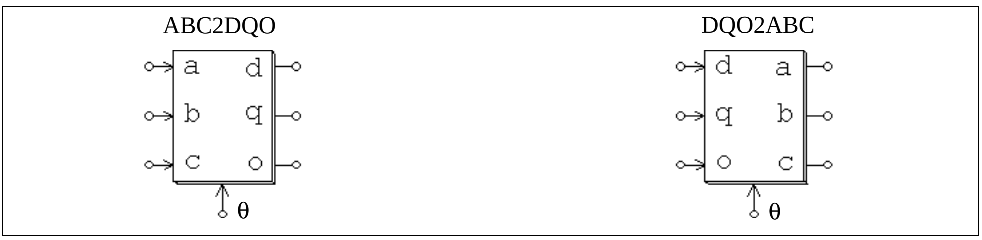
The transformation equations from abc to dqo are:

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

The transformation equations from dqo to abc are:

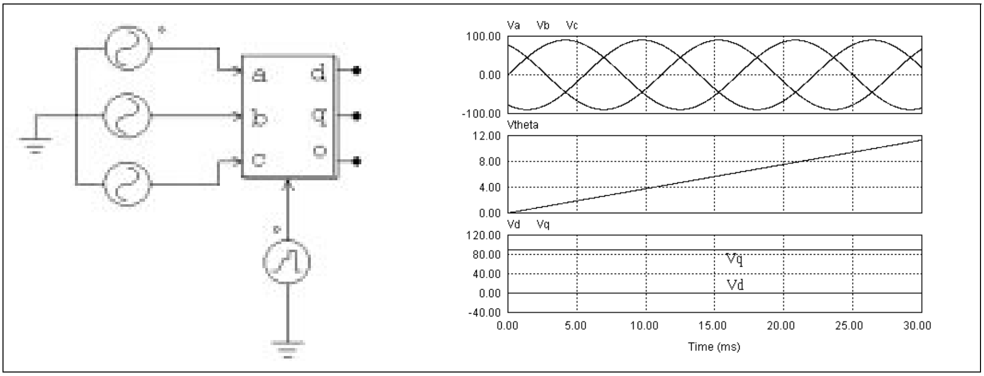
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix}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

Images:



Example:

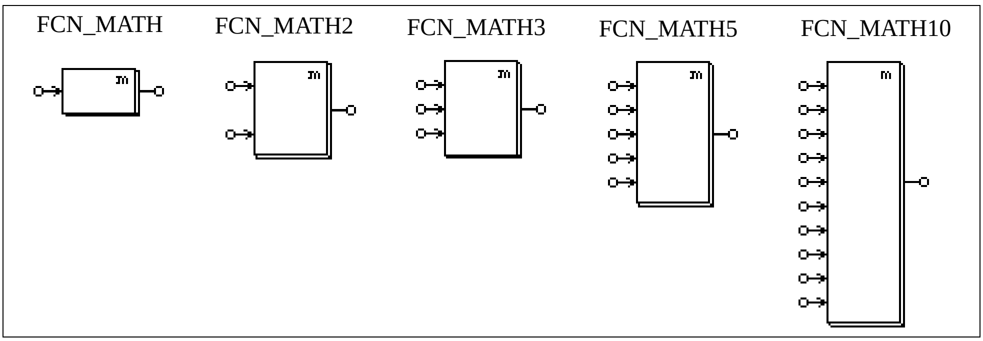
In this example, three symmetrical ac waveforms are transformed into dqo quantities. The angle θ is defined as θ = ωt where ω = 2π*60. Since the angle θ changes linearly with time, a piecewise linear voltage which has a ramp waveform is used to represent θ. The simulation waveforms show the three-phase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



4.6.3 Math Function Blocks

The output of a math function block is expressed as the mathematical function of the inputs. With this block, one can implement complex and nonlinear relationship easily. Blocks with 1, 2, 3, 5, and 10 inputs are provided.

Images:



Attributes:

Parameters	Description
Expression $f(x_1,x_2,...,x_n)$	Expression of the output versus inputs where *n* is the number of inputs
Expression df/dx_i	Expression of the derivative of the function *f* versus the i_{th} input

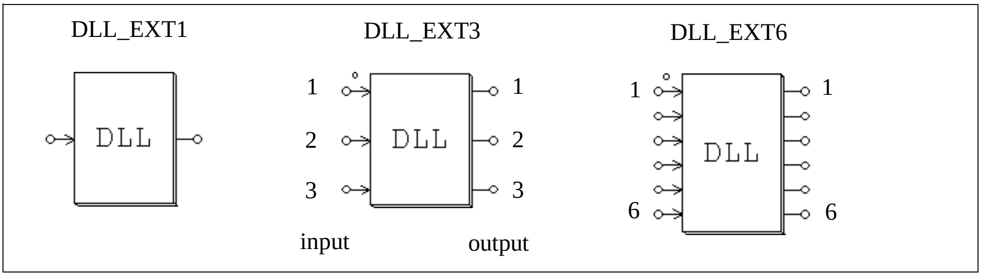
The derivative df/dx_i can be set to zero.

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

4.6.4 External DLL Blocks

An external DLL (dynamic link library) block allows users to write code in C/C++ or Fortran language, compile it into DLL using either Microsoft C/C++, Borland C++, or Digital Visual Fortran, and link it with PSIM. These blocks can be used in either the power circuit or the control circuit.

Images:



Attribute:

Parameter	Description
File Name	Name of the DLL file

The node with a dot is for the first input (in[0]).

The name of the DLL file can be arbitrary. The DLL file, however, must be in the same directory as the schematic file that uses the DLL file.

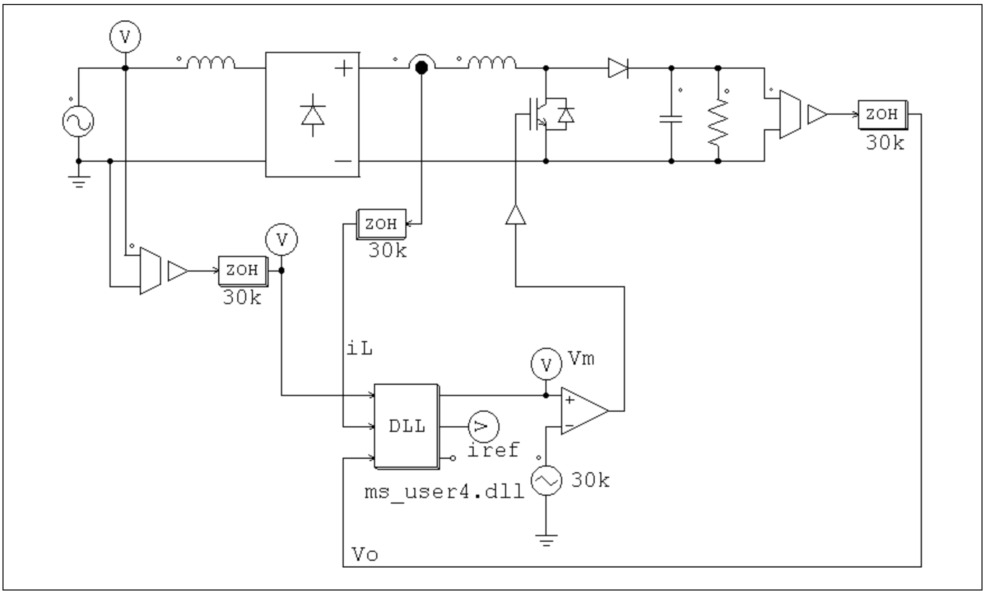
A DLL block receives the values from PSIM as the input, performs the calculation, and sends the output back to PSIM. PSIM calls the DLL routine at each simulation time step. However, when the inputs of the DLL block are connected to one of these discrete elements (zero-order hold, unit delay, discrete integrators and differentiators, z-domain transfer function blocks, and digital filters), the DLL block is called only at the discrete sampling times.

Sample files are provided for Microsoft C/C++, Borland C++, and Fortran routines. 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 the 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
}
```

5

Analysis Specification

5.1 Transient Analysis

Parameters for the transient analysis are defined by selecting **Simulation Control** in the **Simulate** menu in PSIM, as follows.

Time Step	Simulation time step, in sec.
Total Time	Total simulation time, in sec.
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 Version 6.0, 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.doc”.

5.2 AC Analysis

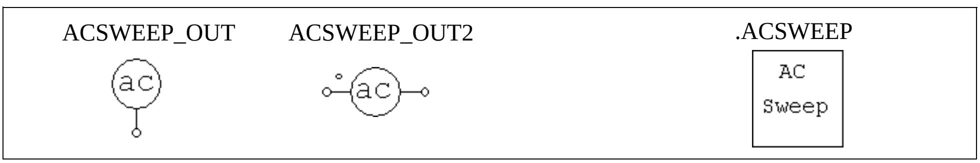
The frequency response of a circuit or a control loop can be obtained with the ac analysis. A key feature of the ac analysis in PSIM is that, a circuit can be in its original switchmode form, and no average model is required. Nevertheless, with the average model, the time it takes to perform the ac analysis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

The following are the steps to set up the ac analysis:

- Identify a sinusoidal source (VSIN) as the excitation source for the ac sweep.
- Place ac sweep probes (ACSWEEP_OUT) at the desired output location. To measure the loop response of a closed control loop, use the node-to-node probe (ACSWEEP_OUT2).
- Place the .ACSWEEP element on the schematic, and define the parameters of the ac sweep.
- Run PSIM.

Below are the images of the ac sweep probes and the .ACSWEEP 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

Parameters	Description
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.

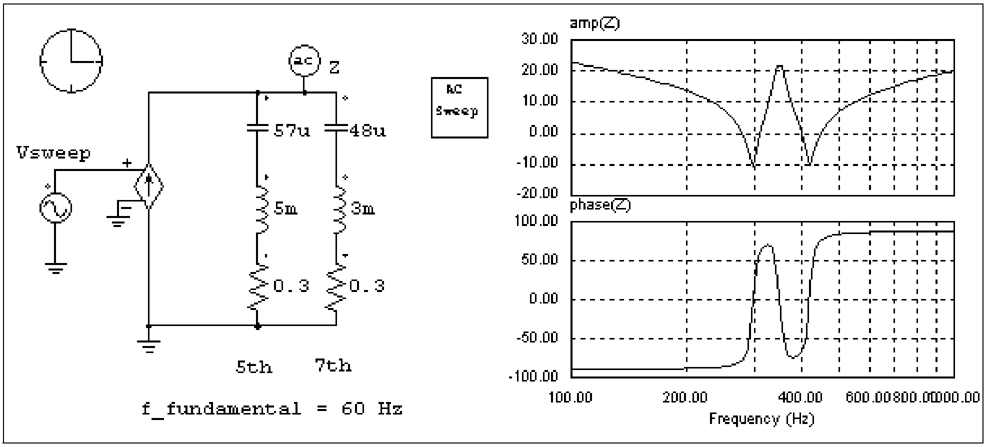
Sometimes, after ac analysis is complete, a warning message is displayed as follows:

> **Warning:** The program did not reach the steady state after 60 cycles. See File “message.doc” 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.doc” 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.

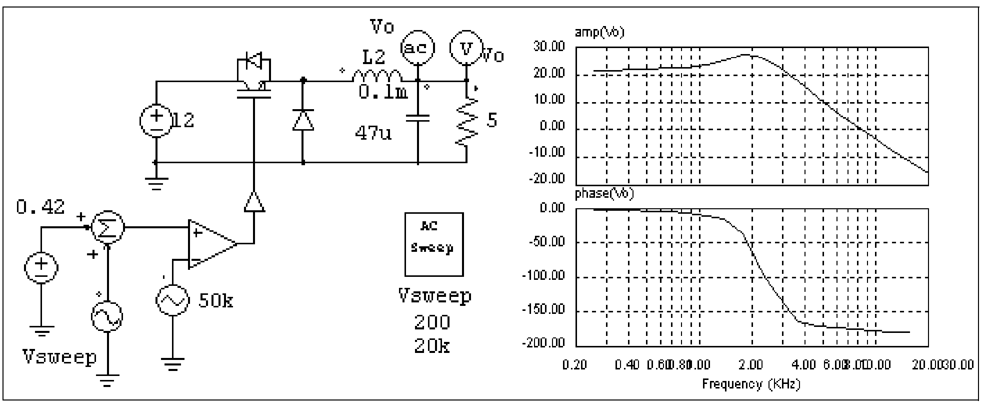
Example: Impedance of Shunt Filters

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

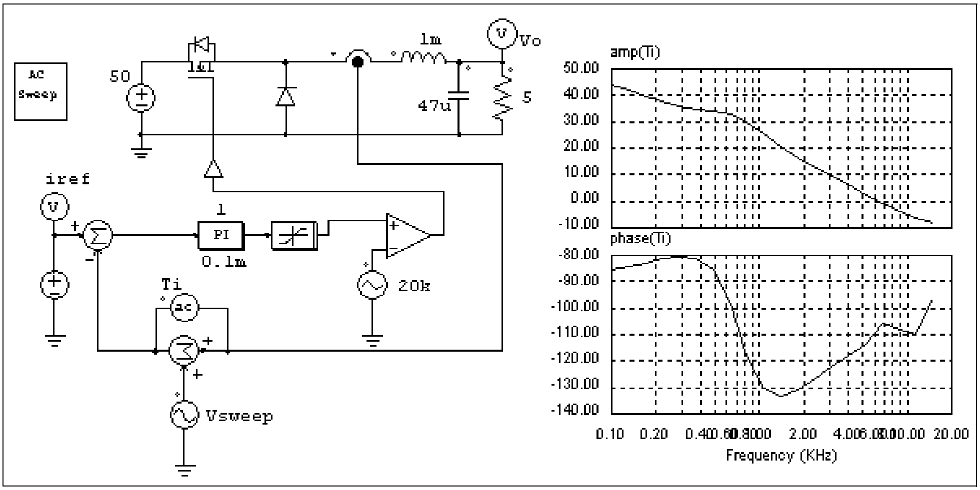
The circuit on the left is an one-quadrant buck converter. An excitation source is injected to the modulation signal, and the output voltage is measured. The result of the ac analysis, on the right, shows the open-loop response of the output voltage versus the modulation signal.



Example: Loop Transfer Function of a Closed-Loop Circuit

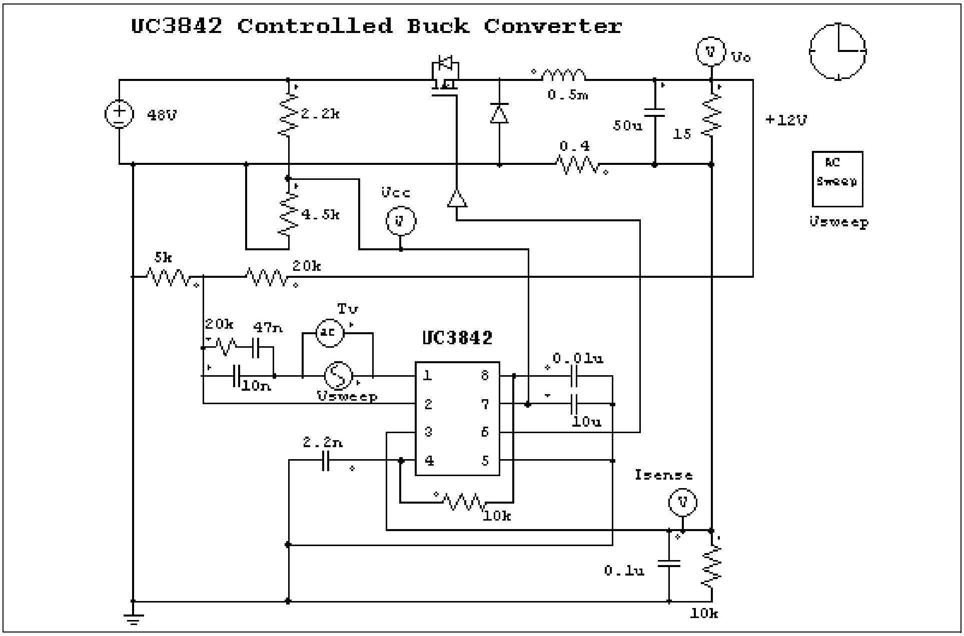
The ac analysis can be used to find out the loop response of a closed-loop system. The circuit below shows a buck converter with average current mode control. By injecting the excitation signal into the current feedback path, and using the node-to-node ac sweep probe (ACSWEEP_OUT2), we can obtain the 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 Unitrode UC3842. The excitation source can be inserted in the feedback path, before the op. amp. output.



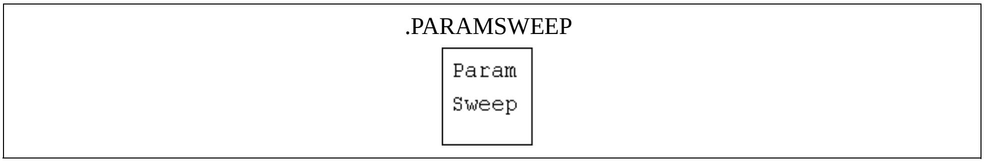
5.3 Parameter Sweep

Parameter sweep can be performed for the following parameters:

- Resistance, inductance, and capacitance of RLC branches
- Gain of proportional blocks (P)
- Time constant of integrators (I)
- Gain and time constant of proportional-integral controllers (PI)
- Gain, cut-off frequency, and damping ratio of 2nd-order low-pass and high-pass filters (FILTER_LP2 / FILTER_HP2)
- Gain, center frequency, and passing and stopping band of 2nd-order bandpass and bandstop filters (FILTER_BP2/ FILTER_BS2)

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

Image:



Attributes:

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

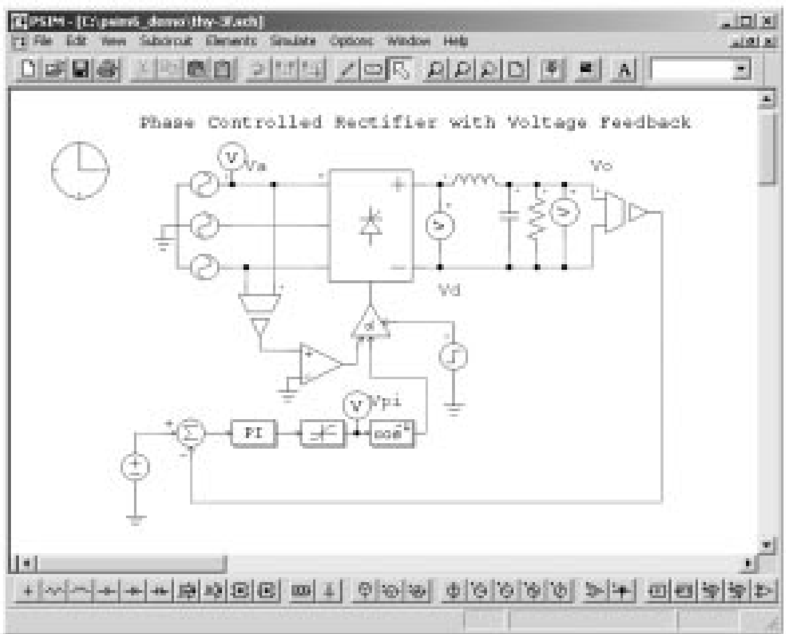
For example, let the resistance of a resistor be "Ro". To sweep the resistance from 2 Ohm to 10 Ohm, with a step of 2 Ohm, the specification will be:

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

6

Circuit Schematic Design

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



In PSIM, all the elements are stored under the menu **Elements**. 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 in PSIM for circuit creation.

Get To get an element from the element library, click on the **Elements** menu. Choose the submenu and highlight the element to be selected.

For example, to get a dc voltage source, click on **Element** , **Sources**, and **Voltage**, then highlight **DC**.

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 Once an element is selected, select **Rotate** to rotate the element.

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

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

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

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

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

6.2 Editing a Circuit

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

Select To select an element, click on the element. A rectangle will appear around the element.

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.

Copy To copy an element or a block of the circuit, select the element or the region, and choose **Copy**. Then choose **Paste** place the element or circuit.

Delete To delete an element, a block of a circuit, or a wire, select the item, and choose **Cut**, or hit the <Delete> key. Note that if **Cut** is used, the last deleted item can be pasted back. This is equivalent to un-do.

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

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

Disable 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.

Enable To enable a previously disabled element or circuit.

Zoom Select **Zoom In** to zoom in the circuit, or **Zoom In Selected** to zoom in to a selected region. Choose **Zoom Out** to zoom out, or **Fit to Page** to zoom out to fit the entire circuit to the screen.

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

6.3 Subcircuit

The following functions are provided for subcircuit editing and manipulation.

New Subcircuit To create a new subcircuit

Load Subcircuit To load an existing subcircuit. The subcircuit will appear on the screen as a block.

Edit Subcircuit To edit the size and file name of the subcircuit

Set Size To set the size of the subcircuit

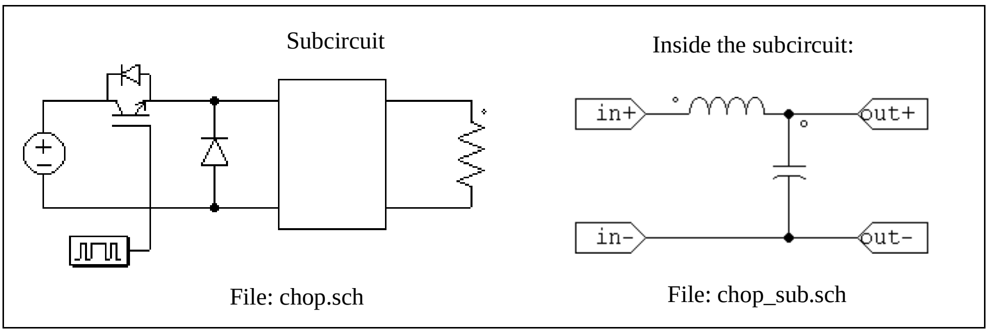
Place Port To place the connection port between the main circuit and the subcircuit

Display Port To display the connection port of the subcircuit

Edit Default Variable List To edit the default variable list of the subcircuit

Edit Image	To edit the subcircuit image
Display Subcircuit Name	To display the subcircuit name
Show Subcircuit Ports	To display the port names of the subcircuit in the main circuit
Hide Subcircuit Ports	To hide the port names of the subcircuit in the main circuit
Subcircuit List	To list the file names of the main circuit and the subcircuits
One Page up	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.



6.3.1 Creating Subcircuit - In the Main Circuit

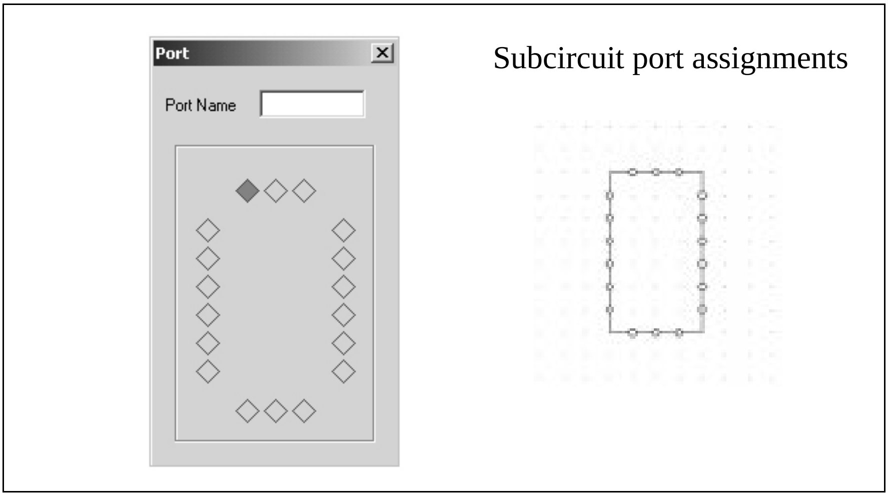
The following are the steps to create the subcircuit “chop_sub.sch” in the main circuit “chop.sch”.

- 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.

6.3.2 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.



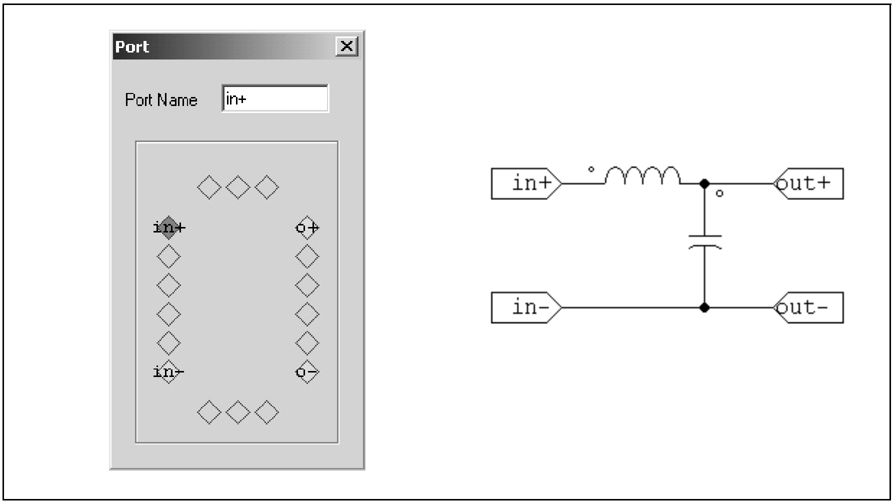
The diamonds on the four sides represent the connection nodes and the positions of the subcircuit. They correspond to the connection nodes of the subcircuit block on the right. There are no diamonds at the four corners since connections to the corners are not permitted.

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.

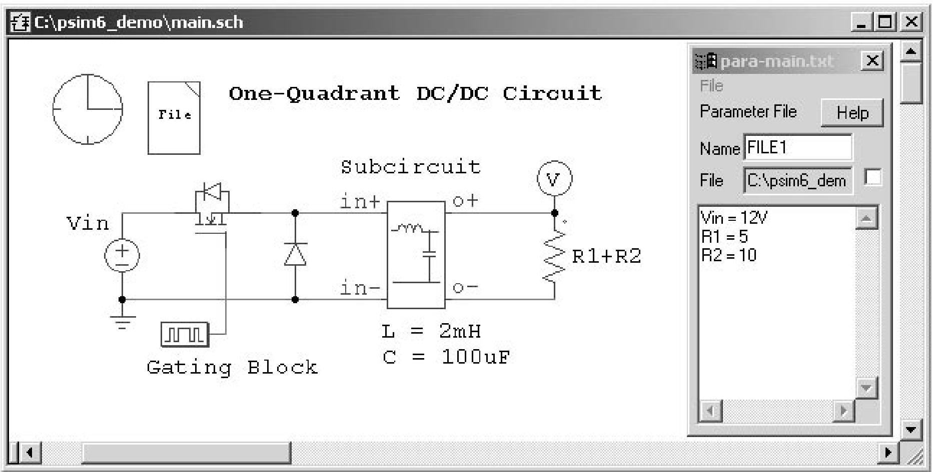
6.3.3 Connecting Subcircuit - In the Main Circuit

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 Port** in the Subcircuit menu to display the port names as defined inside the subcircuit.
- Connect the wires to the connection points accordingly.

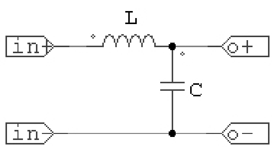
6.3.4 Other Features of the Subcircuit

This section describes other features of the subcircuit through an example.



File: main.sch

Inside the subcircuit:



File: sub.sch

6.3.4.1 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.

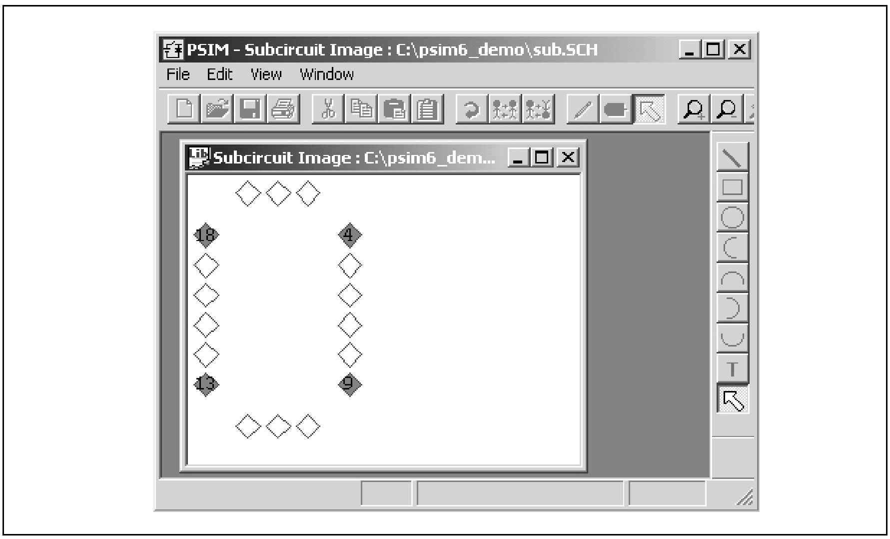
This feature allows the parameters of a subcircuit to be defined at the main circuit level. In the case where the same subcircuit is used several times in one main circuit, different parameters can be assigned to the same variable. For example, if the subcircuit "sub.sch" is used two times in above example, 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.

6.3.4.2 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

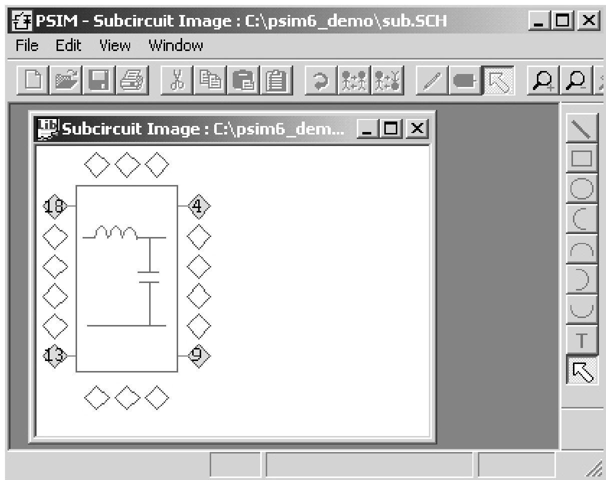


In the window, the diamonds marked red are the connection nodes of the subcircuit block, in exactly the same positions as appearing in the main circuit.

- 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.

6.3.4.3 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 **Element** 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 **Element** 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.

6.4 Other Options

6.4.1 Running the Simulation

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

To view the simulation results, choose **Run SIMVIEW** from the **Simulate** menu. Refer to Chapter 7 for the use of SIMVIEW.

6.4.2 Generate and View the Netlist File

To generate the netlist, choose **Generate Netlist File** from the **Simulate** menu. 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 **View Netlist File** from the **Simulate** menu.

6.4.3 Define Runtime Display

One can view selected waveforms as the simulation runs. This is useful if one wishes to monitor and abort a simulation if needed. The waveforms that can be displayed in the runtime will be selected from the list of outputs defined in the circuit.

6.4.4 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 message** is checked, warning messages generated during the simulation will be suppressed. Otherwise, warning messages will be shown before waveforms are displayed in SIMVIEW.

6.4.5 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 Select** .

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**.

6.5 Editing PSIM Library

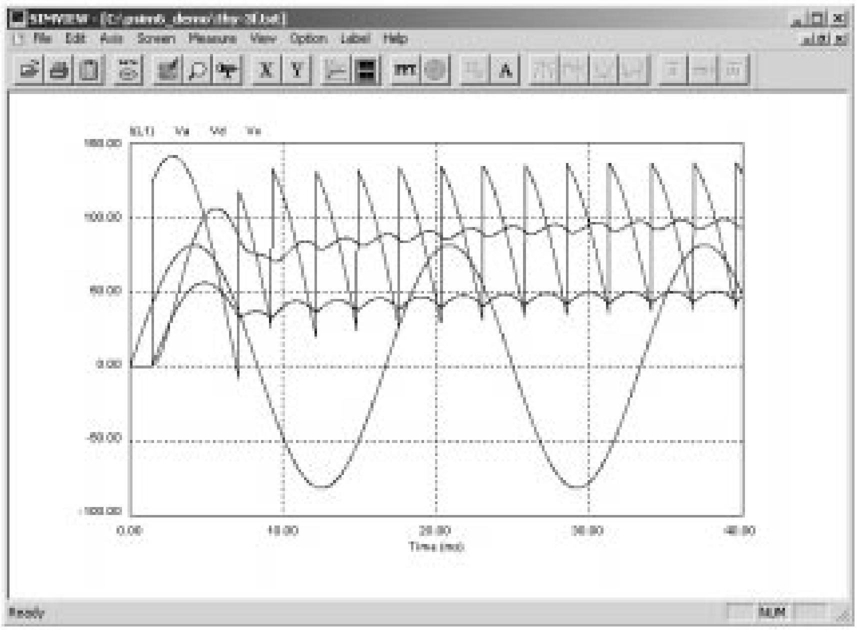
The PSIM library consists of two parts: one is the image library (psimimage.lib) and the other is the netlist library (psim.lib). The netlist library can not be modified. But users can modify the image library, or create their own image library.

To create or modify the image library, go to **Edit / Edit Library / Edit Library Files**, and follow the instructions on the screen.

7

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 I(L1) V(o) V(a) V(pi)
 0.1000000E-04 0.000000E+00 -0.144843E-18 0.307811E+00 0.100000E+01
 0.2000000E-04 0.000000E+00 -0.289262E-18 0.615618E+00 0.100000E+01
 0.3000000E-04 0.000000E+00 -0.576406E-18 0.923416E+00 0.100000E+01
 0.4000000E-04 0.000000E+00 -0.860585E-18 0.123120E+01 0.100000E+01
 0.5000000E-04 0.000000E+00 -0.114138E-17 0.153897E+01 0.100000E+01
... ... ... ...
```

Functions in each menu are explained below.

7.1 File Menu

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

Open	Load text data file
Open Binary	Load SIMVIEW binary file
Merge	Merge another data file with the existing data file for display
Re-Load Data	Re-load data from the same text file
Save	In the time display, save waveforms to a SIMVIEW binary file with the .smv extension. In the FFT display, save the FFT results to a text file with the .fft extension. The data range saved will be the same as shown on the screen.
Save As	In the time display, save waveforms to a SIMVIEW binary file specified by the user. In the FFT display, save the FFT results to a text file specified by the user.
Print	Print the waveforms
Print Setup	Set up the printer
Print Page Setup	Set up the hardcopy printout size
Print Preview	Preview the printout
Exit	Quit SIMVIEW

When the data of a text file are currently being displayed, after new data of the same file have become available, by selecting **Re-Load Data**, waveforms will be re-drawn based on the new data.

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. Note that 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:

Copy to Clipboard	Copy the waveforms to the clipboard

Edit Title Edit the printout title. By default, the title shows the file name and path.

7.3 Axis Menu

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

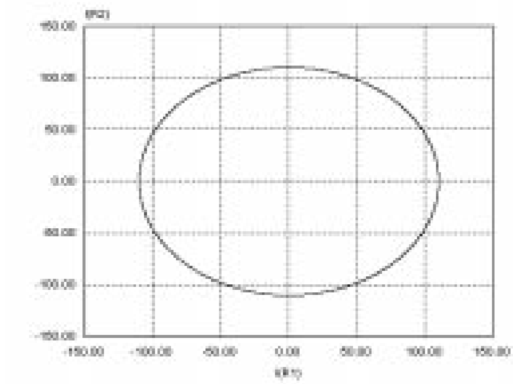
X Axis	Change the settings of the X axis
Y Axis	Change the settings of the Y axis
Axis Label Setting	Change the settings of the X/Y axis label
Edit Default	If the item is checked, the first column, which is usually Time, will be
Variable List	used as the X axis

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

If the *Auto-Scale* box is checked and the *Grid Division* is chosen as default, 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.

In the **Axis Label Setting**, the label font size can be changed, and the display of the label can be disabled.

By default, the option **Default X-Axis: Time** is selected. That is, the first column of the data, which is usually Time, is used as the X axis. If this option is not selected, 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.



Note that this option can only be selected or de-selected when there are no documents in the SIMVIEW environment.

7.4 Screen Menu

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

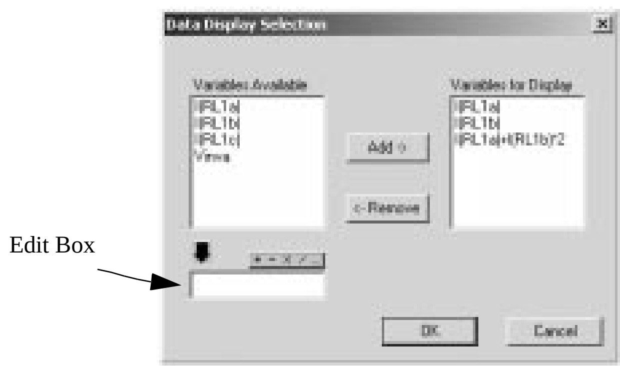
Add/Delete Curves Add or delete curves from the selected screen

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 dialog box of the **Add/Delete Curves** function 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.

7.5 View Menu

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

Zoom To zoom into a selected region

Re-Draw To re-draw the waveform using the auto-scale

Measure To measure the values of the waveforms

Escape To escape from the Zoom or Measure mode

Max To find the global maximum of a selected curve

Min To find the global minimum of a selected curve

Next Max To find the next local maximum of a selected curve

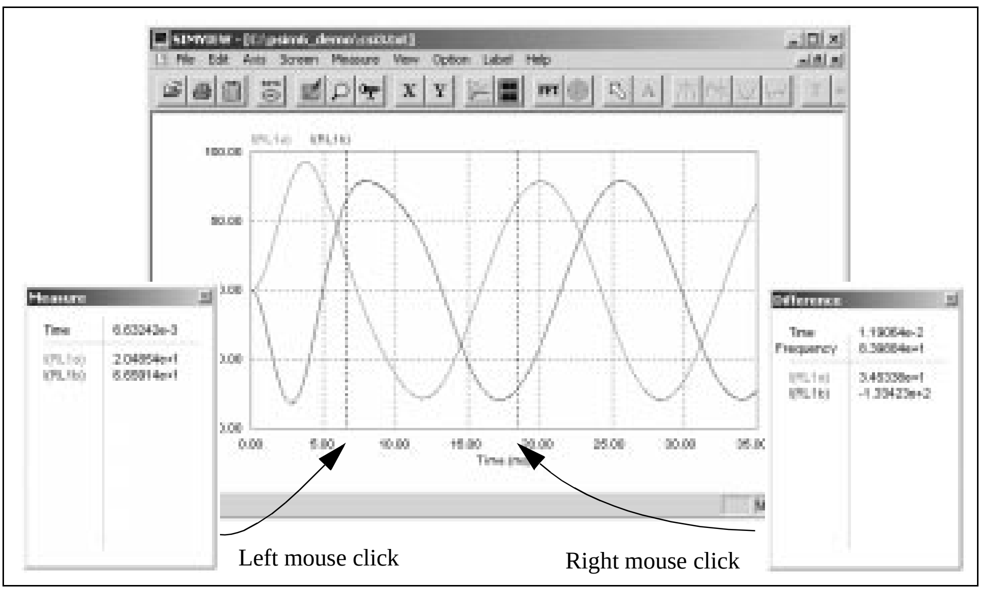
Next Min To find the next local minimum of a selected curve

Toolbar To enable/disable toolba

Status Bar To enable/disable status bar

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 name of the curve at the left top of the graph, and the four functions, **Max**, **Min**, **Next Max**, and **Next Min** can be used to evaluate the curve. Note that these four functions are only enabled in the **Measure** mode and after a curve is selected.

In the zoom-in mode, waveforms can be shifted horizontally or vertically. There are left and right arrows below the x-axis, and up and down arrows in the far right axis. By clicking on the arrow, the waveforms will be shifted by one division.

7.6 Option Menu

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

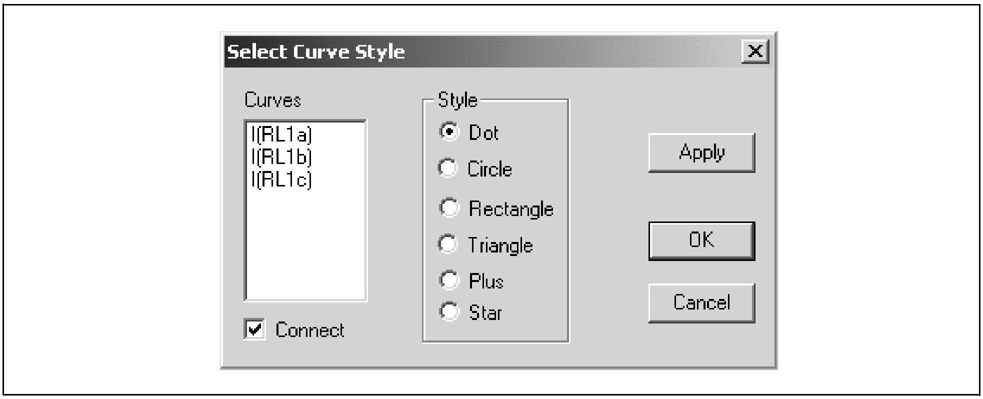
FFT	Perform the Fast Fourier Transform analysis
Time	Switch from the frequency spectrum display to time domain display
Set Text Fonts	Change the text font type and size
Set Curves	Change the display of curves
Set Background	Set the screen background to be either Black (default) or White
Grid	Enable or disable the grid display
Color	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.

The display of a curve can be changed through **Set Curves**. The data points of a curve can have either no symbol, or one of the following symbols: Circle, Rectangle, Triangle, Plus, and Star. Also, data points can be either connected or discrete.

To change the settings of a curve, first select the curve using the left mouse, then choose the proper settings, and click on *Apply*. After all the settings are selected, Click on *OK*.

The dialog box of the **Set Curves** function is shown below.



Once “Color” is de-selected, the display becomes black-and-white. If the waveform screen is copied to the clipboard, the bitmap image will be in monochrome. This will result a much smaller memory size as compared to the image in color display.

7.7 Label Menu

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

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

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

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

7.8 Exporting Data

As stated in Section 7.1, 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**.

8

Error/Warning Messages and Other Simulation Issues

8.1 Simulation Issues

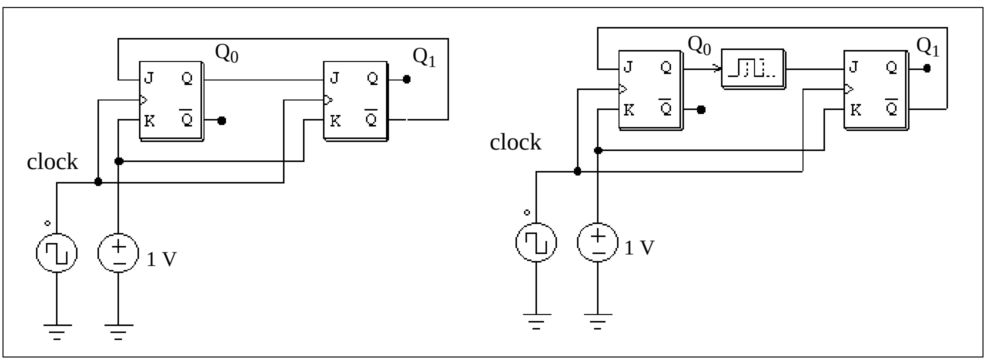
8.1.1 Time Step Selection

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

8.1.2 Propagation Delays in Logic Circuits

The logic elements in PSIM are ideal, i.e. there is no propagation delay. If a logic circuit uses the propagation delays for its operation, a function block in PSIM, called the Time Delay block (TDELAY), needs to be added to represent the effect of the propagation delay.

To illustrate this, take a two-bit counter circuit as an example.



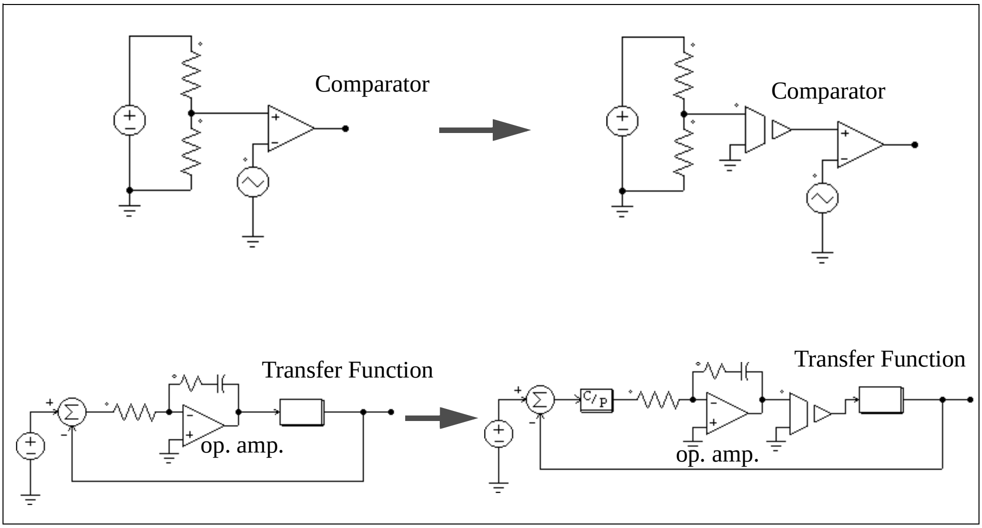
In the circuit on the left, the initial values of both Q0 and Q1 are assumed to be zero. At the clock rising edge, Q0 will change to 1. Without delay, the position of Q1, which should remain at 0, will toggle to 1 at the same time.

To prevent this, a time delay element with the delay period of one time step needs to be inserted between Q0 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 (CTOP) will be automatically inserted. This is illustrated in the examples below.



It should be noted that, in PSIM, the power circuit and the control circuit are solved separately. There is one time step delay between the power and the control circuit solutions.

8.1.4 FFT Analysis

When using FFT for the harmonic analysis, one should make sure that the following requirements are satisfied:

- The waveforms have reached the steady state;

- The length of the data selected for FFT should be the multiple integer of the fundamental period.

For a 60-Hz waveform, for example, the data length should be restricted to 16.67 msec. (or multiples of 16.67 msec.). Otherwise, the FFT results will be incorrect. The data is selected by clicking on **X Axis** in SIMVIEW, de-selecting **Auto-scale** in **Range**, and specifying the starting time and the final time. The FFT analysis is only performed on the data that are displayed on the screen.

Note that the FFT results are discrete. The FFT results are determined by the time interval between two consecutive data points, Δt, and the data length T_{length}. The data point interval Δt is equal to the simulation time step multiplied by the print step. In the FFT results, the frequency incremental step will be $1/T_{length}$, and the maximum frequency will be $1/(2*\Delta t)$.

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

E-3 Error message: No. of an element exceeds the limit.

This error message occurs when the total number of a particular element exceeds the limit specified by the program. This problem can only be solved by re-compiling the PSIM simulator with increased array dimensions. Please contact Powersim Technologies Inc. for assistance.

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.