
PSIM

Version 9.1

For Power Electronics & Motor control

User's Guide

Powersim Inc.

Mywayプラス株式会社

PSIM[®] User's Guide

PSIM Version 9.1

Release 1

© Copyright 2001-2011 Powersim Inc., Myway Plus Corporation

All rights reserved. No part of this manual of the software may be photocopied or reproduced in any form or by any means without the written permission of Myway Plus Corporation.

Disclaimer

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

お問い合わせ先

Myway プラス株式会社

〒222-0022 神奈川県横浜市西区花咲町 6-145 横浜花咲ビル

Tel 045-548-8836, Fax 045-548-8832

Email: sales@myway.co.jp

URL: <http://www.myway.co.jp/>

目次

第1章	概要	8
1.1	はじめに.....	8
1.2	回路構成.....	9
1.3	PSIMの実行に必要な動作環境.....	10
1.4	プログラムのインストール.....	10
1.5	シミュレーション例.....	11
1.6	素子パラメータの仕様および形式.....	11
第2章	電気回路素子	14
2.1	抵抗-インダクタ-キャパシタ.....	14
2.1.1	抵抗 - インダクタ - キャパシタ.....	14
2.1.2	レオスタット.....	15
2.1.3	可飽和リアクトル.....	15
2.1.4	非線形素子.....	16
2.2	スイッチ.....	17
2.2.1	ダイオード・LED・ツェナーダイオード・DIAC.....	18
2.2.2	サイリスタ・トライアック.....	19
2.2.3	GTO・トランジスタ.....	21
2.2.4	双方向スイッチ.....	23
2.2.5	線形スイッチ.....	24
2.2.6	スイッチ・ゲート信号ブロック.....	26
2.2.7	単相スイッチ・モジュール.....	27
2.2.8	3相スイッチ・モジュール.....	28
2.3	相互結合インダクタ.....	31
2.4	変圧器.....	32
2.4.1	理想変圧器.....	32
2.4.2	単相変圧器.....	32
2.4.3	三相変圧器.....	34
2.4.4	三相変圧器（可飽和）.....	35
2.5	三相ACケーブル.....	36
2.6	磁気要素.....	38
2.6.1	巻き線.....	38
2.6.2	漏れ磁束通路.....	38
2.6.3	エアギャップ.....	39
2.6.4	線形コア.....	40
2.6.5	可飽和コア.....	41
2.7	その他エレメント.....	42
2.7.1	オペアンプ.....	42
2.7.1.1	オペアンプ（理想オペアンプ）.....	42
2.7.1.2	オペアンプモデル（非線形オペアンプ）.....	43
2.7.2	TL431 シャントレギュレータ.....	44
2.7.3	フォトカプラ.....	45
2.7.4	dv/dt ブロック.....	46
2.7.5	リレー.....	47

2.8	熱モジュール(Thermalモジュール)	48
2.8.1	デバイスデータベースエディタ	48
2.8.2	データベースのダイオードデバイス	57
2.8.3	ダイオード損失計算	58
2.8.4	データベースのIGBTデバイス	60
2.8.5	IGBT損失計算	62
2.8.6	データベースのMOSFETデバイス	64
2.8.7	MOSFET損失計算	65
2.9	モータ駆動モジュール	68
2.9.1	機械系の基準方向	68
2.9.2	直流機	69
2.9.3	誘導機	71
2.9.4	飽和付かご型誘導機	74
2.9.5	ブラシレス直流機	75
2.9.6	他励式同期機	79
2.9.7	永久磁石同期機	81
2.9.8	飽和付永久磁石同期機	84
2.9.9	スイッチトリラクタンスモータ	87
2.9.10	非線形スイッチトリラクタンスモータ	89
2.10	MagCouplerモジュール	91
2.10.1	MagCoupler ブロック	92
2.10.2	MagCoupler-DLブロック	96
2.11	MagCoupler-RT モジュール	97
2.12	機械素子及びセンサ	103
2.12.1	機械負荷	103
2.12.1.1	定トルク負荷	103
2.12.1.2	定電力負荷	104
2.12.1.3	定速度負荷	105
2.12.1.4	一般負荷	105
2.12.1.5	外部コントロール負荷	106
2.12.2	ギアボックス	106
2.12.3	メカニカル・カップリング・ブロック	107
2.12.4	機械系-電気系インタフェース・ブロック	107
2.12.5	速度・トルクセンサ	109
2.12.6	位置センサ	111
2.12.6.1	アブソリュートエンコーダ	111
2.12.6.2	インクリメンタルエンコーダ	112
2.12.6.3	レゾルバ	112
2.12.6.4	ホールセンサ	113
2.13	Renewable Energy Models	115
2.13.1	太陽電池モジュール	115
2.13.2	風車	118
第3章	制御回路素子	120
3.1	伝達関数ブロック	120
3.1.1	比例制御器	121
3.1.2	積分器	121

3.1.3	微分器.....	123
3.1.4	比例積分 (PI) 制御器.....	123
3.1.5	単極制御器.....	124
3.1.6	二極比例積分 (PI) 制御器.....	124
3.1.7	Type-3 制御器.....	125
3.1.8	組み込みフィルタ・ブロック.....	126
3.2	数値演算関数ブロック.....	127
3.2.1	加算器.....	127
3.2.2	乗算器と除算器.....	127
3.2.3	平方根ブロック.....	128
3.2.4	指数／累乗／対数ブロック.....	128
3.2.5	二乗平均平方根 (RMS) ブロック.....	129
3.2.6	絶対値ブロック.....	129
3.2.7	三角関数ブロック.....	129
3.2.8	高速フーリエ変換 (FFT) ブロック.....	130
3.2.9	最大/最小機能ブロック.....	131
3.3	その他の関数ブロック.....	132
3.3.1	比較器.....	132
3.3.2	リミッタ.....	132
3.3.3	微分 (dv/dt) リミッタ.....	133
3.3.4	台形および方形波ブロック.....	133
3.3.5	サンプル・ホールド・ブロック.....	134
3.3.6	丸めブロック.....	134
3.3.7	時間遅れブロック.....	135
3.3.8	マルチプレクサ.....	136
3.3.9	高調波歪み (THD) ブロック.....	136
3.4	デジタル素子.....	138
3.4.1	論理ゲート.....	138
3.4.2	セット・リセット・フリップフロップ.....	138
3.4.3	J-K フリップフロップ.....	139
3.4.4	Dフリップフロップ.....	139
3.4.5	単安定マルチバイブレータ.....	139
3.4.6	パルス幅カウンタ.....	140
3.4.7	Up/Downカウンタ.....	140
3.4.8	A/D および D/A 変換器.....	141
3.5	デジタル制御モジュール.....	142
3.5.1	ゼロ次ホールド.....	142
3.5.2	z 領域伝達関数ブロック.....	143
3.5.2.1	積分器 (離散型).....	143
3.5.2.2	微分器 (離散型).....	144
3.5.2.3	デジタル・フィルタ.....	145
3.5.3	単位遅れブロック.....	147
3.5.4	離散化ブロック.....	147
3.5.5	循環バッファ.....	149
3.5.6	畳込みブロック.....	150
3.5.7	メモリ読み出しブロック.....	150

3.5.8	データ配列	151
3.5.9	スタック	151
3.5.10	多重サンプリングシステム	152
3.6	SimCouplerモジュール	153
3.6.1	PSIMとSimulinkでの設定	153
3.6.2	Simulinkにおける Solver Typeとタイムステップの設定	155
第4章	その他の素子	158
4.1	パラメータ・ファイル	158
4.2	電圧源・電流源	159
4.2.1	時間	159
4.2.2	定数	159
4.2.3	直流電源	159
4.2.4	正弦波電源 (Sinusoidal Source)	160
4.2.5	方形波電源 (Square-Wave Source)	160
4.2.6	三角波電源／のこぎり波電源	161
4.2.7	ステップ電源	162
4.2.8	区分線形電源 (Piecewise Linear)	162
4.2.9	ランダム電源	163
4.2.10	数式関数電源	164
4.2.11	制御付き電圧源・電流源	164
4.2.12	非線形電圧制御電源	165
4.3	電圧・電流センサ	166
4.4	プローブとメーター	166
4.5	電圧/電流スコープ	168
4.6	初期値	170
4.7	スイッチ制御器	170
4.7.1	オンオフ制御器	170
4.7.2	点弧角制御器	171
4.7.3	PWM ルックアップテーブル制御器	172
4.8	ファンクションブロック	174
4.8.1	制御・パワー変換ブロック	174
4.8.2	変換ブロック	174
4.8.2.1	abc-dq α 変換ブロック	175
4.8.2.2	abc- $\alpha\beta$ 変換ブロック	175
4.8.2.3	$\alpha\beta$ -dq変換ブロック	176
4.8.2.4	直交座標-極座標変換ブロック	177
4.8.3	数式関数ブロック	177
4.8.4	ルックアップテーブル (Look-up Table)	178
4.8.5	Cブロック	180
4.8.6	シンプルC ブロック	182
4.8.7	外部DLLブロック	182
4.8.8	組み込みソフトウェアブロック	187
第5章	解析	188
5.1	過渡解析	188
5.2	周波数特性解析	189

5.3	パラメータのスィープ.....	193
第6章	PSIMによる回路図入力.....	194
6.1	回路の作成.....	194
6.2	ファイルメニュー.....	195
6.3	回路の編集 (Edit).....	196
6.4	Viewメニュー.....	197
6.5	サブ回路 (Subcircuit).....	197
6.5.1	サブ回路作成 — メイン回路から.....	199
6.5.2	サブ回路作成 — サブ回路内部から.....	199
6.5.3	サブ回路の接続 — メイン回路で.....	200
6.5.4	サブ回路のその他の機能.....	200
6.5.4.1	メイン回路からサブ回路への変数値の受け渡し.....	201
6.5.4.2	サブ回路シンボルの作成.....	201
6.5.4.3	サブ回路をPSIMの素子リストに追加する.....	202
6.6	Simulateメニュー.....	202
6.7	Optionsメニュー.....	207
6.8	Utilitiesメニュー.....	211
6.9	PSIMライブラリの管理.....	212
6.9.1	2番目のイメージ図作成.....	212
6.9.2	ライブラリへ新しいサブ回路要素の追加.....	213
6.9.3	ライブラリへ新規DLL要素の追加.....	216
6.10	シンボルライブラリの作成.....	218
第7章	SIMVIEW による波形処理.....	221
7.1	File メニュー.....	221
7.2	Edit メニュー.....	222
7.3	Axis メニュー.....	222
7.4	Screen メニュー.....	222
7.5	Measure メニュー.....	223
7.6	Analysisメニュー.....	224
7.7	View メニュー.....	225
7.8	Options メニュー.....	225
7.9	Label メニュー.....	226
7.10	Settingsメニュー.....	226
7.11	データの書き出し (Export).....	226
7.12	実際のSIMVIEW使用の流れ.....	227
7.12.1	SIMVIEWの立ち上げ.....	227
7.12.2	波形表示.....	228
7.12.3	波形観測.....	230
7.12.4	測定.....	232
7.12.5	描画.....	233
第8章	エラー・警告と全般に関する注意.....	235
8.1	全般に関する注意.....	235
8.1.1	タイムステップ.....	235
8.1.2	論理回路の伝搬遅れ.....	235

8.1.3	パワー回路と制御回路のインタフェース	236
8.1.4	FFT による高調波解析	236
8.2	エラーおよび警告メッセージ	237
8.3	デバッグ	238

第1章 概要

1.1 はじめに

PSIM はパワーエレクトロニクスおよびモータ制御のために開発されたシミュレーション・パッケージです。PSIM は高速シミュレーション、使いやすいユーザインタフェース、波形解析機能などにより、パワーエレクトロニクスの解析、制御系設計、モータドライブの研究などに優れたシミュレーション環境を提供します。

PSIM には本体 (Professional) の他に、以下のアドオン・オプションが用意されています。

- モータ駆動モジュール (Motor Drive Module)
- デジタル制御モジュール (Digital Control Module)
- SimCoupler モジュール
- Thermal モジュール
- MagCoupler モジュール
- MagCoupler-RT モジュール
- SimCoder モジュールおよび関連のハードウェアターゲット
- Renewable Energy モデル
- SmartCtrl
- ModCoupler モジュール

(SmartCtrl と ModCoupler モジュールは PSIM とは別の 3rd パーティ製品ですが、便宜上一つのモジュールとして扱います。)

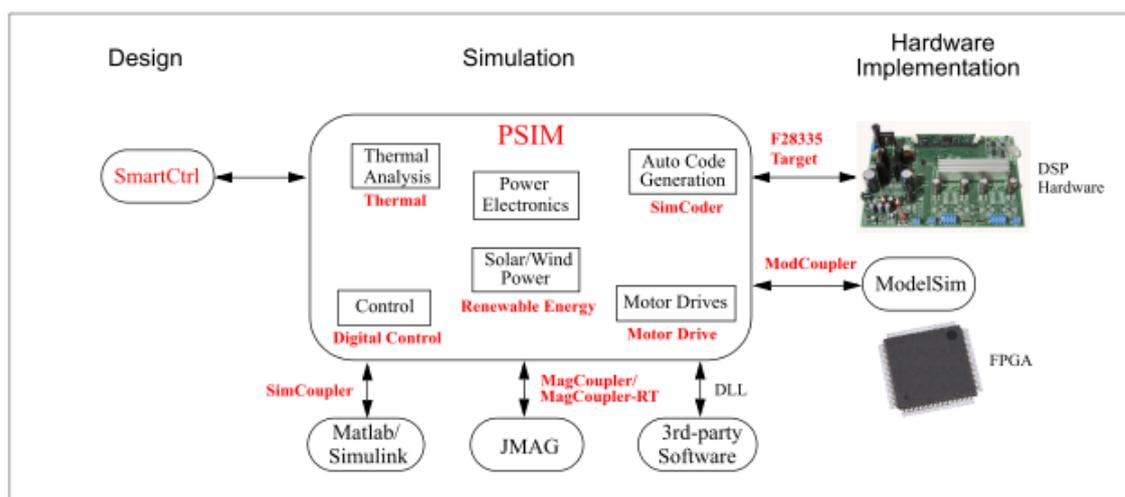
モータ駆動モジュールはモータドライブシステムの解析のためにモータモデルと機械負荷モデルが用意されています。

デジタル制御モジュールはゼロ次ホールドなどの離散要素、 z -領域伝達関数、離散化ブロックを提供し、デジタル制御系を解析できます。

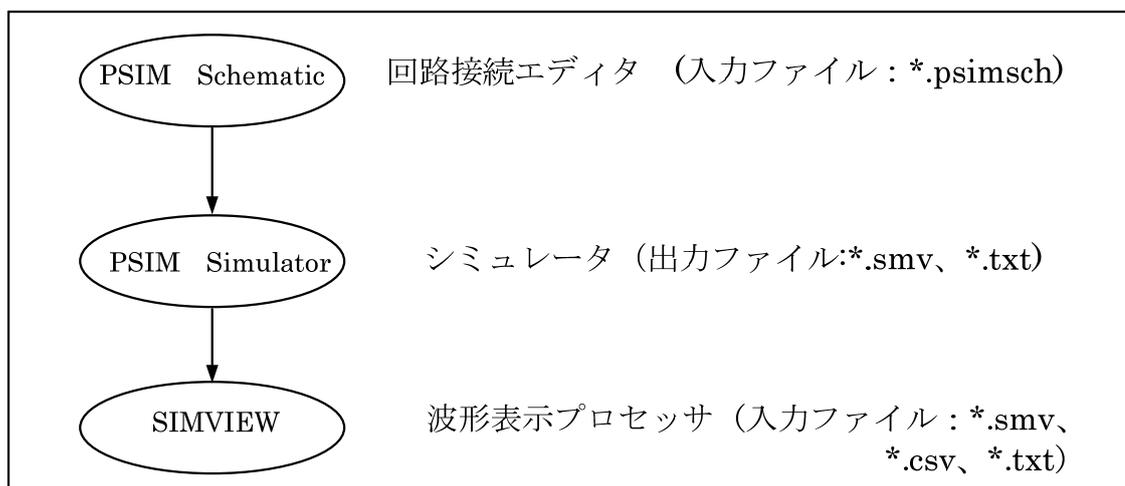
SimCoupler モジュールは PSIM と MATLAB/Simulink を同時に使ったシミュレーションを行うためのインタフェースを提供します。

SimCoupler モジュールは PSIM と MATLAB/Simulink を同時に使ったシミュレーション、そして MagCoupler/MagCoupler-RT モジュールは PSIM と電磁解析ソフトウェア JMAG/JMAG-RT との連成シミュレーションを実現します。MATLAB/Simulink と JMAG とのリンク機能を使えば、PSIM がパワーエレクトロニクス、制御と電磁氣的要素を一元的に取り扱える統合開発環境を提供します。

SimCoder モジュールおよび SmartCtrl については、それぞれのユーザーマニュアルを参照して下さい。



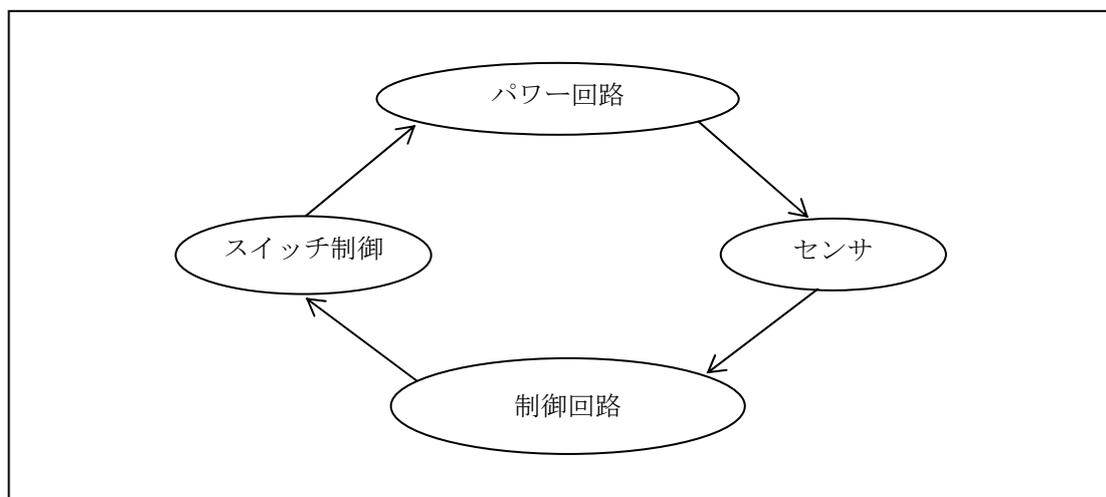
PSIM パッケージは次の 3 つのソフトウェアから構成されています。PSIM 回路接続エディタ、PSIM シミュレータ、そして波形表示処理の SIMVIEW です。下の図にシミュレーション環境を示します。



本マニュアルの第 1 章は回路の構成法、実行に必要なソフトウェアとハードウェア、およびインストール手順について解説します。第 2 章から第 4 章までは、パワー回路素子および制御素子について述べます。第 5 章では過渡解析と交流回路解析の手順を解説します。PSIM を使った回路図の作成法と SIMVIEW の使用法は第 6 章と第 7 章で解説します。第 8 章はエラー・メッセージの一覧をまとめてあります。

1.2 回路構成

PSIM での回路はパワー回路、制御回路、センサ、スイッチ制御の 4 つのブロックで構成されます。以下の図に各ブロックの相互間系を示します。



パワー回路はスイッチング素子、RLC 回路、変圧器、および相互結合インダクタからなっています。制御回路は s 領域あるいは z 領域の伝達関数のブロック線図で表現します。またアナログ素子（乗算器やサンプラなど、非線形素子を含む）およびデジタル素子（論理ゲートおよびフリップフロップ）を制御回路で使うことができます。センサはパワー回路の電圧・電流値を計測し、これらの値を制御回路ブロックに伝えます。制御回路で作られたゲート信号はスイッチ制御ブロックを経て、パワー回路のスイッチング素子に送られます。

1.3 PSIM の実行に必要な動作環境

- ・ハードウェア：128MB 以上のメモリを搭載した PC/AT 互換機
- ・OS：Microsoft Windows XP/Vista
- ・Internet Explorer：Ver.6.0 以降が必要（MagCoupler を使用する場合）

[DLL 機能を使用する場合]

- ・対応コンパイラ：Microsoft C/C++、Visual C++ .net

[SimCoupler モジュールを利用する場合]

- ・MATLAB R13以降が必要（最新の情報はPSIM専用HPをご参照ください。）

[MagCoupler モジュールを利用する場合]

- ・JMAG Ver.10 以降が必要（最新の情報は PSIM 専用 HP をご参照ください。）

1.4 プログラムのインストール

PSIM のインストール手順は別冊 PSIM Quick Guide をご覧ください。

PSIM のファイル構成は以下のようになっています。

ファイル	内容
psim.exe	PSIM 回路図エディタ

SIMVIEW.exe	SIMVIEW 波形表示プログラム
PcdEditor.exe	デバイスデータベースエディタ
SetSimPath.exe	SimCoupler モジュールのパス設定プログラム

PSIM で使われるファイル拡張子は以下の通りです。

*.psimsch	PSIM 回路図ファイル
*.lib	PSIM ライブラリファイル
*.fra	PSIM AC スイープ解析出力ファイル (テキスト)
*.dev	デバイスデータベースファイル
*.txt	シミュレーション出力ファイル (テキスト) ※いずれか選択
*.smv	シミュレーション出力ファイル (バイナリ) ※いずれか選択

PSIM 回路図ファイルの拡張子は PSIM8.0 までは .sch ですが、PSIM9.0 では他のソフトウェアとの混同を避けるため psimsch に変更されました。

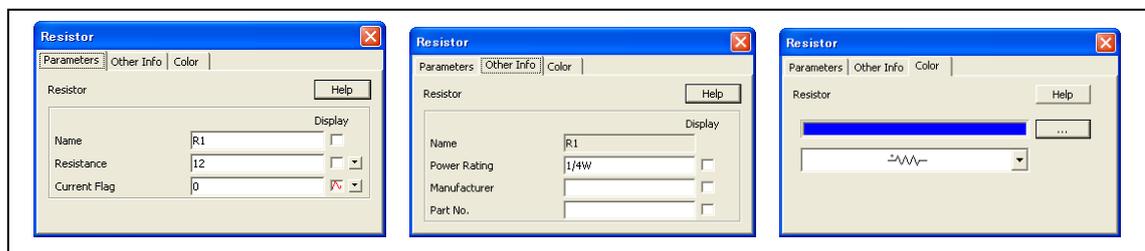
1.5 シミュレーション例

プログラムに添付のサンプル chop.sch を実行するには以下のように操作します。

- 1) PSIM を起動します。
- 2) **File** メニューから **Open** を選び、chop.sch を選びます。
- 3) **Simulate** から **Run PSIM** を選び PSIM を起動します。PSIM シミュレータは netlist ファイルを読み込み、シミュレーションを開始します。シミュレーション結果は chop.txt ファイルに保存されます。また、シミュレーション中に生じた警告メッセージは message.doc ファイルに保存されます。
- 4) **Simulate** メニューから **Run SIMVIEW** を選び SIMVIEW を起動します。表示したい波形を選んでください。**Options** メニューで Auto-run SIMVIEW が選ばれている場合は、SIMVIEW は自動的に起動します。

1.6 素子パラメータの仕様および形式

PSIM で使用できる素子のパラメータ・ダイアログ・ウィンドウには 3 つのタブあります。ひとつはパラメータ (Parameters)、他の情報 (Other Info)、そして色 (Color) です。これらを以下の図に示します。



パラメータ (Parameters) のタブで指定した情報はシミュレーションに使います。その他の情報 (Other Info) はシミュレーションには使いません。後者は報告書作成用で、素子の定格、製造元、パーツ番号などの情報を Other Info のタブに保存しておく、PSIM の **View** メニューの **Element List** から選ぶパーツリストに掲載されます。

パラメータ (Parameters) のタブで指定できる情報は数値または数式です。たとえば抵抗は次のいずれかの方法で指定できます：

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

ここで、R1、R2、Vo、Io はパラメータ・ファイルで (4.1参照) もしくはこの抵抗がサブ回路にある場合 (6.5参照) はメイン回路にて定義します。

また、PSIM では 10 の累乗も指定できます。以下の単位をサポートしています。

G 10⁹
 M 10⁶
 k または K 10³
 m 10⁻³
 u 10⁻⁶
 n 10⁻⁹
 p 10⁻¹²

数式にはカッコを使うこともできますが、大文字と小文字の区別はありません。PSIM で使うことができるのは次の関数です。

+	加算	
-	減算	
*	乗算	
/	除算	
^	べき乗	[例 : 2^3=2*2*2]
SQRT	平方根	
SIN	サイン	
COS	コサイン	
ASIN	アーク・サイン	
ASIN	アーク・コサイン	
TAN	タンジェント	
ATAN	アークタンジェント	
ATAN2	アークタンジェント 2	[-π <= atan2(y,x) <= π] atan2(y,x) = atan(y/x)
SINH	ハイパボリックサイン	
COSH	ハイパボリックコサイン	
EXP	指数	[例 : EXP(x) = e ^x]
LOG	自然対数	[LOG(x)=ln(x)]
LOG10	常用対数	(基底 10)
ABS	絶対値	

SIGN

符号関数

[例 : SIGN(1.2)=1; SIGN(-1.2)=-1]

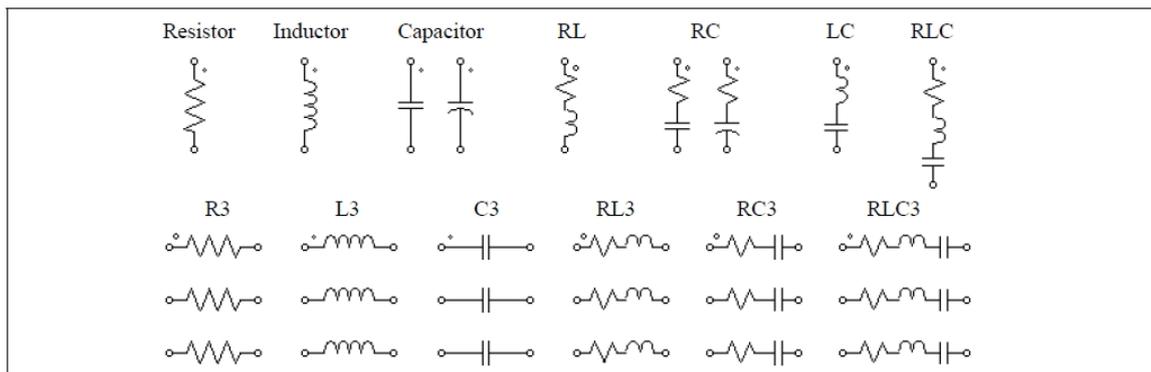
第2章 電気回路素子

2.1 抵抗-インダクタ-キャパシタ

2.1.1 抵抗 - インダクタ - キャパシタ

PSIMには、抵抗（Resistor）、インダクタ（Inductor）、キャパシタ（Capacitor）の各ブランチと集合RLCブランチの両方が備わっています。インダクタ電流とキャパシタ電圧は初期条件として設定できます。3相回路の設定を容易にするために、対称3相RLCブランチ、R3、RL3、RC3、RLC3が備わっています。対称3相RLCブランチでは初期インダクタ電流とキャパシタ電圧は全てゼロに設定されます。

シンボル



シンボルのすぐ上の名称は netlist に登録された素子の名前と同じです。たとえば、抵抗器は Library メニューで Resistor と表示され、名称は R です。

3相ブランチの場合は、ドットの付いた相は A 相になります。

仕様

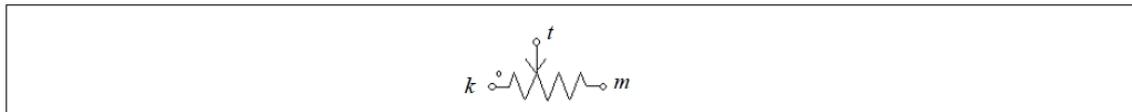
パラメータ	機能
Resistance	抵抗 (Ω)
Inductance	インダクタンス (H)
Capacitance	キャパシタンス (F)
Initial Current	初期インダクタンス電流 (A)
Init.Cap.Coltage	初期キャパシタンス電圧 (V)
Current Flag	ブランチ電流出力フラグ。フラグが0のとき、電流出力なし。フラグが1のとき、電流はディスプレイ用出力ファイルに保存される。ブランチのドットの付いた端子に電流が流れ込むときを正とする。
Current Flag_A	それぞれ3相ブランチの相 A、B、C のフラグ
Current Flag_B	
Current Flag_C	

ブランチの抵抗値、インダクタンス、キャパシタンスはすべて同時には零にできません。少なくともパラメータの一つは0以外でなければなりません。

2.1.2 レオスタット

レオスタット (Rheostat) はタップ付きの抵抗器です。

シンボル



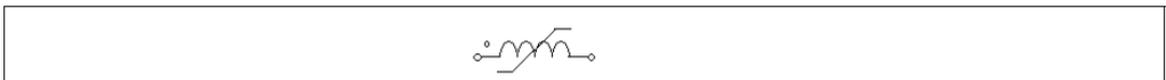
仕様

パラメータ	機能
Total Resistance	全抵抗 (Ω) 端子 k から端子 m までの抵抗
Tap Position	タップ位置 (0-1) 端子 k から t までの抵抗は $R \cdot \text{Tap}$
Current Flag	端子 k に流れ込む電流のフラグ

2.1.3 可飽和リアクトル

可飽和リアクトル (Saturable Inductor) はインダクタの鉄心の飽和を反映できます。

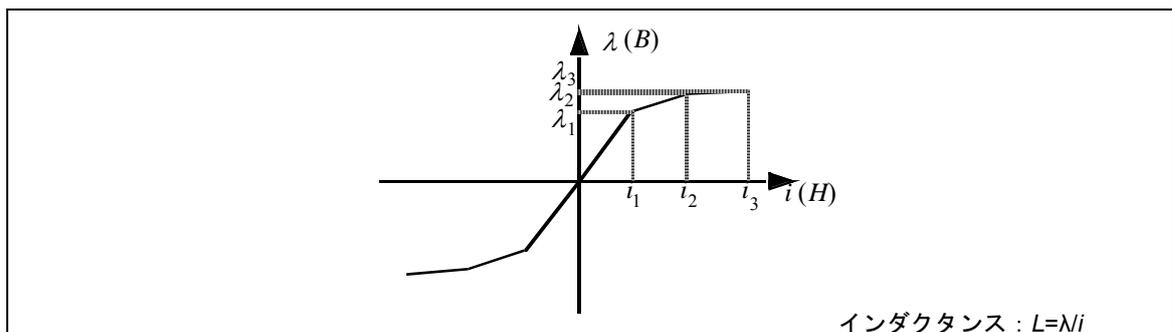
シンボル



仕様

パラメータ	機能
Current v.s. Inductance	電流対インダクタンス特性 $(i_1, L_1), (i_2, L_2), \dots$ のように指定
Current Flag	端子 k に流れ込む電流のフラグ

非線形の B-H 曲線は区分線形特性で近似します。磁束密度 B は鎖交磁束 λ に比例し、磁化力 H は電流に比例するので、B-H 曲線は以下に示すように $\lambda - i$ 曲線で等価に表すことができます。



インダクタンスは $L=\lambda/i$ で求められ、これは $\lambda - i$ 曲線の傾斜に相当します。したがって、飽和特性は異なる点のデータの対を (i_1, L_1) , (i_2, L_2) , (i_3, L_3) , などのように指定することにより表現できます。

λ は i の増加に伴って、 $L_1*i_1 < L_2*i_2 < L_3*i_3$ のように単調に増加する必要があります。同様に、実際の飽和特性では $\lambda - i$ 特性の各区間の傾きは電流の増加に伴って単調に減少するべきです。

条件によっては可飽和リアクトルを含むシミュレーションは収束しないかもしれません。そのような場合、可飽和リアクトルと並列に微小なコンデンサを接続してみてください。

2.1.4 非線形素子

電圧 - 電流の非線形特性を表現する素子 (Nonlinear Elements) は以下の4つが用意されています。

抵抗素子 (NONV) $[v = f(i)]$

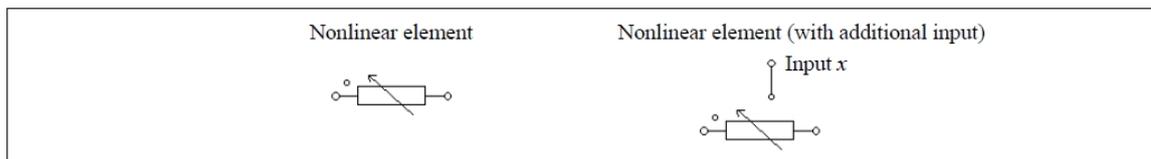
補助入力付き抵抗 (NONV_1) $[v = f(i, x)]$

コンダクタンス素子 (NONI) $[i = f(v)]$

補助入力付きコンダクタンス (NONI_1) $[i = f(v, x)]$

補助入力は電圧信号に限ります。

シンボル



仕様

(抵抗型素子の場合)

パラメータ	機能
Expression $f(i)$	NONV 素子のための関数
Expression $f(i, x)$	NONV_1 素子のための関数
Expression df/di	電流に関する電圧の微分式
Initial Value i_0	電流 i の初期値
Lower Limit of i	電流 i の下限値
Upper Limit of i	電流 i の上限値

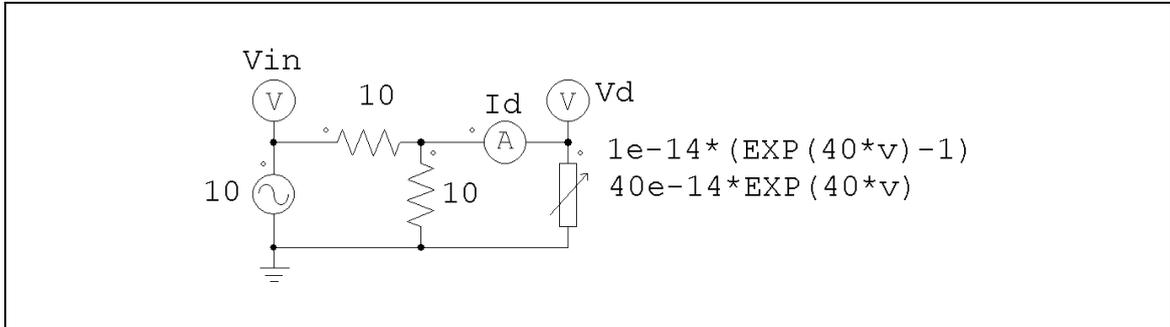
(コンダクタンス型素子の場合)

パラメータ	機能
Expression $f(v)$	NONI 素子のための関数
Expression $f(v, x)$	NONI_1 素子のための関数
Expression df/dv	電圧に関する電流の微分式
Initial Value v_0	電圧 v の初期値

Lower Limit of v	電圧 v の下限値
Upper Limit of v	電圧 v の上限値

電圧/電流の初期値および上下限値を適切に設定することで、解の収束が得やすくなります。

例：非線形ダイオード



上の回路で、非線形素子 (NONI) が非線形ダイオードを模擬しています。ダイオードの電流は電圧の関数で $i = 10^{-14} * (e^{40v} - 1)$ と与えています。

PSIM ではこの非線形素子の特性は以下のように指定します。

Expression $f(v)$	$1e-14 * (EXP(40*v) - 1)$
Expression df/dv	$40e-14 * EXP(40*v)$
Initial Value v_0	0
Lower Limit of v	-1e3
Upper Limit of v	1

2.2 スイッチ

PSIM のスイッチ素子にはふたつの基本タイプがあります。ひとつはスイッチ・モード素子で、カット・オフ (オフ) または飽和 (オン) のいずれかの状態で動作するもの。もうひとつは線形スイッチ素子で、カット・オフ・線形・飽和のいずれかの領域で動作します。

スイッチ・モード素子には次のようなスイッチがあります。

ダイオード (DIODE/DIAC)

サイリスタ (THY/TRIAC)

自己転流スイッチ :

Gate Turn-Off サイリスタ (GTO)

バイポーラ・トランジスタ (npn/pnp Transistor)

IGBT (IGBT)

MOSFET (MOSFET_P、MOSFET_N)

双方向スイッチ (SSWI)

(カッコの中は PSIM での名称です)

スイッチモデルは理想型で、オン・オフ時の過渡動作は無視されます。スイッチには $10\mu\Omega$ のオン抵抗と $10M\Omega$ のオフ抵抗があります。スナバ回路は必要としません。

線形スイッチ素子には以下のタイプがあります。

npn バイポーラ接合トランジスタ (npn Transistor(3-state))

pnp バイポーラ接合トランジスタ (pnp Transistor(3-state))

n-チャンネル MOSFET(MOSFET(3-state))

p-チャンネル MOSFET(p-MOSFET(3-state))

2.2.1 ダイオード・LED・ツェナーダイオード・DIAC

ダイオード (Diode)、LED

発光ダイオード (LED) は導通すると発光します。ダイオードや LED の導通は回路操作条件によって決定されます。ダイオードは閾値電圧を超えて正方向にバイアスされるとオンし、電流が 0 になるとオフします。

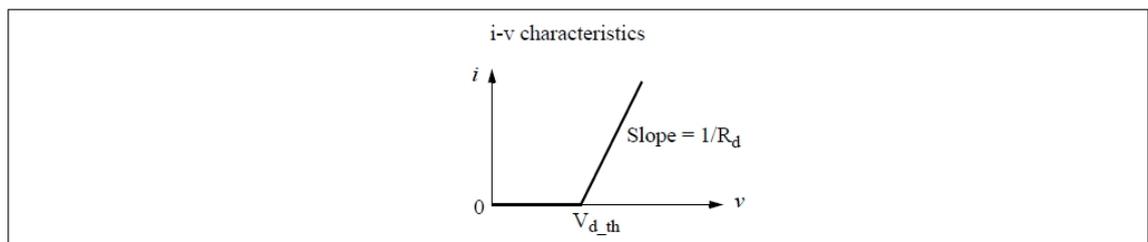
シンボル



仕様

パラメータ	機能
Diode Threshold Voltage	ダイオードの閾値電圧 ($V_{d,th}$, 単位 V)。正のバイアス電圧が $V_{d,th}$ 以上になると、LED が導通します。
Diode Resistance	導通時のダイオード抵抗値 R_d (単位 Ω)
Initial Position	ダイオード初期状態フラグ(1:オン; 0:オフ)
Current Flag	ダイオードの電流フラグ

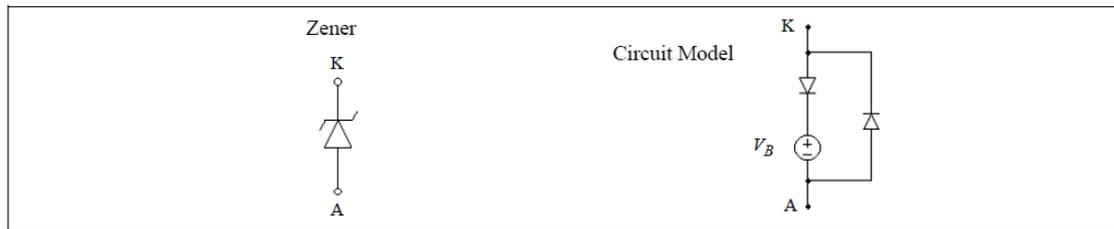
ダイオードおよび LED の i - v 特性は以下の通りです。



ツェナーダイオード

PSIM のツェナーダイオードは以下に示すような回路によりモデル化されています。

シンボル



仕様

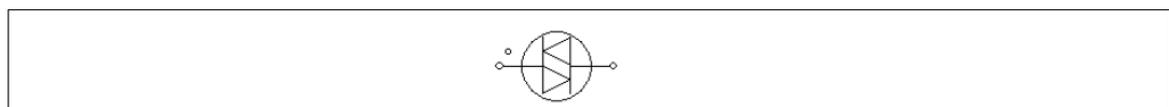
パラメータ	機能
Breakdown Voltage	ツェナーダイオードのブレイク電圧 (V_B , 単位 V)
Forward Threshold Voltage	順方向の閾値電圧 (アノードからカソード間) (単位 V)
Forward Resistance	順方向のオン抵抗 (単位 Ω)
Current Flag	ツェナーダイオードの電流フラグ (アノードからカソード方向が 1)

ツェナーダイオードが順方向にバイアスされた場合は、通常のダイオードとして動作します。逆方向にバイアスされた場合は、カソード電圧 V_{KA} がブレイク電圧 V_B 以下である限り、導通を阻止します。電圧 V_{KA} が V_B より高い場合には、電圧 V_{KA} が V_B にクランプされます。(ツェナーダイオードがクランプすると、ダイオードのオン抵抗が $10\mu\Omega$ なので、カソード-アノード間の電圧は実際には $V_{KA} = V_B + 10\mu\Omega \cdot I_{KA}$ となり、 I_{KA} の値によっては V_{KA} は V_B よりも少し高くなります。電流 I_{KA} が非常に大きい場合は、電圧 V_{KA} は V_B よりもかなり高くなる可能性があります。)

DIAC

DIAC は双方向性のダイオードです。DIAC はブレイクオーバー (breakover) 電圧がかかるまで電流を通しません。DIAC が導通するようになったあとの電圧はブレイクバック (breakback) 電圧と呼ばれます。

シンボル



仕様

パラメータ	機能
Breakover Voltage	DIAC が導通を始める電圧 (V)
Breakback Voltage	導通時電圧降下 (V)
Current Flag	DIAC の電流フラグ

2.2.2 サイリスタ・トライアック

サイリスタはオン時刻が制御できます。オフ時刻は電流条件で決定されます。

トライアックは両方向に電流を流すことができるサイリスタです。トライアックは逆向きの 2 つのサイリスタが並列接続された構造をしています。

シンボル



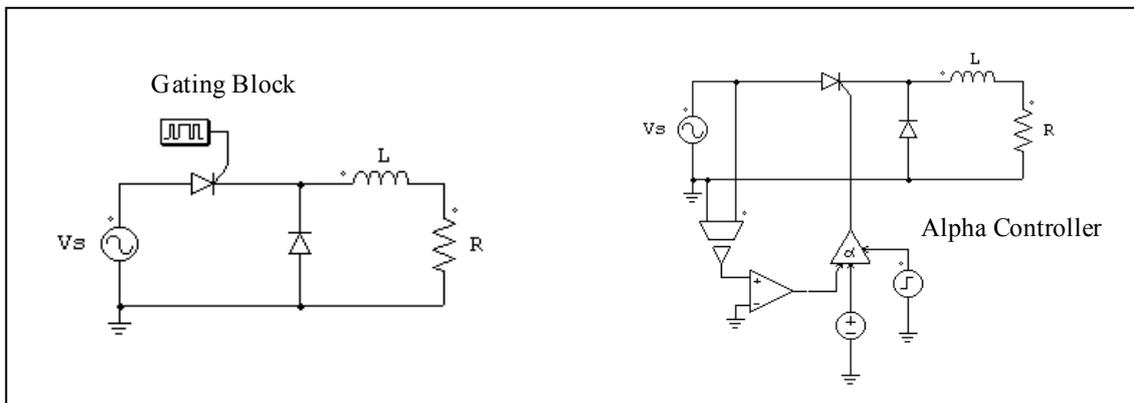
仕様

パラメータ	機能
Voltage Drop	サイリスタ導通時電圧降下
Holding Current	サイリスタの最小導通電流; 電流がこの値以下になるとサイリスタの状態は OFF になります (サイリスタのみに適用)。
Latching Current	サイリスタを ON 状態に保つ最小電流 (サイリスタのみに適用)。
Initial Position	スイッチ初期状態フラグ
Current Flag	スイッチ電流出力フラグ

TRIAC の Holding Current と Latching Current はいずれもゼロです。

サイリスタおよび TRIAC を制御するには 2 つの方法があります。一つはゲート・ブロック (GATING) を使うことです。もう一つはスイッチ・コントローラを使うことです。サイリスタまたは TRIAC のゲート極はこれらのいずれかに接続されていなければなりません。

例：サイリスタ・スイッチの制御



左側の回路はスイッチ・ゲート信号ブロック (2.2.6参照) を使っています。スイッチング・ゲートパターンと周波数はシミュレーション開始前に設定し、シミュレーション中に変化することはありません。右側の回路は点弧角制御器 (4.7.2参照) を使っています。遅れ角 α (度) は回路の直流源により指定されません。

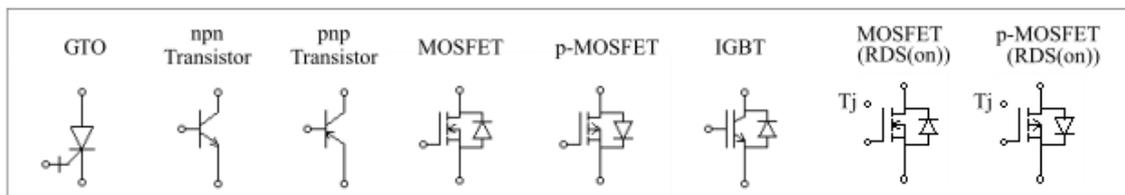
2.2.3 GTO・トランジスタ

GTO, IGBT, MOSFET のような自己転流スイッチはゲートが High (スイッチ・コントローラなどから論理値 1 が入るとき) に順方向バイアス (コレクターエミッタまたはドレインソース間電圧が正) されたときにオンします。またゲート電圧が Low の場合や、電流が 0 になった場合はいつでもオフします。これに対し npn バイポーラ接合トランジスタ (BJT) と p チャネル MOSFET (MOSFET_P) はゲート信号が Low で、かつ逆方向バイアスが掛かった (コレクターエミッタまたはドレインソース間電圧が負) ときにオフします。

GTO スイッチは正方向と逆方向の両方の阻止能力を持った対称素子です。IGBT と MOSFET / MOSFET_P スイッチのモジュールは能動スイッチと逆阻止並列ダイオードから構成されています。

PSIM の BJT スイッチは実際のデバイスとは異なり、(GTO のように) 逆方向電圧をブロックします。また、電流ではなくゲート信号電圧により制御されます。

シンボル



仕様 (GTO)

パラメータ	機能
Voltage Drop	GTO 導通電圧降下 (V)
Initial Position	スイッチ初期状態フラグ (0:オフ、1:オン)
Current Flag	スイッチ電流出力フラグ (0:非表示、1:表示)

仕様 (nnp および pnp BJT)

パラメータ	機能
Saturation Voltage	飽和電圧 V_{ce_sat} (nnp の場合) または V_{ec_sat} (pnp の場合) (V)
Initial Position	スイッチ初期状態フラグ (0:オフ、1:オン)
Current Flag	スイッチ電流出力フラグ (0:非表示、1:表示)

仕様 (n チャネルおよび p チャネル MOSFET)

パラメータ	機能
On Resistance	MOSFET のオン抵抗 R_{ds_on} (Ω)
Diode Threshold Voltage	逆並列ダイオード閾値電圧 (V)
Diode Resistance	逆並列ダイオードのオン抵抗 (Ω)
Initial Position	スイッチ初期状態フラグ (0:オフ、1:オン)
Current Flag	スイッチ電流出力フラグ (0:非表示、1:表示)

仕様 (n チャネルおよび p チャネル MOSFET (R_{ds(on)}))

パラメータ	機能
T _j at Test	試験時のジャンクション温度 T _{j_b} (°C)
RDS(on) at Test	試験時のオン抵抗 R _{ds_on} (Ω)
Temperature Coefficient	オン抵抗の温度係数 (1/°C)
Diode Threshold Voltage	逆並列ダイオード閾値電圧 (V)
Diode Resistance	逆並列ダイオードのオン抵抗 (Ω)
Initial Position	スイッチ初期状態フラグ (0:オフ、1:オン)
Current Flag	スイッチ電流出力フラグ (0:非表示、1:表示)

仕様 (IGBT)

パラメータ	機能
Saturation Voltage	IGBT の飽和電圧 V _{ce_sat} (V)
Transistor Resistance	トランジスタのオン抵抗 (Ω)
Diode Threshold Voltage	逆並列ダイオード閾値電圧 (V)
Diode Resistance	逆並列ダイオードのオン抵抗 (Ω)
Initial Position	スイッチ初期状態フラグ (0:オフ、1:オン)
Current Flag	スイッチ電流出力フラグ (0:非表示、1:表示)

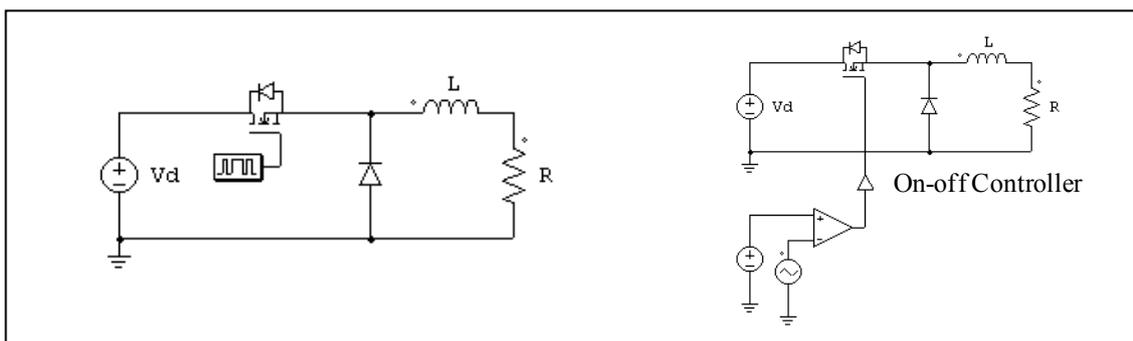
MOSFET (RDS (ON)) と p- MOSFET (RDS (ON)) の画像は、ゲート上部のノードは、ジャンクション温度を定義するためのものです。このノードの電圧信号は、ジャンクション温度 T_j (°C) を定義します。

例えば、25V の電圧値は 25°C の接合部温度を意味します。MOSFET のオン抵抗は以下のとおり表されるジャンクション温度の関数になります：

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

自己転流スイッチはゲート・ブロック (GATING) かスイッチ・コントローラのいずれかで制御されます。これらはスイッチのゲート (ベース) 極に接続されていなければなりません。

例：MOSFET スwitchの制御

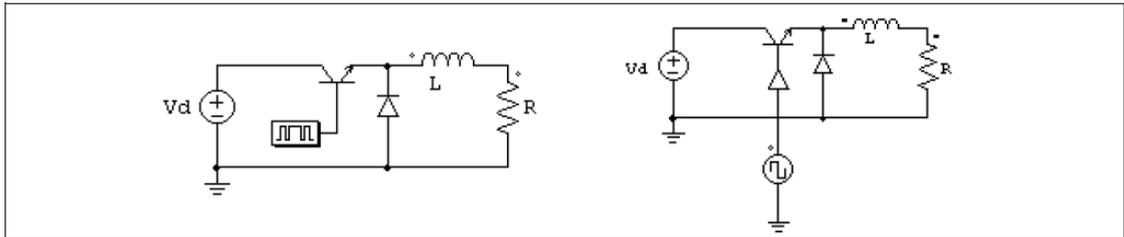


左側の回路はゲート・ブロックを使用し、右側の回路はオンオフ制御器 (4.7.1参照) を使用します。ゲート信号は比較器の出力で決定されます。

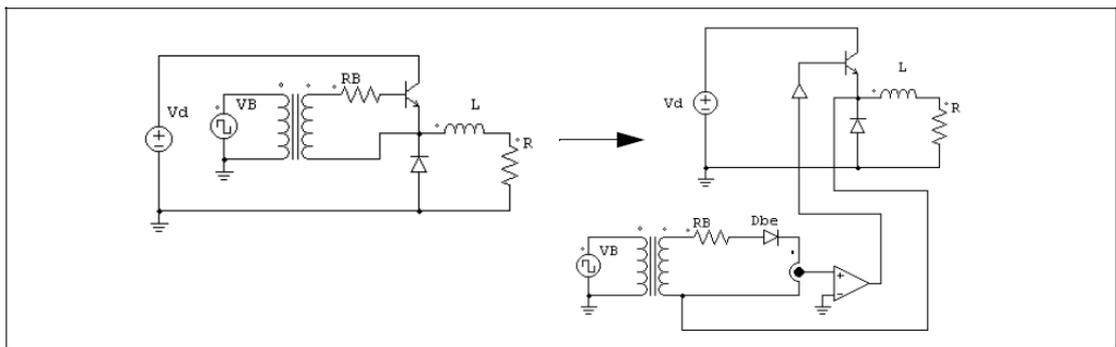
例：npn バイポーラ接合トランジスタ (BJT) の制御

左側の回路はゲート・ブロックを、右側の回路はオンオフ・スイッチ・コントローラを使用しています。

以下の例は npn スイッチを制御するもうひとつの例です。左の回路は実際の回路での npn スイッチを制御する接続を示します。この場合、ゲート電圧 V_B を、変圧器を介してトランジスタのベース駆動回路に加えると、ベース電流がトランジスタの導通状態を決めます。



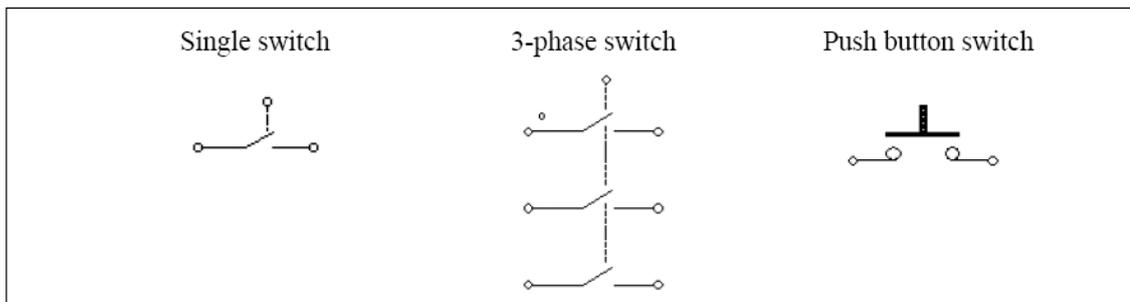
PSIM ではこれを右のように構成します。ダイオード D_{be} を使ってベース-エミッタ間 0.7V の電圧降下を表現しています。ベース電流が 0（もしくは一定のしきい値；この場合、ベース電流は直流電源と比べます）を越えると、比較器の出力が 1 になり、オンオフ・スイッチ・コントローラを介してトランジスタにターンオン・パルスを加えます。



2.2.4 双方向スイッチ

PSIM ではシングルスイッチ、3相スイッチ及びプッシュボタンスイッチの3タイプの双方向スイッチを用意しています。

シンボル



仕様

パラメータ	機能
Initial Position	スイッチ初期状態フラグ(シングルスイッチのみ)
Initial Position (for Phase A/B/C)	3相スイッチのA・B・C相の初期状態フラグ
Switch position	プッシュボタンのスイッチ状態(オンかオフ)
Current Flag	スイッチ電流フラグ(シングルスイッチのみ)
Current Flag (for Phase A/B/C)	A・B・C相のスイッチ電流フラグ

3相スイッチのイメージで、ドット付スイッチがA相です。シングルスイッチ及び3相双方向のスイッチに関しては、電圧バイアス状態にかかわらず、ゲート信号が High の時オンであり、ゲート信号が Low のときオフです。プッシュボタンスイッチにおいて、スイッチ状態はパラメータ入力で設定されます。

2.2.5 線形スイッチ

PSIMには線形スイッチとして npn および pnp バイポーラ接合トランジスタと n-チャンネル及び p-チャンネル MOSFET モデルを用意しています。

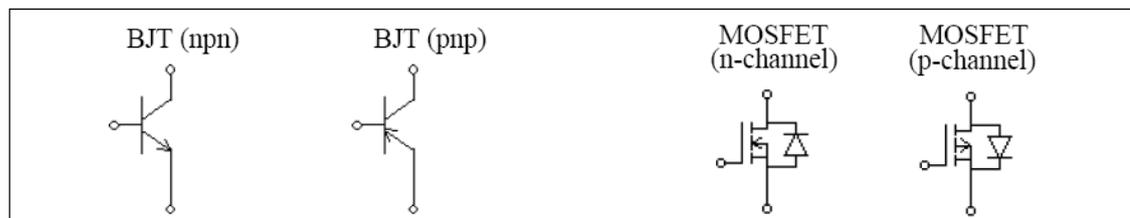
線形 npn バイポーラ接合トランジスタ (nnp Transistor(3-state))

線形 pnp バイポーラ接合トランジスタ (pnp Transistor(3-state))

線形 n-チャンネル MOSFET(MOSFET(3-state))

線形 p-チャンネル MOSFET(p-MOSFET(3-state))

シンボル



BJT の仕様

パラメータ	機能
Current Gain beta	トランジスタの電流ゲイン β ($\beta = I_c / I_b$ として定義)
Bias Voltage V_f	順方向バイアス電圧 (NPN の場合ベース-エミッタ間, PNP の場合エミッタ-ベース間)
$V_{ce,sat}$ [PNP の場合は $V_{ec,sat}$]	NPN のコレクター-エミッタ間飽和電圧および PNP のエミッタ-コレクタ間飽和電圧

MOSFET の仕様

パラメータ	機能
On Resistance	MOSFET のオン抵抗、 $R_{ds(on)}$ (単位: Ω)
Threshold voltage $V_{gs(th)}$	ゲート-ソース電圧のしきい電圧 (単位: V)
Transconductance g_m	MOSFET の相互コンダクタンス

Diode Voltage Drop | 逆並列ダイオードの順方向電圧降下(単位:V)

線形BJTスイッチはベース電流 I_b で制御されます。スイッチの状態はカット・オフ（オフ）、線形、飽和（オン）のいずれかの領域です。以下にNPNトランジスタの各領域電圧・電流特性を示します。

- カット・オフ領域： $V_{be} < V_r; I_b = 0; I_c = 0$
- 線形領域： $V_{be} = V_r; I_c = \beta * I_b; V_{ce} < V_{ce, sat}$
- 飽和領域： $V_{be} = V_r; I_c < \beta * I_b; V_{ce} = V_{ce, sat}$

ここで、 V_{be} はベース-エミッタ間電圧、 V_{ce} はコレクター-エミッタ間電圧、 I_c はコレクタ電流です。

PNPトランジスタの場合の各領域の電圧・電流特性を以下に示します。

- 遮断領域： $V_{eb} < V_r; I_b = 0; I_c = 0$
- 線形領域： $V_{eb} = V_r; I_c = \beta * I_b; V_{ec} < V_{ec, sat}$
- 飽和領域： $V_{eb} = V_r; I_c < \beta * I_b; V_{ec} = V_{ec, sat}$

ここで、 V_{eb} はエミッター-ベース間電圧、 V_{ec} はエミッター-コレクタ間電圧、 I_c はコレクタ電流です。

線形MOSFETスイッチはゲート-ソース間電圧、 V_{gs} で制御されます。スイッチの状態は遮断（オフ）、線形(リニア)、飽和（オン）のいずれかの領域です。

以下にn-チャンネルMOSFETの場合の各領域の特性を示します。

- 遮断領域： $V_{gs} < V_{gs(th)}; I_d = 0$
- 能動領域： $V_{gs} > V_{gs(th)}$ 及び $V_{gs} - V_{gs(th)} < V_{ds}; I_d = g_m * (V_{gs} - V_{gs(th)})$
- オーム領域： $V_{gs} > V_{gs(th)}$ 及び $V_{gs} - V_{gs(th)} > V_{ds}; I_d = V_{ds} / R_{ds(on)}$

ここで、 V_{gs} はゲート-ソース間電圧、 V_{ds} はドレイン-ソース間電圧、 I_d はドレイン電流です。

p-チャンネルMOSFETの場合の各領域の特性を示します。

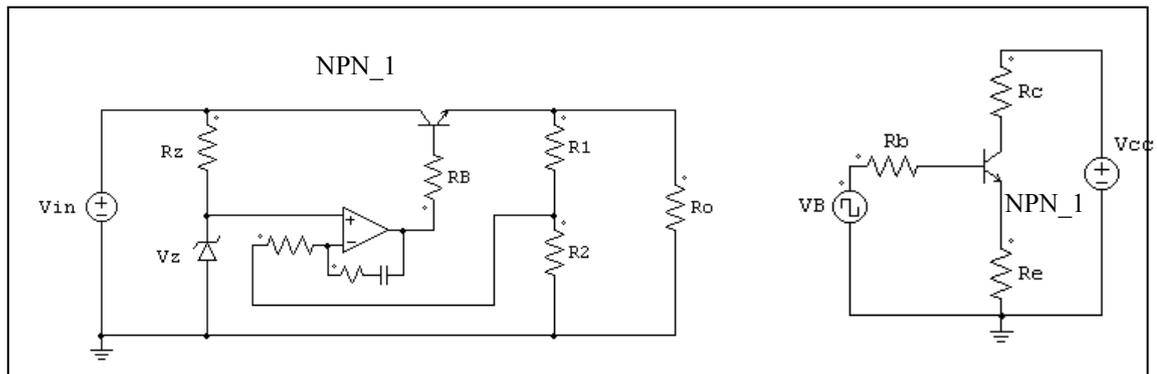
- 遮断領域： $V_{gs} > V_{gs(th)}; I_d = 0$
- 能動領域： $V_{gs} < V_{gs(th)}$ 及び $V_{gs} - V_{gs(th)} > V_{ds}; I_d = g_m * (V_{gs} - V_{gs(th)})$
- オーム領域： $V_{gs} < V_{gs(th)}$ 及び $V_{gs} - V_{gs(th)} < V_{ds}; I_d = V_{ds} / R_{ds(on)}$

線形スイッチのNPN/PNP BJT及びn-チャンネル/p-チャンネルMOSFETではゲート（ベース）ノードはパワー端子です。必ずパワー回路素子（抵抗や電源など）に接続するようにしてください。ゲート・ブロックやスイッチ制御回路に接続することはできません。

ご注意：BJT及びMOSFETの線形スイッチモデルは単純な回路では問題ありませんが、回路が複雑になると正しく動作しないことがあります。これらのモデルをご使用になるときは充分ご注意ください。

以下に線形スイッチモデルの使用例を示します。左側の回路は線形レギュレータ回路で、トランジスタは線形領域で動作します。右側の回路は簡単なテスト回路です。

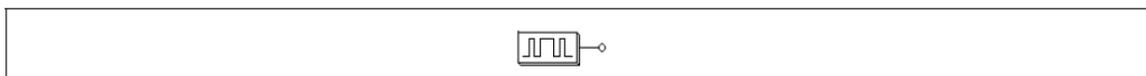
例：線形スイッチ NPN_1 を使った回路



2.2.6 スイッチ・ゲート信号ブロック

スイッチ・ゲート信号ブロックはスイッチやスイッチ・モジュールのゲート信号パターンを定義します。また、スイッチ素子のゲート端子にのみ接続できます。その他の素子には接続できません。

シンボル



仕様

パラメータ	機能
Frequency	ゲート信号ブロックに接続されたスイッチやスイッチ・モジュールの動作周波数 (Hz)
No. of Points	スイッチングポイント数 (<i>Gating Block</i> のみ)
Switching Points	スイッチングポイント (度数)。周波数が 0 の場合は、スイッチングポイントは秒を意味します。 (<i>Gating Block</i> のみ)
File for Gating Table	File 名指定による参照テーブル (<i>Gating Block (file)</i> のみ)

スイッチングポイント数は 1 周期内のスイッチング動作の総数を意味します。例えば、スイッチが 1 周期内に 1 度オンオフされた場合は、スイッチングポイント数は 2 になります。

Gating Table File は sch ファイルと同じフォルダに置いてください。Gating Table File のフォーマットは次のようになります。

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

Gn

G1、G2、Gnはスイッチングポイントです。

例：



スイッチが2000Hzで動作し、1周期内で以下のようなゲート・パターンを持つと仮定します。

PSIMでは、このスイッチのゲート・ブロックの仕様は以下となります。

周波数	2000
ポイント数	6
スイッチングポイント	35, 92, 175, 187, 345, 357

ゲート・パターンには6個所のスイッチングポイント（3パルス）があります。対応するスイッチング角はそれぞれ35°、92°、175°、187°、345°、357°となります。

Gating Block (file) を使う場合は次のようになります。

周波数	2000
スイッチングポイントファイル	test.tbl

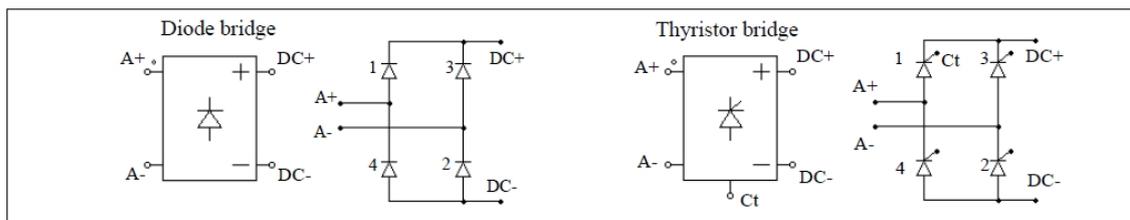
test.tblは次のようになります。

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

2.2.7 単相スイッチ・モジュール

PSIMには内蔵の単相ダイオードブリッジモジュール（single-phase diode bridge module）とサイリスタブリッジモジュール（thyristor bridge module）があります。シンボルとモジュールの内部接続を以下に示します。

シンボル



仕様

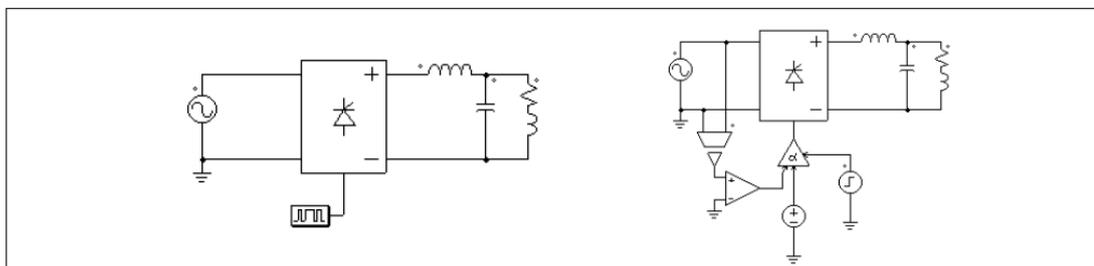
パラメータ	機能
Diode Voltage Drop or Voltage Drop	ダイオードかサイリスタの順方向電圧降下
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

サイリスタモジュールの底部にある極 Ct はスイッチ 1 用のゲート制御極です。サイリスタモジュールについては、スイッチ 1 用のゲート制御だけを指定します。他のスイッチのゲート制御はプログラム内部で参照されます。

単一サイリスタ・スイッチと同様に、サイリスタ・ブリッジも、以下の例に示すようにゲート信号ブロックか点弧角コントローラのいずれかで制御されます。

例：サイリスタ・ブリッジの制御

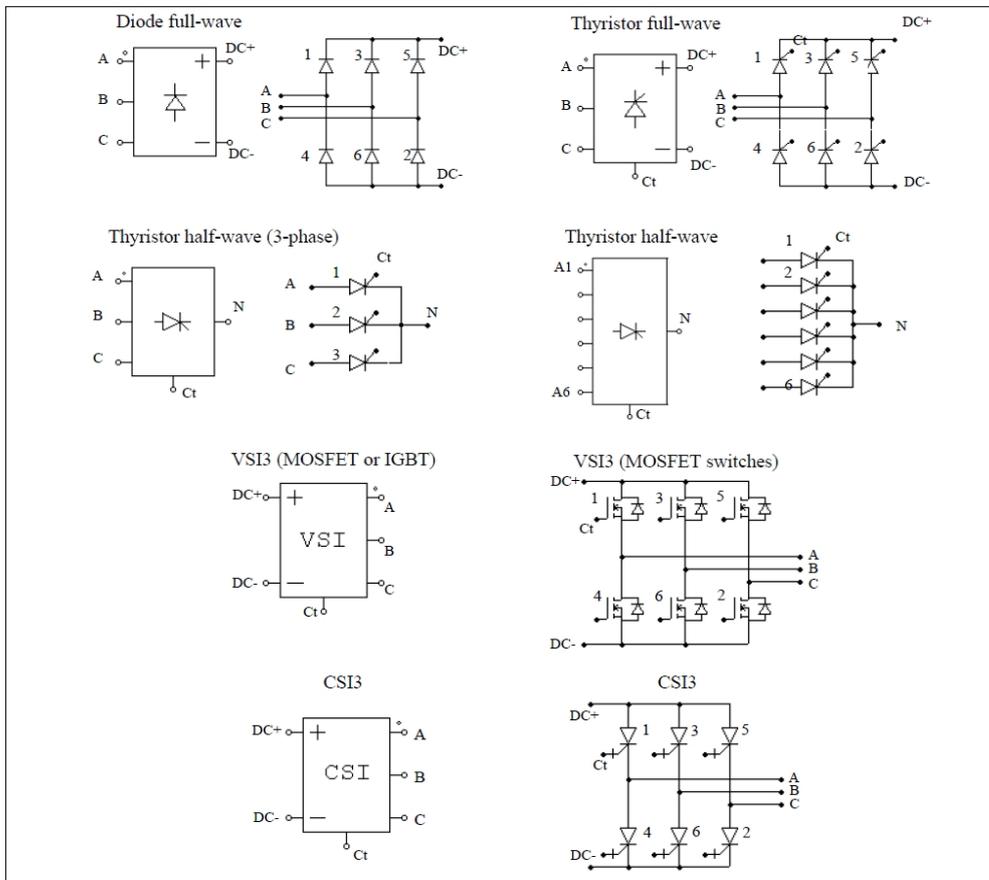
左側の電流ゲート制御はゲート信号ブロックで指定され、右側は点弧角コントローラで制御されます。点弧角コントローラの大きな利点は、サイリスタ・ブリッジの遅れ角 α を角度で直接に制御できる点です。



2.2.8 3相スイッチ・モジュール

以下の図に 3 相スイッチ・モジュールと内部接続を示します。

シンボル



仕様 (ダイオードブリッジ)

パラメータ	機能
Diode Threshold Voltage	ダイオードの閾値電圧降下 (V)
Diode Resistance	ダイオードのオン抵抗 (Ω)
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態フラグ
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

仕様 (サイリスタブリッジ)

パラメータ	機能
Voltage Drop	サイリスタの導通電圧降下 (V)
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態フラグ
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

仕様 (VSI3 ブリッジ) (※MOSFET スイッチ)

パラメータ	機能
On Resistor	MOSFET のオン抵抗 (Ω)
Diode Threshold Voltage	ダイオードの閾値電圧降下 (V)
Diode Resistance	ダイオードのオン抵抗 (Ω)
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態フラグ
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

仕様 (VSI3 ブリッジ) (※IGBT スイッチ)

パラメータ	機能
Saturation Voltage	IGBT トランジスタの飽和電圧 V_{ce_sat} (Ω)
Transistor Resistance	IGBT トランジスタのオン抵抗 (Ω)
Diode Threshold Voltage	逆並列ダイオードの閾値電圧降下 (V)
Diode Resistance	逆並列ダイオードのオン抵抗 (Ω)
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態フラグ
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

仕様 (CSI3 ブリッジ)

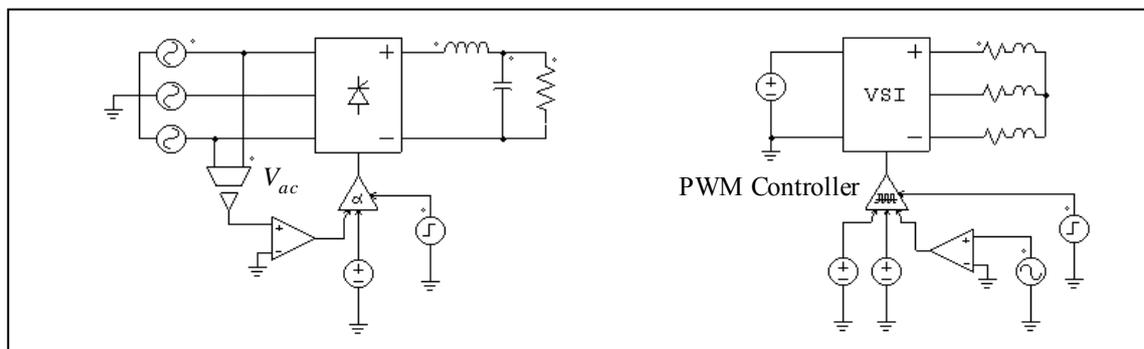
パラメータ	機能
Voltage Drop	スイッチの順方向導通電圧降下 (V)
Resistance	スイッチの順方向オン抵抗 (Ω)
Init.Position_ <i>i</i>	スイッチ <i>i</i> 用初期状態フラグ
Current Flag_ <i>i</i>	スイッチ <i>i</i> 用電流フラグ

単相モジュールと同様に、3 相モジュールについてはスイッチ 1 用のゲート制御だけを指定します。他のスイッチのゲート制御は自動的に参照されます。半波サイリスタ・ブリッジについては、隣接二相間で 120 度ずれています。その他の全てのブリッジについては 60 度ずれています。

サイリスタ・ブリッジは点弧角コントローラで制御されます。同様に PWM 電圧電流源インバータは PWM ルックアップテーブルコントローラで制御されます。

以下に 3 相電圧源インバータモジュールの制御の例を示します。

例：3 相 VSI モジュールの制御



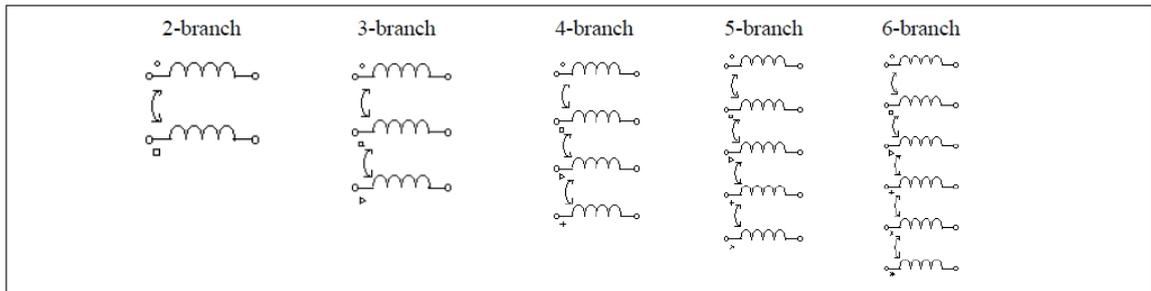
左側のサイリスタ回路は点弧角コントローラを使用します。3 相回路については、電圧 V_{ac} のゼロクロス時が遅れ角 $\alpha=0$ に対応します。そのため、この信号はコントローラに同期を与えるために使用されます。

右側の回路は PWM ルックアップテーブルコントローラを使用します。PWM パターンはテキスト・ファイルのルックアップテーブルに保存されます。ゲート・パターンは変調指標に基づいて選択されます。PWM ルックアップテーブル制御器の入力としては他に遅れ角、同期化、有効・無効信号があります。PWM ルックアップテーブル制御器の詳細は 4.7.3 において説明します。

2.3 相互結合インダクタ

PSIM では、2 から 6 までのブランチ 5 種類の相互結合インダクタが提供されています。

シンボル

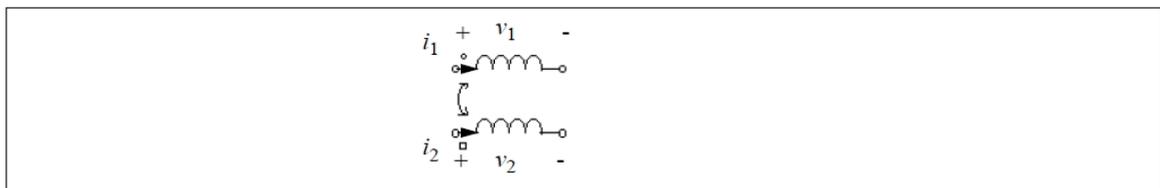


仕様

パラメータ	機能
L_{ii} (self)	インダクタ i の自己インダクタンス (H)
L_{ij} (mutual)	インダクタ i と j の間の相互インダクタンス
$i_{i_initial}$	インダクタ i の初期電流
IFlag_i	インダクタ i の電流出力フラグ

シンボルでは、○がインダクタ 1、□が 2、+が 3、+が 4、×が 5、*が 6 を示します。

下図には 2 ブランチの相互結合インダクタを示しています。



L_{11} と L_{22} をそれぞれブランチ 1 と 2 の自己インダクタンス、また L_{12} と L_{21} をそれぞれ相互インダクタンスとします。するとブランチ電圧と電流には以下のような関係になります。2 巻線間の相互インダクタンスは常に等しい($L_{12}=L_{21}$)と仮定しています。

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

例 :

相互結合インダクタは次のような自己インダクタンスと相互インダクタンスを持っているとします。

$L_{11}=1\text{mH}$ 、 $L_{22}=1.1\text{mH}$ 、 $L_{12}=L_{21}=0.9\text{mH}$ 。PSIM では MUT2 素子の仕様は以下のようです。

L_{11} (自己)	1.e-3
L_{12} (相互)	0.9e-3
L_{22} (自己)	1.1e-3

2.4 変圧器

2.4.1 理想変圧器

理想変圧器 (Ideal Transformer) には損失、漏れ磁束がありません。

シンボル



大きなドットの付いた巻線が1次側を表し、もう一方の巻線が2次側を表します。

仕様

パラメータ	機能
Np(primary)	1次巻線巻数
Ns(secondary)	2次巻線巻数

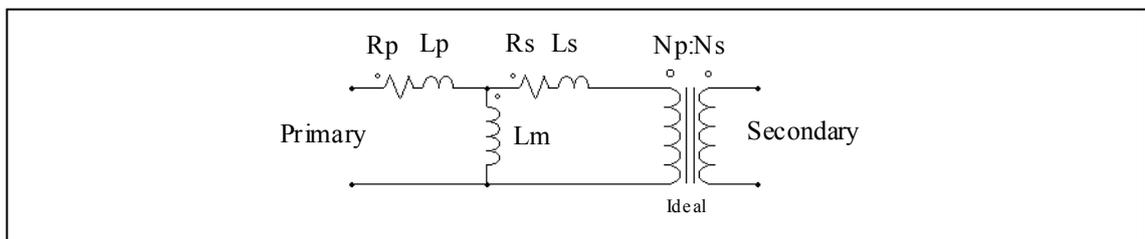
巻線比は定格電圧比に等しいので、巻数は各側の定格電圧で置き換えることができます。

2.4.2 単相変圧器

PSIM では、単相変圧器モジュールは以下のように備わっています。

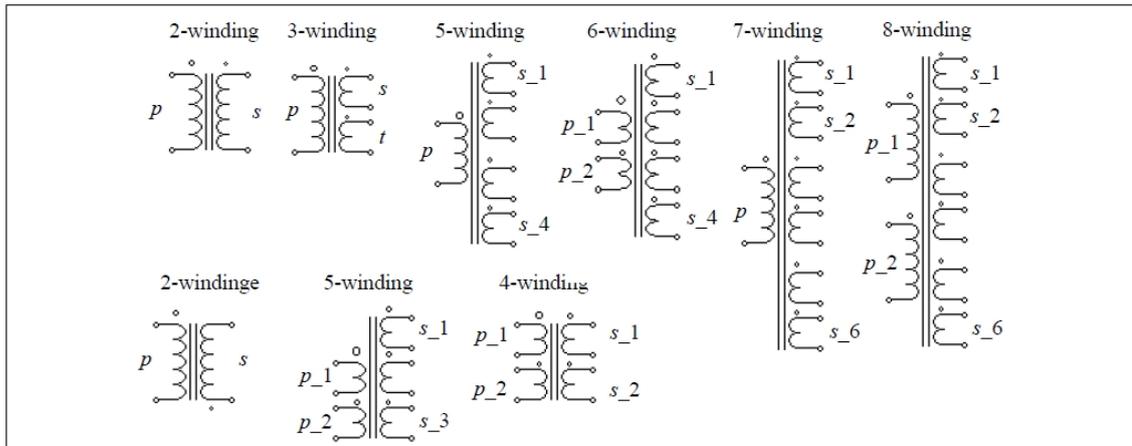
TF_1F/TF_1F_1	1次巻線が1と2次巻線が1の変圧器
TF_1F_3W	1次巻線が1と2次巻線が2の変圧器
TF_1F_4W	1次巻線が2と2次巻線が2の変圧器
TF_1F_5W/TF_1F_5W_1	1次巻線が1と2次巻線が4の変圧器
TF_1F_6W	1次巻線が2と2次巻線が4の変圧器
TF_1F_7W	1次巻線が1と2次巻線が6の変圧器
TF_1F_8W	1次巻線が2と2次巻線が6の変圧器

単相2巻線変圧器は以下のようにモデル化されています。



ここで R_p と R_s は1次/2次巻線抵抗、 L_p と L_s は1次/2次巻線漏れインダクタンス、 L_m は磁化インダクタンスです。全ての値は1次側から参照した値で表現されています。

シンボル



シンボルでは、p は 1 次側を、s は 2 次側を、t は 3 次側を表します。

大きなドットの付いた巻線が 1 次巻線（1 次巻線が 2 と 2 次巻線が 2 の変圧器 TF_IF_4W の 1 次巻線）を表します。多巻線変圧器については、巻線の順番は上から下です。

2 または 3 巻線変圧器についての仕様は以下のようになります。

仕様

パラメータ	機能
R_p (primary)	1 次巻線、2 次巻線、3 次巻線の抵抗 (Ω)。1 次側に換算
R_s (secondary)	
R_t (tertiary)	
L_p (pri.leakage)	1 次巻線、2 次巻線、3 次巻線の漏れインダクタンス (H)。1 次側に換算
L_s (sec.leakage)	
L_t (tertiary)	
L_m (magnetizing)	磁化インダクタンス (H)。1 次側に換算
N_p (primary)	1 次巻線、2 次巻線、3 次巻線の巻数
N_s (secondary)	
N_t (tertiary)	

すべての抵抗およびインダクタンスが 1 次巻線側に参照されます。

2 つ以上の 1 次巻線、または 3 以上の 2 次巻線を持つ変圧器についての仕様は以下のようになります。

仕様

パラメータ	機能
R_{p_i} (primary i)	i 番目の 1 次巻線、2 次巻線、3 次巻線の抵抗 (Ω)。1 次側に換算
R_{s_i} (secondary i)	
R_{t_i} (tertiary i)	
L_{p_i} (pri. i leakage)	i 番目の 1 次巻線、2 次巻線、3 次巻線の漏れインダクタンス (H)。1 次側に換算
L_{s_i} (sec. i leakage)	
L_{t_i} (tertiary i)	
L_m (magnetizing)	磁化インダクタンス (H)。1 次側に換算
N_{p_i} (primary i)	i 番目の 1 次巻線、2 次巻線、3 次巻線の巻数
N_{s_i} (secondary i)	

すべての抵抗およびインダクタンスが 1 次巻線側に参照されます。

例：

単相2巻線変圧器が巻線抵抗 0.002Ω 、1次側2次側両方の漏れインダクタンス 1mH (全て1次側に換算)、磁化インダクタンス 100mH 、巻線比 $N_p:N_s=220:440$ とします。PSIMでは、変圧器はTF_1Fを使って以下のような仕様を設定します。

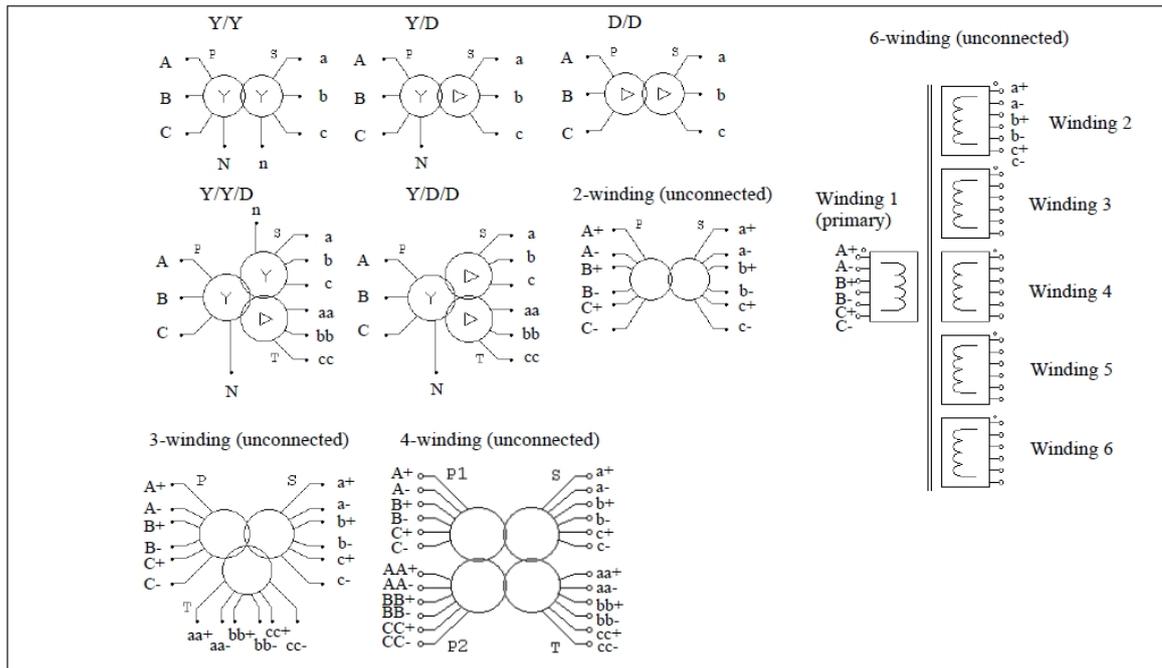
R_p (1次)	2m
R_s (2次)	2m
L_p (1次)	1m
L_s (2次)	1m
L_m (磁化)	100m
N_p (1次)	220
N_s (2次)	440

2.4.3 三相変圧器

PSIMには、以下に示すような2巻線と3巻線の変圧器モジュールが提供されており、全て三脚鉄心です。

- 三相変圧器 (巻線接続無し)
- 三相 Y/Y、Y/ Δ 接続変圧器
- 三相三巻線変圧器 (巻線接続無し)
- 三相三巻線 Y/Y、Y/ Δ 接続変圧器
- 三相四巻線変圧器 (巻線接続無し)
- 三相六巻線変圧器 (巻線接続無し)

シンボル



仕様

パラメータ	機能
R_p (primary)	1次巻線、2次巻線、3次巻線の抵抗 (Ω)。1次側に換算
R_s (secondary)	
R_t (tertiary)	
L_p (pri.leakage)	1次巻線、2次巻線、3次巻線の漏れインダクタンス (H)。1次側に換算
L_s (sec.leakage)	
L_t (ter.leakage)	
L_m (magnetizing)	磁化インダクタンス (H)。1次側に換算
N_p (primary)	1次巻線、2次巻線、3次巻線の巻数
N_s (secondary)	
N_t (tertiary)	

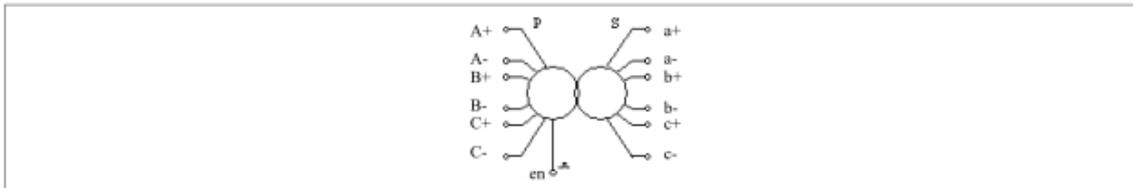
シンボルで、pは1次側を、sは2次側を、tは3次側を表します。

3相変圧器は単相変圧器と同様にモデル化されています。全てのパラメータは1次側に換算した値で表されています。

2.4.4 三相変圧器（可飽和）

飽和と残留磁束を考慮可能な三相3巻線の変圧器モジュールが提供されています。

シンボル



仕様

パラメータ	機能
Operating Voltage (phase)	変圧器 1 次側の一相あたりの電圧実効値 (V)
Operating Frequency	変圧器の動作周波数 (Hz)
R_p (primary)	1 次巻線抵抗 (Ω)
L_p (pri.leakage)	1 次巻線の漏れインダクタンス (H)
R_s (secondary)	2 次巻線抵抗 (Ω)、1 次側に換算。
L_s (sec.leakage)	2 次巻線の漏れインダクタンス (H)、1 次側に換算。
R_m (core losses)	変圧器のコア損失を表す抵抗 (Ω)
I_m vs. L_m	磁化電流 I_m (A) に対する一次巻線から見た磁化特性を定義する磁化インダクタンス L_m (H) の値
Phase A Residual Flux	A 相の単位当たりの残留磁束
Phase B Residual Flux	B 相の単位当たりの残留磁束
Phase C Residual Flux	C 相の単位当たりの残留磁束
N_p (primary)	1 次巻線、2 次巻線の巻数
N_s (secondary)	

すべての抵抗およびインダクタンスが 1 次巻線側に換算されます。

画像では、p は一次側を指し、s は二次側を指します。ノード en は、制御ロジック信号に接続する、制御ノードです。0 から 1 へ信号が変わる瞬間が、変圧器が通電されている瞬間を示します。

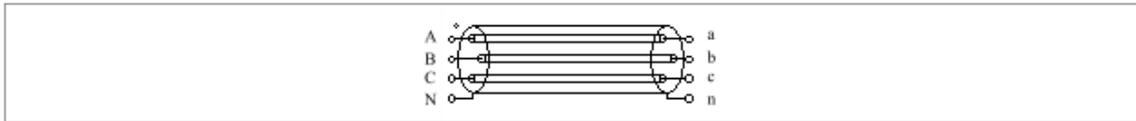
Operating Voltage は実際の動作条件における相あたりの電圧を指していて、定格電圧とは限らないことに注意してください。例えば、変圧器の定格は 63kV（一次側線間電圧実効値）で、一次巻線が Y に接続されている場合、相あたりの動作電圧は、 $63000/\sqrt{3}$ V となります。一次巻線がデルタに接続されている場合は、相あたりの動作電圧は 63000V になります。

同じ変圧器をデルタに接続して、例えば実際の動作電圧が 60000V である場合、相あたりの動作電圧は 60000V になります。

2.5 三相 AC ケーブル

三相 AC ケーブルのモデルは、誘導結合と相の間の容量が考慮されます。

シンボル



仕様

パラメータ	機能
Cable Length	ケーブルの長さ (m)
Operating Frequency	ケーブルの動作周波数 (Hz)
Resistance R_d	ケーブル正相抵抗 (Ω)
Reactance X_d	ケーブル正相リアクタンス (Ω)
Resistance R_0	ケーブル零相抵抗 (Ω)
Reactance X_0	ケーブル零相リアクタンス (Ω)
Capacitance C_0	ケーブル零相キャパシタンス (F)

画像では、ケーブルの両側の下部のノード N または n はケーブルの皮膜です。一般的にケーブルのグラウンドに接続されています。

ケーブルパラメータは、メーカーのデータシートから入手できるはずですが、それらが利用できない場合、ケーブルの各相は、R の抵抗、L の自己インダクタンス、および M の相互インダクタンスを持ち、正相および零相のパラメータは次のように計算できると仮定します。

$$R_d = R$$

$$X_d = \omega \cdot (L - M)$$

$$R_0 = R$$

$$X_0 = \omega \cdot (L + 2M)$$

ここで、 $\omega = 2\pi f$ で、f は動作周波数 (Hz) になります。

2.6 磁気要素

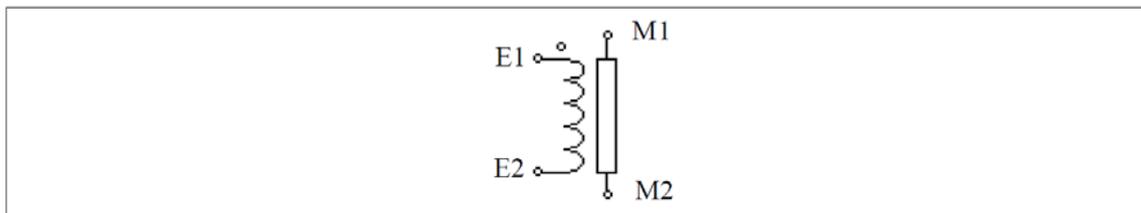
異なる単位系が磁気回路解析で用いられています。下の表は SI 系 (System International) と混成 CGS 系 (Centimeter Gram Second) 単位系の磁気量と二つの単位系の変換を示しています。

量	SI Unit	CGS Unit	CGS-SI 変換
磁束 ϕ	Weber	Maxwell	1 Weber=10 ⁴ Maxwell
磁束密度 B	Tesla	Gauss	1 Tesla=10 ⁴ Gauss
磁場強度 H	A*Turns/m	Oerstad	1 A*Turns/m= $\frac{1000}{4\pi}$ Oerstad
起磁力 (mmf) F	A*Turns	Gillbert	1 A*Turns= $\frac{10}{4\pi}$ Gillbert
透磁率 (真空) μ_0	4 π *10 ⁻⁷	1	4 π *10 ⁻⁷

2.6.1 巻き線

巻き線要素は電気回路と磁気等価回路とのインタフェースです。

シンボル



仕様

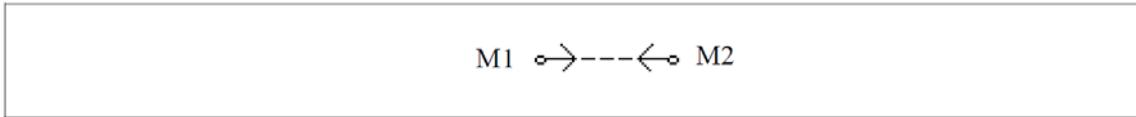
パラメータ	機能
Number of Turns	巻き線の巻数
Winding Resistance	巻き線抵抗

この要素は磁気コアの巻き線を示す。2つの磁気ノード (E1 と E2) は、電気回路に接続されます。2つの磁気ノード (M1 と M2) は、他の磁気要素(例えば漏れ磁束の通路、エアギャップと磁気コア)に接続されます。

2.6.2 漏れ磁束通路

この要素は漏れ磁束の通路をモデル化したものです。

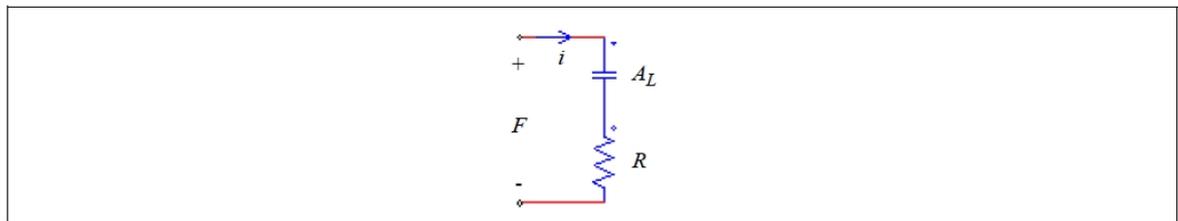
シンボル



仕様

パラメータ	機能
Inductance Factor A_L	コアのインダクタンスファクター(A_L)
Resistance for Losses	抵抗 $R(\Omega)$ は漏れ磁束による損失
Current Flag	抵抗に流れる電流のフラグ

抵抗 R は漏れ磁束による損失を表します。漏れ磁束経路に印加された起磁力が F であると仮定すると、漏れ磁束経路の電気等価回路は以下のように表せます。



起磁力は、電圧源の形でコンデンサ(静電容量は AL)と抵抗 R の間に印加されます。このブランチを通じて流れる電流は以下の通りになります。

$$i = \frac{F}{R^2 + \frac{1}{(\omega \cdot A_L)^2}}$$

ここで ω は起磁力のac周波数を示します。

漏れ磁束と抵抗 R による損失の関係は次のようになります。

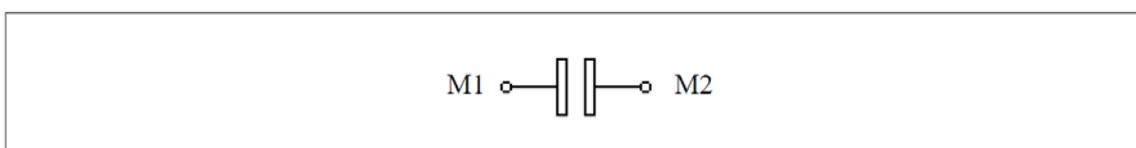
$$P_{loss} = I_{rms}^2 \cdot R$$

ここで I_{rms} は電流 i の実効値を示します。

2.6.3 エアギャップ

エアギャップ要素のシンボルと仕様は以下の通りです。

シンボル



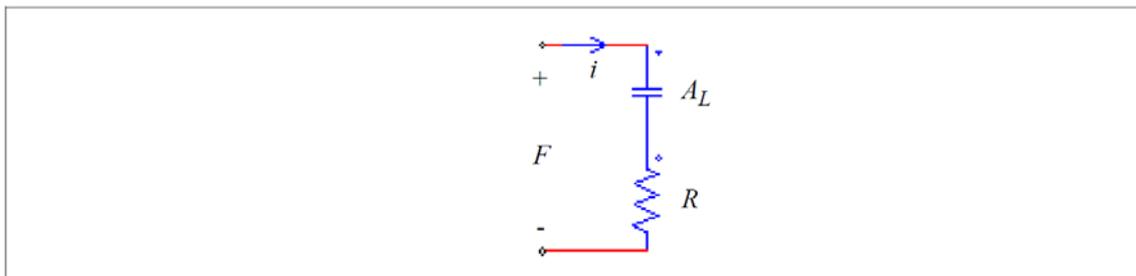
仕様(Air Gap)

パラメータ	機能
Air Gap Length	エアギャップの長さ、 l_g (m)
Cross Section Area	エアギャップの断面積、 A_c (m^2)
Resistance for Losses	抵抗 $R(\Omega)$ はエアギャップフリンジ効果による損失を表します
Current Flag	抵抗に流れる電流のフラグ

仕様(Air Gap(AL))

パラメータ	機能
Inductance Factor A_L	コアのインダクタンスファクター(A_L)
Resistance For Losses	抵抗 $R(\Omega)$ はエアギャップフリンジ効果による損失を表します
Current Flag	抵抗に流れる電流のフラグ

抵抗 R はエアギャップフリンジ効果による損失を表します。漏れ磁束経路に加えられた起磁力が F であると仮定すると、エアギャップの電気等価回路は以下の通りです。



起磁力は、電圧源の形でコンデンサ(静電容量は AL)と抵抗 R の間に加えられます。インダクタンスファクターは次のように定義される。

$$A_L = \frac{\mu_0 \cdot A_c}{l_g}$$

ここで、 $\mu_0 = 4\pi \times 10^{-7}$ とします。

抵抗における損失はフリンジ効果による損失を表す。この損失は次の通りである。

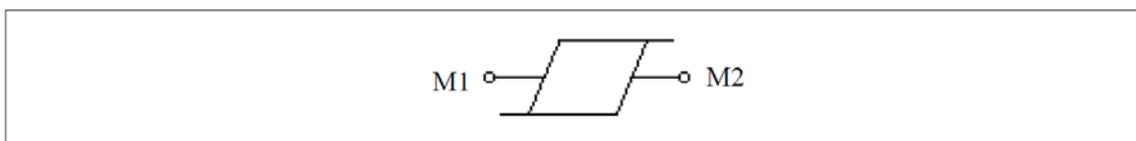
$$P_{loss} = I_{rms}^2 \cdot R$$

ここで、 I_{rms} は抵抗 R を通じて流れる電流 i の実効値を表します。

2.6.4 線形コア

この要素は線形無損失コアを表します。

シンボル



仕様

パラメータ	機能
Inductance factor, A_L	コアのインダクタンス・ファクター(A_L)

コアの長を L 、断面積を A_c とするならばインダクタンス要因 A_L は次のように表現します。

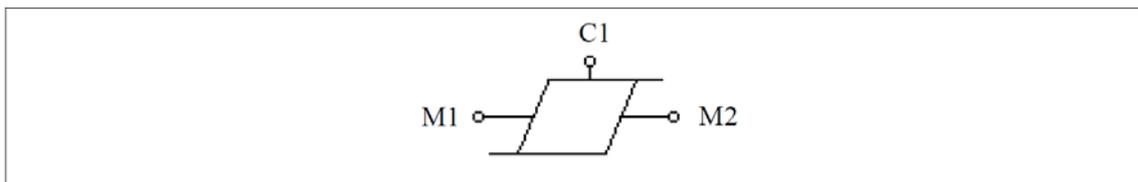
$$A_L = \frac{\mu_o \cdot \mu_r \cdot A_c}{L}$$

ここで、 μ_r はコア材料の相対浸透率を示します。

2.6.5 可飽和コア

この要素は飽和およびヒステリシスを用いる磁気コアをモデル化したものです。

シンボル



仕様

パラメータ	機能
Inductance Factor A_L	コアのインダクタンス・ファクター(A_L)
Resistance for Losses	コアによる損失を表す抵抗 $R(\Omega)$
Coefficient ϕ_{sat}	コア B-H カーブの係数 Φ_{sat} (Weber)
Coefficient K_1	コア B-H カーブの係数 K_1
Coefficient K_{exp1}	コア B-H カーブの係数 K_{exp1}
Coefficient K_2	コア B-H カーブの係数 K_2
Coefficient K_{exp2}	コア B-H カーブの係数 K_{exp2}
Initial Flux ϕ_o	コアの初期磁束 Φ_o (Weber)
Current Flag	抵抗 R に流れる電流のフラグ。損失計算に電流を使用することが可能

シンボル中のノード M1 と M2 は他の磁気要素(巻線、漏れ磁束経路、エアギャップ等)に接続するノードです。またノード C1 はコアの磁束を示す出力ノードです。このノードに電圧プローブを接続することによって、コアの磁束を表示することが出来ます。ノード C1 は制御回路ノードを示します。

係数の K_1 、 K_{exp1} 、 K_2 、および K_{exp2} は、実際の磁性材料の B-H カーブに合うのに使用されます。係数 K_1 は、コア材料によって値が異なりますが通常 0.7~1 の値をとります。

係数 K_{exp1} は主にコアの飽和率に影響を与え、10~200 の範囲(低透過性フェライトの場合 10、metglas の場合 200) になります。

係数 K_2 と K_{exp2} は非常にまれな時(例えば、鉄共振の調節装置)に使用されます。 K_2 、 K_{exp2} は B-H カーブに影響を与えないように以下の通りに設定されます。

$$K_2 > 2$$

$$K_{exp2} > 20$$

コア係数を決定するためのチュートリアルが PSIM インストールフォルダの "doc" フォルダにあります。また、コアの B-H カーブをプロットするためのプログラム (Utilities->B-H Curve) が用意されています。

2.7 その他エレメント

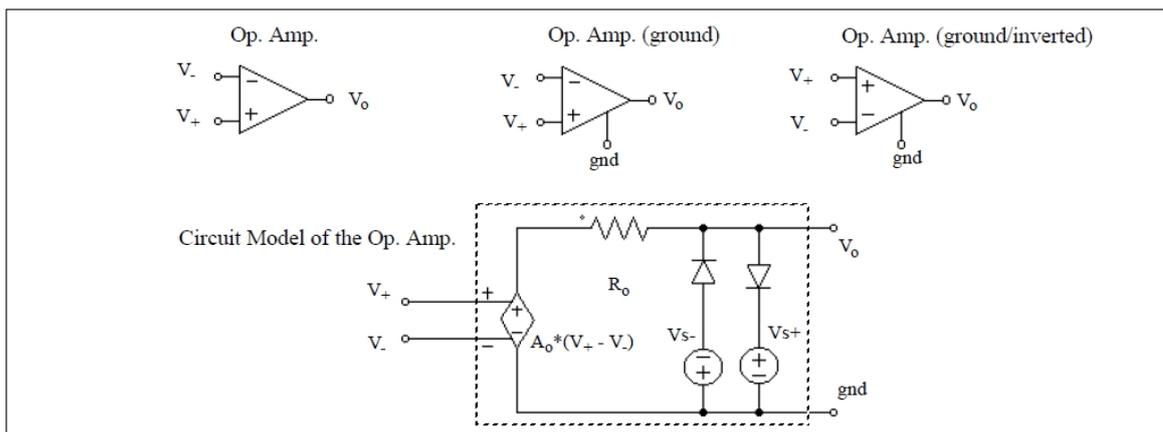
2.7.1 オペアンプ

2種類のアンプモデル (理想アンプ、非線形アンプ) が提供されています。

2.7.1.1 オペアンプ (理想アンプ)

PSIM では次の 3 種類の理想演算増幅器 (オペアンプ) があります ; *Op.Amp*、*Op.Amp(ground)*、*Op.Amp(ground/inverted)*。理想アンプは以下の図に示すパワー回路素子を使ってモデリングされています。

シンボル



ここで、

- V_+ ; V_- 非反転および反転入力電圧
- V_o 出力電圧
- A オペアンプのゲイン (プログラム上は $A=100,000$)
- R_o 出力抵抗値 (プログラム上は $R_o=80\Omega$)

仕様

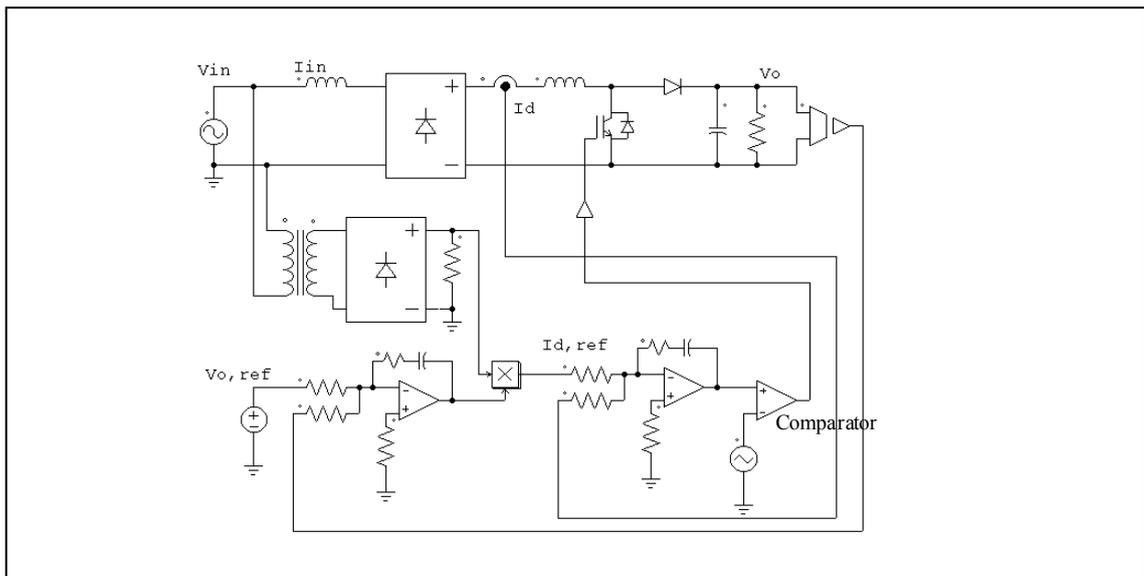
パラメータ	機能
Voltage Vs+	オペアンプの上限値電源電圧
Voltage Vs-	オペアンプの下限値電源電圧

Op.Amp、*Op.Amp(ground)*、*Op.Amp(ground/inverted)*の違いはGNDです。*Op.Amp*では基準接地ノードがパワー回路のGNDに接続されているのに対して、*Op.Amp(ground)*と*Op.Amp(ground/inverted)*のGND端子は独立に接続可能で、浮かしておくことも可能です。

オペアンプのシンボルは比較器に似ているので注意してください。オペアンプでは、反転入力（-）が左上にあり、非反転入力（+）は右下です。比較器ではこれが逆になります。

例.力率改善ブースト回路

以下の回路は力率改善ブースト回路の例です。電流ループが内ループ、電圧ループが外ループを形成しています。オペアンプを使って電圧・電流ループともPIレギュレータを構成しています。

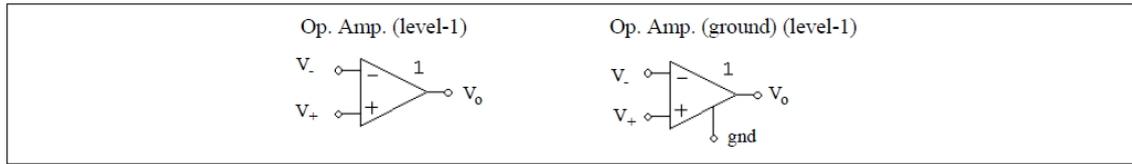


2.7.1.2 オペアンプモデル（非線形オペアンプ）

理想オペアンプモデルに対し、本オペアンプモデルにはバンド幅制限と出力電流制限を定義することができます。

PSIMは *Op.Amp.(level-1)* と *Op.Amp(ground)(level-1)* の2つオペアンプモデルを提供しています。*Op.Amp.(level-1)* と *Op.Amp(ground)(level-1)* との違いはGNDです。*Op.Amp.(level-1)*では基準接地ノードがパワー回路のGNDに接続されているのに対して、*Op.Amp(ground)(level-1)*のGND端子は独立に接続可能で、浮かしておくことも可能です。

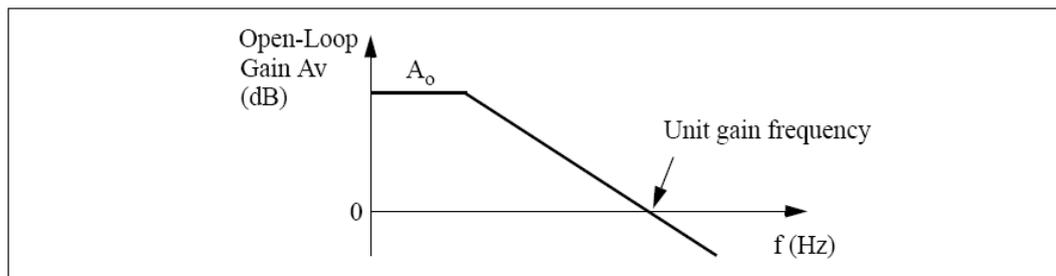
シンボル



仕様

パラメータ	機能
Input Resistance	オペアンプの入力抵抗 (単位: Ω)
DC Gain A_0	オペアンプの直流ゲイン
Unit Gain Frequency	オペアンプのゲインが 1 である周波数 (単位:Hz)
Output Resistance R_O	オペアンプの出力抵抗 (単位: Ω)
Maximum Output Current	オペアンプの最大出力電流(単位:A)
Voltage V_{s+}	オペアンプの上限値電源電圧
Voltage V_{s-}	オペアンプの下限値電源電圧

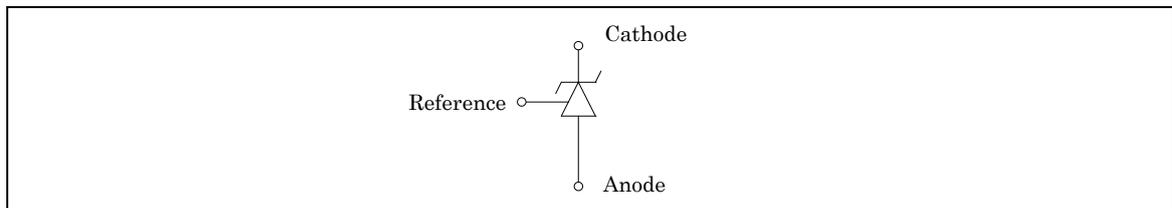
オペアンプのオープンループゲイン特性を以下に示します。



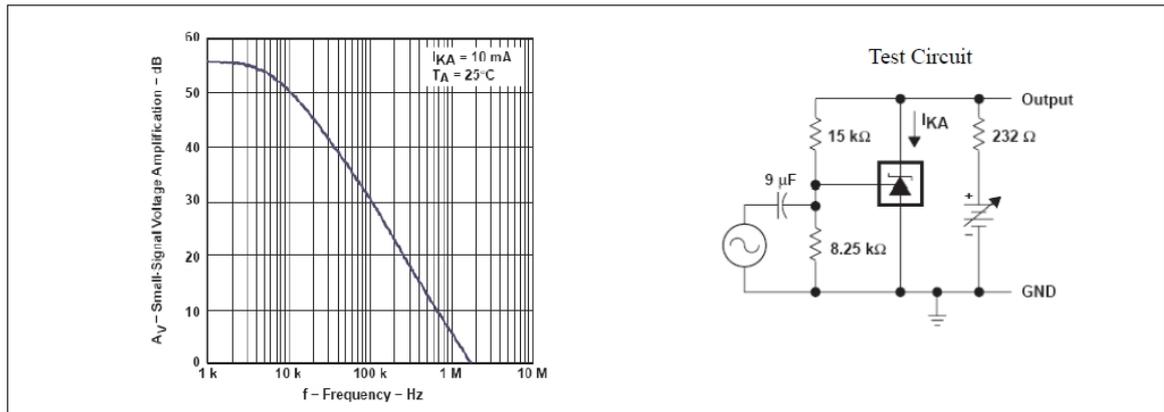
2.7.2 TL431 シャントレギュレータ

TL431 シャントレギュレータ (TL431 shunt regulator) のシンボルは、以下の通りです。

シンボル



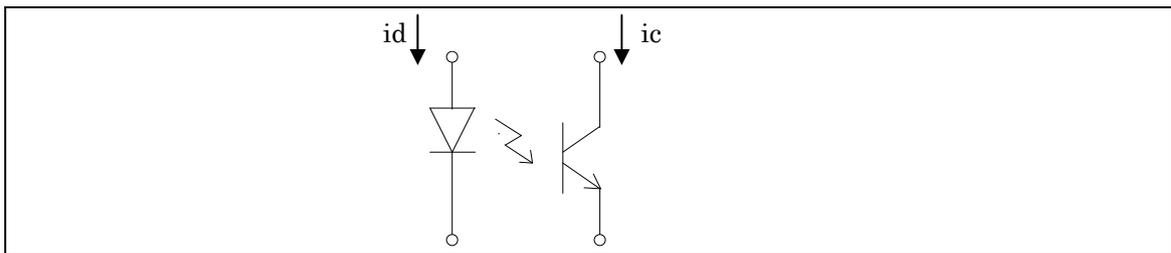
TL431 レギュレータは基準ノードの電圧を約 2.495V に保ちます。このモデルでは、TL431 の有限帯域幅がモデル化されます。すなわち、TL431 モデルの AC 周波数特性はデータシートの図のようになります。



2.7.3 フォトカプラ

フォトカプラ(Optocoupler)のシンボルおよびパラメータは以下の通りです。

シンボル



仕様

パラメータ	機能
Current Transfer Ratio	トランジスタ電流 i_c とダイオード電流 i_d 間の電流伝達率。 $CTR=i_c/i_d$
Diode Resistance	ダイオード抵抗 R_d (Ω) ※必ず 0 より大きい値に設定して下さい。
Diode Threshold Voltage	ダイオード順方向閾値電圧 V_{d_th} (V)
Transistor Vce_sat	トランジスタ飽和電圧 V_{ce_sat} (V)
Transistor-side Capacitance	トランジスタのコレクタ-エミッタ間のキャパシタンス C_p (F)

これらのパラメータは各メーカーのデータシートから直接得たり、データシートの情報から計算したりすることで得られます。Motorola 社のフォトカプラ 4N25 を例に挙げます。データシートより、電流伝達率およびトランジスタ飽和電圧は

$$CTR = 70\% = 0.7; v_{ce_sat} = 0.15 \text{ V}$$

データシートの“LED 順方向電圧 vs 順方向電流”の特性から、ユーザはダイオード抵抗 R_d と閾値電圧 V_{d_th} を計算することが可能です。ダイオード順方向電圧は、以下の比例直線に近似することが可能です。

$$v_d = V_{d_th} + R_d * i_d$$

$T_A=25^\circ\text{C}$ の時、この曲線から $i_d=10\text{mA}$ 、 $v_d=1.15\text{V}$; $i_d=40\text{mA}$ 、 $v_d=1.3\text{V}$ の2つが読み取れます。これらの2つの点に基づき、 R_d と V_{d_th} の値は次のように計算されます： $R_d=5\Omega$ 、 $V_{d_th}=1.1\text{V}$ 。

このフォトカプラモデルは遅れを考慮していませんが、キャパシタ C_p を通ってトランジスタのコレクタ-エミッタ間のターンオン/ターンオフ過渡電流をモデリングしています。このキャパシタンスは立下り時間 t_{fall} とスイッチング時間試験条件より、以下のように求められます。

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

ここで R_L はスイッチング時間試験回路中の抵抗負荷です。4N25 では、データシートより $t_{fall}=1.3\mu\text{s}$ 、 $R_L=100\Omega$ となり、これよりキャパシタンスは $C_p=6\text{nF}$ と求められます。

2.7.4 dv/dt ブロック

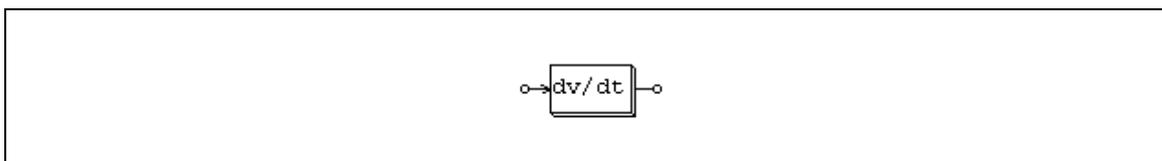
dv/dt ブロックは制御回路の微分器と同等の機能を提供します。唯一の違いは、この素子はパワー回路用というだけです。dv/dt ブロックの出力は入力電圧の時間に関する微分です。

微分は以下のように計算しています。

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

ここで、 Δt はシミュレーションタイムステップ、 $v_{in}(t)$ と $v_{in}(t-\Delta t)$ はそれぞれ現在時刻の入力と1ステップ前の入力です。

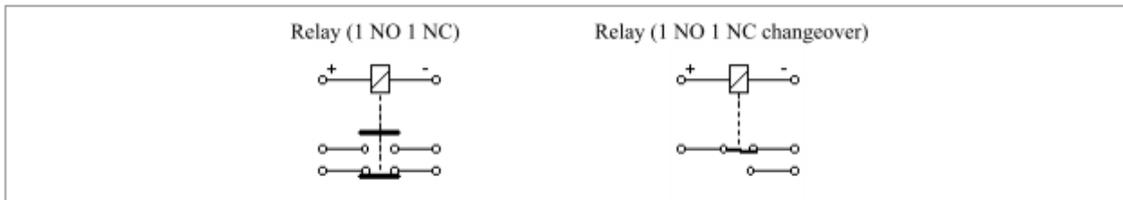
シンボル



2.7.5 リレー

ノーマルオープン（NO）スイッチとノーマルクローズ（NC）スイッチと2つのリレーブロックが用意されています。

シンボル



仕様

パラメータ	機能
Rated Coil Voltage	リレーコイルの定格電圧 (V)
Coil Resistance	コイルの抵抗 (Ω)
Operate Voltage	いずれかのリレーが動作する電圧 (V)
Release Voltage	いずれかのリレーが初期位置に戻る電圧 (V)
Operate Time	動作電圧に到達してからスイッチが動作するまでの時間 (s)
Release Time	解除電圧に到達してからスイッチが解除されて初期位置に戻るまでの時間 (s)

リレーは1つのノーマルオープンスイッチおよび1つのノーマルクローズスイッチを備えています。画像に示すような極性で直流電圧がリレーコイルに印加された時、電圧が[Operate Voltage]に達すると、[Operate Time]によって定義された時間遅延後に、NOスイッチは閉じ、NCスイッチは開きます。コイル電圧が[Release Voltage]に下がると、[Release Time]で定義された時間遅延後に、2つのスイッチは初期位置に戻ります。

2.8 熱モジュール(Thermal モジュール)

熱モジュールはPSIMソフトウェアのアドオン・モジュールです。半導体デバイス(ダイオード、IGBTおよびMOSFET)の損失を計算することができます。

損失は電流として出力されます。温度上昇値を知るためには熱抵抗($^{\circ}\text{C}/\text{W}$)の値を抵抗素子で模擬します。抵抗の電圧降下が温度上昇値になります。

熱モジュールのコアはデバイスデータベースです。PcdEditor.exeというデバイスデータベースエディタを利用して、データベースへ新しいデバイスを加えることができます。データベースを容易に管理することも可能です。デバイスのデータベースは損失計算のためにシミュレーションで使用されます。

データベースのデバイスをシミュレーションで使用する方法と、損失を計算する方法を以下に示します：

- － デバイスのビヘイビアモデルをシミュレーションで使用します。ビヘイビアモデルは、動特性(ターンオンおよびターンオフの過渡現象)ではなく、デバイスの静特性(順方向電圧降下またはオン抵抗など)を考慮に入れます。
- － 計算された電圧と電流に基づいて、PSIMはデバイスデータベースにアクセスし、導通損失あるいはスイッチング損失を計算します。デバイスの静特性は次のシミュレーションのために更新されます。

本損失計算は近似計算です。結果の正確さはデバイスデータの正確さに依存します。実回路動作条件でのデバイスをテストして得た結果が厳密にデバイスデータに反映されているかどうかです。ユーザは、シミュレーションによる損失計算結果とハードウェアから測定した結果を検証する必要があります。

次のセクションでは、データベースへデバイスをどのように加えることができるか、また、それをシミュレーションでどのように使用することができるかを説明します。

2.8.1 デバイスデータベースエディタ

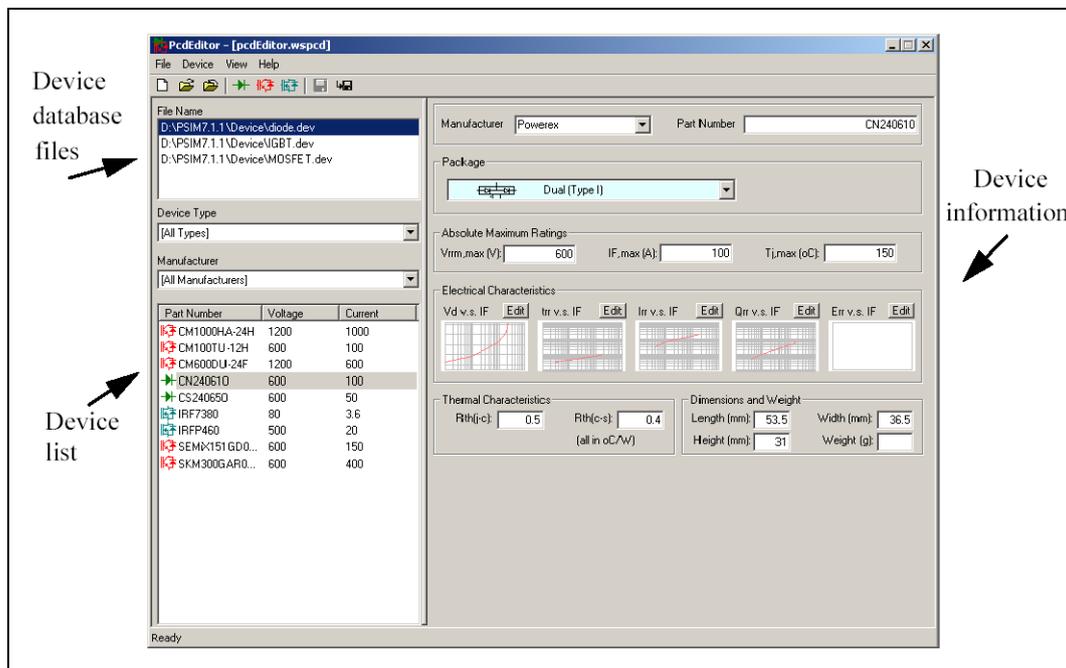
デバイスデータベースエディタ(PcdEditor.exe)は、デバイスを追加、編集、及び管理するためのものです。データベースエディタの図を以下に示します。

左側には、データベースエディタへロードされるデバイスデータベースファイルおよびデバイスのリストがあります。デバイスはデバイスのタイプかメーカーのいずれかに基づいて表示することができます。さらに、リストのタイトルバーをクリックすることにより、部品番号、定格電圧あるいは定格電流によってデバイスリストをソートすることができます。

右側に各デバイスの情報が示され主に次の情報が各デバイスに定義されます：

- ・ メーカーと部品番号
- ・ パッケージ型式
- ・ 絶対最大定格

- ・ 電気的特性
- ・ 熱的特性
- ・ 寸法と質量



新しいデバイスファイルを作成するためには、**File > New Device File**を選んでください。デバイスファイルをエディタにロードするためには、**File > Open Device File**を選んでください。デバイスファイルを閉じるには、**File > Close Device File**を選んでください。

デバイスファイルにはダイオード、IGBTおよびMOSFETの3タイプのデバイスを加えることができます。しかしながら、通常のIGBTデバイスと比べて、デュアルIGBT-ダイオードモジュールには異なったパラメータがあるので、それらは別々のタイプ（IGBT-ダイオードタイプ）として扱われます。

新しいデバイスを作成するためには、**Device(デバイス)**メニューに移動して、**New Diode(新しいダイオード)**、**New IGBT(新しいIGBT)**、**New IGBT-DIODE(新しいIGBTダイオード)**あるいは**New MOSFET(新しいMOSFET)**を選んでください。

同じデータベースファイルに既存のデバイスのコピーを作るために、リストのデバイスをハイライトして**Device > Save Device As**を選んでください。既存のデバイスのコピーを作って別のデータベースファイルに保存するために、最初にリスト中のデバイスを強調して、次に、File Nameリスト中のファイル名をハイライトしてください。それから、**Device > Save Device As**を選んでください。

データベースにデバイスを追加:

デバイスデータベースファイルに装置を加える方法を明らかにするために、デバイスデータベースファイル「diode.dev」へPowerexのディスクリットダイオードCS240650(600V、50A)を追加する手順を以下に示します。

– PcdEditor.exeを起動してください。Fileメニューの**New Device File**を選択し、“diode.dev”と呼ばれるファイルを作成してください。このファイルはデフォルトでPSIMプログラム・フォルダの下のデバイスサブ・フォルダに置かれます。

– 「File Name」リストでファイル名「Diode」をハイライトしたまま、**Device ->New Diode**を選んでください。メーカーが「New」と部品番号が「New」状態でダイオードはデータベースファイルに追加されます。

– スクリーンでPowerex CS240650データシートのPDFファイルを表示してください。

– PcdEditorで、データシートからこのデバイスについて次の情報を入力してください。

メーカー: Powerex
 部品型式: CS240650
 パッケージ: Discrete

絶対最大定格 (**Absolute Maximum Ratings**) :

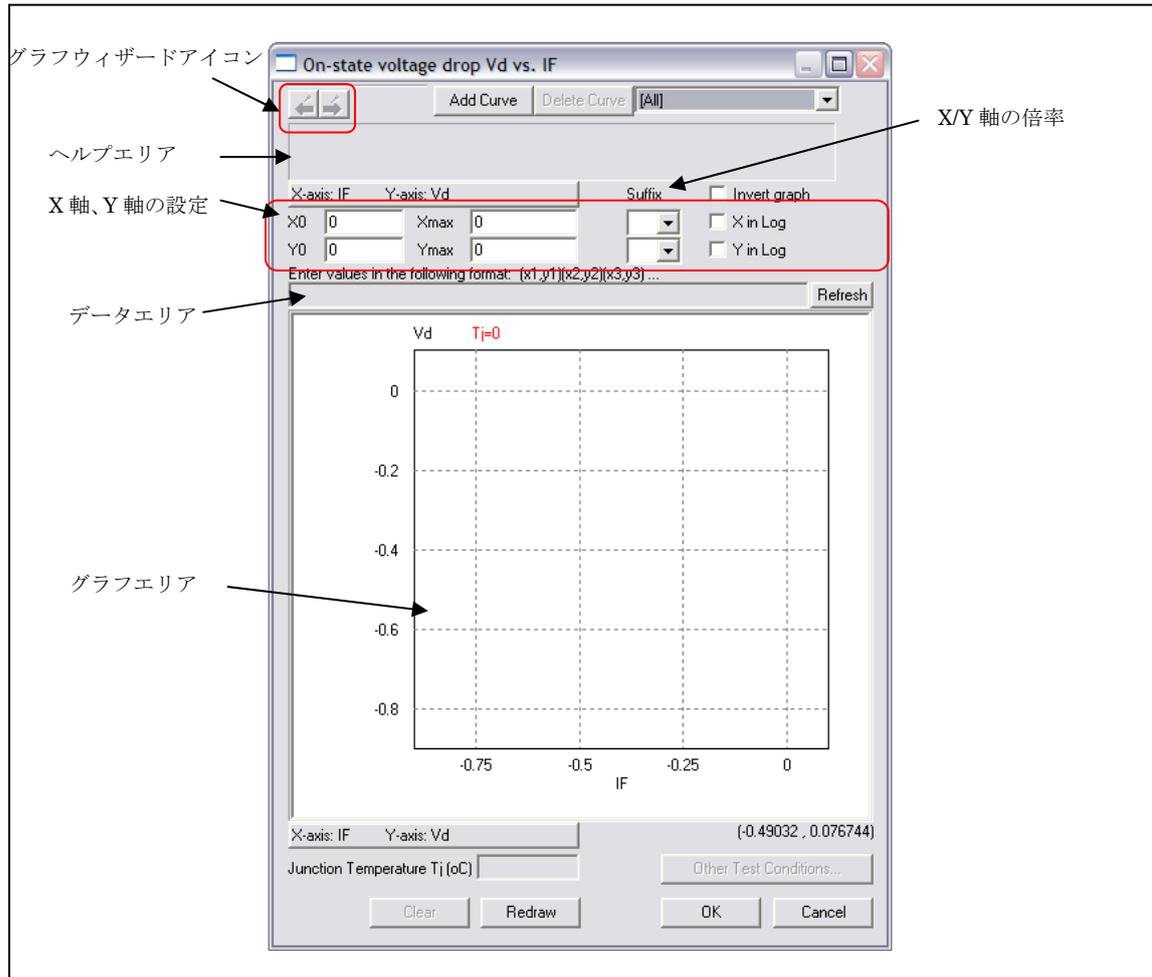
$V_{rrm,max}$ (V): 600
 $I_{F,max}$ (A): 50
 $T_{j,max}$ (°C): 150

– V_d v.s. I_F グラフ領域の上にある**Edit**ボタンをクリックして電気的特性、順電圧特性 V_d v.s. I_F を定義してください。

以下のダイアログウィンドウが表示されます。ダイアログウィンドウは2ページあります。グラフ (**Graph**) と条件 (**Conditions**) 。

グラフページはx軸とy軸セッティング、データポイントおよびグラフを含んでいます。この場合、y軸は順方向電圧降下 V_d でx軸は順方向電流 I_F です。軸は倍率 (10^{-3} はm、 10^{-6} はuを使用可能) を持つことができます。

条件ページは、グラフが得られる条件を含んでいます。



グラフを定義するには2つの方法があります。一つの方法は手動でグラフデータ・ポイントを入力することです。もう一つの方法はグラフウィザードを使用し、データシートからグラフを直接得ることです。データ・ポイントが少ない場合は、手動でグラフを定義した方が簡単です。しかしながら、グラフがあるならばグラフウィザードを使用するとより簡単になります。

手動でグラフを定義:

データシートの「Maximum On-State Characteristics」グラフを参照して、x/y軸セッティングに対する値を以下のように入力してください。

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

一目視によりグラフを観察して、数データ・ポイントを選択してください。それから、データ域に次のデータ・ポイントを入力してください。

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

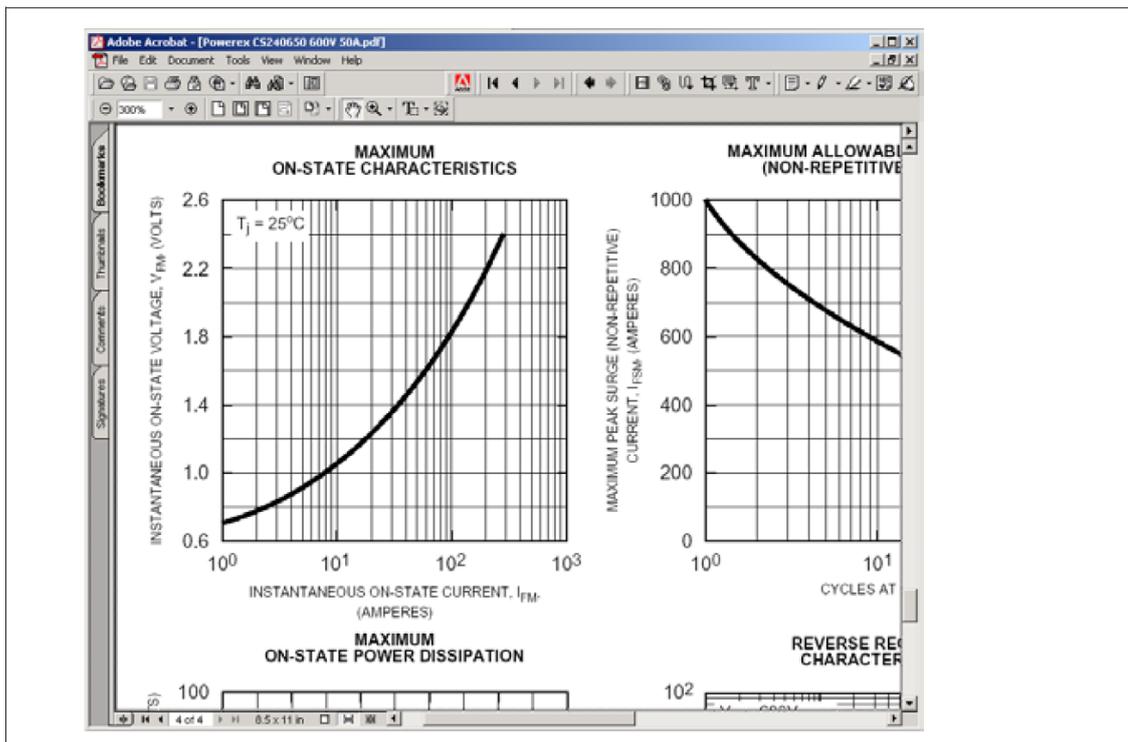
それからグラフを表示するために、**Refresh**ボタンをクリックしてください。

—**Conditions** タブをクリックして、25°Cの**Junction Temperature**を入力してください。

グラフウィザードを利用してグラフを定義:

—グラフウィザードを始めるために、フォワードウィザードアイコン  をクリックしてください。

—以下の通りにデータシートのグラフをスクリーンに表示してください。



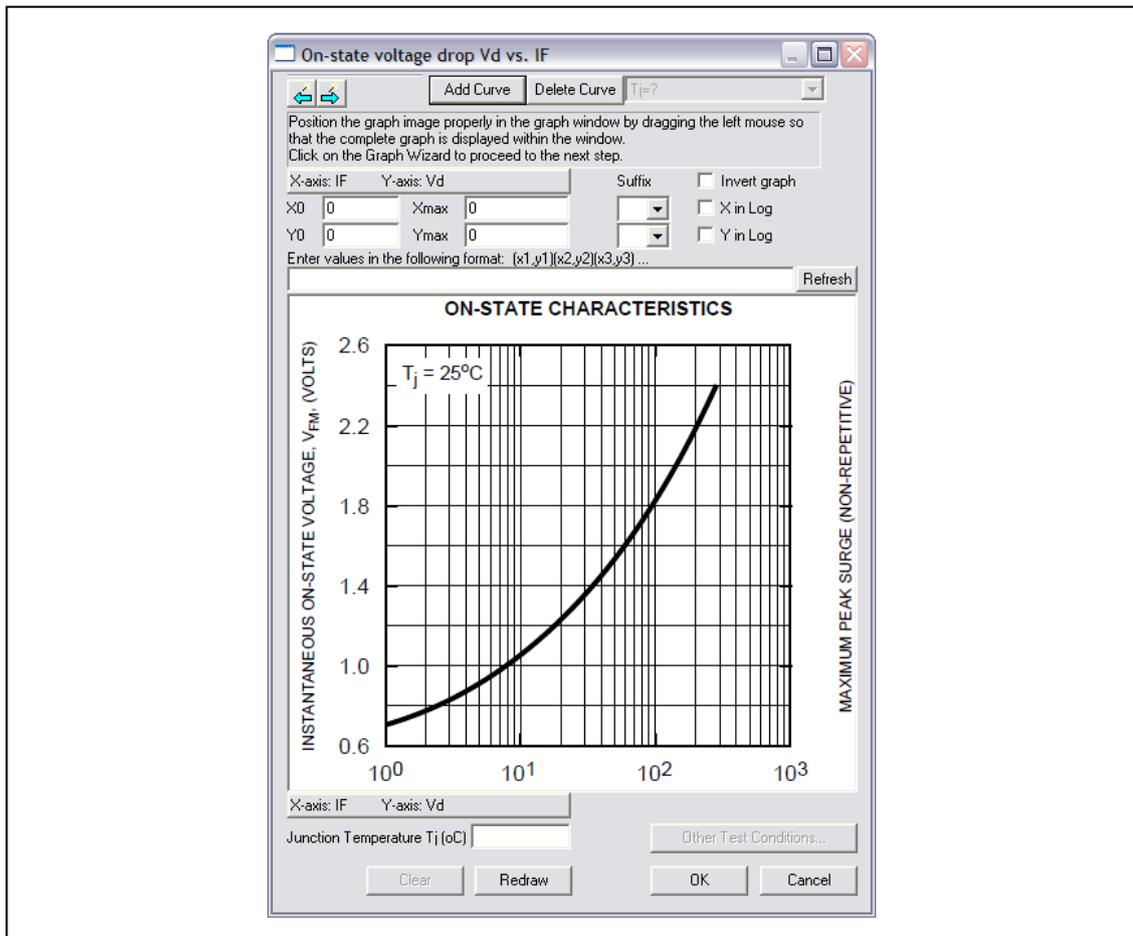
そして、プリントスクリーンキー(キーボード上の表記「Prt Scrn」)を押して、クリップボードにスクリーン図をコピーしてください。

スクリーン図をPcdEditorのグラフ・ウィンドウに貼り付けるために、フォワードウィザードアイコン



をクリックしてください。ウィンドウの中で完全なグラフが表示されるように、マウスの左ボタンを押し続けてドラッグしてきちんとグラフをグラフウィンドウに置いてください。

グラフが大き過ぎるか、または小さ過ぎるなら、バックワードウィザードアイコン  をクリックして、以前のステップに戻ってください。次に、Adobe Acrobatでグラフの図をリサイズして、もう一度クリップボードにスクリーン図をコピーしてください。グラフダイアログウィンドウは以下のように表示されます。

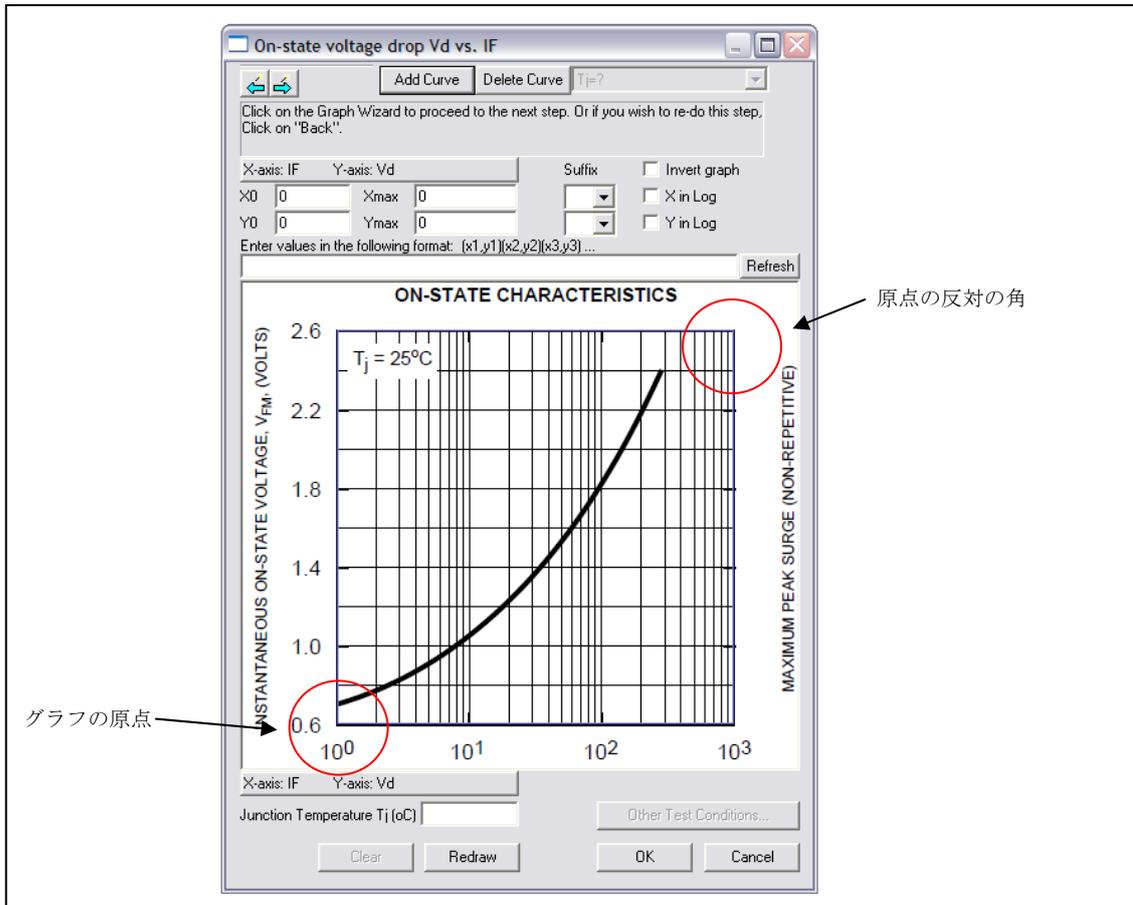


フォワードウィザードアイコン  をクリックして次のステップに進んでください。

—このステップでは、グラフ領域の境界を定義します。グラフの原点（通常左下隅）でマウスの左ボタンをクリックし、原点の反対の角（通常右上隅）でマウスの右ボタンをクリックします。グラフの原点が左下隅でなく各コーナーのうちのいずれか1つである場合もあります。

グラフ原点の位置をより正確に見つけるために、マウスの右ボタンをクリックして拡大してください。ズームを解除するためには、Esc(Escape)キーを押してください。

この後に、青い長方形がグラフの境界の近くに現われます。また、ダイアログ・ウィンドウが以下のように表示されます。



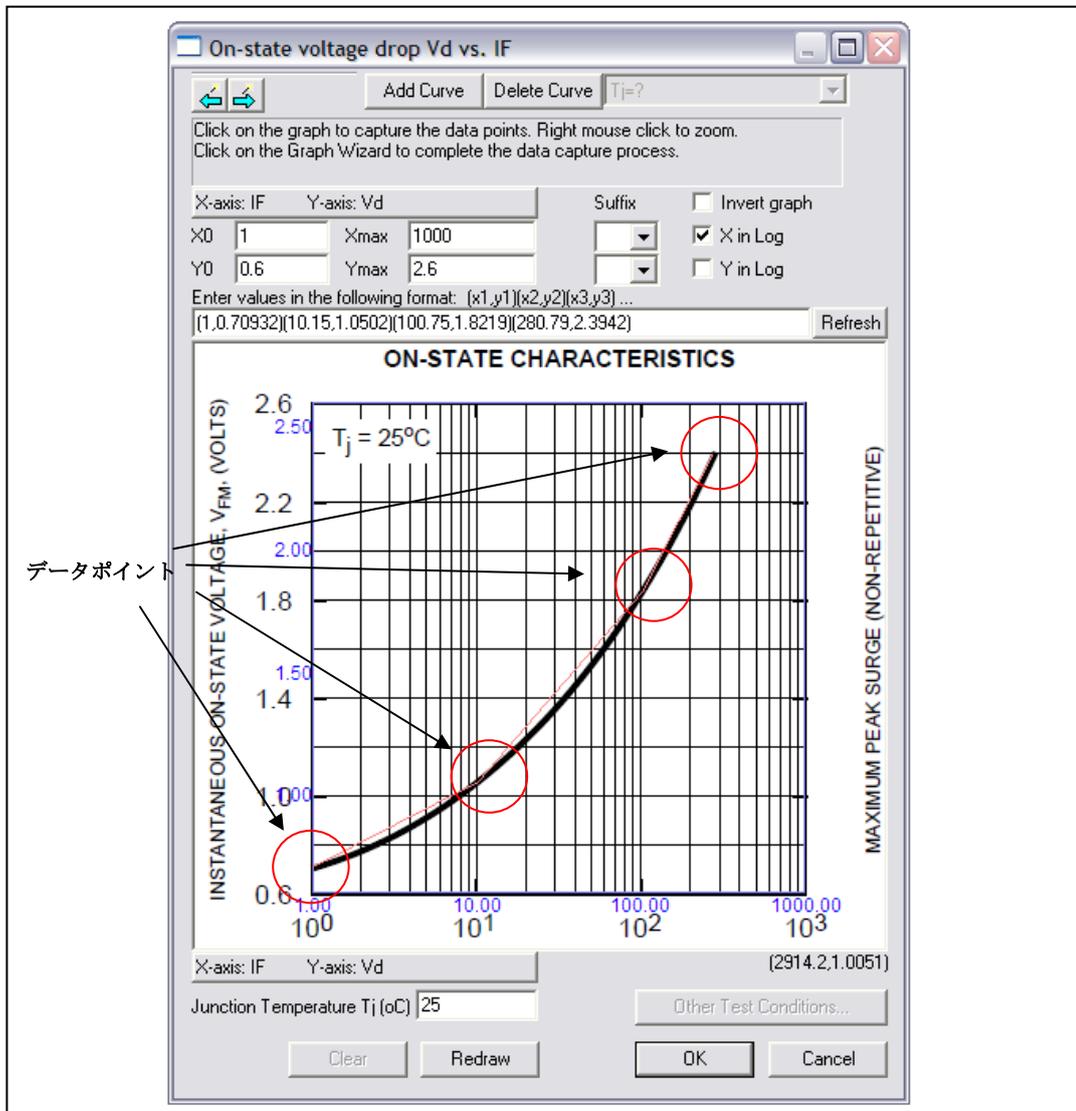
次のステップへ移動するためにフォワードウィザードアイコンをクリックしてください。

—このステップでは、x軸とy軸の設定を定義します。以下の通りに設定を入力してください。

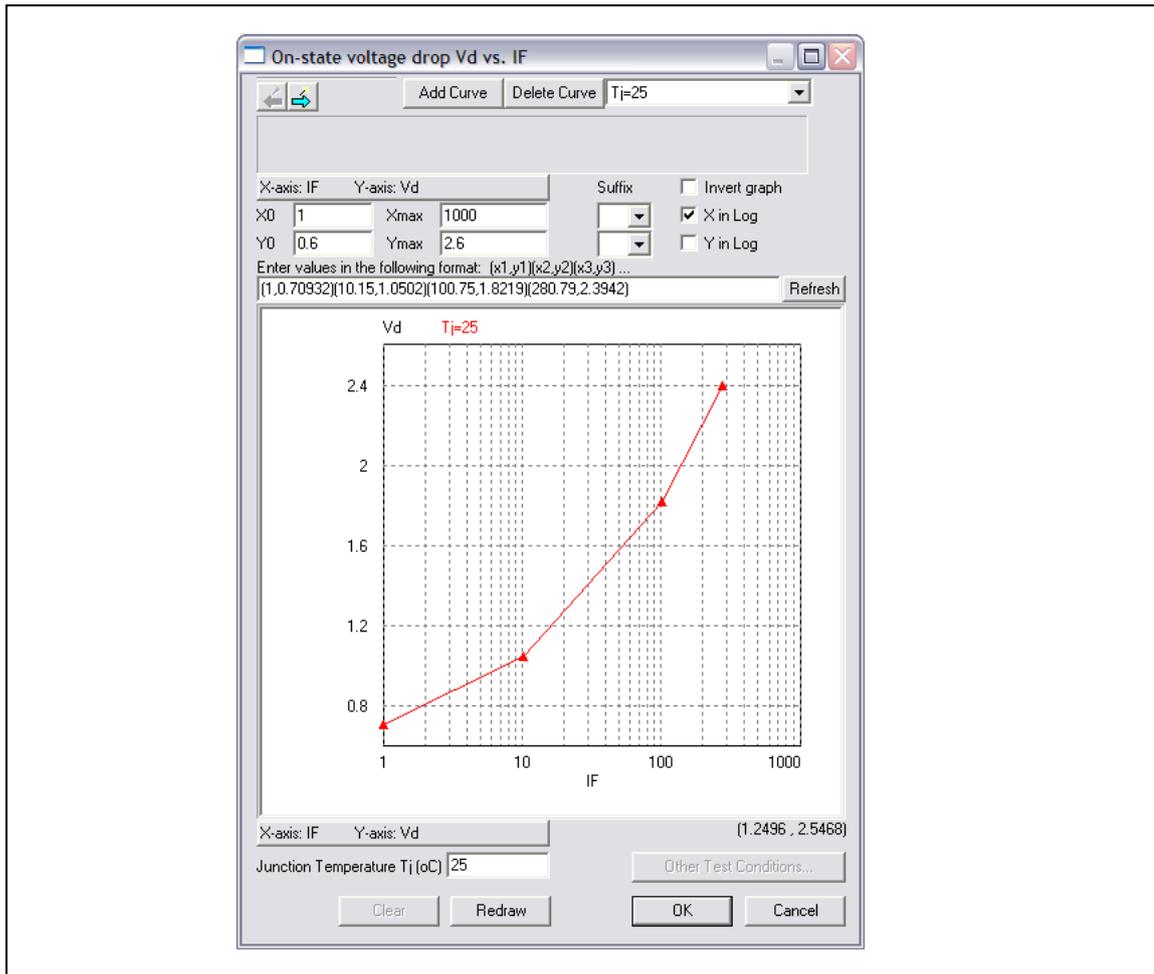
X0: 1
 Xmax: 1000
 Y0: 0.6
 Ymax: 2.6
 X in log: checked

次のステップに移るためにフォワードウィザードアイコンをクリックしてください。

—波形の上でマウスの左ボタンをクリックして、データ・ポイントを得てください。この場合、1A、10A、100A、および300A周辺で4つデータ・ポイントを得ています。もう一度、画像を拡大するためにマウスの右クリックをしてください。



データ・ポイントを収集しているときにデータ・ポイントを接続する赤い線が現れます。データ・ポイント収集プロセスを完了するために、フォワードウィザードアイコン  をクリックしてください。最終的なグラフダイアログウィンドウは、以下の通りに表示されます。



グラフ上のデータ・ポイントのx軸とy軸の値を見るためには、グラフエリアの内部にカーソルを置いてください。カーソル・図は反対に変わります。カーソルのx軸とy軸はダイアログウィンドウの右上に表示されます。x軸とy軸の値を読み込むために、カーソルを波形の最上部に置いてください。

—同じプロセスで、逆回復特性 t_{rr} v.s. I_F , I_{rr} v.s. I_F , 及び Q_{rr} v.s. I_F を定義してください。

—**熱的特性**を入力してください。

$R_{th(j-c)}$: 0.6

$R_{th(c-s)}$: 0.4

—**寸法と質量**を入力してください。

Length (mm): 53

Width (mm): 36

Height (mm): 29

デバイス情報を保存するために**Device -> Save Device**を選んでください。これで、データベースへダイオードを追加するプロセスの完了になります。

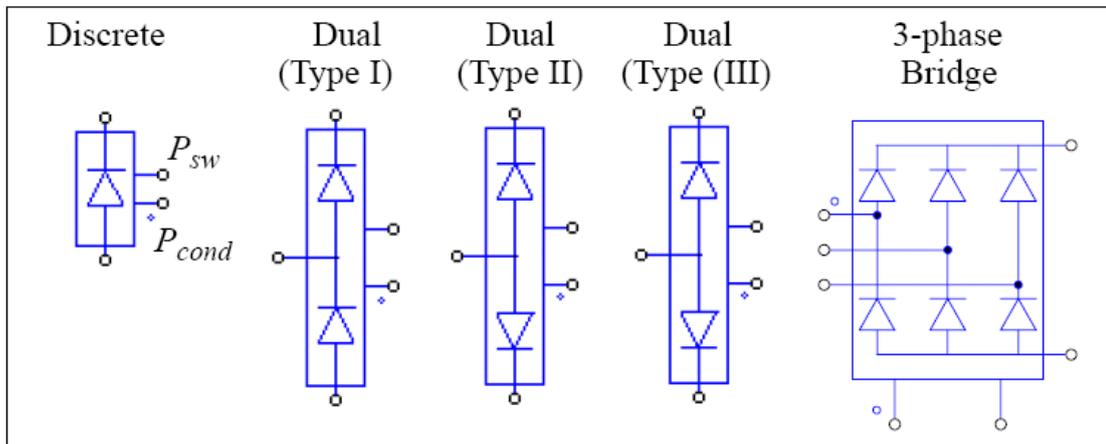
2.8.2 データベースのダイオードデバイス

データベースには以下のダイオードデバイスの情報が定義されています。

一般情報:

Manufacturer: デバイスメーカー
 Part Number: メーカーの部品番号
 Package: ディスクリットパッケージ、デュアルパッケージ、3相ブリッジパケ

ージ



図では、ダイオードのアノードとカソード端子以外に、2つの余分な端子があります。ドットがある端子はダイオード導通損失 P_{cond} のためであり、ドットのない端子はダイオードスイッチング損失 P_{sw} のためのもので

絶対最大定格:

$V_{rrm,max}$ (V): ピーク逆方向阻止電圧
 $I_{F,max}$ (A): 最大直流電流
 $T_{j,max}$ (°C): 最大接合部温度

電気的特性:

V_d v.s. I_F : 順方向導通電圧降下 V_d v.s. 順方向電流 I_F
 t_{rr} v.s. I_F : 逆方向回復時間 t_{rr} v.s. 電流 I_F
 I_{rr} v.s. I_F : ピーク逆方向回復電流 I_{rr} v.s. 電流 I_F
 Q_{rr} v.s. I_F : 逆方向回復電荷量 Q_{rr} v.s. 電流 I_F
 E_{rr} v.s. I_F : 逆方向回復電力損失 E_{rr} v.s. 電流 I_F

熱的特性:

$R_{th(j-c)}$: 接合・ケース間熱抵抗、°C/W
 $R_{th(c-s)}$: ケース・シンク間熱抵抗、°C/W

寸法と質量:

Length (mm): デバイスの長さ、mm

Width (mm): デバイスの幅さ、mm

Height (mm): デバイスの高さ、mm

Weight (g): デバイスの質量、g

Err対v.s. IF特性の「Reverse blocking voltage VR (V)」定数が損失の計算をはかるのに利用されますので、正しい値を入れてください。

絶対最大定格、熱的特性、寸法と質量の情報は損失計算には使用されません。

損失 P_{cond} 、 P_{sw} の単位はワットで、これらの端子から流れる電流の形で表されます。したがって、損失を測定して表示するためには電流計を P_{cond} または P_{sw} ノードと GND の間に接続する必要があります。使用しないときは、これらの端子はフローティングにせず、GND に接続してください。

2.8.3 ダイオード損失計算

損失計算はシミュレーションでデータベースのダイオードデバイスを選択して使用することができます。熱モジュールのライブラリのダイオードデバイスには次のパラメータがあります。

仕様

パラメータ	機能
Device	デバイスデータベースから選択された特定デバイス
Frequency	損失が計算される周波数(Hz)
Pcond Calibration Factor	トランジスタ伝導損失 P_{cond} の補正係数 K_{cond}
Psw Calibration Factor	トランジスタ・スイッチング損失 P_{sw} の補正係数 K_{sw}

パラメータ *Frequency* は損失が計算される周波数です。例えば、デバイスが 10kHz のスイッチング周波数で作動して、パラメータ *Frequency* が 10kHz に設定されると、損失はスイッチング周期の値になります。しかしながら、パラメータ *Frequency* が 60Hz に設定されると、損失は 60Hz 周期の値になります。

パラメータ *Pcond Calibration Factor* はトランジスタ伝導損失のための修正率です。例に関して、修正の前に計算された導通損失は P_{cond_cal} より、

$$P_{cond} = K_{cond} * P_{cond_cal}$$

同様に、パラメータ *Psw Calibration Factor* はトランジスタ・スイッチング損失のための修正率です。例に関して、修正の前に計算されたスイッチング損失が P_{sw_cal} であるなら、

$$P_{sw} = K_{sw} * P_{sw_cal}$$

導通損失:

ダイオード導通損失は次のように計算されます。

$$\text{導通損失} = V_d * I_F$$

ここで、 V_d はダイオード電圧降下、及び I_F はダイオード順電流です。

スイッチング損失:

スイッチング損失を計算する際に、ダイオードのターンオン損失は無視され、考慮されません。逆回復によるダイオードターンオフ損失は次のように計算されます。

$$P_{sw_off} = E_{rr} * f$$

または

$$P_{sw_off} = 1/4 * Q_{rr} * V_R * f$$

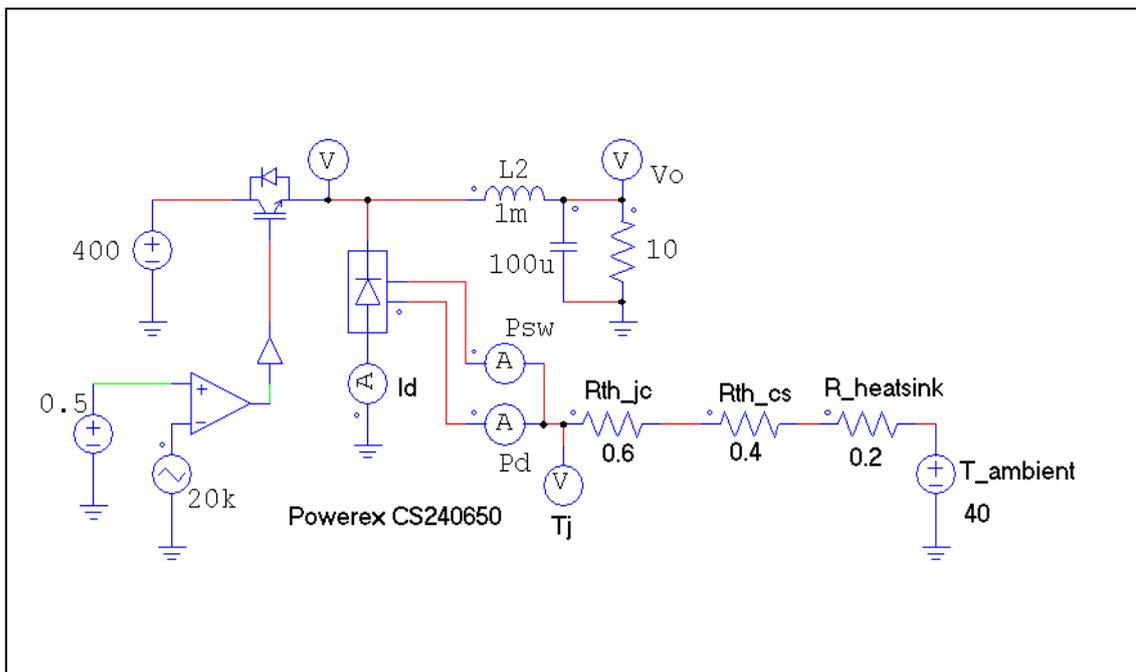
ここで、 E_{rr} は逆方向回復エネルギー損失、 Q_{rr} は逆方向回復電荷量、 V_R は逆阻止電圧、そして f は入力パルメータ *Frequency*で定義された周波数です。逆方向回復電荷量 Q_{rr} は以下のように定義されます。

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

E_{rr} がデバイスデータベースに与えられているならば、それを基に損失の計算をします。もし E_{rr} が与えられていないが、 Q_{rr} がデバイスデータベースで与えられる場合、損失は Q_{rr} に基づいて計算されます。しかし Q_{rr} も与えられないとき、損失は t_{rr} と I_{rr} に基づいて計算されます。それら両方とも与えられなければ、損失は0とみなされます。

例: ダイオード損失計算

下記の回路は、Powerex のディスクリット・ダイオード CS240650(600V、50A)を使用するサンプル回路を示します。導通損失及びスイッチング損失は2つの電流計によって測定されます。損失の情報を知ることができれば、熱等価回路の構築によって、デバイス接合部温度を計算することができます。熱の過渡現象を考慮しない熱回路を示します。



2.8.4 データベースの IGBT デバイス

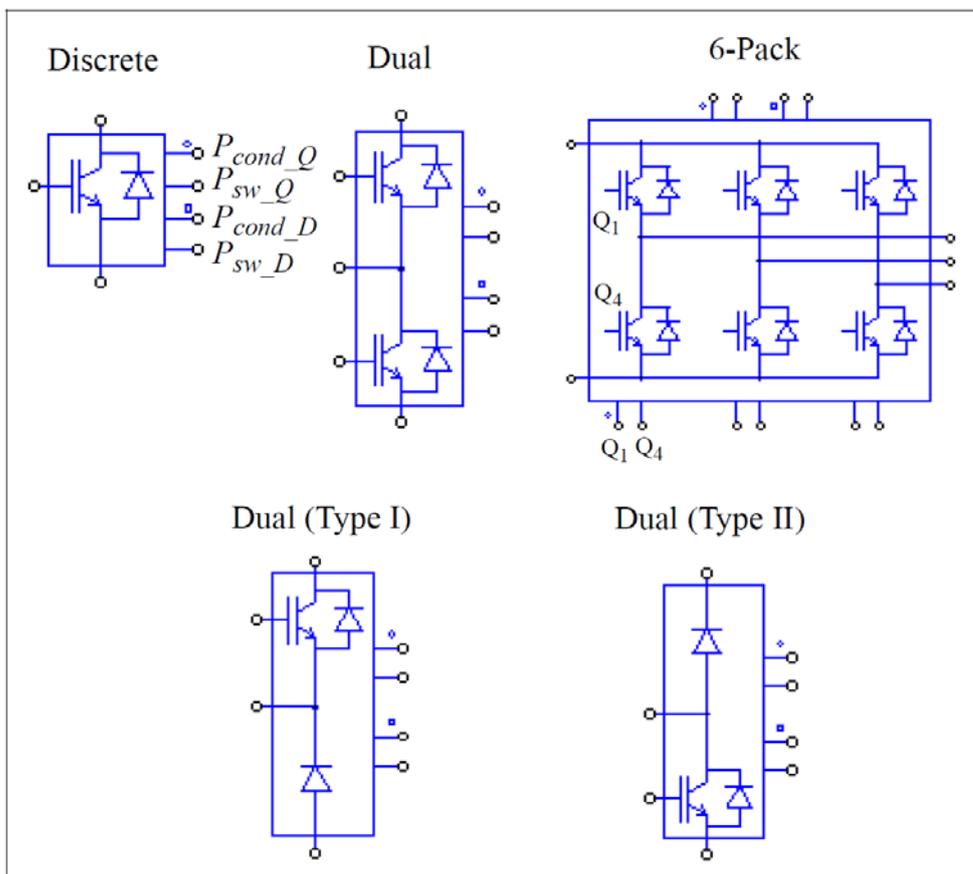
デュアルパッケージでは、ハイサイド側およびローサイド側のスイッチの両方が IGBT(フルブリッジ)である場合、スイッチのうちの1つは IGBT であり、他はフリーホイーリングダイオード(ハーフブリッジ)である場合があります。ハーフブリッジのデュアル IGBT 装置では、フリーホイーリングダイオードのパラメータが逆並列ダイオードのパラメータと異なる場合がありますので、この種のデバイスは IGBT ダイオードデバイスと呼ばれ、シミュレーションで異なったタイプとして扱われます。しかし、ここでは説明をわかりやすくするため、両方のデバイスを IGBT デバイスとして扱います。

IGBTデバイスには、ディスクリート、デュアルパッケージ、または6-パックの3タイプのパッケージがあります。

データベースに IGBT デバイスの以下の情報が定義されています。

一般情報:

Manufacturer:	デバイスメーカー
Part Number:	メーカーの部品番号
Package:	ディスクリートパッケージ、デュアルパッケージ、6-パックパッケージ



図では、IGBTとダイオード端子以外に、上部から下部まで4つの余分な端子があります(6パックのパッケージの場合、左側から右側)。それらは、トランジスタ導通損失 $P_{\text{cond_Q}}$ (円のドットがある)の端子、トランジスタ・スイッチング損失 $P_{\text{sw_Q}}$ のための端子、ダイオード伝導損失 P_{cond} (正方形のドットがある)の端子、ダイオードスイッチング損失 P_{sw} の端子です。

絶対最大定格:

$V_{\text{ce,max}}$ (V): 最大コレクタ・エミッタ電圧

$I_{\text{c,max}}$ (A): 最大コレクタ電流

$T_{\text{j,max}}$ (°C): 最大接合温度

電気的特性 - トランジスタ:

$V_{\text{ce(sat)}}$ v.s. I_{c} : コレクタ・エミッタ飽和電圧 $V_{\text{ce(sat)}}$ v.s. コレクタ電流 I_{c}

E_{on} v.s. I_{c} : ターンオンエネルギー損失 E_{on} v.s. コレクタ電流 I_{c}

E_{off} v.s. I_{c} : ターンオフエネルギー損失 E_{off} v.s. コレクタ電流 I_{c}

電気的特性 - ダイオード(また逆並列ダイオード):

V_{d} v.s. I_{F} : 順伝導電圧降下 v.s. 順電流 I_{F}

t_{rr} v.s. I_{F} : 逆方向回復時間 t_{rr} v.s. 電流 I_{F}

I_{rr} v.s. I_{F} : ピーク逆方向回復電流 I_{rr} v.s. 電流 I_{F}

Q_{rr} v.s. I_{F} : 逆方向回復電荷 Q_{rr} v.s. 電流 I_{F}

電気的特性 - フリーホイールダイオード (for IGBT-Diode device only):

V_{d} v.s. I_{F} : 順伝導電圧降下 v.s. 順電流 I_{F}

t_{rr} v.s. I_{F} : 逆方向回復時間 t_{rr} v.s. 電流 I_{F}

I_{rr} v.s. I_{F} : ピーク逆方向回復電流 I_{rr} v.s. 電流 I_{F}

Q_{rr} v.s. I_{F} : 逆方向回復電荷 Q_{rr} v.s. 電流 I_{F}

E_{rr} v.s. I_{F} : 逆方向回復電力損失 E_{rr} v.s. 電流 I_{F}

熱的特性:

$R_{\text{th(j-c)}}$ (transistor): トランジスタ接合 - ケース間熱抵抗、°C/W

$R_{\text{th(j-c)}}$ (diode): ダイオード接合 - ケース間熱抵抗、°C/W

$R_{\text{th(c-s)}}$: ケース - シンク間熱抵抗、°C/W

寸法と質量:

Length (mm): デバイスの長さ、mm

Width (mm): デバイスの幅さ、mm

Height (mm): デバイスの高さ、mm

Weight (g): デバイスの質量、g

損失 P_{cond_Q} 、 P_{sw_Q} 、 P_{cond} 、と P_{sw_D} の単位はワットです。これらの端子から流れる電流の形で表されます。したがって、損失を測定して表示するためには、電流計を端子と GND 間に接続しなければなりません。これらの端子を使用しない場合、これらの端子はフローティングにせず GND に接続して下さい。

2.8.5 IGBT 損失計算

損失計算のシミュレーションにおいてデータベースの IGBT デバイスを選択して使用することができます。熱モジュールのライブラリの IGBT デバイスには次のパラメータがあります。

仕様

パラメータ	機能
Device	デバイスデータベースから選択された特定デバイス
Frequency	損失が計算される周波数(Hz)
Pcond_Q Calibration Factor	トランジスタ伝導損失 P_{cond_Q} の補正係数 K_{cond_Q}
Psw_Q Calibration Factor	トランジスタ・スイッチング損失 P_{sw_Q} の補正係数 K_{sw_Q}
Pcond_D CalibrationFactor	ダイオード伝導損失 P_{cond_D} の補正係数 K_{cond_D}
Psw_D CalibrationFactor	ダイオードスイッチング損失 P_{sw_D} の補正係数 K_{sw_D} T

パラメータ *Frequency* は損失が計算される周波数です。例えば、デバイスが 10kHz のスイッチング周波数で作動して、パラメータ *Frequency* が 10kHz に設定されると、損失はスイッチング周期の値になります。しかしながら、パラメータ *Frequency* を 60Hz に設定すると、損失は 60Hz 周期の値になります。

パラメータ *Pcond_Q Calibration Factor* はトランジスタ導通損失のための修正率です。例に関して、修正の前に計算された伝導損失は $P_{cond_Q_cal}$ より、

$$P_{cond_Q} = K_{cond_Q} * P_{cond_Q_cal}$$

同様に、パラメータ *Psw_Q Calibration Factor* はトランジスタ・スイッチング損失のための修正率です。例に関して、修正の前に計算されたスイッチング損失は $P_{sw_Q_cal}$ より、

$$P_{sw_Q} = K_{sw_Q} * P_{sw_Q_cal}$$

パラメータ *Pcond_D Calibration Factor* と *Psw_D Calibration Factor* はトランジスタ損失と同様に働きますが、これらはダイオードの損失に関係します。

導通損失：

トランジスタ伝導損失は次のように計算されます。

$$\text{トランジスタ伝導損失} = V_{ce(sat)} * I_c$$

ここで、 $V_{ce(sat)}$ はトランジスタのコレクタ・エミッタ間飽和電圧、及び I_c はコレクタ電流です。

スイッチング損失：

トランジスタターンオン損失は次のように計算されます。

$$\text{トランジスタターンオン損失} = E_{on} * f * V_{cc} / V_{cc_datasheet}$$

ここで、 E_{on} はトランジスタターンオンエネルギー損失、 f は入力パラメータ *Frequency* で定義された周波数、 V_{cc} は実際の DC バス電圧です。 $V_{cc_datasheet}$ は試験条件に「DC バス電圧(V)」として定義されたデータシートの E_{on} 対 E_{off} の特性の DC バス電圧です。

トランジスタターンオフ損失は次のように計算されます。

$$\text{トランジスタターンオフ損失} = E_{off} * f * V_{cc} / V_{cc_datasheet}$$

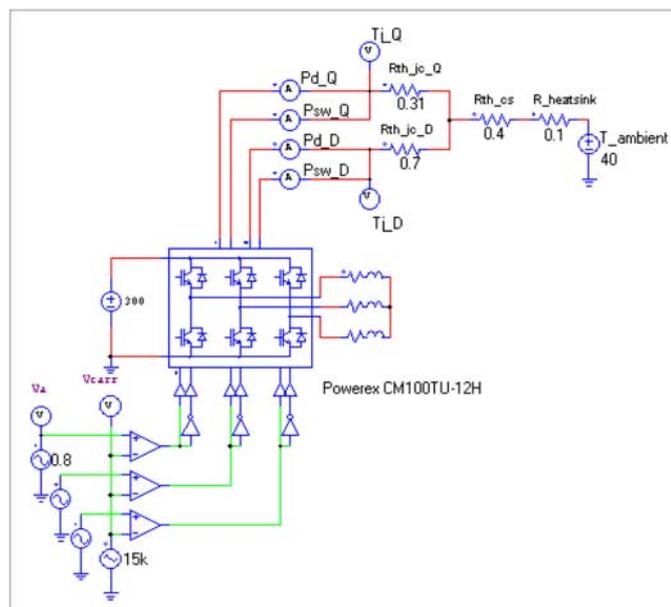
ここで、 E_{off} はトランジスタターンオフエネルギー損失です。

逆並列ダイオードまたはフリーホイールダイオードの損失の計算方法は、セクション 2.7.3 に述べたものと同じ方法です。

計算した損失、特にスイッチング損失の結果が近似値で表現されます。結果の正確さはデバイスデータベースのデータの正確さと実際の回路の動作条件に依存します。ハードウェアによる損失測定で、設計値と実際の損失結果を確認する必要があります。

例: IGBT 損失計算

下記の回路は、Powerexの6パックIGBTモジュールCM100TU-12H(600V、100A)を使用するサンプル回路を示します。トランジスタとダイオードの導通損失及びスイッチング損失は、別々に加えられます。また、熱等価回路は温度上昇を計算するためのものです。熱モジュールを利用すると、様々な条件でデバイスの熱伝導能力を迅速に確認することができるので、異なる複数種類のデバイスを比較することができます。

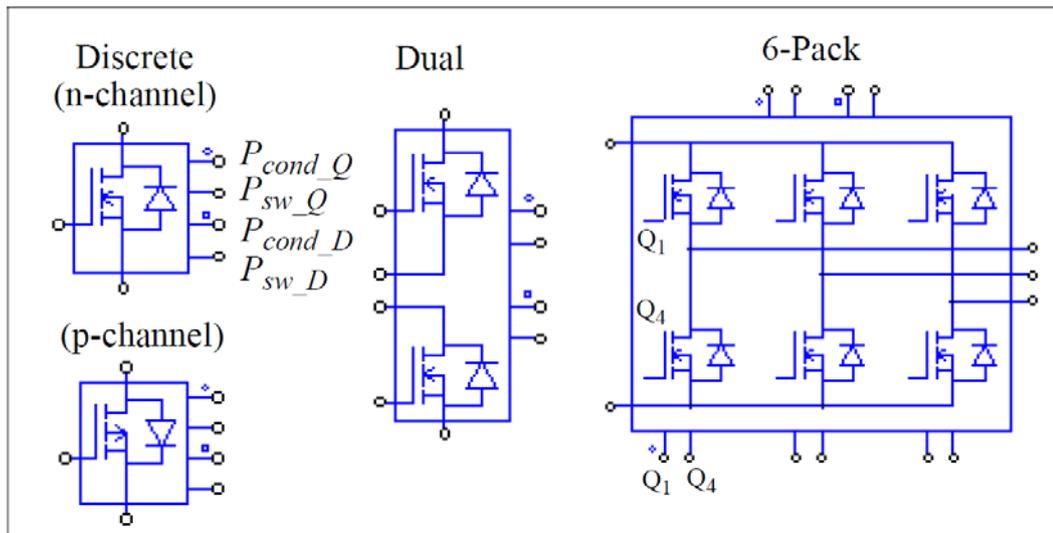


2.8.6 データベースの MOSFET デバイス

データベースには MOSFET デバイスの以下の情報が定義されています。

一般情報:

Manufacturer:	デバイスメーカー
Part Number:	メーカーの部品番号
Package:	ディスクリットパッケージ、デュアルパッケージ、6-パックパッケージ



図では、MOSFETとダイオード端子以外に、上部から下部まで4つノードがあります(または、6パックのパッケージの場合、左側から右側)。それらは、トランジスタ伝導損失 P_{cond_Q} の端子(円のドッドがある)、トランジスタ・スイッチング損失 P_{sw_Q} の端子、ダイオード伝導損失 P_{cond} の端子(正方形のドッドがある)、ダイオードスイッチング損失 P_{sw} の端子です。

絶対最大定格:

$V_{DS,max}$ (V):	最大ドレイン・ソース間電圧
$I_{D,max}$ (A):	最大ドレイン電流
$T_{j,max}$ (°C):	最大接合部温度

電気的特性 – トランジスタ:

RDS(on) (ohm):	試験時のドレイン・ソース間オン抵抗 $R_{DS(on)_b}$ (試験条件:ジャンクション温度 $T_{j,b}$ (°C、通常は25°C)、ゲート・ソース間電圧 V_{GS} 、(V); 及びドレイン電流 I_D 、(A))
Temperature Coefficient:	オン抵抗の温度係数 K_T (1/°C)
VGS(th) (V):	ゲート閾値電圧 $V_{GS(th)}$ (測定条件:ドレイン電流 I_D 、(A))
gfs (S):	順方向相互コンダクタンス g_{fs} (測定条件:ドレイン・ソース間電圧 V_{DS} 、(V); 及びドレイン電流 I_D 、(A))

t_r (ns) and t_f (ns): 上昇時間 t_r 及び下降時間 t_f 測定条件: ドレイン・ソース間電圧 V_{DS} 、V; ドレイン電流 I_D 、A; 及びゲート抵抗 R_g 、 Ω)

Qg, Qgs, and Qgd: Total gate charge Qg, ゲート・ソース間電荷量 Q_{gs} , とゲート・ドレイン間電荷量(「ミラー」) Q_{gd} , nC (測定条件: ドレイン・ソース間電圧 V_{DS} 、V; ゲート・ソース間電圧 V_{GS} 、V; 及びドレイン電流 I_D 、A)

Ciss, Coss, and Crss: 入力容量 C_{iss} , 出力容量 C_{oss} , と帰還容量 C_{rss} , pF (測定条件: ドレイン・ソース間電圧 V_{DS} 、V; ゲート・ソース間電圧 V_{GS} 、V; 及びテスト周波数、MHz)

電気的特性 –ダイオード :

Vd v.s. IF: 順伝導電圧降下 V_d v.s. 順電流 I_F

trr and Qrr: 逆回復時間 t_{rr} 、ns, 及び逆回復電荷量 Q_{rr} , μ C (測定条件: 順電流 I_F 、A; 電流の変化率 di/dt 、A/ μ s ; およびと接合部温度 T_j 、 $^{\circ}$ C)

熱的特性:

$R_{th(j-c)}$: 接合・ケース間熱抵抗、 $^{\circ}$ C/W

$R_{th(c-s)}$: ケース・シンク間熱抵抗、 $^{\circ}$ C/W

寸法と質量:

Length (mm): デバイスの長、mm

Width (mm): デバイスの幅さ、mm

Height (mm): デバイスの高さ、mm

Weight (g): デバイスの質量、g

損失 P_{cond_Q} 、 P_{sw_Q} 、 P_{cond} 、と P_{sw_D} の単位はワットです。これらの端子から流れる電流の形に表されます。したがって、損失を測定して表示するために、電流計を端子と GND 間に接続しなければなりません。これら端子が使用されていないとき、これらの端子はフローティングにせず GND に接続してください。

2.8.7 MOSFET 損失計算

損失計算にシミュレーションでデータベースの IGBT 装置を選択して、使用することができます。熱モジュールのライブラリの MOSFET には次のパラメータがあります。

仕様

パラメータ	機能
Device	デバイスデータベースから選択された特定デバイス
Frequency	損失が計算される周波数(Hz)
VGG+ (upper level)	ゲート・ソース間電圧の上位レベル、V
VGG- (lower level)	ゲート・ソース間電圧の下位レベル、V
Rg_on (turn-on)	ターンオンの時ゲート抵抗
Rg_off (turn-off)	ターンオフの時ゲート抵抗。ほとんどの場合、ターンオンゲート抵抗

	R_{g_on} とターンオフゲート抵抗 R_{g_off} は同一です。しかし、いくつかの場合、これらの2つの抵抗は異なることがあります。
RDS(on) CalibrationFactor	オン抵抗 $R_{DS(on)}$ の較正係数
gfs Calibration Factor Pcond_Q CalibrationFactor	順方向相互コンダクタンス g_{fs} の較正係数
Psw_Q CalibrationFactor	トランジスタ伝導損失 P_{cond_Q} の較正係数 K_{cond_Q}
Pcond_D CalibrationFactor	トランジスタ・スイッチング損失 P_{sw_Q} の較正係数 K_{sw_Q}
Psw_D CalibrationFactor	ダイオード伝導損失 P_{cond_D} の較正係数 K_{cond_D}
	ダイオードスイッチング損失 P_{sw_D} の較正係数 K_{sw_D}

パラメータ *Frequency* は損失が計算される周波数です。例えば、装置が 10kHz のスイッチング周波数で作動して、パラメータ *Frequency* が 10kHz に設定されると、損失はスイッチング周期の値になります。しかしながら、パラメータ *Frequency* を 60Hz に設定すると、損失は 60Hz 周期の値になります。

パラメータ *Pcond_Q Calibration Factor* はトランジスタ導通損失のための修正率です。例に関して、修正の前に計算された導通損失は $P_{cond_Q_cal}$ より、

$$P_{cond_Q} = K_{cond_Q} * P_{cond_Q_cal}$$

同様に、パラメータ *Psw_Q Calibration Factor* はトランジスタ・スイッチング損失のための修正率です。例に関して、修正の前に計算されたスイッチング損失は $P_{sw_Q_cal}$ より、

$$P_{sw_Q} = K_{sw_Q} * P_{sw_Q_cal}$$

パラメータ *Pcond_D Calibration Factor* と *Psw_D Calibration Factor* はトランジスタの損失と同様に働きますが、これらはダイオードの損失に関係します。

導通損失:

トランジスタ伝導損失は次のように計算されます。

$$\text{トランジスタ伝導損失} = I_D^2 * R_{DS(on)}$$

ここで、 I_D はドレイン電流、及び $R_{DS(on)}$ はオン抵抗です。

スイッチング損失:

トランジスタターンオン損失は次のように計算されます。

$$\text{トランジスタターンオン損失} = E_{on} * f$$

ここで、 E_{on} はトランジスタターンオンエネルギー損失、及び f は入力パラメータ *Frequency* で定義された周波数です。

トランジスタターンオフ損失は次のように計算されます。

$$\text{トランジスタターンオフ損失} = E_{off} * f$$

ここで、 E_{off} はトランジスタターンオフエネルギー損失です。エネルギー損失 E_{on} と E_{off} は MOSFET 装置のゲート電流、入力/出力/帰還容量、およびゲート電荷量の情報に基づいて計算されます。

逆並列ダイオードまたはフリーホイールダイオードの損失の計算方法はセクション 2.7.3 に述べた方法と同じです。

計算された損失、特にスイッチング損失の結果は近似値で求められます。結果の正確さは装置データベースのデータの正確さと実際の回路の動作条件に依存します。ハードウェアによる測定結果で、設計値と実際の損失を確認することが必要です。

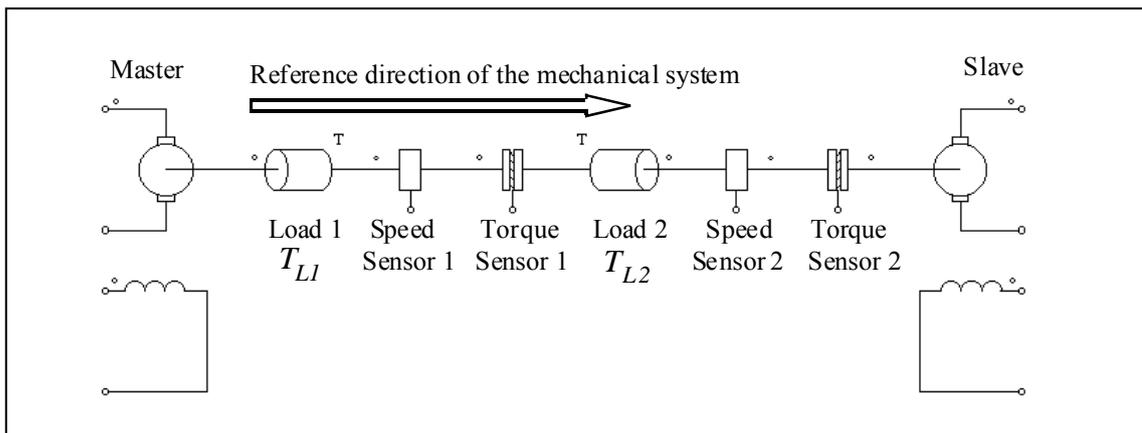
2.9 モータ駆動モジュール

モータ駆動モジュール (Motor Drive Module) は PSIM のアドオン・オプションで、モータモデルと機械負荷のモデルを提供します。

2.9.1 機械系の基準方向

モータはマスタかスレーブのいずれかのモードに設定します。機械系に 1 台だけのモータがある場合、このモータはマスタモードに設定しなければなりません。2 台以上のモータが系統にある場合、1 台だけをマスタモードに、その他はスレーブモードに設定してください。

マスタモードのモータはマスタモータと呼ばれ、機械系の基準方向を定義します。基準方向は、以下に示すようにマスタモータのシャフトノードから、シャフトに沿って、機械系の他のモータへ向かう方向として定義されます。



この機械系では、左側のモータがマスタで右側がスレーブです。それゆえ、機械系の基準方向は機械シャフトに沿って左側から右側に定義されます。さらに、基準方向が素子のドットで表した側に入ると、その素子は基準方向に沿っていると言えます。そうでなければ、基準方向に対して逆ということになります。例えば、負荷 1、スピードセンサ 1、トルクセンサ 1 は基準方向に沿っています。そして負荷 2、スピードセンサ 2、トルクセンサ 2 は基準方向に対して逆となっています。

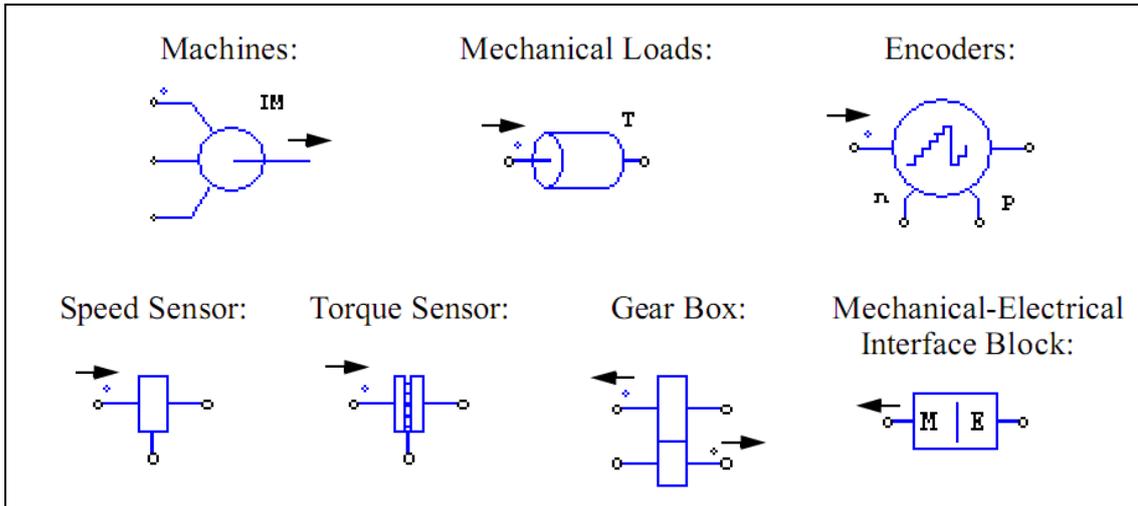
さらにマスタモータの電機子と界磁電流の両方が正のとき、機械速度は正と仮定します。

この記述に基づけば、スピードセンサが機械系の基準方向に沿っていれば、マスタモータにより発生した正の速度は、正のスピードセンサ出力を与えます。そうでなければ、スピードセンサの出力は負になります。例えば、上の例のマスタモータのスピードが正の場合、スピードセンサ 1 の読値は正で、スピードセンサ 2 の読値は負です。

基準方向は機械負荷がモータとどのように反応するかも決定します。この系統では、2 つの定トルク機械負荷、大きさがそれぞれ T_{L1} 、 T_{L2} があります。負荷 1 は基準方向に沿っており、負荷 2 は基準方向と

逆です。そのため、マスタモータに対する負荷 1 の負荷トルクは T_{L1} で、マスタモータに対する負荷 2 の負荷トルクは $-T_{L2}$ です。

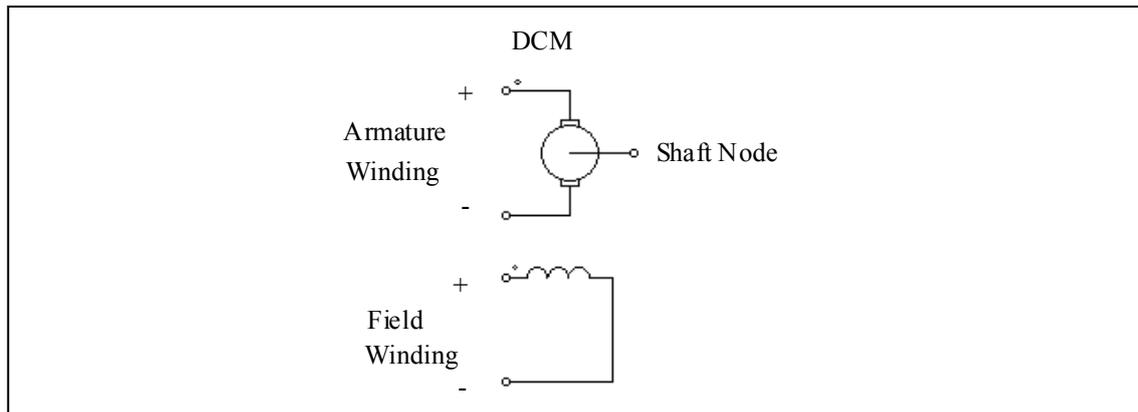
その他の機械系エレメントの基準方向は以下ようになります。



2.9.2 直流機

直流機のシンボルとパラメータは以下ようになります。

シンボル



仕様

パラメータ	機能
R_a (armature)	電機子巻線抵抗 (Ω)
L_a (armature)	電機子巻線インダクタンス (H)
R_f (field)	界磁巻線抵抗 (Ω)
L_f (field)	界磁巻線インダクタンス (H)
Moment of Inertia	モータの慣性モーメント ($kg \cdot m^2$)

$V_t(\text{rated})$	定格電機子電圧 (V)
$I_a(\text{rated})$	定格電機子電流 (A)
$n(\text{rated})$	定格回転数 (rpm)
$I_f(\text{rated})$	定格界磁電流 (A)
Torque Flag	内部トルク T_{em} の出力フラグ
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ, 0: スレーブ)

トルクフラグを 1 に設定すると、モータによって発生される内部トルクが表示用のデータファイルに保存されます。

直流機の動特性は以下の方程式で記述されています。

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

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

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

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

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

v_t , v_f , i_a , i_f はそれぞれ電機子電圧、界磁電圧、電機子電流、界磁電流です。 E_a は内部誘起電圧、 ω_m は機械系の回転速度 (rad/sec) です。 T_{em} は内部発生トルクで T_L は負荷トルクです。内部誘起電圧と内部トルクは以下のようにも記述できます。

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

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

ここで L_{af} は電機子と界磁巻線間の相互インダクタンスです。これは定格運転条件より以下のように算出されます。

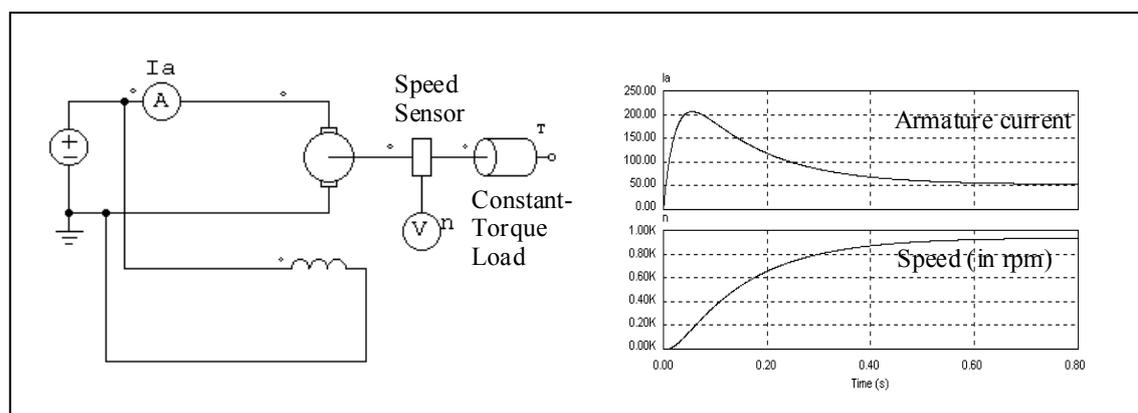
$$L_{af} = \frac{(V_t - I_a \cdot R_a)}{I_f \cdot \omega_m}$$

直流機モデルは線形磁化特性を仮定しています。磁気飽和は考慮されていません。

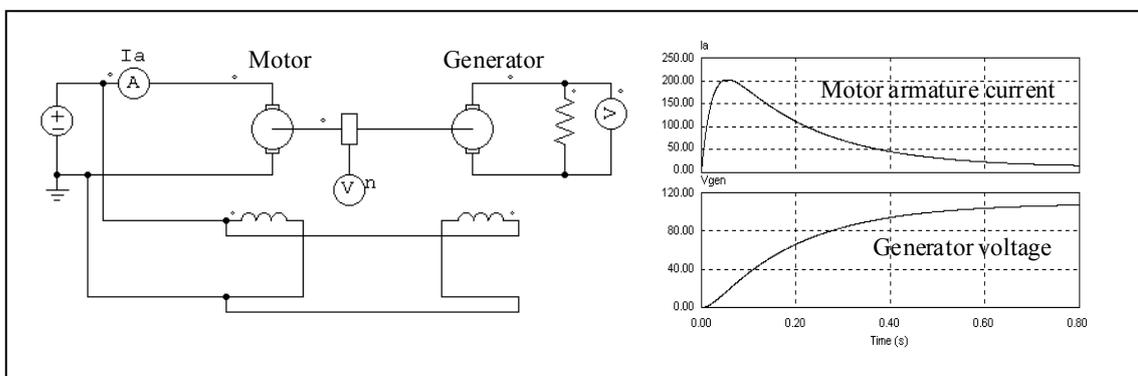
例：定トルク負荷と直流機

以下に定トルク負荷 T_L とシャント励磁直流機の回路を示します。負荷は機械システムの基準方向に沿っているため、負荷トルクは T_L です。またスピードセンサも基準方向に沿っています。これは正の速度に対して正の出力を与えます。

電機子電流と速度のシミュレーション波形は以下の右側に示されています。

**例：直流モータと発電機の組み合わせ**

以下に直流モータと発電機の組み合わせを示します。左側のモータはマスタモードに設定されており、右側の発電機はスレーブモードに設定されています。モータ電機子電流と発電機電圧のシミュレーション波形は始動過渡特性を表しています。

**2.9.3 誘導機**

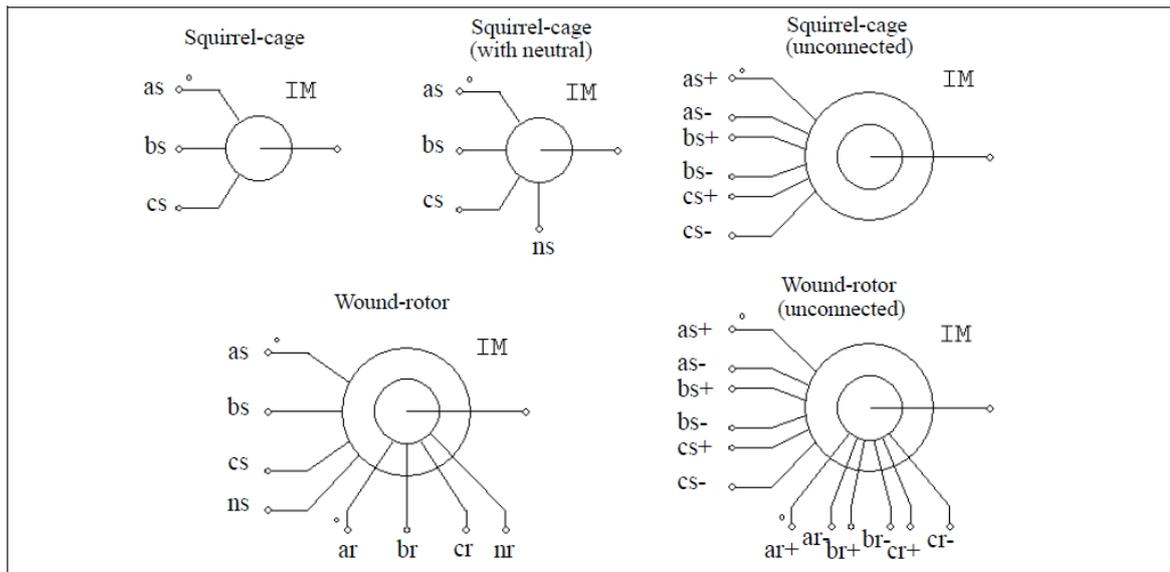
かご型誘導機と巻線型誘導機のそれぞれに線形および非線形の2つのモデルが備わっています。線形モデルはさらに汎用タイプと対称タイプに分かれます。この節では線形モデルのみ説明します。

線形モデルには以下の5つがあります。

- 対称3相かご型誘導機 (INDM_3S)

- 対称 3 相中性点付かご型誘導機 (INDM_3SN)
- 汎用 3 相かご型誘導機 (INDM3_S_LIN)
- 対称 3 相巻線型誘導機 (INDM3_WR)
- 汎用 3 相巻線型誘導機 (INDM3_WR_LIN)

シンボル



仕様

パラメータ	機能
R_s (固定子)	固定子巻線抵抗 (Ω)
L_s (固定子)	固定子巻線漏れインダクタンス (H)
R_r (回転子)	回転子巻線抵抗 (Ω)
L_r (回転子)	回転子巻線漏れインダクタンス (H)
L_m (磁化)	磁化インダクタンス (H)
N_s/N_r Turns Ratio	固定子と回転子の巻線比 (巻線型のみ)
No. of Poles	モータの極数 (偶数)
Moment of Inertia	モータの慣性モーメント J ($kg \cdot m^2$)
Torque Flag	内部トルク T_{em} の出力フラグ。フラグが 1 に設定されると、内部トルクの出力が要求される。
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ; 0: スレーブ)

すべてのパラメータは固定子側に換算した値を使用します。

Master/Slave Flagは機械系の基準方向を定義するために使用します。詳細は 2.9.1を参照してください。

また、モータは相順通りに励磁したときに正の速度で回転します。

INDM_3SN は INDM_3S に固定子中性点端子を追加したものです。

PSIM の三相誘導機は以下の式で表されています。

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

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

ここで、

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

かご型機では、 $v_{a,r} = v_{b,r} = v_{c,r} = 0$ です。

ここで、パラメータ行列は以下のように定義されています。

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

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

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

ここで、 M_{sr} は固定子と回転子巻線間の相互インダクタンスで、 θ は機械角です。相互インダクタンスは磁化インダクタンスと以下の関係があります。

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

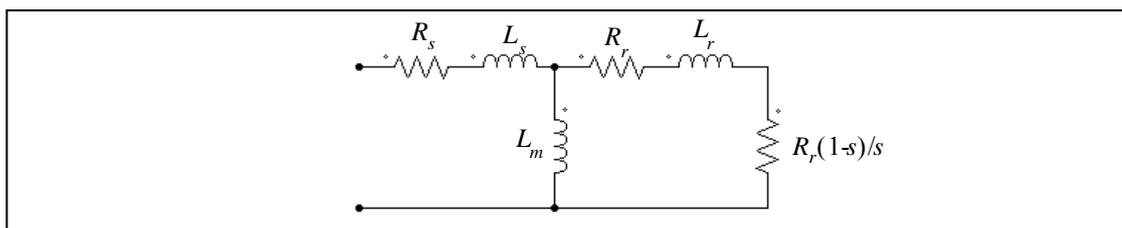
機械系方程式は以下のように表されます。

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

ここで電気トルク T_{em} は以下のように定義されます。

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

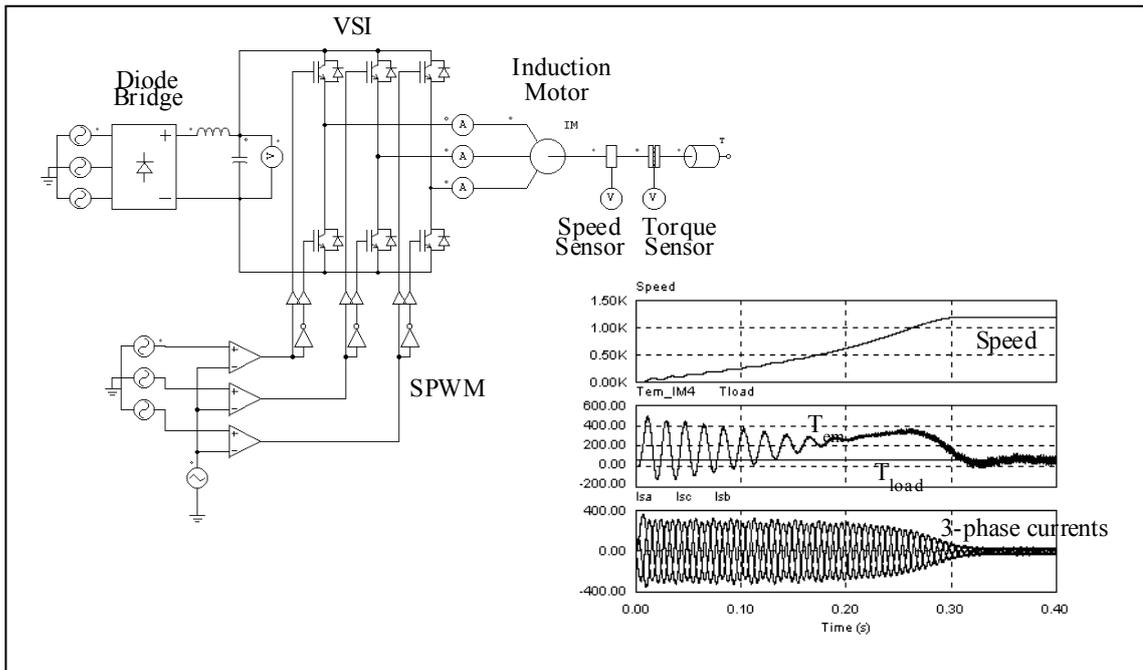
対称かご型誘導機では、モータの定常状態での等価回路は以下のように示されます。s はすべりです。



例 : VSI 誘導電動機ドライブシステム

下の図にオープンループ誘導電動機ドライブシステムを示します。誘導電動機は 6 極で、正弦波 PWM により電圧源インバータから供給されています。直流母線はダイオードブリッジを通じて電源を供給されています。

シミュレーション波形は機械系回転速度 (rpm)、発生トルク T_{em} 、負荷トルク T_{load} 、3 相入力電流の始動時の過渡特性を表しています。

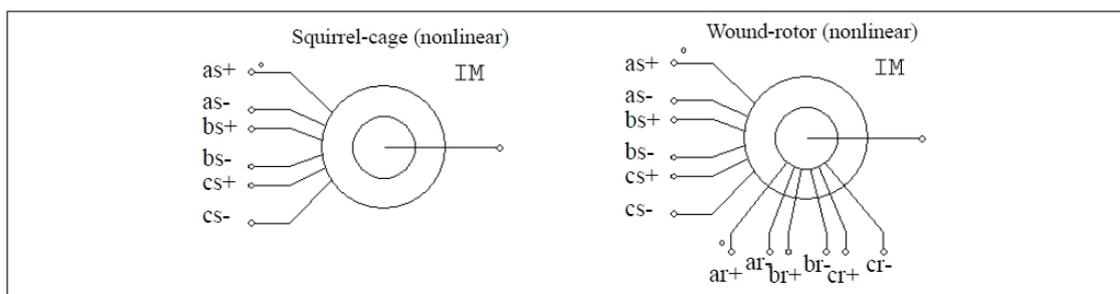


2.9.4 飽和付かご型誘導機

飽和特性付誘導機には以下の 2 種類があります。

- 非線形 3 相かご型誘導機 (Squirrel-cage Ind. Machine (nonlinear))
- 非線形 3 相巻線型誘導機 (Wound-rotor Ind. Machine (nonlinear))

シンボル



仕様

パラメータ	機能
R _S (固定子)	固定子巻線抵抗 (Ω)
L _S (固定子)	固定子巻線漏れインダクタンス (H)
R _r (回転子)	回転子巻線抵抗 (Ω)
L _r (回転子)	回転子巻線漏れインダクタンス (H)
L _m (磁化)	磁化インダクタンス (H)
N _S /N _r	固定子と回転子の巻線比 (巻線型のみ)
Number of Poles	モータの極数 (偶数)
Moment of Inertia	モータの慣性モーメント J (kg*m ²)
Torque Flag	内部トルク T _{em} の出力フラグ。フラグが 1 に設定されると、内部トルクの出力が要求される。
Master/Slave Flag	マスタ/スレーブモードのフラグ (1 : マスタ ; 0 : スレーブ)
I _m v.s. L _m (I _{m1} , L _{m1})...	励磁電流と励磁インダクタンスの関係を区分線形の点の組で与えます [(I _{m1} , L _{m1}), (I _{m2} , L _{m2}), ...]

全てのパラメータは固定子側に換算した値で表されています。

飽和付き 3 相誘導機の動特性は以下の方程式で記述されています。

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

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

ここで、

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

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

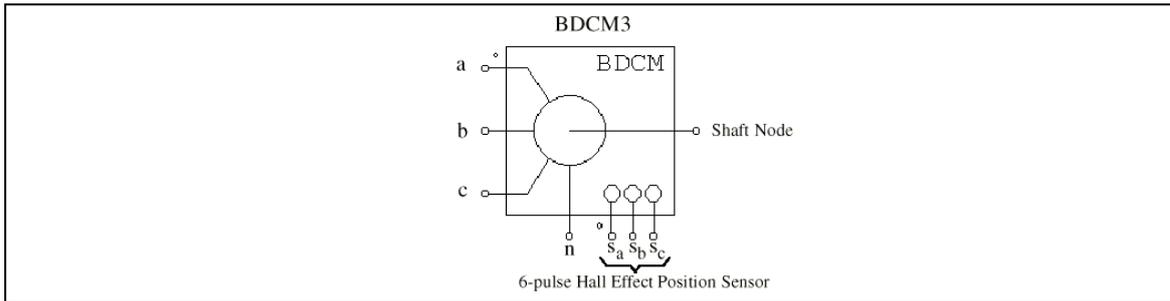
この場合、インダクタンス M_{sr} は定数ではなく、励磁電流 I_m の関数になります。

2.9.5 ブラシレス直流機

3 相ブラシレス直流機は永久磁石を使った同期機的一种と考えることができます。固定子には 3 相巻線を、固定子には永久磁石を持っています。PSIM のブラシレス直流機のモデルでは誘導起電力が台形状の波形となります。

3 相ブラシレス直流機のシンボルとパラメータを以下に示します。

シンボル



仕様

パラメータ	機能
R (Stator Resistance)	固定子の相抵抗 (Ω)
L (Stator self ind.)	固定子の自己インダクタンス (H)
M (Stator mutual ind.)	固定子の相互インダクタンス (H) 相互インダクタンス M は負の値です。巻線の構成により、相互インダクタンス M と自己インダクタンス L の比は通常 $-1/3$ から $-1/2$ の間になります。もし M が不明の場合は、 $M = -0.4 * L$ としたものが無難なデフォルト値として使えます。
Vpk / krpm	線間のピーク誘起電圧定数、単位は V/krpm (機械系の回転速度) です。
Vrms / krpm	線間の実効誘起電圧定数、単位は V/krpm (機械系の回転速度) です。 Vpk/krpm と Vrms/krpm の値はモータのデータシートから得られるはずですが、これらの値が不明の場合は、ブラシレス機を 1000rpm で発電機として運転し、線間電圧のピーク値と実効値を測ることでわかります。
No. of poles	極数 P
Moment of Inertia	モータの慣性モーメント J ($\text{kg} \cdot \text{m}^2$)
Shaft Time Constant	軸の時定数 (T_{shaft})
theta_0 (deg.)	回転角の初期値 θ_r (電気角・度) 回転角の初期値とは時刻 $t=0$ の時の回転角です。回転角 0° の位置は、回転数が正の値のとき、A 相の誘起電圧が (負から正へ) 零点を横切るときの位置として定義できます。
theta_advance (deg.)	位置センサの進み角 θ_{advance} (電気角・度) 進み角とは A 相の上側のスイッチが 120° 導通モードのとき、ターンオン角と 30° の角度の差として定義できます。たとえば、A 相が 25° でオンになった場合は、進み角は 5° です ($30-25=5$)。
Conduction Pulse Width	位置センサの導通パルス幅 (電気角・度) 導通パルスが正のとき、全波ブリッジ・インバータの下側スイッチがオンになります。 120° 導通モードのときは導通パルス幅が 120° となります。
Torque Flag	内部電気トルク T_{em} の出力フラグ (1: 出力あり; 0: なし)
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ, 0: スレーブ)。マスタ/スレーブのフラグの設定により、モータの動作モードが変わります。詳しくは 2.9.1 を参照ください。

シンボルの端子は以下のとおりです。a, b, c はそれぞれ a, b, c 相の固定子巻線端子；固定子巻線は Y 結線され、n は中性点端子です。軸端子は機械軸を接続するためのもの。これらはすべてパワー端子なので、パワー回路に接続しなければなりません。

S_a, S_b, S_c はそれぞれ a, b, c 相のホール効果 6 パルス位置センサの端子、センサ出力は 3 値の転流パルス (1, 0, -1) です。センサ出力の端子はすべて制御端子なので制御回路に接続して下さい。

3 相ブラシレス直流機の方程式は以下の通りです。

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

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

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

ここで、 v_a, v_b, v_c は相電圧、 i_a, i_b, i_c は相電流、R, L, M はそれぞれ固定子の各相抵抗、自己インダクタンス、相互インダクタンスです。また、 E_a, E_b, E_c はそれぞれ a, b, c 相の誘起電圧です。

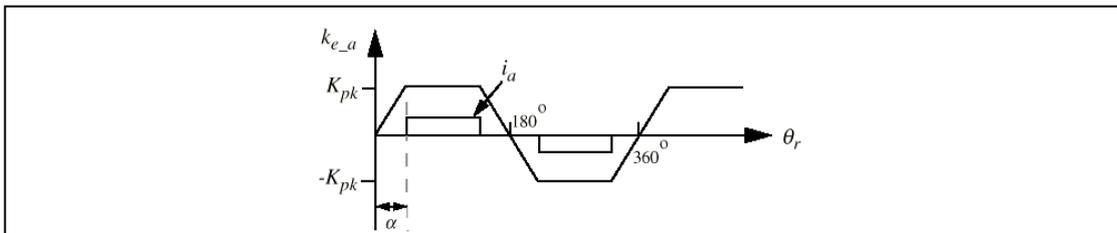
誘起電圧は回転子の機械回転速度 ω_m と回転角 θ_r の関数で次のようになります。

$$E_a = k_{e_a} \cdot \omega_m$$

$$E_b = k_{e_b} \cdot \omega_m$$

$$E_c = k_{e_c} \cdot \omega_m$$

係数 $k_{e_a}, k_{e_b}, k_{e_c}$ は回転角 θ_r によって決まります。このモデルでは下図に示す理想的な台形波形を想定しています。同じ図に A 相の電流を示します。



この図で、 K_{pk} は台形のピーク値 (V/(rad./sec.)) で、 $k_{pk} = \frac{v_{pk}/krpm}{2} \cdot \frac{1}{1000 \cdot 2\pi/60}$

により定義されます。Vpk/krpm と Vrms/krpm の値を与えると、角度 θ は PSIM により自動的に決まります。

モータの電気トルクは、

$$T_{em} = (E_a \cdot i_a + E_b \cdot i_b + E_c \cdot i_c) / \omega_m$$

機械系の方程式は、

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

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

ここで、 B は係数、 T_{load} は負荷トルク、および P は極数です。この係数 B は慣性モーメント J と機械系の時定数 τ_{mech} から以下のように計算されます。

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

ホール効果センサについての追加説明

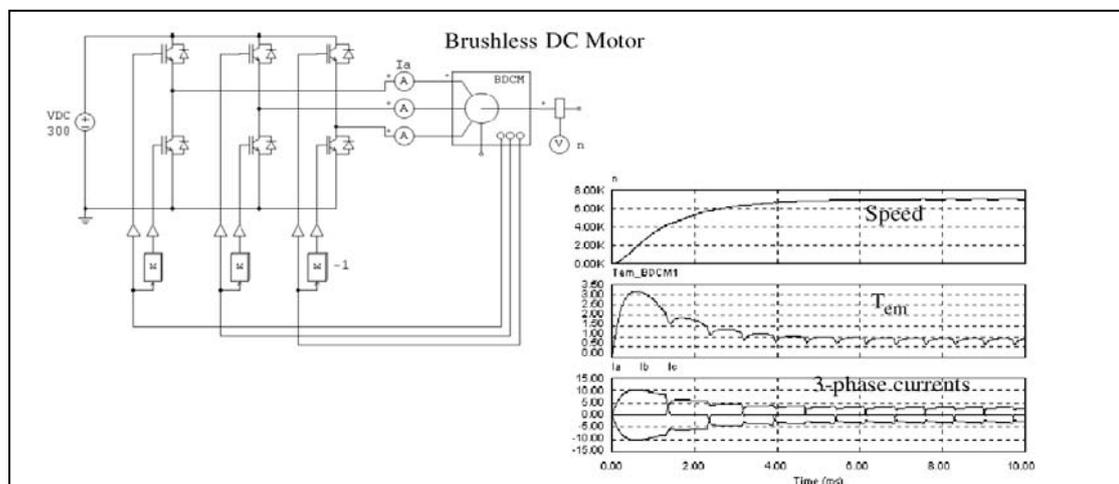
ホール効果位置検出センサはホール・スイッチとトリガ磁石から構成されています。ホール・スイッチは半導体スイッチで（たとえば MOSFET や BJT）、磁界があるしきい値よりも高いか低い場合に開閉します。これはホール効果の応用で、スイッチに外部から電流を流すと磁束密度に比例した電圧が誘起されます。ホール・スイッチと一体になった（もしくは近接した）信号条件回路により誘起電圧を検出するのが一般的です。この回路により、立ち上がり鋭くノイズ耐性が高い TTL レベルのパルスを発生し、シールドケーブルにより制御回路に接続することができます。3相ブラシレス直流モータの場合、3つのホール素子を電気角で 120° ずつずらして固定子に配置します。

トリガ磁石は別の磁石でも、あるいは回転子の永久磁石も使うことができます。別置のトリガ磁石を使う場合、回転子の極から十分に離して配置する必要があります。そして、ホール素子に充分近くなるように軸に取り付けます。回転子の磁石を兼用する場合は、ホール素子を回転子の磁石に充分近く取り付けます。これは適正な回転子の位置で漏れ磁束により励起されるようにするためです。

例：開ループ・ブラシレス直流モータの始動

以下の図に開ループのブラシレス直流モータの駆動回路を示します。モータには3相電圧型インバータにより電源を供給しています。ホール素子位置センサの出力を使って、インバータのゲート信号を調整し、結果として6パルス運転を実現しています。

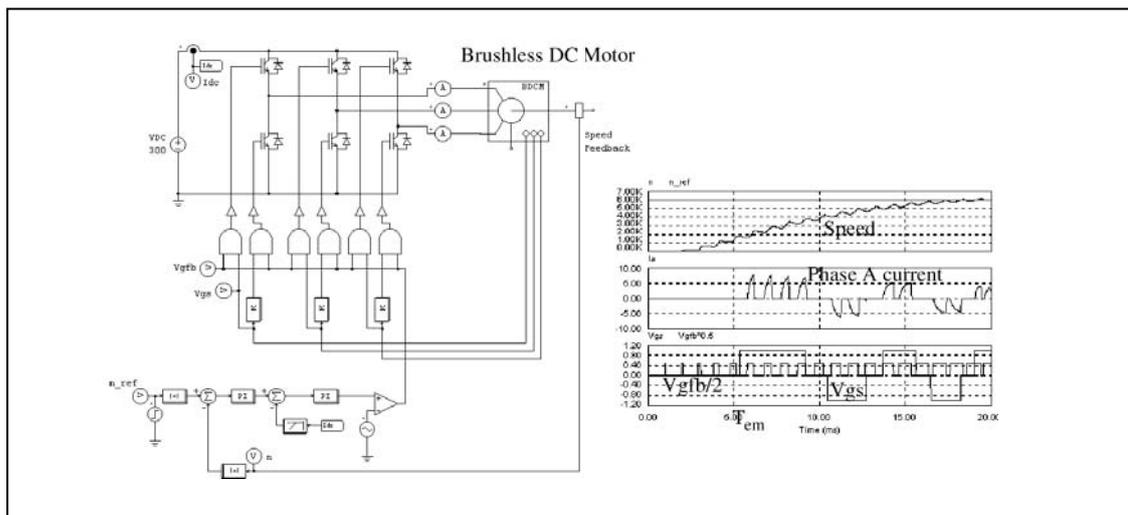
シミュレーション波形はモータ始動時の機械系回転速度 (rpm)、電気トルク T_{em} および3相入力電流を示します。



例：速度フィードバック付きブラシレス直流モータ

以下の図は速度フィードバック付きブラシレス直流モータの駆動回路です。センサの転流パルス（この場合 A 相に対して V_{gs} ）ともうひとつの高周波パルス（A 相に対する V_{gfb} ）を調整することにより、速度制御を実現しています。高周波パルスは直流電流フィードバックにより発生させています。

シミュレーション波形は基準および実際の機械系回転速度（rpm）、A 相の電流、および V_{gs} と V_{gfb} の制御信号を示しています。ここでは表現上、 V_{gfb} は実際の半分の大きさに描いてあります。



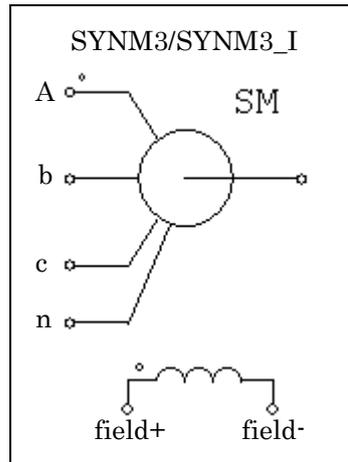
2.9.6 他励式同期機

従来型の同期機の構造は 3 相固定子巻線と突極または円筒型の回転子の界磁巻線、それに機種により制動巻線からなります。

同期機モデルは、固定子回路の内部モデルによって、電圧型 (SYNM3) と電流型 (SYNM3_I) の 2 通りのモデルが存在します。電圧型インタフェースのモデルは、固定子側に置かれた制御付き電圧源からなり、このモデルは同期機が発電機として動作している場合、および固定子外部回路が誘導性ブランチに直列接続されているときに適します。一方、電流型インタフェースは固定子側に置かれた制御付電流源により、このモデルは同期機が電動機として動作している場合、および固定子回路が容量性のブランチと並列接続されているときに効果的です。

このモデルのシンボルと仕様を以下に示します。

シンボル



仕様

パラメータ	機能
R _s (Stator)	固定子巻線抵抗 (Ω)
L _s (Stator)	固定子巻線漏れインダクタンス (H)
L _{dm} (d-axis mag. ind.)	d 軸磁化インダクタンス (H)
L _{qm} (q-axis mag. ind.)	q 軸磁化インダクタンス (H)
R _f (field)	界磁巻線抵抗 (Ω)
L _{fl} (field)	界磁巻線漏れインダクタンス (H)
R _{dr} (damping cage)	制動巻線 d 軸抵抗 (Ω)
L _{drl} (damping cage)	制動巻線 d 軸漏れインダクタンス (H)
R _{qr} (damping cage)	制動巻線 q 軸抵抗 (Ω)
L _{qrl} (damping cage)	制動巻線 q 軸漏れインダクタンス (H)
N _s /N _f (effective)	固定子と界磁回路の実効巻線比
Number of Poles	モータの極数 (偶数)
Moment of Inertia	モータの慣性モーメント J (kg*m ²)
Torque Flag	内部トルク T _{em} の出力フラグ。フラグが 1 に設定されると、内部トルクの出力が要求される。
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ; 0: スレーブ)

全てのパラメータは固定子側に参照した値で表されています。

3 相同期機の動特性は以下の方程式で記述されています。

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

ここで、

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

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

ここで、インダクタンス行列は以下のように定義しています。

$$\begin{aligned}
 [L] &= \begin{bmatrix} [L_{11}] & [L_{12}] \\ [L_{12}]^T & [L_{22}] \end{bmatrix} \\
 [L_{11}] &= \begin{bmatrix} L_s + L_o + L_2 \cos(2\theta_r) & -\frac{L_o}{2} + L_2 \cos\left(2\theta_r - \frac{2\pi}{3}\right) & -\frac{L_o}{2} + L_2 \cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ -\frac{L_o}{2} + L_2 \cos\left(2\theta_r - \frac{2\pi}{3}\right) & L_s + L_o + L_2 \cos(2\theta_r) & -\frac{L_o}{2} + L_2 \cos(2\theta_r) \\ -\frac{L_o}{2} + L_2 \cos\left(2\theta_r + \frac{2\pi}{3}\right) & -\frac{L_o}{2} + L_2 \cos(2\theta_r) & L_s + L_o + L_2 \cos\left(2\theta_r - \frac{2\pi}{3}\right) \end{bmatrix} \\
 [L_{12}] &= \begin{bmatrix} L_{sf} \cos(2\theta_r) & L_{sd} \cos(2\theta_r) & -L_{sq} \sin(2\theta_r) \\ L_{sf} \cos\left(2\theta_r - \frac{2\pi}{3}\right) & L_{sd} \cos\left(2\theta_r - \frac{2\pi}{3}\right) & -L_{sq} \sin\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{sf} \cos\left(2\theta_r + \frac{2\pi}{3}\right) & L_{sd} \cos\left(2\theta_r + \frac{2\pi}{3}\right) & -L_{sq} \sin\left(2\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \\
 [L_{22}] &= \begin{bmatrix} L_f & L_{fdr} & 0 \\ L_{fdr} & L_{dr} & 0 \\ 0 & 0 & L_{qr} \end{bmatrix}
 \end{aligned}$$

ここで、 θ_r は回転子角度です。また、電気トルクは以下の通りです。

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

機械系の方程式は以下のように表されます。

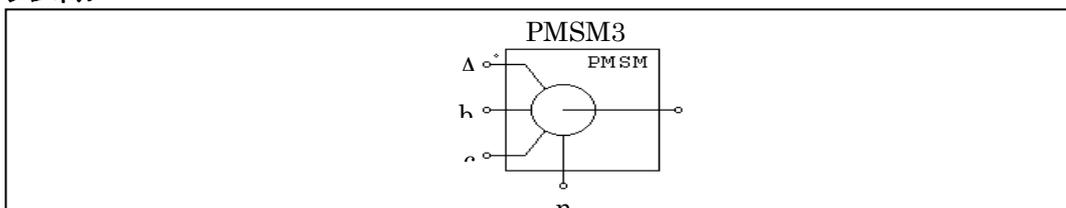
$$\begin{aligned}
 J \cdot \frac{d\omega_m}{dt} &= T_{em} - T_{load} \\
 \frac{d\theta_r}{dt} &= \frac{P}{2} \cdot \omega_m
 \end{aligned}$$

2.9.7 永久磁石同期機

3 相の永久磁石を使った同期機は固定子に 3 相巻線を、回転子には永久磁石を使っています。ブラシレス直流機と違い、同期機は誘起電圧が正弦波です。

永久磁石同期機のシンボルとパラメータを以下に示します。

シンボル



仕様

パラメータ	機能
R_S (Stator Resistance)	固定子の巻線抵抗 (Ω)
L_d (d-axis ind.)	固定子の d 軸インダクタンス (H)
L_q (q-axis ind.)	固定子の q 軸インダクタンス (H) d 軸は磁石の中心を通り、q 軸はふたつの磁石の間を通ります。q 軸は d 軸に比べて進んでいます。
Vpk / krpm	線間のピーク誘起電圧定数、単位は V/krpm (機械系の回転速度) です。 Vpk/krpm の値はモータのデータシートから得られるはずですが、これらの値が不明の場合は、同期機を 1000rpm で発電機として運転し、線間電圧のピーク値を測ることでわかります。
No. of poles	極数 P
Moment of Inertia	モータの慣性モーメント J ($\text{kg}\cdot\text{m}^2$)
Shaft Time Constant	軸の時定数 (τ_{shaft})
Torque Flag	内部電気トルク T_{em} の出力フラグ (1: 出力あり; 0: なし)
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ, 0: スレーブ)。マスタ/スレーブのフラグの設定により、モータの動作モードが変わります。詳しくは 2.9.1 を参照ください。

シンボルの端子は以下のとおりです。a, b, c はそれぞれ a, b, c 相の固定子巻線端子；固定子巻線は Y 結線され、n は中性点端子です。軸端子は機械軸を接続するためのもの。これらはすべてパワー端子なので、パワー回路に接続しなければなりません。

永久磁石同期機の方程式は以下で表されます。

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

ここで、 v_a, v_b, v_c は相電圧、 i_a, i_b, i_c は相電流、 R_S は固定子の各相抵抗です。また、 $\lambda_a, \lambda_b, \lambda_c$ はそれぞれ a, b, c 相の交鎖磁束です。交鎖磁束は以下の式で表せます。

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

ここで、 θ_r は回転子の電気角、 λ_{pm} は以下の式で決まる係数です。

$$\lambda_{pm} = \frac{60 \cdot V_{pk} / krpm}{\pi \cdot P \cdot 1000 \cdot \sqrt{3}}$$

Pは極数です。

固定子の自己および相互インダクタンスは回転子の位置により変化し、以下の式で決まります。

$$\begin{aligned} L_{aa} &= L_{sl} + L_o + L_2 \cdot \cos(2\theta_r) \\ L_{bb} &= L_{sl} + L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ L_{cc} &= L_{sl} + L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{ab} &= L_{ba} = -\frac{1}{2}L_o + L_2 \cdot \cos\left(2\theta_r - \frac{2\pi}{3}\right) \\ L_{ac} &= L_{ca} = -\frac{1}{2}L_o + L_2 \cdot \cos\left(2\theta_r + \frac{2\pi}{3}\right) \\ L_{bc} &= L_{cb} = -\frac{1}{2}L_o + L_2 \cdot \cos(2\theta_r) \end{aligned}$$

ここで、 L_{sl} は固定子の漏れインダクタンスです。

また、d軸およびq軸インダクタンスは以下のように書けます。

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

電気トルクは次の式で計算されます。

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

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

一方、機械系の方程式は

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

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

ここで、B は係数、 T_{load} は負荷トルク、および P は極数です。この係数 B は慣性モーメント J と機械系の時定数 T_{mech} から以下のように計算されます。

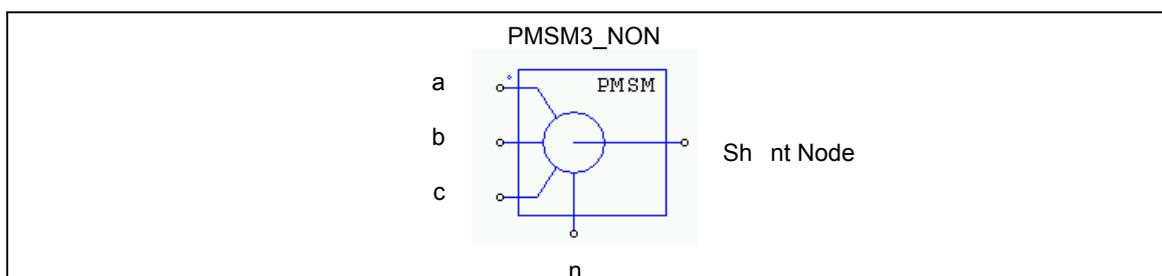
$$B = \frac{J}{\tau_{mech}}$$

2.9.8 飽和付永久磁石同期機

3 相の飽和付永久磁石同期機は、d 軸/q 軸インダクタンスを d 軸/q 軸電流の関数として設定することができる永久磁石同期機です。d 軸/q 軸インダクタンスと d 軸/q 軸電流の関係は別ファイルに記述したルックアップテーブルによって決定されます。

飽和付永久磁石同期機のシンボルとパラメータを以下に示します。

シンボル



仕様

パラメータ	機能
R_S (stator Resistance)	固定子の巻線抵抗 (Ω)
L_S (stator leakage ind.)	固定子の漏れインダクタンス (H)
Vpk / krpm	線間のピーク誘起電圧定数、単位は V/krpm (機械系の回転速度) です。Vpk/krpm の値はモータのデータシートから得られるはずですが、これらの値が不明の場合は、同期機を 1000rpm で発電機として運転し、線間電圧のピーク値を測ることでわかります。
No. of poles	極数 P
Moment of Inertia	モータの慣性モーメント J ($kg \cdot m^2$)
Shaft Time Constant	軸の時定数 T_{shaft} (s)。 摩擦係数 B と関係する。ただし、 $B=J / T_{shaft}$
L_d Lookup Table File	L_{dm} 用ルックアップテーブルのファイル名
L_q Lookup Table File	L_{qm} 用ルックアップテーブルのファイル名
dq Flag	ルックアップテーブル用のフラグ(0: L_{dm} , L_{qm} は i_d , i_q の関数; 1: L_{dm} , L_{qm} は電流ベクトル I_m と角度の関数)
Transformation Flag	変換係数フラグ (詳細は以下を参考のこと)
Torque Flag	内部電気トルク T_{em} の出力フラグ (1: 出力あり; 0: なし)

Master/Slave Flag	マスタ/スレーブモードのフラグ（1：マスタ，0：スレーブ）。マスタ/スレーブのフラグの設定により、モータの動作モードが変わります。詳しくは2.9.1を参照ください。
-------------------	--

d 軸, q 軸のインダクタンス L_d , L_q と磁化インダクタンス L_{dm} , L_{qm} との関係は以下の通りになっています。

$$L_d = L_s + L_{dm}$$

$$L_q = L_s + L_{qm}$$

ただし、 L_s は固定子の漏れインダクタンスです。通常は L_s が非常に小さいので、 L_d を L_{dm} 、そして L_q を L_{qm} と見なすことができます。

Transformation Flagはabcの三相静止座標からdqの回転座標への座標変換の変換係数を決定します。

Transformation Flagが0の時：

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

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

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

電流ベクトルの位相角の単位は degree で、範囲は-180°から 180°になっています。

また、Transformation Flag が 1 の時：

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

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

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

電流ベクトルの位相角の単位は degree で、範囲は 0°から 360°になっています。

L_{dm} と L_{qm} のルックアップテーブルは以下のフォーマットを持っています：

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

ただし、 m は行の数、 n は列の数、 V_r は行のベクトル、 V_c は列のベクトル、そして L_{ij} は i 行目と j 列目における L_{dm} 或いは L_{qm} の値[H]であります。ベクトル V_r と V_c は単調増加である必要があります。

dq Flag が 0 の場合、ベクトル V_r には I_d のデータを、ベクトル V_c には I_q のデータを設定し、 L_{ij} には該当する I_d, I_q の場合のインダクタンス値を設定します。

dq Flag が 1 の場合、ベクトル V_r には I_m のデータを、ベクトル V_c には θ_m のデータを設定し、 L_{ij} には該当する I_m, θ_m の場合のインダクタンス値を設定します。

入力がルックアップテーブルで定義された二つのポイントの間にあるとき、補間された値が計算に使われます。また、入力が最小値より小さいまたは最大値より大きいとき、入力に最小値または最大値が代入されます。ルックアップテーブルの L_{dm} と L_{qm} を I_d と I_q の線形関数として定義すれば、この飽和付永久磁石同期機モデルは線形永久磁石同期機モデルとして使うことが可能です。

以下にルックアップテーブルの例を示します：

```

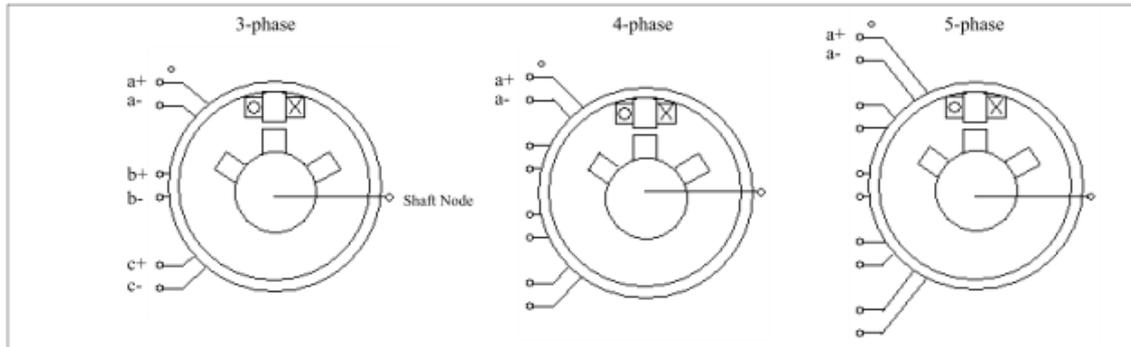
4,15
-5.7155 -4.8990 -4.0825 -3.2660
-5.7155 -4.8990 -4.0825 -3.2660 -2.4495 -1.6330 -0.8165 0 0.8165 1.6330 2.4495 3.2660 4.0825
4.8990 5.7155
0.0109 0.0109 0.0107 0.0104 0.0102 0.0100 0.0098 0.0098 0.0098 0.0100 0.0102 0.0104 0.0107
0.0109 0.0109
0.0109 0.0109 0.0109 0.0106 0.0109 0.0106 0.0105 0.0105 0.0105 0.0106 0.0109 0.0106 0.0109
0.0109 0.0109
0.0109 0.0109 0.0109 0.0109 0.0111 0.0108 0.0106 0.0106 0.0106 0.0108 0.0111 0.0109 0.0109
0.0109 0.0109
0.0110 0.0110 0.0111 0.0110 0.0110 0.0109 0.0108 0.0107 0.0108 0.0109 0.0110 0.0110 0.0111
0.0110 0.0110

```

2.9.9 スイッチトリラクタンスモータ

3相、4相、5相の3つのタイプのスイッチトリラクタンスモータモデルが用意されました。

シンボル



仕様

パラメータ	機能
Resistance	固定子各相抵抗 (Ω)
Inductance Lmin	各相インダクタンスの最小値 (H)
Inductance Lmax	各相インダクタンスの最大値 (H)
Theta_min, θ _{min}	インダクタンスが最低にある期間 (度)
Theta_max, θ _{max}	インダクタンスが最大にある期間 (度)
Stator Pole Number	固定子極数
Rotor Pole Number	回転子極数
Moment of Inertia	モータの慣性モーメント J (kg*m ²)
Torque Flag	内部トルク T _{em} の出力フラグ。
Master/Slave Flag	マスタ/スレーブモードのフラグ (1 : マスタ, 0 : スレーブ)

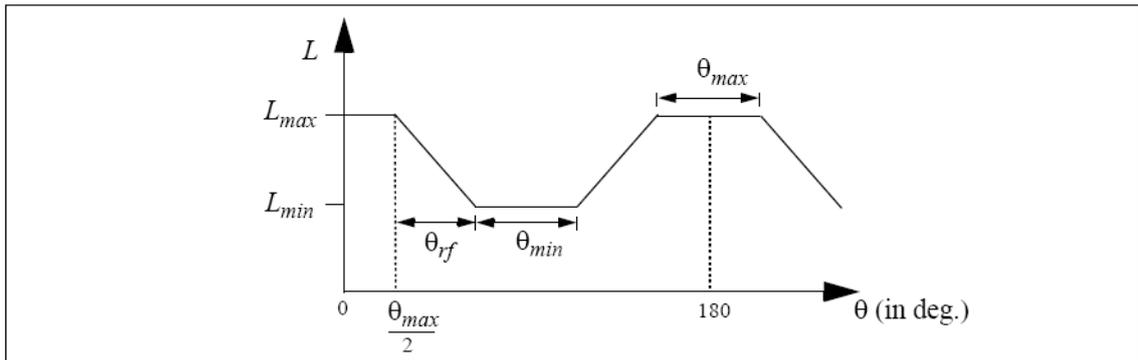
マスタ/スレーブのフラグの設定により、モータの動作モードが変わります。(2.8.1 参照)

ノードの役割は以下のとおりです。a+, a-, b+, b-, c+, c- はそれぞれ a, b, c 相の固定子巻線端子。軸ノード (shaft node) は機械軸の接続端子。これらの端子はパワー端子なのでパワー回路に接続しなければなりません。

スイッチトリラクタンスモータの方程式は、各相あたり次のようになります。

$$v = i \cdot R + \frac{d(L \cdot i)}{dt}$$

ここで、v は相電圧、i は相電流、R は相回路抵抗、L は相回路インダクタンスです。相回路インダクタンス L は以下の図に示すように回転の関数です。インダクタンスプロファイルは半波対称で 180 度の後は繰り返しとなります。



回転角は、固定子と回転子の歯が完全にかみ合っていない場合を $\theta = 0$ とします。インダクタンスの値は、上昇 (rising)、頂上 (flat-top)、降下 (falling)、底 (flat-bottom) の4段階のいずれかになります。

ここで定数 k を $k = \frac{L_{max} - L_{min}}{\theta}$ と定義すると、インダクタンス L を以下のように回転角の関数として表現できます。

$$\begin{aligned}
 L &= L_{max} && \text{for } 0 \leq \theta \leq \frac{\theta_{max}}{2} \\
 L &= L_{max} - k \cdot \left(\theta - \frac{\theta_{max}}{2} \right) && \text{for } \frac{\theta_{max}}{2} \leq \theta \leq \frac{\theta_{max}}{2} + \theta_{rf} \\
 L &= L_{min} && \text{for } \frac{\theta_{max}}{2} + \theta_{rf} \leq \theta \leq \frac{\theta_{max}}{2} + \theta_{rf} + \theta_{min} \\
 L &= L_{min} + k \cdot \left(\theta - \frac{\theta_{max}}{2} - \theta_{rf} - \theta_{min} \right) && \text{for } \frac{\theta_{max}}{2} + \theta_{rf} + \theta_{min} \leq \theta \leq \frac{\theta_{max}}{2} + 2\theta_{rf} + \theta_{min} \\
 L &= L_{max} && \text{for } \frac{\theta_{max}}{2} + 2\theta_{rf} + \theta_{min} \leq \theta \leq 180^\circ
 \end{aligned}$$

各相の電気トルクは以下により求めています。

$$T_{em} = \frac{1}{2} \cdot i^2 \cdot \frac{dL}{d\theta}$$

上記のようなインダクタンスの表現により、各段階での電気トルクは、

$$T_{em} = i^2 \cdot \frac{k}{2} \quad \text{[上昇]}$$

$$T_{em} = 0 \quad \text{[頂上]}$$

$$T_{em} = -i^2 \cdot \frac{k}{2} \quad \text{[降下]}$$

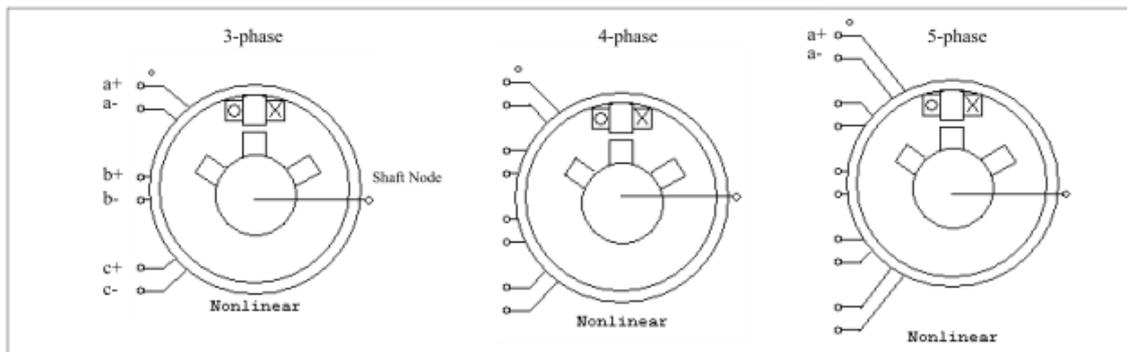
$$T_{em} = 0 \quad \text{[底]}$$

なお、このモデルでは飽和を考慮していません。

2.9.10 非線形スイッチトリラクタンスモータ

以前のセクションのスイッチトリラクタンスモータモデルでは、インダクタンスは回転子位置にだけに依存します。更に、インダクタンスは頂上 (flat-top) 、と底 (flat-bottom) の時は固定です。この3相モデルでは、インダクタンスと回転子位置の関係だけではなく、インダクタンスと電流の関係も非線形に定義することができます。インダクタンスと回転子位置の関係及びインダクタンスと電流の関係は2次元のルックアップテーブルを通して定義されます。

シンボル



仕様

パラメータ	機能
Resistance	固定子各相抵抗 (Ω)
Stator Pole Number	固定子極数
Rotor Pole Number	回転子極数
Inductance Table File	ローター位置と電流に対してインダクタンスの 2 次元のテーブルを保存するファイル名
Moment of Inertia	モータの慣性モーメント J ($\text{kg}\cdot\text{m}^2$)
Torque Flag	内部トルク T_{em} の出力フラグ。フラグを 1 に設定すると内部トルクの出力を要求できます。
Master/Slave Flag	マスタ/スレーブモードのフラグ (1 : マスタ, 0 : スレーブ)

インダクタンステーブルファイルのフォーマットは以下の通りです :

```

m, n
θ1, θ2, ..., θm
I1, I2, ..., In
L1,1, L1,2, ..., L1,n
L2,1, L2,2, ..., L2,n
... ..
Lm,1, Lm,2, ..., Lm,n
    
```

ここで、 m は行の数、 n は列の数、 $\theta_1 \sim \theta_m$ は回転子位置(単位 : degree)の行のベクトル、 $i_1 \sim i_n$ は相電流(単位:A)の列のベクトル、そして L_{ij} は i 行目と j 列目におけるインダクタンス値[H]です。例えば、回転子位置 θ_2 、相電流 i_1 の場合、インダクタンスは $L_{2,1}$ です。半波対称のために、 $0 \sim 180$ 度のインダクタンス値だけが必要です。また、行ベクトルと列ベクトルは増加である必要があります。

2.10 MagCoupler モジュール

MagCouplerモジュールはPSIMとJMAGの連成シミュレーションを行なうためのインタフェースを提供します。JMAGは電気機器、アクチュエータやその他電気・電子機器または部品の電磁界解析ソフトウェアです。MagCouplerモジュールを使えば、PSIMによるパワーエレクトロニクスとその制御、そしてJMAGの電磁界解析を同時に実行することができます。

MagCouplerモジュールはMagCouplerブロックとMagCoupler-DLブロックの二つがあり、これらはそれぞれセクション2.11で紹介する機械素子と速度/トルク/位置のセンサにあたります。

MagCouplerブロックとMagCoupler-DLブロックの違いは以下の通りです。

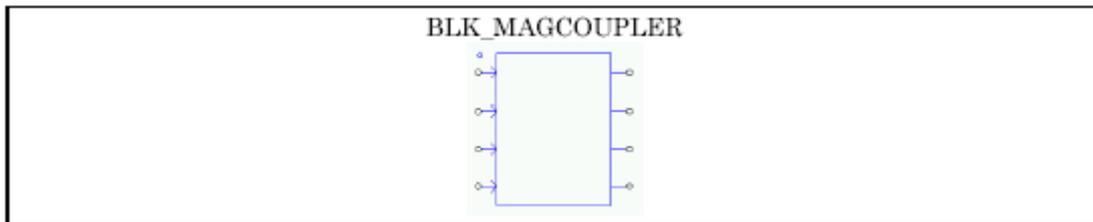
- 「全体のシミュレーション時間短縮可能」
MagCouplerブロックでは、PSIMとJMAGは同じ時間ステップで実行します。MagCoupler-DLブロックは違う時間ステップで実行させることもできます。これにより、比較的時間のかかるJMAG側の時間ステップだけを長くして、全体のシミュレーション時間を短くすることも可能です。
- 「電流入力可能」
MagCouplerブロックのインタフェースポートは信号ポートであり、電流を入出力することが出来ません。よって電流入力をブロックに接続するためには、まずこれを電圧信号に変換しなければなりません。一方、MagCoupler-DLブロックの入出力は電力の入出力ポートであり、通常の電力回路のノードと同様の働きを持ちます。（つまり、電圧が加わり、電流がノードに流入、流出する現象が表現されるということです。）この違いから、MagCoupler-DLブロックのPSIMとJMAG間のインタフェースはさらに現実に則したものになっているといえます。

MagCouplerを使用する場合、PSIM上でJMAGがインストールされているディレクトリか、JMAGのDLLファイル（「jbdll.dll」と「RT_Dll.dll」）のあるディレクトリのPathを設定する必要があります。PSIMにJMAGのディレクトリのPathを設定するには、Option→Set Pathを選択し、Add Folderボタンを押して、JMAGのディレクトリを選択してください。

また、MagCouplerを使用するためには、Internet Explorer Ver.6.0以降が必要なことにご注意ください。Ver.5.x以前のInternet ExplorerがインストールされているコンピュータではMagCouplerは正常に動作しません。

2.10.1 MagCoupler ブロック

シンボル



仕様

パラメータ	機能
Link Table File	PSIM と JMAG との接続を定義する XML ファイル (.xml の拡張子) です。
JMAG Input File	JMAG 用の JCF 入力ファイル (.jcf 拡張子) です。 .xml ファイルと .jcf ファイルは同じディレクトリ上にある必要があります。
JMAG Case Text	JMAG 回路用のコメント
IN Nodes	PSIM から JMAG に値を渡すノード
OUT Nodes	JMAG から PSIM に値を渡すノード

使う回路の入出力ノード数によってシンボルの入出力ノード数も変化します。

MagCoupler ブロックは、電圧、電流と位置を入力とし、電圧、電流、位置、トルクと力を出力します。PSIM では、MagCoupler ブロックはパワー回路素子として扱われます。

MagCoupler と他の回路との連結はいずれの入力と出力においても電圧信号です(入力ノードに流れ込む電流は存在しません)。このとき、回路中からの電流を電圧信号へ変換(あるいはその逆の変換)をするには、「電流制御電圧源」あるいは「電圧制御電流源」を使うことで実現できます。

Link Table File は入出力のインターフェースと JMAG に関連する関数を定義します。このファイルは JMAG によって自動的に生成されます。編集フィールドの右側にあるブラウズボタンをクリックしてこのファイルを設置してください。

JMAG Input File は JMAG のソルバに読み込まれる JCF 入力データファイルです。ファイル名は Link Table File で定義されます。

JCF 入力ファイル .jcf は必ず Link table file .xml と同じディレクトリにあることにご注意ください。何らかの素子データベースを JMAG で使用している場合、それらのファイルも .xml ファイルと同じディレクトリに置く必要があります。一方、.xml ファイルは必ずしも回路図ファイルと同じディレクトリに入れる必要

はありません。ただし、同じファイル名の.xml ファイルが回路図ディレクトリに存在する場合、PSIM は優先的に回路図ディレクトリにあるファイルを読み込みます。

JMAG Case Text はユーザがJMAG 回路を識別するための「名前」として利用するテキストであり、自由に設定することができます。

IN Nodes はPSIM からJMAG に値を渡すノードです。MagCoupler ブロックの図では、入力端子の順番はIN Nodes の設定と同じ順序で左上から左下に並びます。この順番はノードをハイライト化し、上あるいは下矢印をクリックすることで変えることができます。

Out Nodes はJMAG がPSIM に値を返すためのノードです。MagCoupler ブロック図において、出力端子の順番はOut Nodes の設定と同じ順序で右上から右下に並びます。この順番はノードをハイライト化し、上あるいは下矢印をクリックすることで変えることができます。

Edit Image ボタンをクリックしますと、MagCoupler ブロック図を編集したり、カスタマイズしたりすることができます。Display File ボタンを押しますと、Microsoft Internet Explorer環境にあるLink Table File を表示され、そしてRead File ボタンを押しますとLink Table Fileが読み込まれる、またはリロードされません。

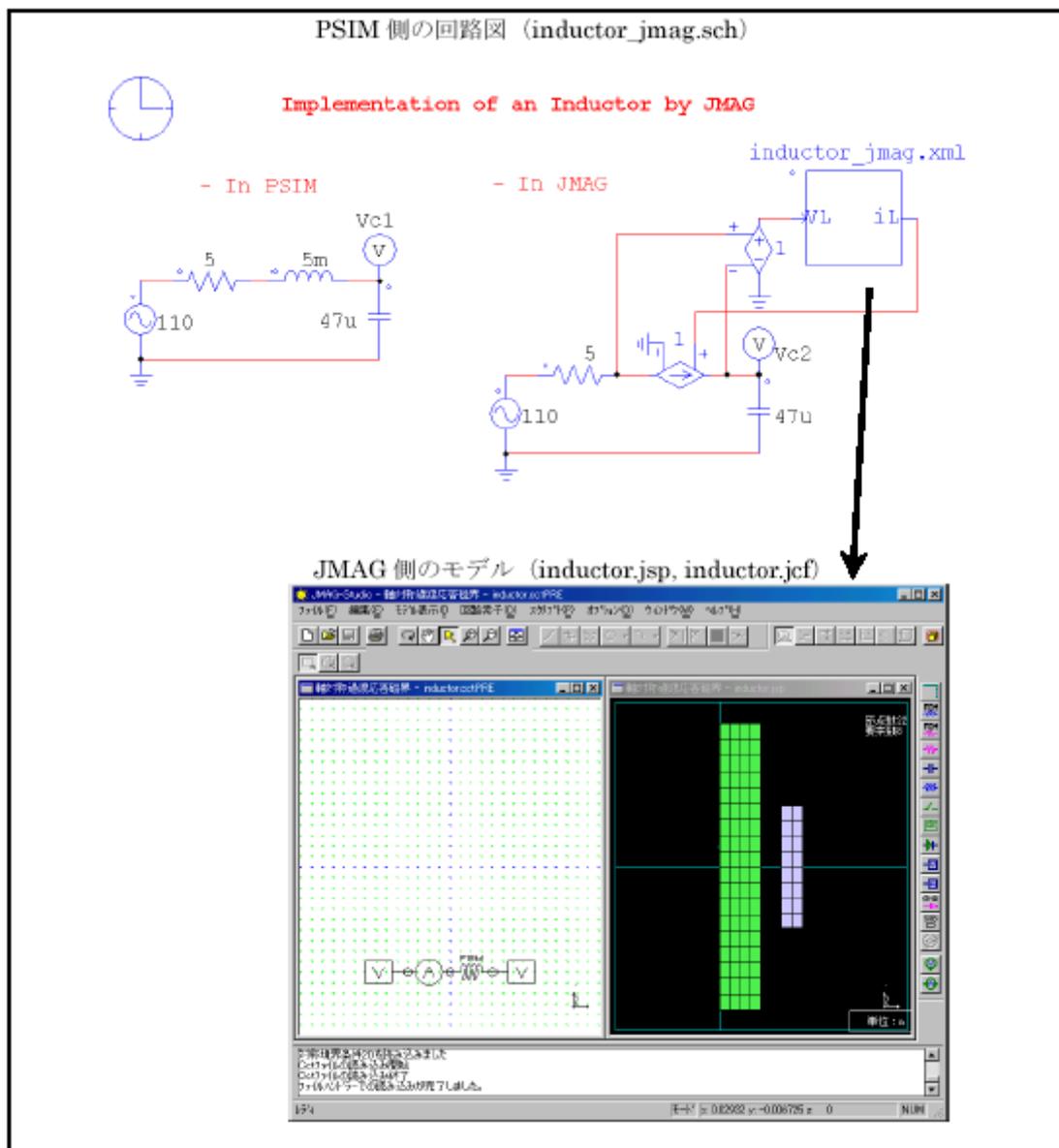
JMAG、PSIM のセットアップ

MagCoupler ブロックを使った連成シミュレーションに向けたJMAG とPSIM間のリンクのセットアップには、大きく二つのステップがあります。まずJMAG 上で回路をセットアップしてLink Table File を生成し、そのLink Table FileをPSIM に読み込む設定をします。

インダクタを例にこの手順について説明します。

この例のPSIM 回路において、左側の回路はPSIM素子 ライブラリに備えつきのインダクタを使います。そして、右側の回路はJMAG で計算されるインダクタを使います。この例では、インダクタはPSIM における制御電流源としてモデル化されます。インダクタの端子電圧はまず電圧制御電圧源を通じてGND を基準とした電圧に変換され、その値はMagCoupler ブロックの入力ノードVL に渡されます。PSIMのシミュレーションタイムステップごとに、入力電圧に基づいてインダクタ電流を計算するJMAG の関数が呼び出されます。JMAG で計算された電流値はPSIM に電圧の形で返され、電圧制御電流源によってインダクタの出力電流になります。

この例のJMAG 側の回路は、Voltage Function(左側)がPSIM から渡された電圧を受け取り、FEM コイルと直列に繋がっている電流プローブを通過します。電流プローブは解析された電流値を観測し、観測した値がPSIM に返されます。JMAG 環境におけるインダクタの構造は右下に示されています。



PSIM でMagCoupler ブロックを通じて、JMAG を呼び出す手順は以下の通りです。

【JMAG 側の操作手順】

- JMAG 回路で、FEM Coil (FEM コイル) の右側にVoltage Function (電源電圧) を接続してください。プロパティウィンドウのElectrical Potential (電位) に属するConstantValue (一定値) を選択し、Constant Value[V] (一定値[V]) を0にセットしてください。
- FEMコイルの左側にCurrent Probe (電流プローブ) を接続します。

- Current Probe（電流プローブ）の左側にもう一つのVoltage Function（電源電圧）を接続します（回路は上図に示す形になります）。Voltage Function（電源電圧）のプロパティウィンドウでCooperates with an external circuit simulator（外部シミュレータと連成する）を選択します。
- インダクタの構造ウィンドウをハイライト化し、Conditions → Create Conditions（条件→条件作成）に移動します。表示されるリストからCoupled External Circuit Simulator（外部回路シミュレータ連成）を選択してOKを押してください。
- Coupled External Circuit Simulator（外部回路シミュレータ連成）ダイアログウィンドウ上に二つのリストが存在します。右側にあるJMAGというリストにPSIMと連動するために使用できるすべてのファンクションが含まれています。一方、左側にあるExternal Circuit Simulator（外部回路シミュレータ）というリストにPSIMと連動するために選択されているファンクションが含まれています。このケースでは、JMAG リストに二つのアイテムがあり、Voltage Function と電流プローブです。
- Voltage Function をハイライト化し、「←」ボタンをクリックし、右側にあるリストから左側にあるリストにアイテムを以降します。電流プローブに対しても同じ手順で行います。すると、これらの二つのアイテムは左側のリストに現れます。
- Voltage Function をハイライト化し、端末名をVLに変更します。また、電流プローブの端子名をiLに変更します。ダイアログウィンドウを閉じます。
- File → Export（ファイル→書き出し）に移動し、JCF...を選択します。JCF ファイル名を“inductor”と設定すると、JCF ファイル“inductor.jcf”とリンクテーブルファイル“inductor_csl.xml”が生成されます。
- 最後にファイル管理の簡略化のために以下の手順を行うことをお勧めします。PSIM 回路図ファイル“inductor_jmag.sch”が含まれるフォルダに、JCF ファイル “inductor.jcf”とリンクテーブルファイル“inductor_csl.xml”をコピーし、リンクテーブルファイルを“inductor_jmag.xml”にファイル名を変更します。

【PSIM 側の操作手順】

- すべての電力回路を作った後、Element → Power → MagCoupler Module に移動し、MagCoupler Block を選択してください。そのブロックを回路図に置いてください。

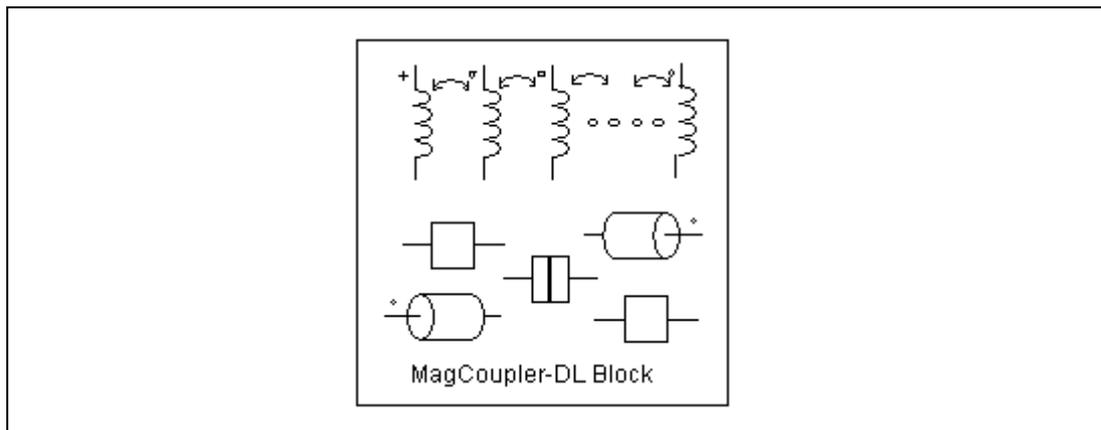
- MagCoupler ブロックをダブルクリックして、プロパティウィンドウを呼び出します。Link Table File エディットフィールドの次にあるブラウザボタンをクリックしファイルを選びます。ここでは"inductor_jmag.xml"を選択します。このファイルが読み込まれた後、プロパティウィンドウはIN ノードVL とOUT ノードiL を表示します。
- MagCoupler ブロックを回路図内の残りの回路に接続します。

これでセットアップは完了です。

2.10.2 MagCoupler-DL ブロック

PMSM デバイスの MagCoupler-DL ブロックは、PMSM デバイスのインタフェースを分かり易く表しています。

シンボル



仕様

パラメータ	機能
Netlist XML File	XML ファイルはPSIMとJMAGのインタフェースを定義しており、.xmlの拡張子を持っています。
JMAG Input File	JMAGの入力ファイルです。拡張子.jcfを持っています。.xmlと.jcfファイルは同じディレクトリに入っていないとなりません。
JMAG Case Text	JMAG回路へのコメントです。
Terminal Names	ブロックのノード名

このブロックシンボル図内では、電源ノード（A、B、そしてC）はブロックの一番上に置かれており、左から右へ配置されています。機械の軸ノードはブロックの左と右に置かれており、右が最初の軸ノード（上図のノードM+）左が二番目の軸ノード（上図のノードM-）です。

電力回路と機械軸ノードは PSIM と JMAG のデータファイル間のインタフェースと同様に Netlist XML File 内に定義されています。このファイルは XML 形式であり、JMAG から生成されます。このファイルを指定するためには、編集部分の右のブラウザボタンをクリックしてください。

JMAG Input File はデバイスをモデル化した JMAG のデータファイルです。このファイルは .jcf 拡張子を持っており、ネットリスト XML ファイル内に定義されています。 .jcf ファイルと .xml ファイルは同じディレクトリに入っていなければなりません。

JMAG Case Text はユーザが JMAG 回路を識別するための「名前」として利用するテキストであり、自由に設定することができます。

特定の XML ファイルを選び、.jcf データファイルが読み込まれると、以下のパラメータがプロパティウインドウ内に表示されます。

パラメータ	機能
dt of JMAG	JMAG 内の時間ステップ
lr	定格電流です。JMAG 内で使われています。
dl_threshold	電流基準値です。JMAG 内で使われています。
dr_threshold	移動の基準点です。JMAG 内で使われています。
MomentofInertia	装置の慣性モーメント
MechTimeConstant	装置の機械的時定数(sec)
Beck emf Flag	全ての FEM コイル電流の表示フラグ (1: 表示、0: 非表示)
Rotor Angle Flag	全軸角の表示フラグ
Speed Flag	全軸速度の表示フラグ
Torque Flag	全軸トルクの表示フラグ

JMAG 側シミュレーションのセットアップ方法は JMAG のマニュアルとドキュメントを参照してください。

2.11 MagCoupler-RT モジュール

MagCoupler-RT モジュールは PSIM と JMAG-RT データファイルとのインタフェースを提供します。JMAG-RT は電磁デバイスをモデル化するためのシミュレータです。JMAG-RT データファイルはあらかじめ JMAG 単体でシミュレーションを実行して作成します。PSIM は JMAG-RT から出力されたデータを使用するため、PSIM でシミュレーションする際に、JMAG がインストールされた PC は必要ありません。

JMAG-RT の最大の利点は、PSIM シミュレーションの実行時に JMAG を実行しないため、PSIM シミュレーションの速度が影響を受けることなく高速シミュレーションが可能になる点です。しかも、JMAG-RT データファイルが JMAG の動的シミュレーションから得られるので、モデルの正確さも MagCoupler を使用した場合と近いレベルになります。

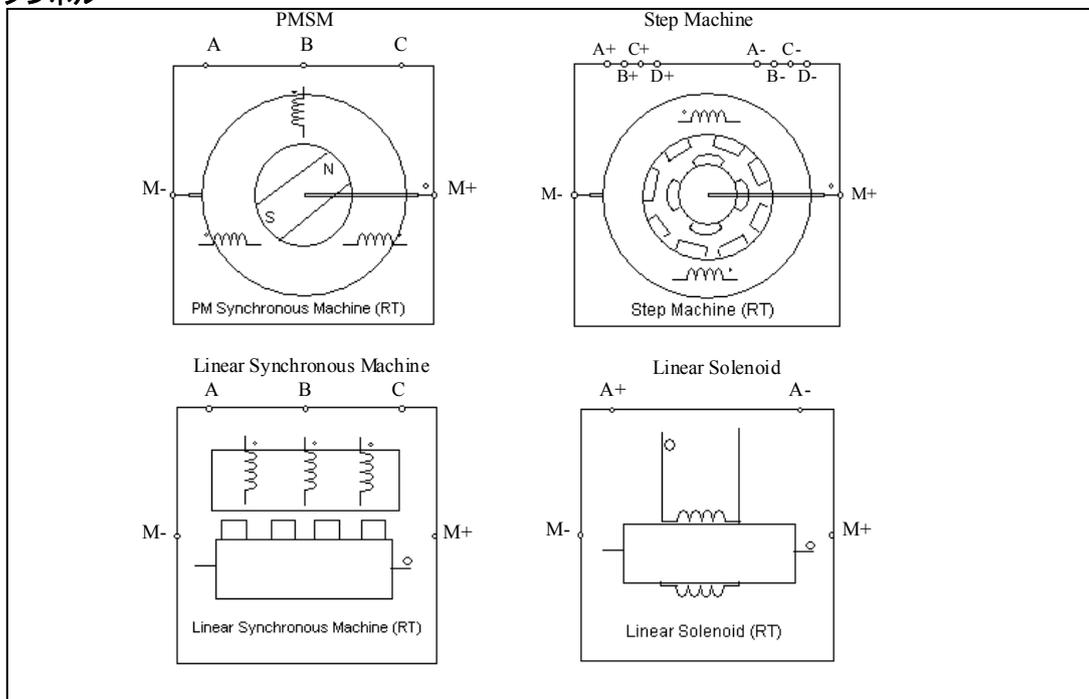
MagCoupler-RT モジュールは、上記の機能のほか、2.11 項に記述されるような機械部と速度/トルク/ポジションセンサを含んでいます。

以下の4つの MagCoupler-RT ブロックが用意されています。

- 3相永久磁石型同期モータ (PMSM)
- 2相ステッピングモータ
- リニア同期モータ
- リニアソレノイド

各ブロックのシンボルは以下の通りです。

シンボル



仕様

パラメータ	機能
Netlist XML File	PSIM と JMAG-RT とのインタフェースを定義する XML ファイル(.xml 拡張子)
JMAG-RT Input File	JMAG-RT データファイル(.rtt 拡張子) .xml ファイルと.rtt ファイルは同じディレクトリ内におく必要があります
JMAG Case Text	ユーザが自由にコメントを記述できます
Terminal Names	ブロックのノード名称

MagCoupler-RT モジュールでは、電気ノード (A, B, C, A+, A-, B+, B-, C+, C-, D+, D-) はブロックの上部に配置されます。ロータシャフトノード(M+, M-)はブロックの左右に配置されます。

電気ノード、ローターシャフトノード、および PSIM と JMAG-RT データファイルとのインターフェースは“Netlist XML File”で定義されています。このファイルは XML 形式で、JMAG-RT マネージャによって生成されます。このファイルを指定するにはモジュールを右クリックし、パラメータ設定ウィンドウにて“.xml”ファイルを参照してください。

このファイル内の“NetlistElement”には、使用する MagCoupler-RT ブロックのタイプごとに決められた名称を設定して下さい。それぞれのブロックの名称は以下の通りです。

- 3 相 PMSM : "PM Synchronous Machine"
- 2 相ステッピングモータ : "Step Machine"
- リニア同期モータ : "Linear Synchronous Machine"
- リニアソレノイド : "Linear Solenoid"

例として、PMSM のネットリスト XML ファイルの最初の数行は、以下のように記述します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<CircuitElement file="imp.rtt" name="MagCoupler-RT">
<NetlistElement type="JMAGRT" element="PM Synchronous Machine" pole_number="4">
```

“JMAG-RT Input File”は JMAG 上でシミュレーションし作成した JMAG-RT データファイルです。このファイルは、“.rtt”拡張子を持ちます。“.rtt”ファイルおよび“.xml”ファイルは同じディレクトリにおく必要があります。

“JMAG Case Text”はユーザが自由にコメントを記述することが可能です。

“Terminal Names”はインターフェースノードの名前となります。ブロック上部の電気ノード (A、B、C) は主回路と接続され、ブロックの左右のローターシャフトノード(M+, M-)は機械系と接続されます。

3 相 PMSM のパラメータは以下の通りです。

パラメータ	機能
shaft1_MomentofInertia	モータの慣性モーメント (kg* m ²)
shaft1_MechTimeConstant	機械的時定数 (sec)
RU_Resistance	U 相の抵抗値 (Ω)
RV_Resistance	V 相の抵抗値 (Ω)
RW_Resistance	W 相の抵抗値 (Ω)
Offsetangle	初期回転角 (°)
coef_inductance	インダクタンス係数 (JMAG-RT で使用)
coef_flux	磁束係数 (JMAG-RT で使用)
coef_torque	トルク係数 (JMAG-RT で使用)
coef_magnet	磁気係数 (JMAG-RT で使用)
coef_material	物質係数 (JMAG-RT で使用)

turns_coil1	コイル1の巻き数 (JMAG-RT で使用)
turns_coil2	コイル2の巻き数 (JMAG-RT で使用)
Current Flag	3相 FEM コイル電流の表示フラグ (1: 表示、0: 非表示)
Back emf Flag	3相 FEM コイルの逆起電力の表示フラグ
Rotor Angle Flag	回転角 (°) の表示フラグ
Speed Flag	シャフトの機械的速度 (rad/sec) の表示フラグ
Torque Flag	発生したトルク (N*m) の表示フラグ
shaft1 Master Flag	モータのマスタ/スレーブフラグ (1: マスタ、0: スレーブ)

2相ステッピングモータのパラメータは以下の通りです。

パラメータ	機能
RA+	A+相の抵抗値 (Ω)
RA-	A-相の抵抗値 (Ω)
RB+	B+相の抵抗値 (Ω)
RB-	B-相の抵抗値 (Ω)
MomentofInertia	モータの慣性モーメント (kg*m ²)
MechTimeConstant	機械的時定数 (sec)
Offsetangle	初期回転角 (°)
turns_coil2	コイル2の巻き数 (JMAG-RT で使用)
coef_Inertia1	慣性1の係数
coef_Inertia2	慣性2の係数
Current Flag	3相 FEM コイル電流の表示フラグ (1: 表示、0: 非表示)
Back emf Flag	3相 FEM コイルの逆起電力の表示フラグ
Rotor Angle Flag	回転角 (°) の表示フラグ
Speed Flag	シャフトの機械的速度 (rad/sec) の表示フラグ
Torque Flag	発生したトルク (N*m) の表示フラグ
shaft1 Master Flag	モータのマスタ/スレーブフラグ (1: マスタ、0: スレーブ)

リニアソレノイドのパラメータは以下の通りです。

パラメータ	機能
RA	ソレノイドの抵抗値 (Ω)
Mass	ソレノイドの質量 (kg)
MechTimeConstant	ソレノイドの機械的時定数 (sec)
SpringConstant	ソレノイドのバネ定数 (JMAG-RT で使用)
DispLimitMax	ソレノイドの変位の上限值 (m)
DispLimitMin	ソレノイドの変位の下限値 (m)
OffsetDisp	初期値の変位 (m)
turns_coil2	コイル2の巻き数 (JMAG-RT で使用)
coef_mass1	質量1係数 (JMAG-RT で使用)
coef_mass2	質量2係数 (JMAG-RT で使用)
Current Flag	3相 FEM コイル電流の表示フラグ (1: 表示、0: 非表示)
Back emf Flag	3相 FEM コイルの逆起電力の表示フラグ
Position Flag	ソレノイドの位置 (m) の表示フラグ
Velocity Flag	ソレノイドの速度 (m/sec) の表示フラグ
Force Flag	発生する力の表示フラグ (N)

mass1 Master Flag	ソレノイドのマスタ/スレーブフラグ (1: マスタ、0: スレーブ)
-------------------	------------------------------------

リニア同期モータのパラメータは以下の通りです。

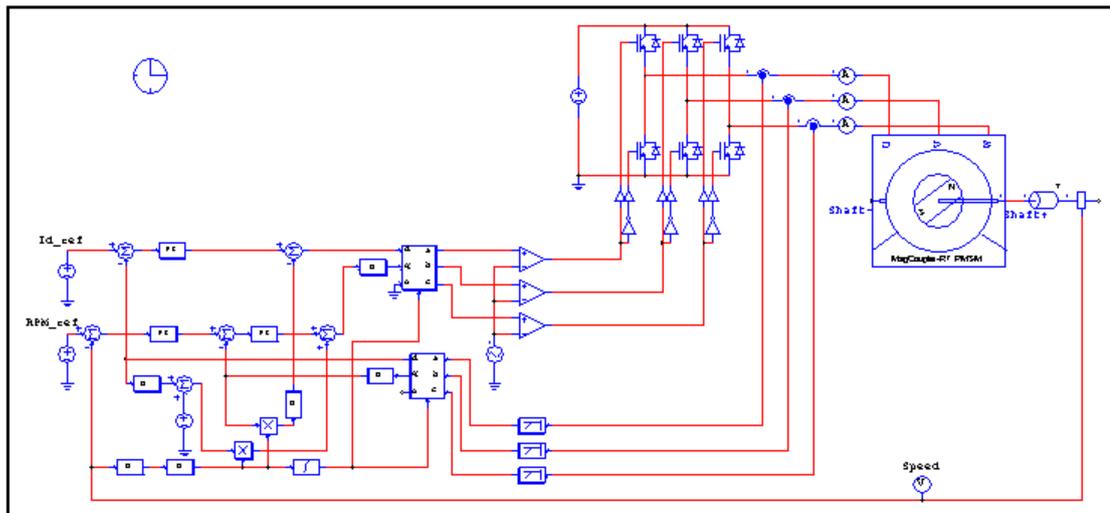
パラメータ	機能
RU	U相の抵抗値 (Ω)
RV	V相の抵抗値 (Ω)
RW	W相の抵抗値 (Ω)
Mass	モータの質量 (kg)
MechTimeConstant	機械的時定数 (sec)
OffsetDisp	初期値の変位 (m)
coef_inductance	インダクタンス係数 (JMAG-RT で使用)
coef_flux	磁束係数 (JMAG-RT で使用)
coef_force	トルク係数 (JMAG-RT で使用)
coef_magnet	磁気係数 (JMAG-RT で使用)
coef_material	物質係数 (JMAG-RT で使用)
turns_coil1	コイル1の巻き数 (JMAG-RT で使用)
turns_coil2	コイル2の巻き数 (JMAG-RT で使用)
coef_mass1	質量1係数 (JMAG-RT で使用)
coef_mass2	質量2係数 (JMAG-RT で使用)
Current Flag	3相 FEM コイル電流の表示フラグ (1: 表示、0: 非表示)
Back emf Flag	3相 FEM コイルの逆起電力の表示フラグ
Position Flag	位置 (m) の表示フラグ
Velocity Flag	速度 (m/sec) の表示フラグ
Force Flag	発生する力の表示フラグ (N)
mass1 Master Flag	モータのマスタ/スレーブフラグ (1: マスタ、0: スレーブ)

上記のうち、フラグを除くすべてのパラメータは XML ファイルで定義された .rtt ファイルから読み取ることができます。表中のすべてのパラメータは PSIM 側で変更することができます。また、“JMAG-RT で使用”と書かれたパラメータを PSIM 側で変更した場合、変更後の値は JMAG-RT に送られます。

MagCoupler-RT のサンプル回路が“examples/MagCoupler-RT”フォルダ内に用意されています。

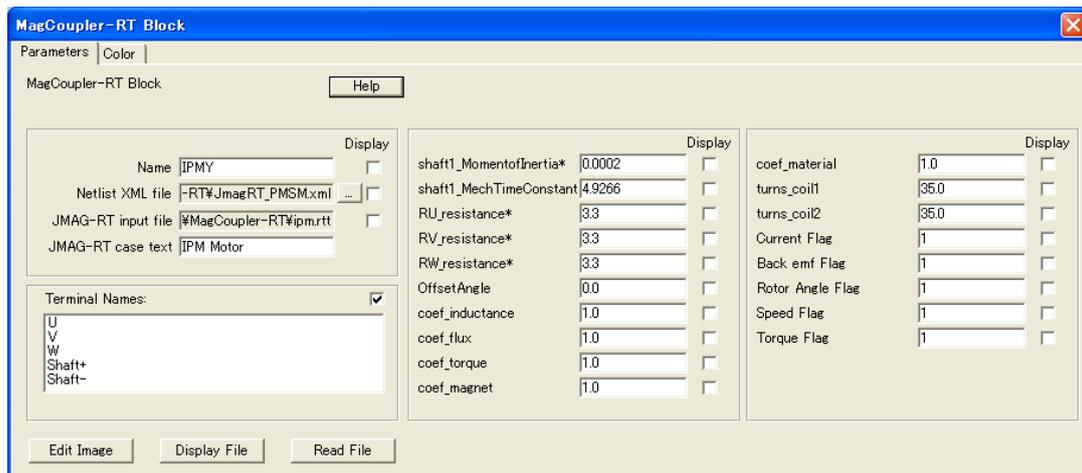
例: JMAG-RT でモデル化した PMSM の駆動回路

下記は JMAG-RT でモデル化した永久磁石同期電動機 (PMSM) の駆動回路です。



また、下記は MagCoupler-RT PMSM ブロックのパラメータ設定ウィンドウです。この例の“.xml”ファイルでは3つの電気ノード(U、V、W)、および2つのローターシャフトノード(shaft+、shaft-)が定義されています。ローターシャフトノードについては、PSIMの機械系モジュールに直接、接続することができます。

パラメータ設定ウィンドウでは“.rtt”ファイルおよび端子名の情報以外に、ユーザは JMAG-RT データファイルのパラメータの値を変更することができます。パラメータ定義と用法については、JMAG-RT のドキュメントを参照してください。(最後の5つのパラメータ(フラグ)を1に設定すると電流、逆起電力、角度、速度、トルクを表示することができます。)



2.12 機械素子及びセンサ

この章では Motor Drive Module、MagCoupler Module、MagCoupler-RT Module に共通する素子について説明します。これらの素子には、機械負荷、ギアボックス、メカニカル・カップリング・ブロック、機械系－電気系インタフェース・ブロック、および様々な速度／トルク／ポジションセンサが含まれます。

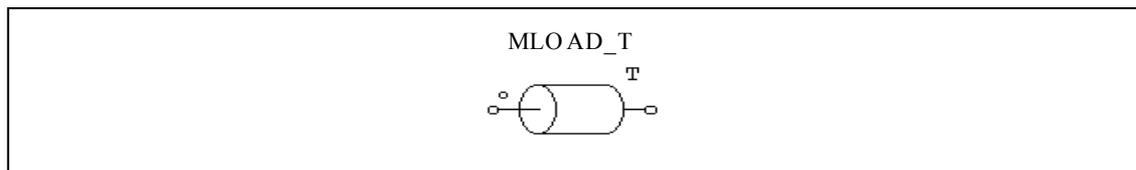
2.12.1 機械負荷

PSIM には機械負荷モデルが標準で備わっています。定トルク負荷、定電力負荷、定速度負荷、一般負荷、及び外部コントロール負荷です。

2.12.1.1 定トルク負荷

定トルク負荷のシンボルは以下ようになります。

シンボル



仕様

パラメータ	機能
Constant Torque	トルク定数 T_{const} (N*m)
Moment of Inertia	モータの慣性モーメント (kg*m ²)

機械系の基準方向が、ドットの付いた端子に入る場合は、負荷は基準方向に沿っているといえ、マスタモータに対する負荷トルクは T_{const} です。そうでない場合、負荷トルクは $-T_{const}$ です。詳細な説明は 2.9.1 を参照してください。

定トルク負荷は以下のように表されます。

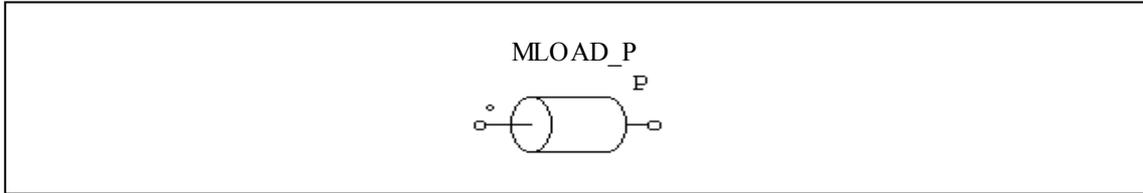
$$T_L = T_{const}$$

この場合、負荷トルクは回転方向に依存しません。

2.12.1.2 定電力負荷

定電力負荷のシンボルを以下に示します。

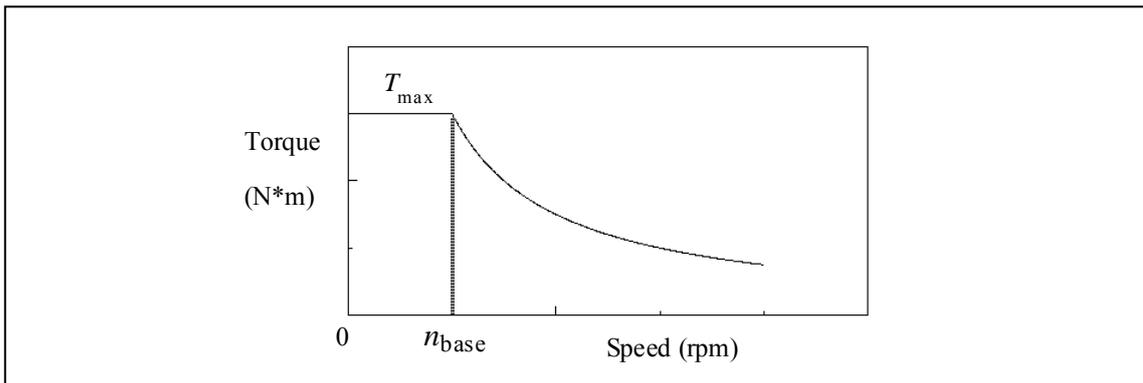
シンボル



仕様

パラメータ	機能
Maximum Torque	負荷の最大トルク T_{\max} (N*m)
Base Speed	負荷の基本速度 (n_{base})
Moment of Inertia	モータの慣性モーメント ($\text{kg}\cdot\text{m}^2$)

定電力負荷のトルク-速度曲線は以下ようになります。



機械速度が基準速度 n_{base} 以下のとき、負荷トルクは

$$T_L = T_{\max}$$

機械速度が基準速度 n_{base} 以上のとき、負荷トルクは

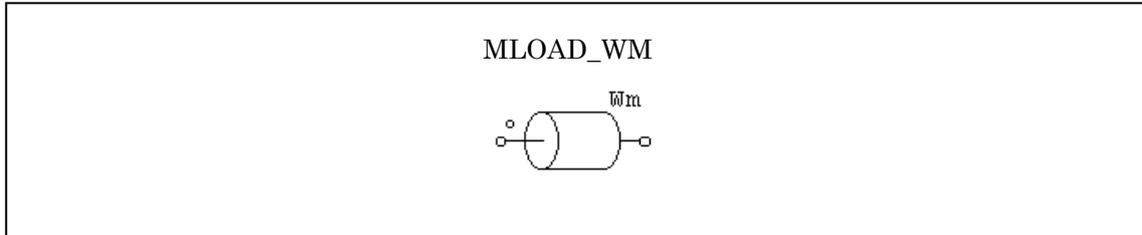
$$T_L = \frac{P}{|\omega_m|}$$

ここで、 $P = T_{\max} * \omega_{\text{base}}$ および $\omega_{\text{base}} = 2\pi * n_{\text{base}} / 60$
 機械系の回転速度 ω_m は rad/sec で表されています。

2.12.1.3 定速度負荷

定速度負荷のシンボルは以下のようになります。

シンボル



仕様

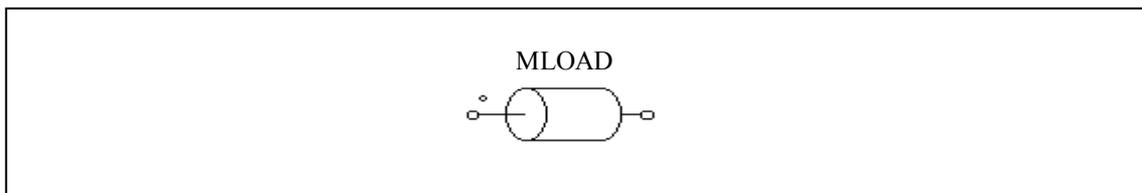
パラメータ	機能
Constant Speed (rpm)	速度定数 (rpm)
Moment of Inertia	モータの慣性モーメント (kg*m ²)

定速度負荷では機械系の回転速度を指定します。速度は速度定数に指定したとおりの一定速度になります。

2.12.1.4 一般負荷

PSIMには、その他に一般負荷が備わっています。

シンボル



仕様

パラメータ	機能
Tc	定トルク項
k ₁ (coefficient)	1次項係数
k ₂ (coefficient)	2次項係数
k ₃ (coefficient)	3次項係数
Moment of Inertia	モータの慣性モーメント (kg*m ²)

一般負荷は以下のように表されます。

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

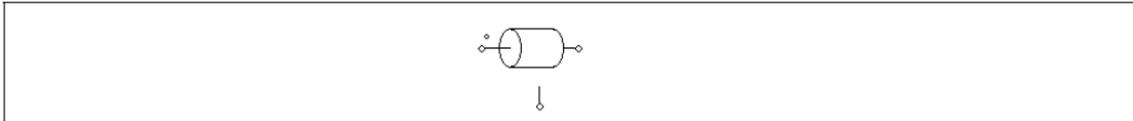
ここで、 ω_m は機械系の回転速度 rad/sec です。

一般負荷のトルクは回転方向に依存します。

2.12.1.5 外部コントロール負荷

外部コントロール負荷は、負荷量を制御ノードから任意に設定するために使用します。

シンボル



仕様

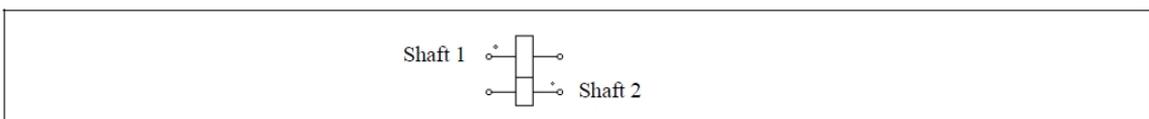
パラメータ	機能
Speed Flag	速度依存性のフラグ Flag =0 のとき、負荷は摩擦方向(回転を妨げる方向)に働きます。 Flag =1 のとき、負荷は回転方向に依存しません。
Moment of Inertia	負荷の慣性モーメント、kg*m ²

機械負荷の大きさは制御ノードの電圧値（1V は 1 N*m に対応する）によって定義されます。このノードは制御回路として取扱います。

2.12.2 ギアボックス

ギアボックスのシンボルを以下に示します。

シンボル



仕様

パラメータ	機能
Gear Ratio	ギア比 a
Shaft 1 Master/Slave Flag	シャフト 1 のマスタ/スレーブフラグ
Shaft 2 Master/Slave Flag	シャフト 2 のマスタ/スレーブフラグ

より大きいドット付きシャフトがシャフト 1 です。

1 段目と 2 段目のギアの歯数をそれぞれ n_1 および n_2 とすると、ギア比は $a = n_1 / n_2$ となります。

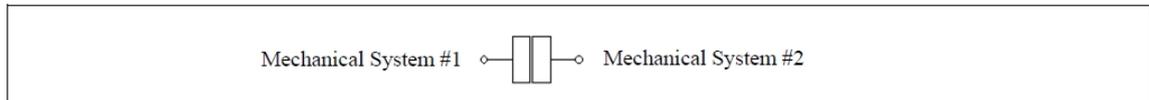
ここで、2つのギアの半径、トルク、回転速度をそれぞれ $r_1, r_2, T_1, T_2, \omega_1, \omega_2$ とすると、次の関係が成り立ちます。

$$T_1 / T_2 = r_1 / r_2 = \omega_2 / \omega_1 = a$$

2.12.3 メカニカル・カップリング・ブロック

2つの機械システムをつなぐためにメカニカル・カップリング・ブロックが使用されます。

シンボル

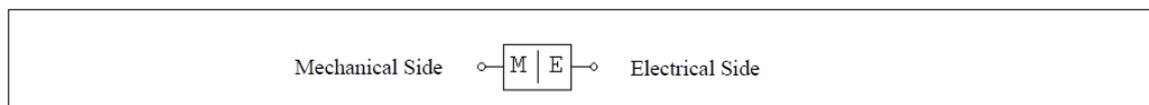


このブロックは両方の機械システムがマスターモードのデバイスを持っている状況で使用されます。機械システムを形成するためにはそれらを一緒に接続する必要がありますが、PSIMの機械システムでは1台のマスターモードのデバイスしか持つことができません。この場合中間にメカニカル・カップリング・ブロックを追加することで解決できます。

2.12.4 機械系－電気系インタフェース・ブロック

このブロックを使うことにより機械系の内部等価回路にアクセスできます。

シンボル



仕様

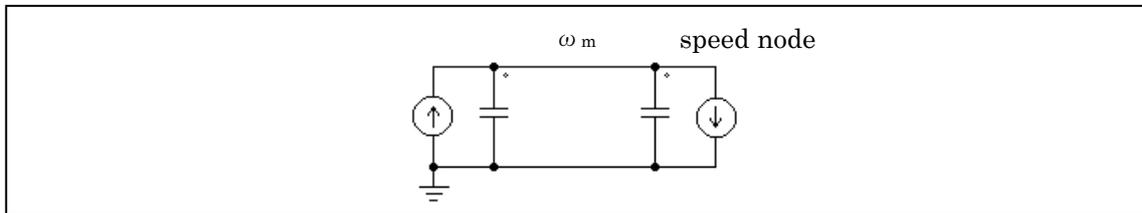
パラメータ	機能
Master/Slave Flag	マスタ/スレーブモードのフラグ (1: マスタ, 0: スレーブ)

モータと同じように機械系－電気系インタフェース・ブロックはマスタ/スレーブモードのフラグにより基準方向を指定することができます。インタフェース・ブロックをマスタに設定すると、基準方向は機械軸に沿って機械系の端子から残りの機械系の素子に向かっての方向になります。ひとつの機械系の中ではインタフェース・ブロックはひとつだけマスタに設定できます。マスタ/スレーブフラグの詳細については2.9.1を参照してください。

駆動系は1台のモータ（電気トルク T_{em} および慣性モーメント J_1 ）と機械負荷（負荷トルク T_{load} と慣性モーメント J_2 ）からなると仮定すると機械系の運動方程式は次のようになります。

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

ここで、 ω_m は機械軸の回転速度です。PSIM では、この方程式は以下に示す等価回路でモデル化されています。)



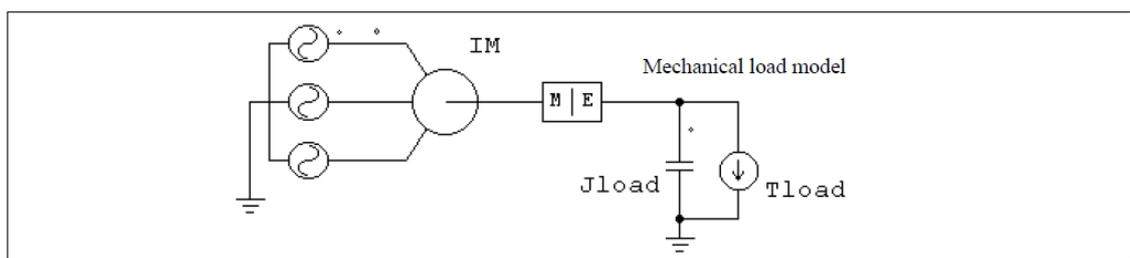
この回路では、2つの電流源を T_{em} と T_{load} の値に設定し、キャパシタで J_1 と J_2 を表現しています。また、ノード対地電圧（速度ノード電圧）は機械系の回転速度 ω_m に対応します。これは $C = J_1 + J_2, V = \omega_m, i = T_{em} - T_{load}$ とすればキャパシタの方程式 $C \cdot dV/dt = i$ に等価です。

PSIM では、モータと機械負荷を表す等価回路はキャパシタに基づく回路モデルを使っています。機械系—電気系インタフェース・ブロックを使えば、この機械系の等価回路の内部にアクセスすることができます。機械系—電気系インタフェース・ブロックの機械系側（MECH の表示）が機械軸に接続されると、電気系側（ELEC の表示）は機械系等価回路の速度ノードに対応します。そこでどのような電気回路もこの端子に接続することができます。

このブロックを使えば、PSIM 内蔵モータや機械負荷をユーザが作成した負荷やモータにつなぐことができます。

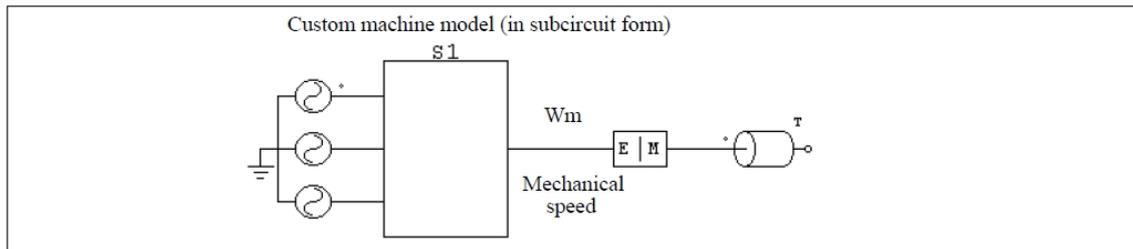
例：カスタム負荷モデルと誘導機

以下の図にユーザが定義した機械負荷を機械系—電気系インタフェース・ブロックで誘導機モデルに接続した例を示します。すでに示したように、電気系側の端子電圧は機械軸の回転速度を示します。また、この端子に接続したキャパシタは慣性モーメントを示します。



例：カスタムモータモデルと定トルク機械負荷

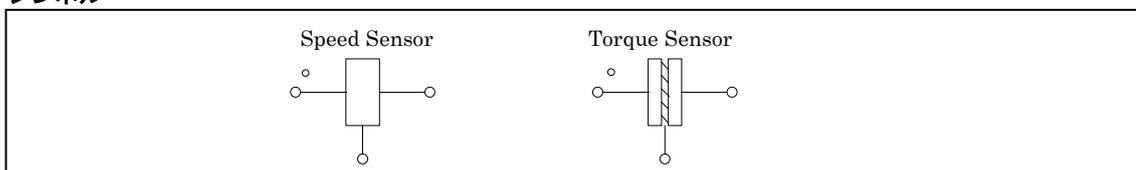
上の例とは逆に、ユーザが作成したカスタムモータモデルを PSIM ライブラリの機械負荷につなげることができます。下の図にその例を示します。カスタムモータは上記キャパシタの等価回路の考え方を使って、機械系の運動方程式をモデル化しています。機械系の回転速度を表すノードは機械系—電気系インタフェース・ブロックの電気系側に接続します。



2.12.5 速度・トルクセンサ

機械系の速度とトルクを測定するために、速度センサとトルクセンサが用意されています。

シンボル

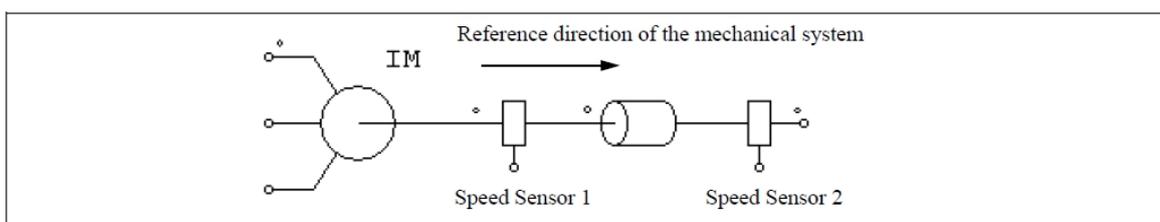


仕様

パラメータ	機能
Gain	センサのゲイン

機械系の基準方向がドットの側からセンサに入っていく方向であるとすると、センサは基準方向に沿っていると言えます。基準方向の詳細は 2.9.1 を参照してください。なお、速度センサの出力は rpm です。

以下の図で速度センサ 1 は基準方向に沿っており、速度センサ 2 は基準方向の逆向きに付いています。モータが正方向に回転している場合、速度センサ 1 の出力は正の値(positive)を読み出し、速度センサ 2 の出力は負の値(negative)を読み出します。

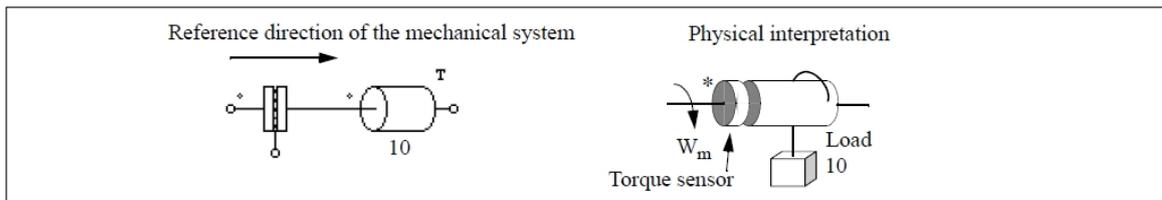


トルクセンサはセンサのドットが付いた端子とドットが付いていない端子との間のトルク差を計測します。この機能を、下図を使って説明します。

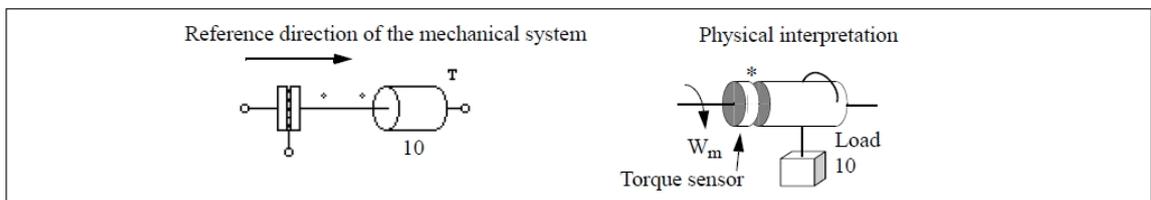
左側の図は 10 (N*m) の負荷が付いたトルクセンサで、機械系の基準方向は左から右へ向かう方向です。右図は実際のシステムに置き換えた場合を示しています。

このケースでは、正方向の回転とトルクは時計回りの方向になります。センサのドットの付いた方向は左側にあり、負荷はシャフトを減速させる働きをします（負荷トルクは反時計回りの方向になります）。

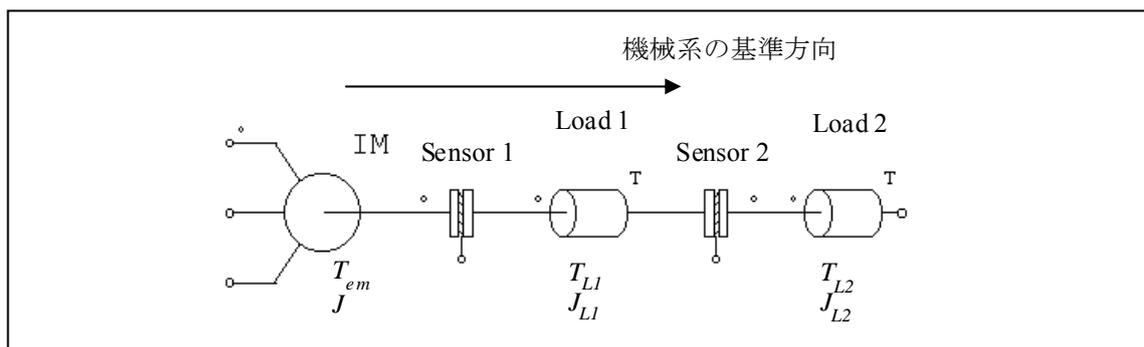
トルクセンサは、センサのドット側が接続されていない場合ドットの付いていない側のトルクテンションを計測し、センサ出力が正の値であればトルクが速度の基準方向と反対であることを意味します。したがって、下図の例では、正方向の回転は時計回りであり、負荷トルクは反時計回りになります。またトルク値は10 (N*m) となります。



同様に、ドット側が接続されているとき、センサはドット側のトルク強度を計測し、回転速度方向は正方向となります。例として、以下の図のように左右が反転しているトルクセンサを考えます。ドット側が接続されているとき、負荷トルクはセンサのドット側に回転速度方向と逆向きにかかります。トルクセンサ出力は-10 (N*m) になります。



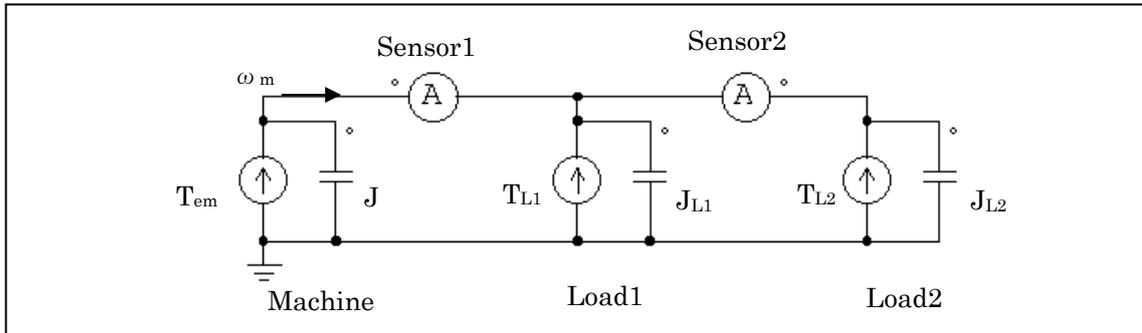
トルクセンサが機械系の等価回路でどのようにモデリングされるかを、以下の図を使って説明します。



この機械系は1つのモータ、2つのトルクセンサ、2つの負荷からなります。モータと負荷のトルク(T)と慣性モーメント(J)は図に示すとおりです。基準方向はこの場合、左から右です。この系の運動方程式は次のようになります。

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

この運動方程式を等価回路で表すと以下のようになります。



等価回路のノード電圧が機械系の速度 ω_m に対応します。左側の電流プローブがトルクセンサ 1(Sensor 1)の出力に相当します。同様に、右側の電流プローブはトルクセンサ 2 (Sensor 2) の計測値を示します。右側の電流プローブは極性が反転していることに注意してください（右から左）。これはトルクセンサ 2 が機械系の基準方向と反対の方向を計測しているからです。

等価回路により機械出力が伝わる様子もわかります。電圧と電流の積を取ることにより、トルクと回転速度の積をとったことと同等になり、これは機械出力に相当します。出力が正ならば、回転速度と同じ方向に伝わります。

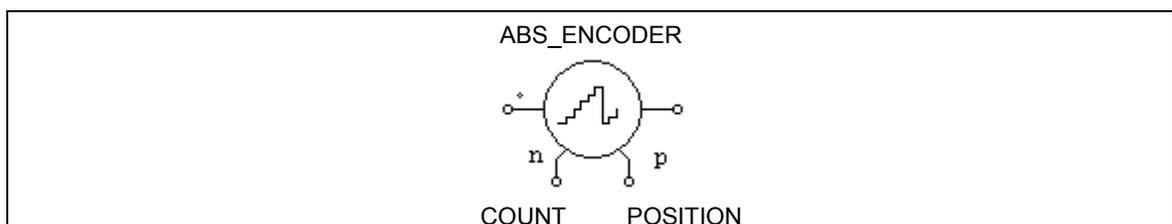
2.12.6 位置センサ

位置センサとして、アブソリュートエンコーダ、インクリメンタルエンコーダ、レゾルバとホールセンサの 4 種類が提供されています。これらのセンサは速度センサ、トルクセンサと同様に、機械的なシャフトに接続されていて、その出力は電圧信号（電圧プローブを接続すると波形表示できます）であり、また制御信号になります。

2.12.6.1 アブソリュートエンコーダ

アブソリュートエンコーダは 360° （機械角）の範囲におけるシャフトの位置を出力する位置センサです。

シンボル



仕様

パラメータ	機能
Initial Position (deg.)	シャフト位置の初期値 (deg.)
No. of Bit of Resolution	分解能のビット数 N

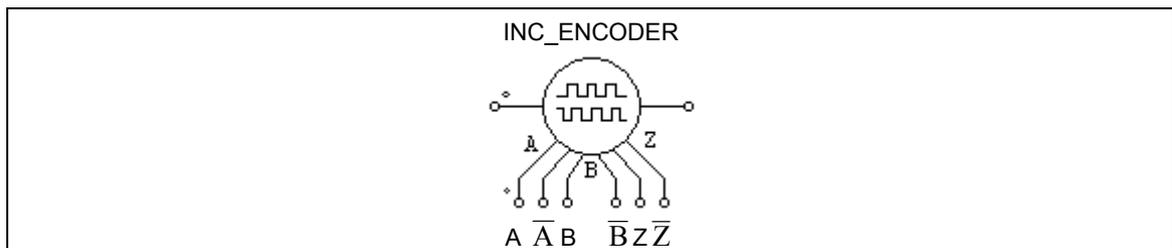
エンコーダの出力の分解能はビット数 N によって決定されます。エンコーダからはカウント数(範囲は 0 から 2^N-1)と、位置としての機械角(範囲は 0 から 360°)が出力されます。

アブソリュートエンコーダを使った例として永久磁石同期機ドライブシステムのサンプルファイル “Absolute Encoder PMSM Drive.sch” に示しています。

2.12.6.2 インクリメンタルエンコーダ

インクリメンタルエンコーダは速度、位相角とシャフトの方向を示すパルス信号を出力します。

シンボル



仕様

パラメータ	機能
Initial Position (deg.)	シャフト位置の初期値 (deg.)
No. of Lines	1回転あたりのパルス数

出力信号 A, B はパルス出力であり、A が B に対して 90°位相が進んでいます。また、Z は基準位置信号であり、通常は 0 が出力され、シャフト角度が 0 になったときに 1 が出力されます。

\bar{A} , \bar{B} , \bar{Z} はそれぞれ A, B, Z を反転した信号です。

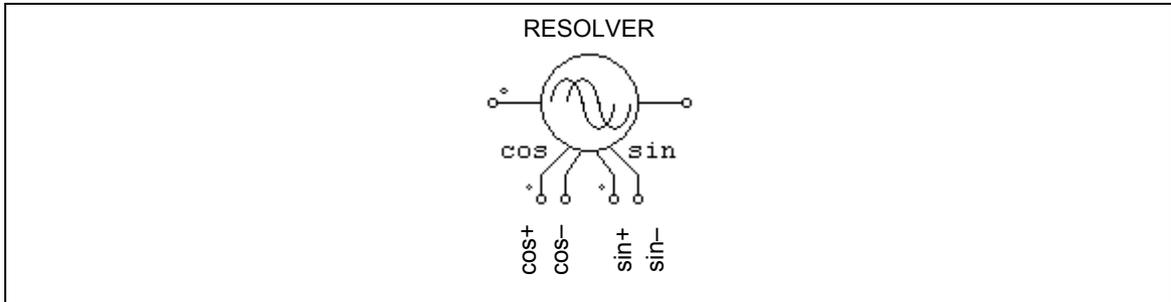
インクリメンタルエンコーダを使った例として誘導機ドライブシステムのサンプルファイルを “Incremental Encoder INDM Drive.sch” に示しています。

2.12.6.3 レゾルバ

レゾルバは主に一つの回転子巻線と二つの固定子巻線を持つ回転式変圧器です。二つの固定子巻線は COS 巻線と SIN 巻線と呼ばれ、互いに 90°離れています。

シャフトが回転すると、COS 巻線と SIN 巻線の出力電圧はシャフトの角度に対して cos と sin の関数として変化します。

シンボル



仕様

パラメータ	機能
Initial Position (deg.)	シャフト位置の初期値 (deg.)
No. of Pole	レゾルバの極数

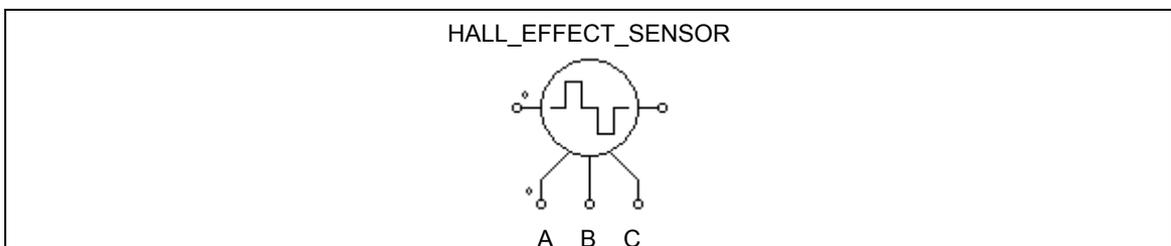
レゾルバは4つの出力、それぞれが cos+、cos-(cos+の反転)、sin+、sin-(sin+の反転)を持っています。すべての出力のピーク値は1です。

レゾルバを使った PMSM ドライブシステムの例を、サンプルファイル“Resolver PMSM Drive.sch”に示しています。

2.12.6.4 ホールセンサ

ホールセンサはシャフトの位置に依存する三つのパルス信号を出力する位置センサです。このセンサは一セットの半導体スイッチとトリガ磁石によって構成されています。磁場の強さが閾値より高い時、或いは低い時に半導体スイッチは開放或いは導通の状態に変わります。

シンボル



仕様

パラメータ	機能
Initial Position (deg.)	シャフト位置の初期値 (deg.)
No. of Pole	センサの極数

ホールセンサは三つのロジック信号 A,B と C を出力します。それぞれの信号は電気角的に 120°離れています。

ホールセンサはブラシレス直流機に内蔵されているホールセンサと同等です。（一般的なホールセンサ IC の出力と異なり、出力を直接ブラシレス直流機の駆動に使用できるようになっています。）

ホールセンサを使ったブラシレス直流機ドライブシステムの例として、サンプルファイル"Hall-Effect Sensor BDCM_Drive.sch" を提供しています。

2.13 Renewable Energy Models

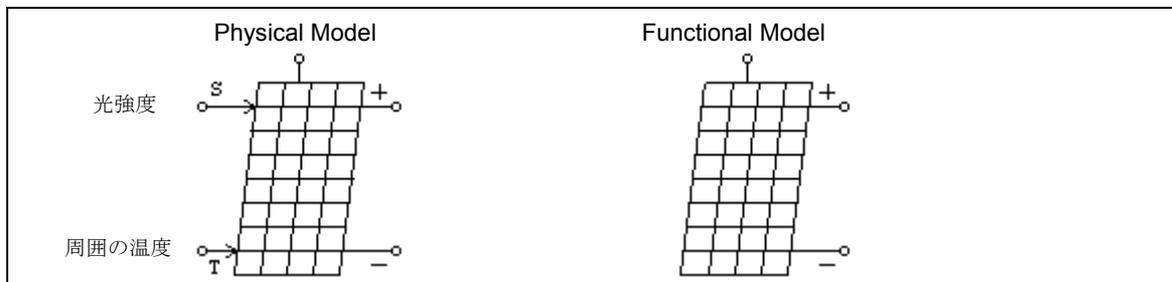
ここには新エネルギーアプリケーションに関連する要素が含まれています。

2.13.1 太陽電池モジュール

詳細モデル(physical model)と簡易モデル(functional model)の2種類の太陽電池モジュール用モデルが用意されています。

これら2つの違いは、詳細モデルは太陽電池の実際の挙動をより正確にシミュレーションでき、光強度や周囲の温度を考慮することができる点にあります。一方で簡易モデルは光強度や温度変化を考慮することはできませんが最小限のパラメータ入力のみでシミュレーションできるため容易に使うことができます。

シンボル



詳細モデル (Physical Model) の仕様

パラメータ	機能
Number of Cells N_s	太陽電池のセル数 N_s 。太陽電池モジュールは N_s 個の連続した太陽電池から構成されます。
Standard Light Intensity S_0	試験環境下での光強度 S_0 (W/m^2)。この値は通常 $1000 W/m^2$ です。(データシートによる)
Ref. Temperature T_{ref}	試験環境下の温度 T_{ref} ($^{\circ}C$)
Series Resistance R_s	各太陽電池の直列抵抗 R_s (Ω)
Shunt Resistance R_{sh}	各太陽電池の並列抵抗 R_{sh} (Ω)
Short Circuit Current I_{sc0}	試験温度 T_{ref} 時の各太陽電池の短絡回路電流 I_{sc0} (A)
Saturation Current I_{s0}	試験温度 T_{ref} 時の各太陽電池のダイオード飽和電流 I_{s0} (A)
Band Energy E_g	各太陽電池のバンドエネルギー (eV)。結晶シリコンでは約 1.12、アモルファスシリコンでは約 1.75 です。
Ideality Factor A	各太陽電池の理想係数 A 。放射係数ともよばれ、結晶シリコンでは約 2、アモルファスシリコンでは 2 より小さい値になります。
Temperature Coefficient C_t	温度係数 C_t ($A/^{\circ}C$ または A/K)
Coefficient k_s	光強度が太陽電池の温度に影響を及ぼす係数 k_s

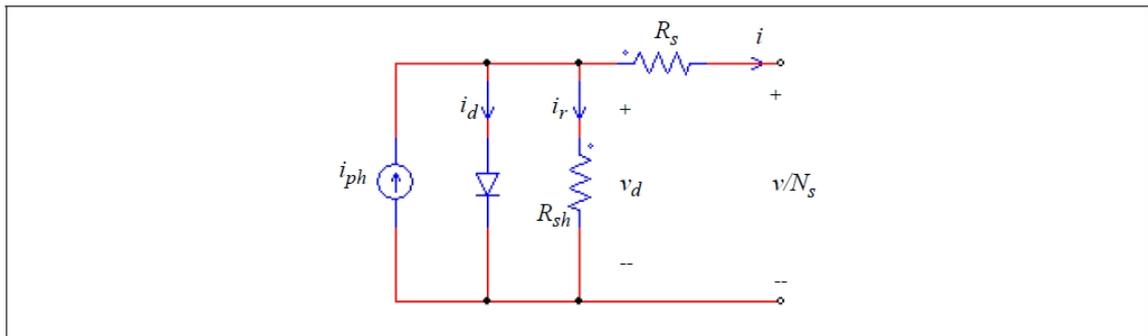
簡易モデルの仕様 (Functional Model)

パラメータ	機能
Open Circuit Voltage V_{oc}	太陽電池の端子を開放したときの測定電圧 (V)
Short Circuit Current I_{sc}	太陽電池の端子を短絡したときの測定電流 (A)
Maximum Power Voltage V_m	最大出力電力時の太陽電池モジュールの端子電圧 (V)
Maximum Power Current I_m	最大出力電力時の太陽電池モジュールの端子電流 (A)

太陽電池モジュール詳細モデルのシンボルで、"+"と"-"のノードはそれぞれ正と負の端子を表しています。"S"で示したノードは光強度入力 (W/m²) を、"T"で示したノードは周囲温度入力 (°C) を表します。上部のノードは、この動作条件で与えられる理論上の最大電力 (W) になります。正負端子のノードがパワー回路用なのに対し、その他のノードはすべて制御回路用になります。

同様に、太陽電池モジュール簡易モデルのシンボルでは"+"と"-"のノードはそれぞれ正と負の端子を表します。上部のノードはこの動作条件で与えられる理論上の最大電力になります。

N_s個のセルからなる太陽電池モジュールの、1つのセルの詳細モデルの等価回路を以下の図に示します。



太陽電池（セル）の方程式は以下の通りです。

$$i = i_{ph} - i_d - i_r$$

$$i_{ph} = I_{sc0} \cdot \frac{S}{S_0} + C_t \cdot (T - T_{ref})$$

$$i_d = I_0 \cdot \left(e^{\frac{qv_d}{AkT}} - 1 \right)$$

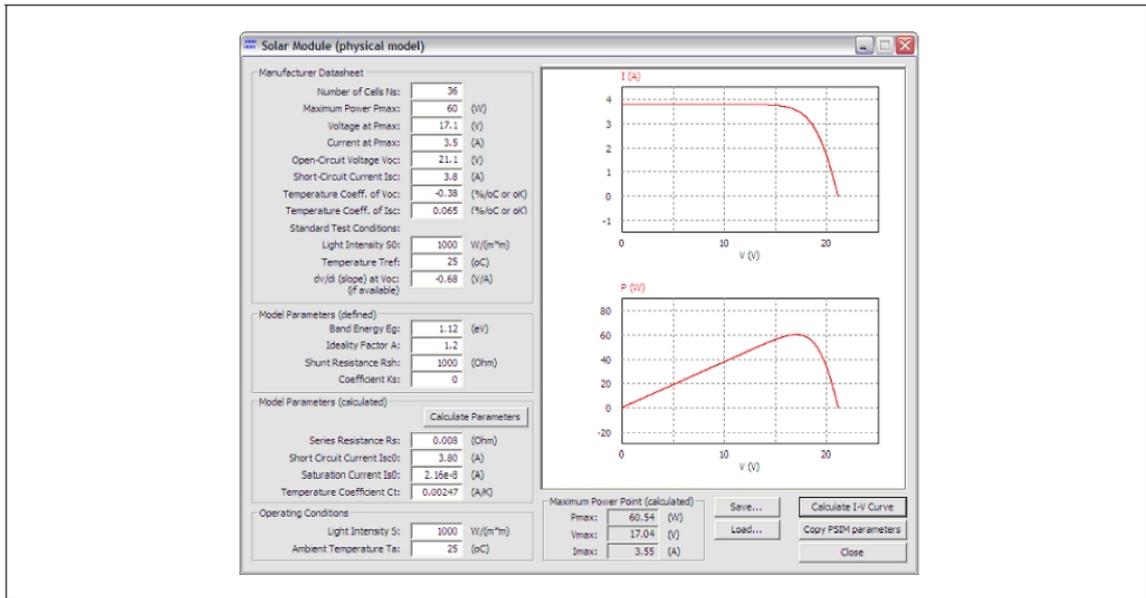
$$I_0 = I_{s0} \cdot \left(\frac{T}{T_{ref}} \right)^3 \cdot e^{\frac{qE_g}{Ak} \left(\frac{1}{T_{ref}} - \frac{1}{T} \right)}$$

$$i_r = \frac{v_d}{R_{sh}}$$

$$T = T_a + k_s \cdot S$$

ここで q : 電子の電荷 ($q = 1.6 \times 10^{-19} \text{C}$)、 k : ボルツマン定数 ($k = 1.3806505 \times 10^{-23}$)、 S : 光強度、 T_a : 周囲の温度、 v : 太陽電池モジュールの端子電圧、 i : 太陽電池モジュールの+端子から流出する電流を表します。

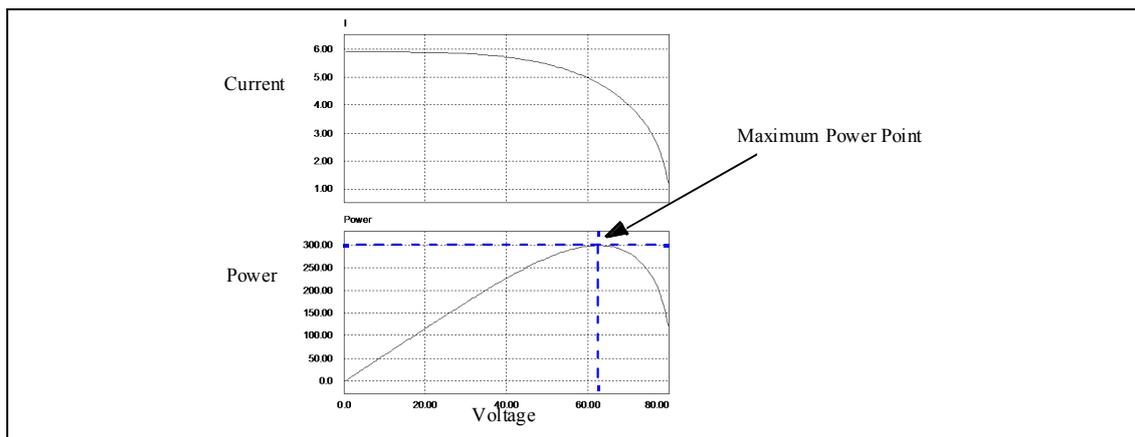
詳細モデルのパラメータの一部はデータシートから得ることができますが、残りは実験データから決定します。詳細モデルのパラメータをデータシートから得るためのツールとして、Utilitiesメニュー内に *Solar Cell (Physical Model)* が用意されています。このツールの画面は以下の通りです。



このツールでパラメータを決定する方法については、PSIM インストールフォルダの doc フォルダ内にある *Tutorial- Solar Module (Physical model).pdf* (英文) をご参照ください。

一方、太陽電池モジュールの簡易モデルでは以下の 4 つのパラメータ（開放電圧、短絡電流、最大電力時の電圧・電流）を用いて生成した i-v 特性に基づいて太陽電池を模擬します。この 4 つのパラメータはデータシートから得られます。

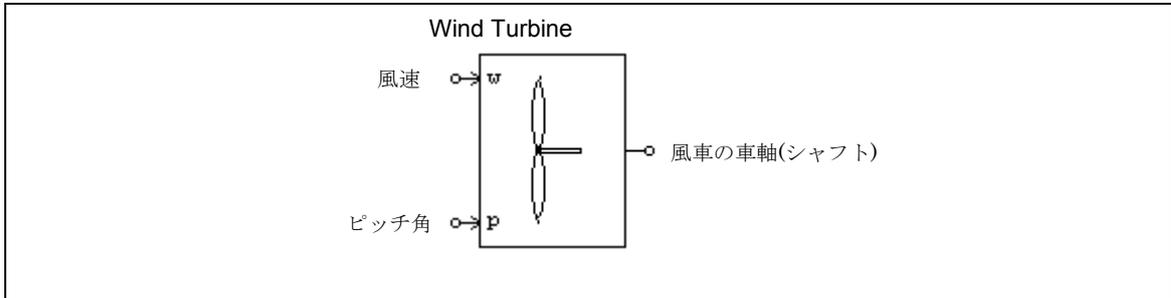
以下の図は、太陽電池モジュールの電圧 vs 電流、および電圧 vs 電力のグラフを示しています。図に示すように、電圧が増加するのに従って太陽電池モジュールの出力電力が増加し、あるポイントで出力電力が最大になることがわかります。異なった運転条件の下で最大出力点(Maximum PowerPoint Tracking、または MPPT)を追従するために多くの制御方式が提案されました。いくつかの MPPT の例が PSIM サンプルにあらかじめ用意されています。



2.13.2 風車

風車のシンボルとパラメータは以下の通りです。

シンボル



仕様

パラメータ	機能
Nominal Output Power	ピッチ角が0°時の風車の最大出力電力 (W) このモデルはピッチ角が0°の時に最も高い電力を出力できるよう設計されています。この値は発電機の定格出力を超えない値に設定して下さい。
Base Wind Speed	風車の最大出力電力を発生させる基本風速 (m/s)
Base Rotational Speed	風車の最大出力電力を発生させる基本回転数 (rpm)
Initial Rotational Speed	風車の初期回転数 (rpm)
Moment of Inertia	風車羽根の慣性モーメント (kg*m ²)
Torque Flag	風車の内部トルクを表示させるフラグ (0:非表示、1:表示)
Master/Slave Flag	接続された機械系のマスタ/スレーブフラグ (0:スレーブ、1:マスタ)

このシンボル中で"w"のノードは風速 (m/s) を表し、"p"のノードは羽根のピッチ角 (deg) を表します。

これらのノードはいずれも制御回路用です。

風車によって生成される電力は、以下の方程式で表されます。

$$P = \frac{1}{2} \cdot A \cdot v_{wind}^3 \cdot \rho \cdot C_p$$

ここで A は回転羽根の面積、 v_{wind} は風速、 ρ は空気密度、 C_p は出力係数です。出力係数 C_p は周速比 λ と羽根のピッチ角 β の関数で、以下のように表されます。

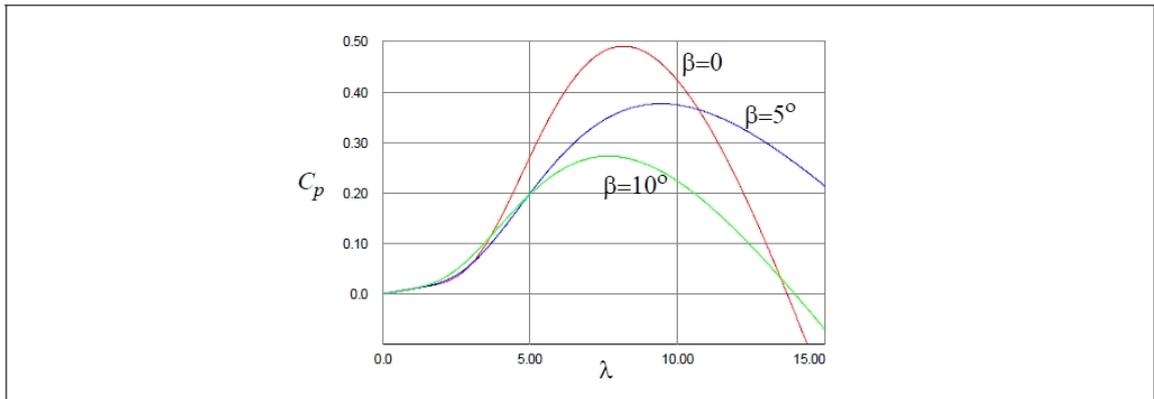
$$C_p = c_1 \cdot (c_2 - c_3\beta - c_4\beta^x - c_5) \cdot e^{-c_6} + c_7$$

ここで $c_1=0.5$ 、 $c_2=116 \cdot \lambda'$ 、 $c_3=0.4$ 、 $c_4=0$ 、 $c_5=5$ 、 $c_6=21 \cdot \lambda'$ 、 $c_7=0.014 \cdot \lambda$ 、また

$$\lambda' = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}$$

となります。

この出力係数 C_p と周速比 λ 、羽根のピッチ角 β との関係は以下のグラフに表されます。



この図は周速比 λ が 8.18 の時に出力係数 C_p が最大値 0.49 をとることを示しています。 C_p が最大値のときの C_p と λ の値を公称値とするため、この図より $C_{p_nom}=0.49$ 、 $\lambda_{nom}=8.18$ と決定されます。風車を制御する方法の一つは、公称値で（または公称値付近で）周速比を維持することであり、このとき風車が生成する電力は最大値となります。

第3章 制御回路素子

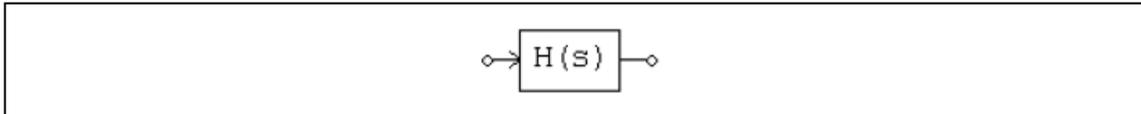
3.1 伝達関数ブロック

伝達関数ブロックは以下のように多項式で表現します。

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

伝達関数ブロックには次の2種類があります：一つは初期値0を持つもの（s-domain Transfer Function）でもう一つは任意の初期値を持つもの（s-domain Transfer Function (initial value)）です。

シンボル



仕様

パラメータ	機能
Order n	伝達関数の次数
Gain	伝達関数のゲイン k
Coeff. B _n ... B ₀	伝達関数の分子の係数
Coeff. A _n ... A ₀	伝達関数の分母の係数
Initial Values x _n ... x ₁	状態変数 x _n ... x ₁ の初期値

入力 U(s)と出力 Y(s)の関係を Y(s)=U(s)*U(s)のように表すと、周波数領域の表現を以下のような微分方程式に変換することができます。

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

出力は以下となります。

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

状態変数の初期値 x_n ... x₁ は TFCTN1 の設定時に入力します。

例.

ここでは、次の2次の伝達関数を考えます。

$$G(s) = 15 \cdot \frac{400 \cdot e^3}{s^2 + 1200 \cdot s + 400 \cdot e^3}$$

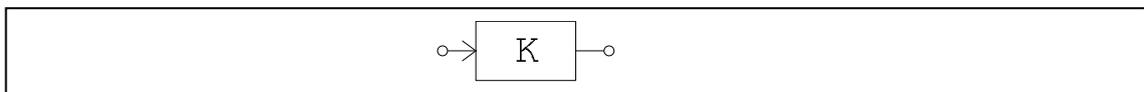
PSIM では、パラメータを以下のように指定します。

次数 n	2
ゲイン k	1.5
係数 Bn ... Bo	0, 0, 400.e3
係数 An ... Ao	1, 1200., 400.e3

3.1.1 比例制御器

比例(P)制御器の出力は入力にゲインを掛けた値になります。

シンボル



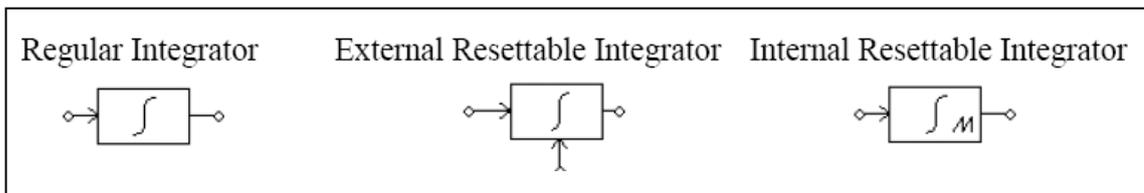
仕様

パラメータ	機能
Gain	伝達関数のゲイン K

3.1.2 積分器

積分器には3種類あります:通常の積分器、リセット機能付き外部積分器、もうひとつはリセット機能付き内部積分器です。

シンボル



仕様

積分器の場合：

パラメータ	機能
Time Constant	積分器の時定数 T(sec)
Initial Output Value	出力の初期値

リセット付き外部積分器の場合：

パラメータ	機能
Time Constant	積分器の時定数 T(sec)
Initial Output Value	出力の初期値
Reset Flag	リセットフラグ (0 : エッジリセット ; 1 : レベルリセット)

リセット付き内部積分器の場合：

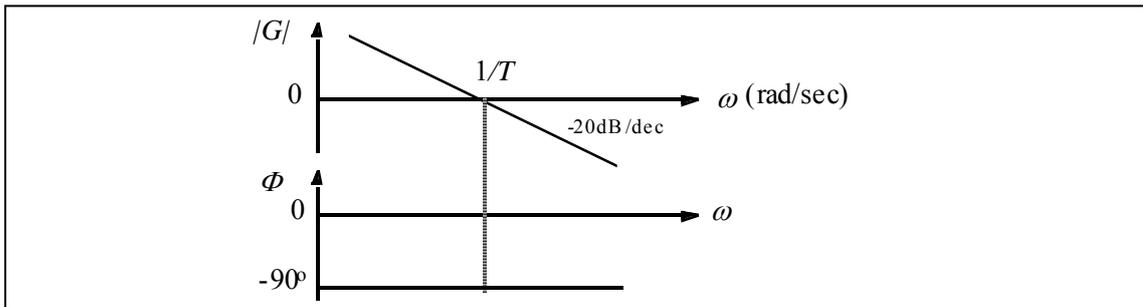
パラメータ	機能
Time Constant	積分器の時定数 T(sec)
Initial Output Value	出力の初期値

Lower Output Limit	出力の下限値
Upper Output Limit	出力の上限値

積分器の伝達関数は、以下の通りです。

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

振幅|G|と位相φのボード線図は以下の通りです。



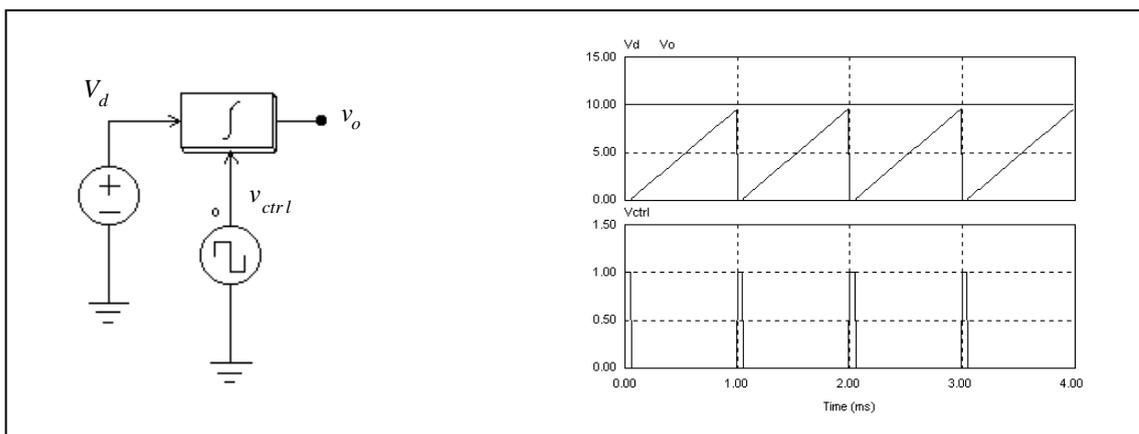
リセット付き外部積分器の出力は制御信号によりリセットすることができます。エッジリセット(リセットフラグ=0)の場合、制御信号の立ち上がりエッジで、積分器の出力は0にリセットされます。レベルリセット(リセットフラグ=1)の場合、制御信号をHigh(1)にすると、積分器の出力は0にリセットされます。

リセット付き内部積分器は出力が下限値か上限値のどちらかに達するとき、出力が0にリセットされます。これは、エッジリセット付き外部積分器と同じ動作をしますがこの場合外部のリセット回路を設定する必要はありません。

積分結果が過大にならないように、積分器の出力にはリミッタを接続する必要があります。また、PSIMの仕様として、積分器とリミッタを接続したとき、リミッタの出力だけでなく、積分器の出力自体もリミットされます。ご注意ください。(下図参照)PI制御器も同様の仕様になっています。

例.

リセット付き積分器の使用例を以下に示します。



上の図で、積分器の入力は直流です。積分器の制御信号にパルス波形を入力し、積分器の出力を各サイクルの終わりにリセットしています。

3.1.3 微分器

微分器の伝達関数は以下になります。

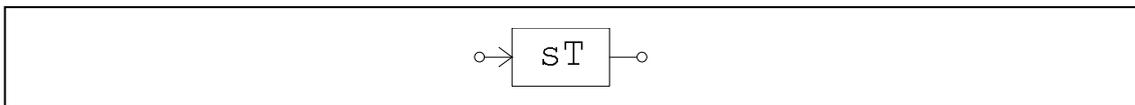
$$G(s) = sT$$

微分は以下のように計算しています。

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

ここで、 Δt はシミュレーションタイムステップ、 $v_{in}(t)$ と $v_{in}(t - \Delta t)$ はそれぞれ現在時刻の入力と1ステップ前の入力です。

シンボル



仕様

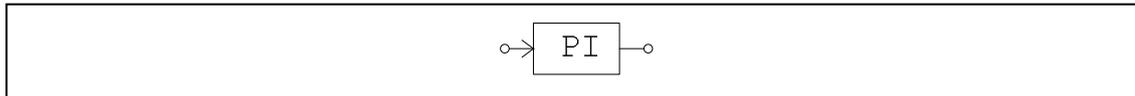
パラメータ	機能
Time Constant	微分器の時定数 T(sec)

入力が急変すると出力がスパイク状になるので、微分器の前にローパス・フィルタを置くことをお勧めします。

3.1.4 比例積分 (PI) 制御器

比例積分 (PI) 制御器は以下のように定義されます。

シンボル



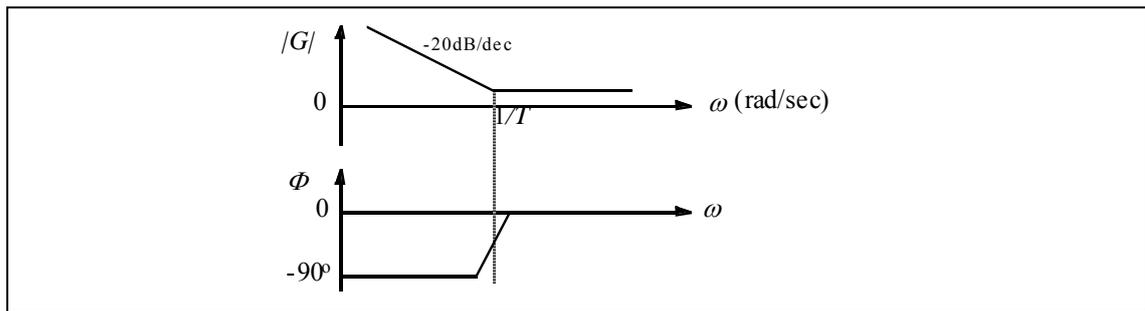
仕様

パラメータ	機能
Gain	伝達関数のゲイン K
Time Constant	PI 制御器の時定数 T(sec)

比例積分 (PI) 制御器の伝達関数は以下の通りです。

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

振幅|G|と位相 ϕ のボード線図は以下の通りです。

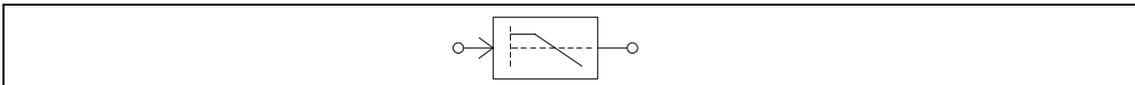


積分結果が過大になるのを防ぐために、PI制御器のあとに必ずリミッタを使用してください。

3.1.5 単極制御器

単極制御器は一つの極を持つ比例制御器です。

シンボル



仕様

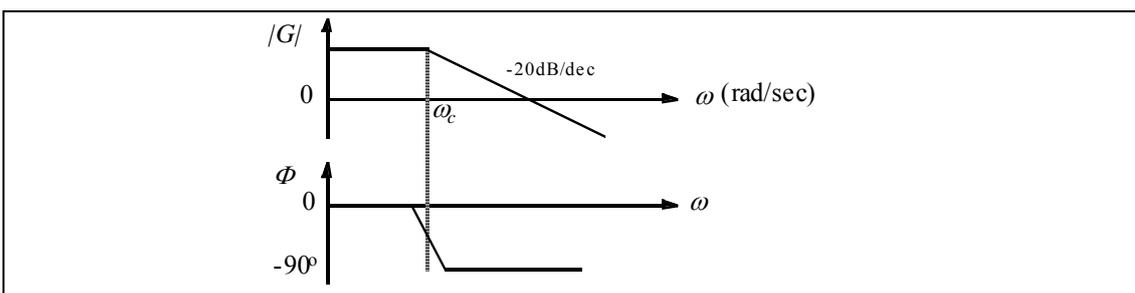
パラメータ	機能
Gain	制御器のゲイン k
Pole Frequency	極の周波数 f_c (Hz)

単極の伝達関数は以下の通り定義されます。

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

ここで $\omega_c = 2\pi f_c$ です。

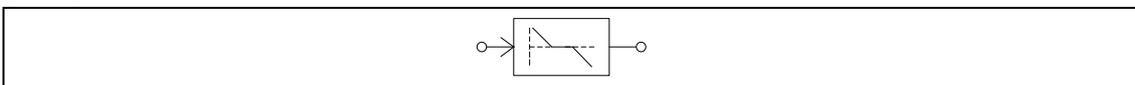
振幅 $|G|$ と位相 Φ のボード線図は以下の図の通りです。



3.1.6 二極比例積分 (PI) 制御器

二極比例積分制御器 (modified PI controller) は極を持った PI で構成されています。

シンボル



仕様

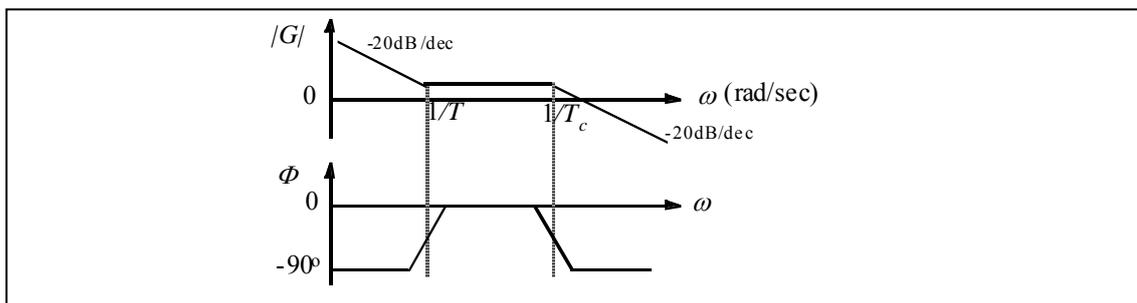
パラメータ	機能
Gain	伝達関数のゲイン k
Time Constant	PI 制御器の時定数 T (sec)
Pole Frequency	極の周波数 f_c (Hz)

この伝達関数は以下の通り定義されます。

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

ここで $T_c=1/\omega_c$, $\omega_c=2\pi f_c$ です。

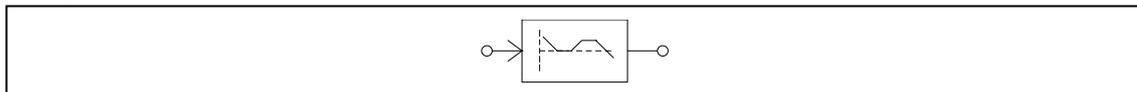
振幅|G|と位相 Φ のボード線図は以下の図の通りです。



3.1.7 Type-3 制御器

Type-3 制御器は2つの零点と2つの極で構成されています。

シンボル



仕様

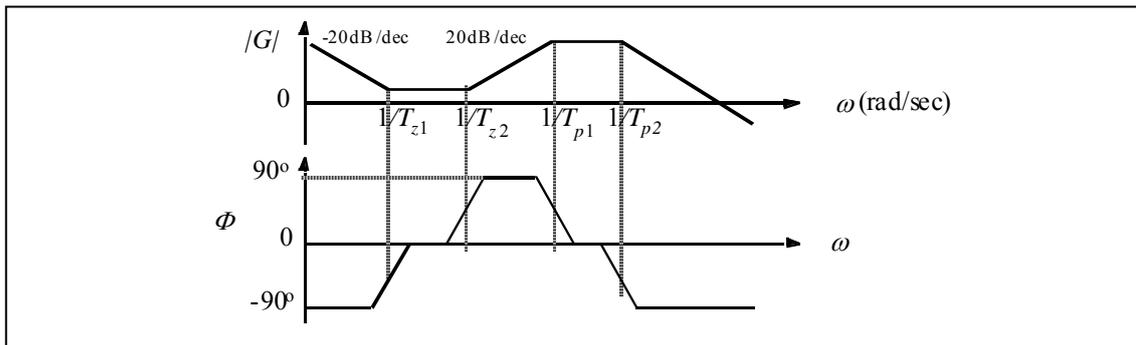
パラメータ	機能
Gain	PI 制御器のゲイン k
Frequency fz1	1つ目の零点の周波数 f_{z1} (Hz)
Frequency fz2	2つ目の零点の周波数 f_{z2} (Hz)
Frequency fp1	1つ目の極の周波数 f_{p1} (Hz)
Frequency fp2	2つ目の極の周波数 f_{p2} (Hz)

この伝達関数は以下の通り定義されます。

$$G(s) = k \cdot \frac{1+sT_{z1}}{sT_{z1}} \cdot \frac{1+sT_{z2}}{(1+sT_{p1}) \cdot (1+sT_{p2})}$$

ここで $T_{z1} = \frac{1}{2\pi f_{z1}}$, $T_{z2} = \frac{1}{2\pi f_{z2}}$, $T_{p1} = \frac{1}{2\pi f_{p1}}$, $T_{p2} = \frac{1}{2\pi f_{p2}}$ です。

振幅|G|と位相 Φ のボード線図は以下の図の通りです。



3.1.8 組み込みフィルタ・ブロック

PSIMには次の4つの2次フィルタと1つの1次フィルタがあります。

2次ローパス・フィルタ

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

2次ハイパス・フィルタ

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

2次バンドパス・フィルタ

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

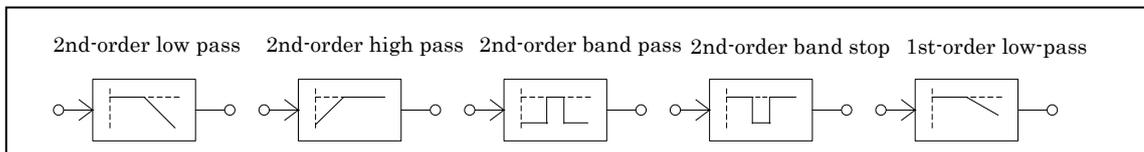
2次バンド阻止・フィルタ

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

1次ローパス・フィルタ

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

シンボル



仕様 (2次フィルタ) :

パラメータ	機能
Gain	伝達関数のゲイン k
Dumping Ratio	減衰係数 ξ (減衰比)
Cut-off Frequency	ローパスおよびハイパス・フィルタのカット・オフ周波数 ($f_c = \omega_c / 2\pi$, Hz)
Center Frequency	バンドパスおよびバンド阻止フィルタの中心周波数 ($f_o = \omega_o / 2\pi$, Hz)

Passing Band; Stopping Band	バンドパスおよびバンド阻止フィルタの周波数バンド幅 ($f_b = B/2\pi$, Hz)
--------------------------------	--

仕様 (1次フィルタ) :

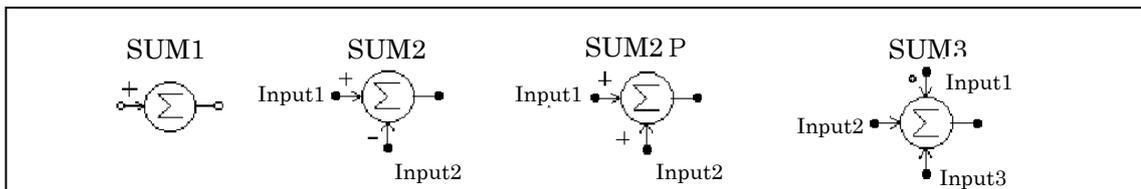
パラメータ	機能
Gain	伝達関数のゲイン k
Cut-off Frequency	ローパス・フィルタのカット・オフ周波数 ($f_c = \omega_c/2\pi$, Hz)

3.2 数値演算関数ブロック

3.2.1 加算器

1入力加算器 (SUM1) または2入力加算器 (SUM2 および SUM2P) の入力はスカラーでもベクトルでも可能です。3入力の加算器はスカラー入力のみです。

シンボル



仕様

パラメータ	機能
Gain_i	i 番目の入力のゲイン k_i

3入力加算器 (SUM3) では、ドット印の付いた入力が最初の入力となります。入力がスカラーの場合は、n入力加算器の出力は次のようになります。

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

入力がベクトルにする場合は2入力加算器の出力も同じくベクトルになります。以下に定義式を示します。

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

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

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

1入力の加算器では入力がベクトルでも出力はスカラーになります。この場合、スカラー出力は入力のベクトル要素をすべて足したものです。すなわち、次のようになります。

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

3.2.2 乗算器と除算器

乗算器 (Multiplier) と除算器 (Divider) の出力はそれぞれ2つの入力の積および商です。

シンボル



除算器ではドット印のついた入力が入数となります。

乗算器の入力はベクトルでもスカラーでも可能です。2 入力ともベクトルの場合、ベクトルの次元は同じでなければなりません。

たとえば、入力を以下のようにした場合、

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

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

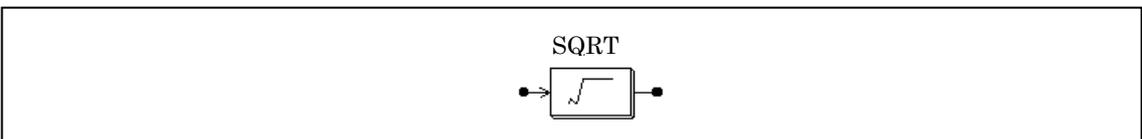
スカラーの出力は、以下のようになります。

$$V_0 = V_1 * V_2^T = a_1 * b_1 + a_2 * b_2 + \dots + a_n * b_n$$

3.2.3 平方根ブロック

入力の平方根を計算します。

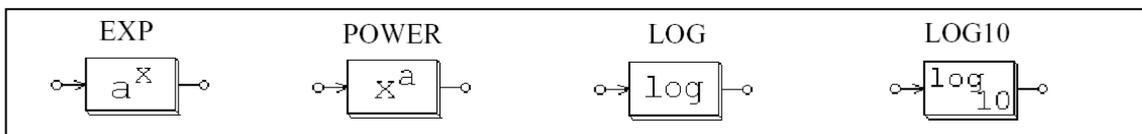
シンボル



3.2.4 指数／累乗／対数ブロック

指数 (EXP)、累乗 (POWER)、および対数 (LOG) の計算をおこないます。

シンボル



仕様

パラメータ	機能
Coefficient k_1	係数 k_1
Coefficient k_2	係数 k_2

指数関数ブロック (EXP) は、出力を次のように計算します。

$$V_o = k_1 \cdot k_2^{V_m}$$

たとえば、 $k_1 = 1$ 、 $k_2 = 2.718281828$ 、および $V_{in} = 2.5$ とすると、出力は $V_o = e^{2.5}$ となります (e は自然対数の基数)。

累乗ブロック (POWER) は出力を以下で計算します。

$$V_o = k_1 \cdot V_{in}^{k_2}$$

LOG ブロックは入力の自然対数を、LOG10 ブロックは常用対数を出力します。

3.2.5 二乗平均平方根 (RMS) ブロック

二乗平均平方根ブロックは、指定された基本周波数 (f_b) により入力の二乗平均平方根を計算します。出力は次のようになります。

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T v_m^2(t) dt}$$

ここで、 $T = 1/f_b$ 。

出力は各平均周期の始めにのみ更新されます。

シンボル



仕様

パラメータ	機能
Base Frequency	基本周波数 f_b (Hz)

3.2.6 絶対値ブロック

絶対値ブロック (ABS) は入力の絶対値を与えます。符号関数ブロック (SIGN) は入力の符号を与えます。たとえば、入力が正のときに出力は 1、入力が負のときに出力は -1 です。

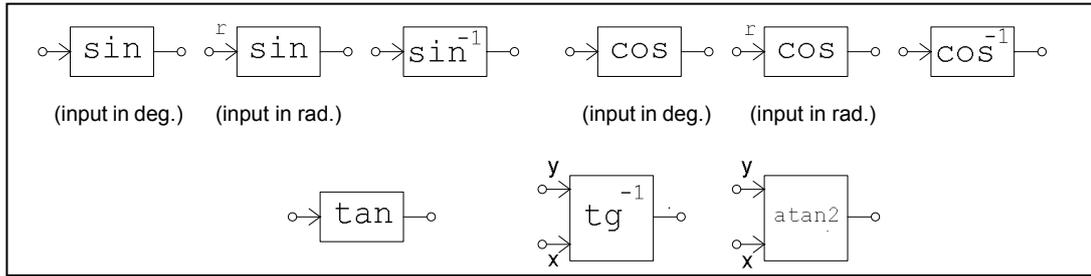
シンボル



3.2.7 三角関数ブロック

2つのサイン (sin)、2つのコサイン (cos)、アークサイン (\sin^{-1})、アークコサイン (\cos^{-1})、タンジェント (tan)、およびアークタンジェント (\tan^{-1})、アークタンジェント 2 (atan2) の9つの三角関数ブロックが利用可能です。出力は入力にそれぞれ対応する三角関数値となります。SIN、COS、TAN ブロックの場合は入力が度(deg)になり、 SIN^{-1} 、 COS^{-1} 、 TG^{-1} ブロックの場合は出力が度(deg)になります。左上に "r" が付いているサイン/コサインブロックおよび atan2 ブロックは、入力がラジアン(rad)になります。

シンボル



\tan^{-1} ブロックの出力は実数値入力 (real) と虚数値入力 (imaginary) の比で計算されます。arc tangent ブロックでは、ドットの付いた入力の実数値の入力、もうひとつの入力が虚数値です。つまり、以下のよう な式になります。

$$\theta = \tan^{-1} \left(\frac{V_{imaginary}}{V_{real}} \right)$$

3.2.8 高速フーリエ変換 (FFT) ブロック

高速フーリエ変換 (FFT) ブロックにより、入力信号の基本波成分を計算できます。

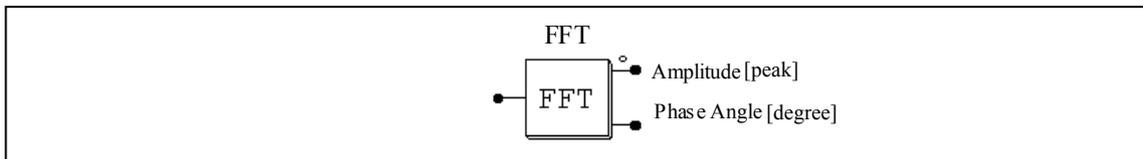
FFT の演算アルゴリズムは、radix-2/decimation-in-Frequency 方法によります。

基本周期の 1 周期間のサンプリング数は 2^N (N は自然数) となるようにします。サンプリング数の上限は 1024 です。

FFT ブロックの出力は入力の基本波の振幅 (最大値) と位相角です。出力は以下の式で計算しています。

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

シンボル



仕様

パラメータ	機能
No. of Sampling Points Fundamental Frequency	サンプリング数 N 基本波周波数 f_b (Hz)

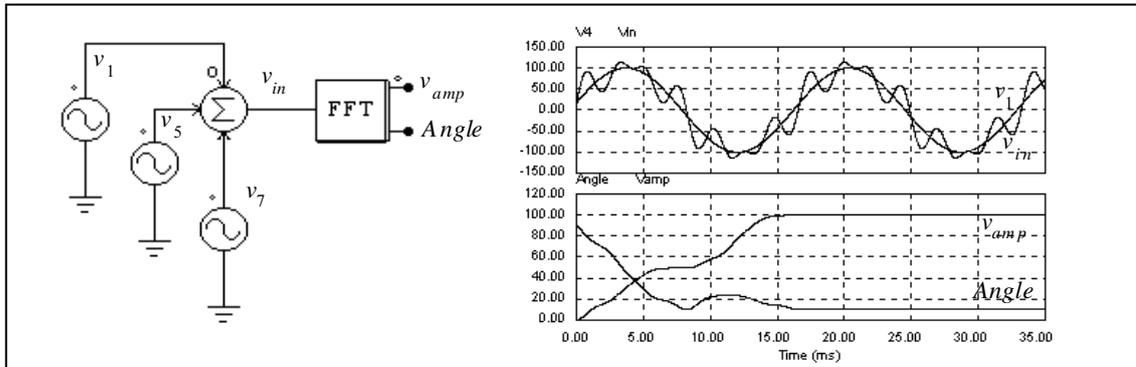
2 つの出力のうち、ドットのついたものが振幅の値です。位相は内部で、 $V_m \sin \omega t$ が位相 0 になるように調整しています。

FFT ブロックは入力信号が DC、基本波、基本波の高調波 (基本波周波数の整数倍の信号) を持つ場合のみ正しく動作します。入力信号が基本波の整数倍ではない成分を持っている場合、FFT ブロックは正しい結果を計算できません。

また、FFT ブロックは基本波成分を計算しますので、例えば入力信号が 60Hz (基本波) と 180Hz の成分を持っている場合、パラメータ「Fundamental Frequency」を 180 にすると正しい結果を計算できません。

例

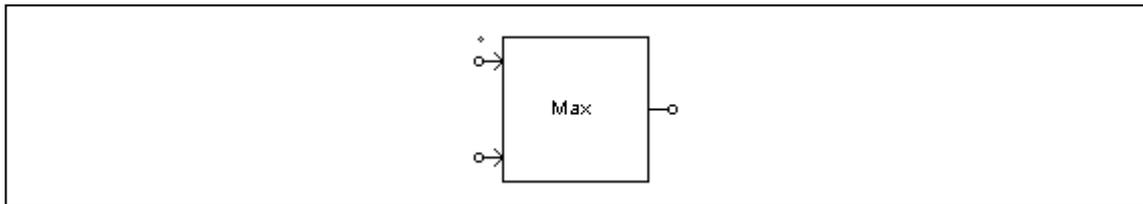
以下の回路で、入力電圧 V_{in} は基本波 V_1 (100V, 60Hz)、5次高調波 V_5 、(25V, 300Hz)、7次高調波 V_7 (25V, 420Hz) を含みます。1サイクル後、FFT の出力は、振幅 100V の定常値となります。



3.2.9 最大/最小機能ブロック

最大/最小機能ブロック (maximum/minimum function block) はそれぞれの入力で最も大きな値を持つもの、または小さな値をもつものの値を出力します。

シンボル



仕様

パラメータ	機能
Number of Inputs	入力数 N
Block Function Type	ブロックの関数型で、 <i>Maximum</i> または <i>Minimum</i> を選択。

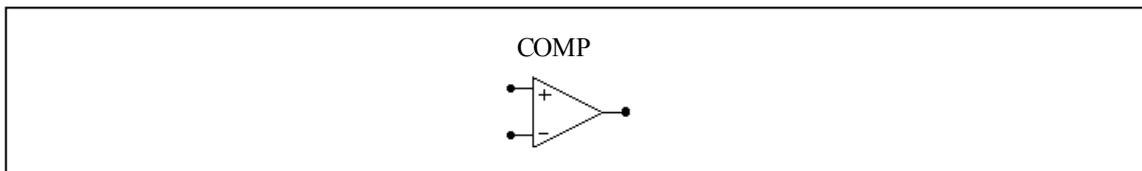
入力数 N はドロップダウンメニューで選択可能です。

3.3 その他の関数ブロック

3.3.1 比較器

正の入力が負の入力よりも大きくなると比較器は High の信号を出力します。正の信号が小さい場合、出力は 0 になります。2 つの入力が等しい場合には、出力は不定となり、前回の値を維持します。

シンボル

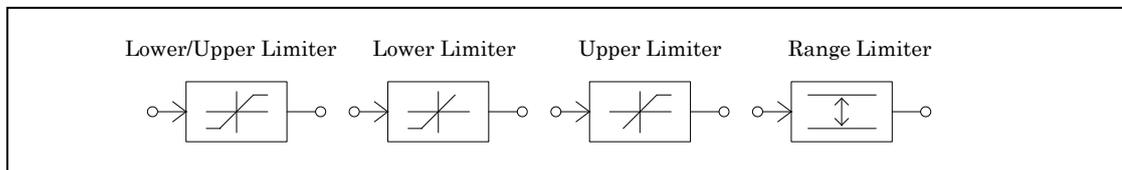


比較器のシンボルはオペアンプのシンボルと似ているので注意してください。比較器では非反転入力は左上で、反転入力が左下になります。オペアンプではこれが逆になります。

3.3.2 リミッタ

Lower/Upper リミッタ、Lower リミッタ、Upper リミッタ、Range リミッタの 4 種類のリミッタがあります。

シンボル



仕様

パラメータ	機能
Lower Limit	リミッタの下限
Upper Limit	リミッタの上限

Range リミッタ以外では、入力が指定範囲を越えるとリミッタ出力[※]は上下限に固定されます。入力が上下限の範囲内であれば、出力は入力と同じ値になります。

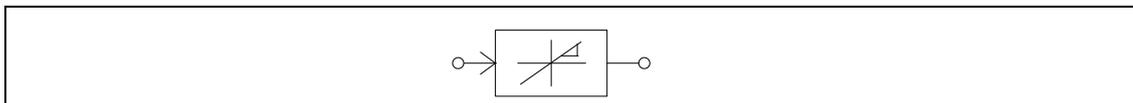
Range リミッタの出力は、上限と下限を超えない範囲に制限されます。リミッタの範囲は $V_{range} = V_{upper} - V_{lower}$ になります。入力が上下限の範囲内であれば、出力は入力と同じ値になります。出力が上限より高い場合、出力は範囲に収まるまで V_{range} の値が引かれます。出力が下限より低い場合、範囲に収まるまで V_{range} の値が足されます。

※注意：PI 素子、積分素子、フィルタ素子の出力にリミッタ素子を接続した場合、リミッタ素子の入力が指定範囲を超えるとリミッタ出力は上下限に固定されるだけでなく、リミッタ素子の入力、すなわち PI 素子等の出力が上下限に固定されます。

3.3.3 微分 (dv/dt) リミッタ

微分リミッタは入力の変化率 (dv/dt) を制限します。変化率が上下限の範囲内の場合、出力は入力と同じです。

シンボル



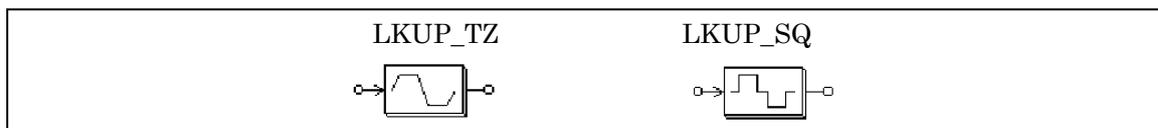
仕様

パラメータ	機能
dv/dt Limit	入力変化率の上限 (dv/dt)

3.3.4 台形および方形波ブロック

台形波形ブロック (LKUP_TZ) および方形波ブロック (LKUP_SQ) はルックアップテーブルの特殊なものです。入力と出力の関係は台形もしくは方形の波形となります。

シンボル



台形波ブロックの仕様は以下のとおりです。

仕様

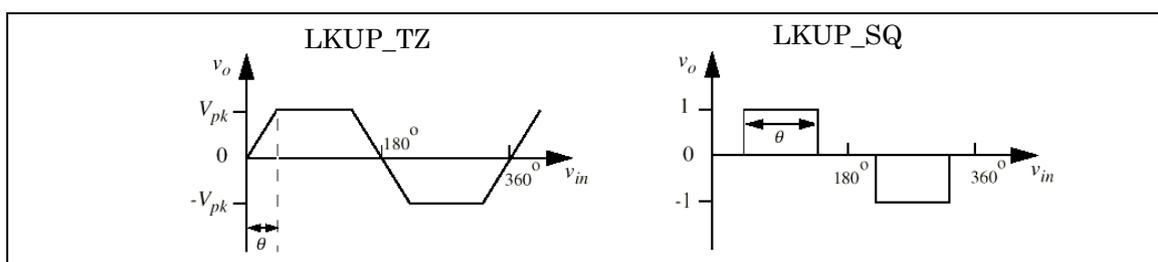
パラメータ	機能
Rising Angle theta	立ち上がり角度 θ (度)
Peak Value	波形のピーク値 V_{pk}

方形波ブロックの仕様は以下のとおりです。

仕様

パラメータ	機能
Pulse Width (deg.)	半周期のパルス幅 (度)

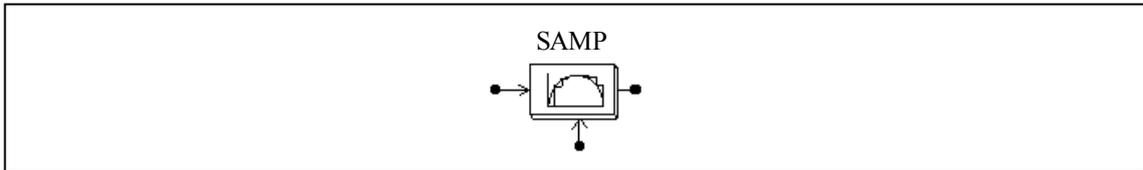
これらのブロックの波形を以下に示します。ここで、入力 v_{in} は角度 (°) で、 -360° から 360° の間でとることができます。いずれの波形も半波および 1/4 波で対称です。



3.3.5 サンプル・ホールド・ブロック

サンプル・ホールド・ブロックは、制御信号が Low から High に（0 から 1 に）変わったときに、入力をサンプルし、サンプル周期のあいだその値を保持します。

シンボル



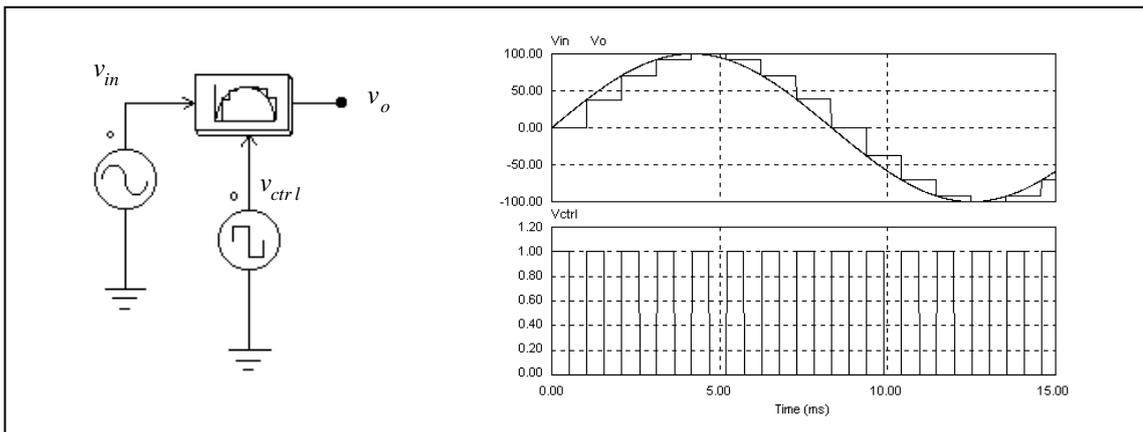
シンボルの下のノードが制御信号です。

ゼロ次ホールド・ブロック（ZOH）と違って、このブロックは連続要素として扱われ、サンプリングを外部信号で制御することができます。一方、ゼロ次ホールドでは離散要素で、サンプリングは固定かつ等間隔となります。

離散系にはゼロ次ホールド・ブロックを使って下さい。

例.

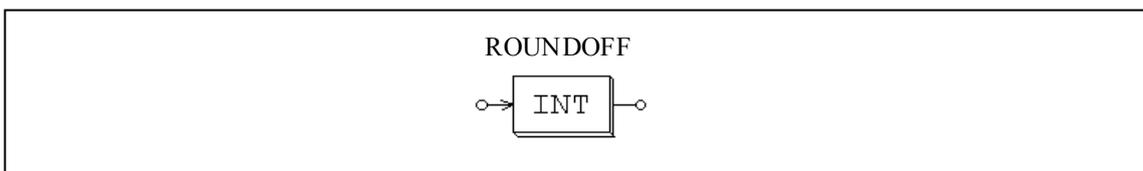
以下の例では、正弦波をサンプル化します。制御信号には振幅 1 の方形波を使っています。



3.3.6 丸めブロック

丸めブロックのシンボルを以下に示します。

シンボル



仕様

パラメータ	機能
No. of Digits	小数点以下の桁数 N
Truncation Flag	切り捨てフラグ (1: 切り捨て; 0: 四捨五入)

丸めブロックの入力を V_{in} とすると、入力はまず、以下の式で変換されます。

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

切り捨てフラグが 1 のとき、まず $V_{in,new}$ を切り捨て、これを 10^N で割ったものを出力とします。そうでない場合は、 $V_{in,new}$ に最も近い整数に等しいとおき、これを 10^N で割ったものを出力とします。

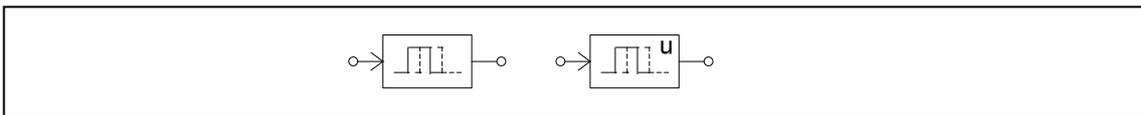
例.

- $V_{in} = 34.5678$; $N = 0$, 切り捨てフラグ = 0 のとき、出力は $V_{out} = 35$.
- $V_{in} = 34.5678$; $N = 0$, 切り捨てフラグ = 1 のとき、出力は $V_{out} = 34$.
- $V_{in} = 34.5678$; $N = 1$, 切り捨てフラグ = 1 のとき、出力は $V_{out} = 34.5$.
- $V_{in} = 34.5678$; $N = -1$, 切り捨てフラグ = 1 のとき、出力は $V_{out} = 30$.

3.3.7 時間遅れブロック

指定時間を遅らせる時間遅れブロック (Time delay block) と、1 タイムステップだけ遅らせる単位時間遅れブロック (Unit time delay block) の計 2 種類の時間遅れブロックがあります。

シンボル



時間遅れブロックは指定の時間間隔分入力信号を遅らせます。例として論理素子の伝搬遅れをモデリングできます。

単位時間遅れブロックは 1 タイムステップ分入力信号を遅らせます。シミュレーションのタイムステップが変更された場合、遅れ時間もこれに従って変更されます。

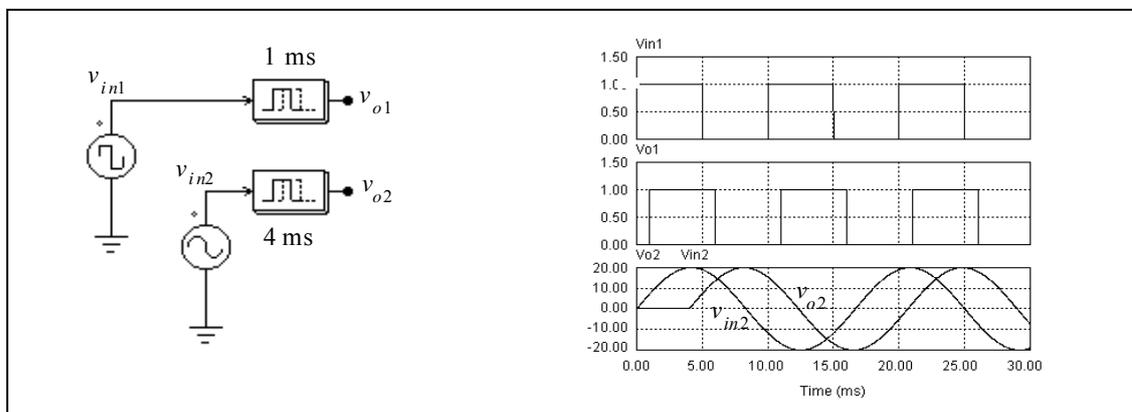
単位時間遅れブロック (Unit time delay block) と Digital Control Module の単位遅れブロック (Unit delay block) の違いは、こちらが連続要素で遅れ時間が 1 タイムステップなのに対し、単位遅れブロックは離散的な要素であり遅れ時間はサンプリング周期と同じになる点です。

仕様

パラメータ	機能
Time Delay	時間遅れ(sec)※Time Delay ブロックのみ

例.

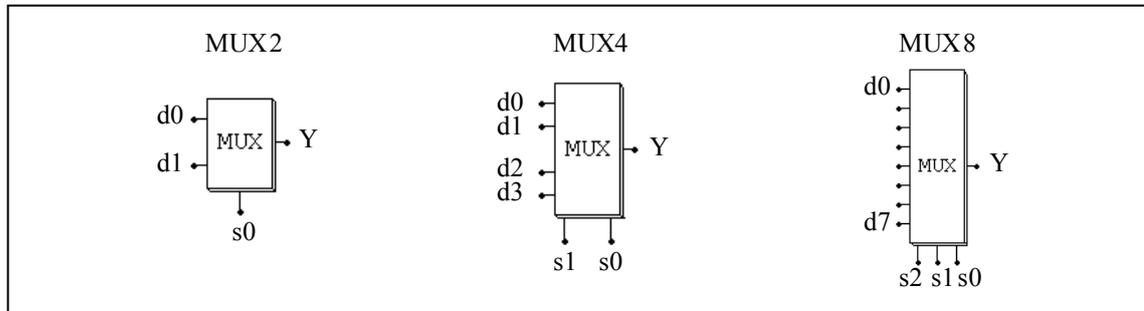
この例では、最初の時間遅れブロックは 1ms の遅れ、2 つめの時間遅れは 4ms と設定しています。この例に示すとおり、時間遅れブロックの入力はアナログでもデジタルでも可能です。



3.3.8 マルチプレクサ

マルチプレクサの出力は制御信号により選ばれた入力となります。マルチプレクサには3種類あります。2入力 (MUX2)、4入力 (MUX4)、および8入力マルチプレクサ (MUX8) です。

シンボル



シンボル図で、 $d_0 \dots d_7$ はデータ入力、 $s_0 \dots s_2$ は制御入力です。マルチプレクサの真理値表は以下のとおりです。

2入力 MUX	
s0	Y
0	d0
1	d1

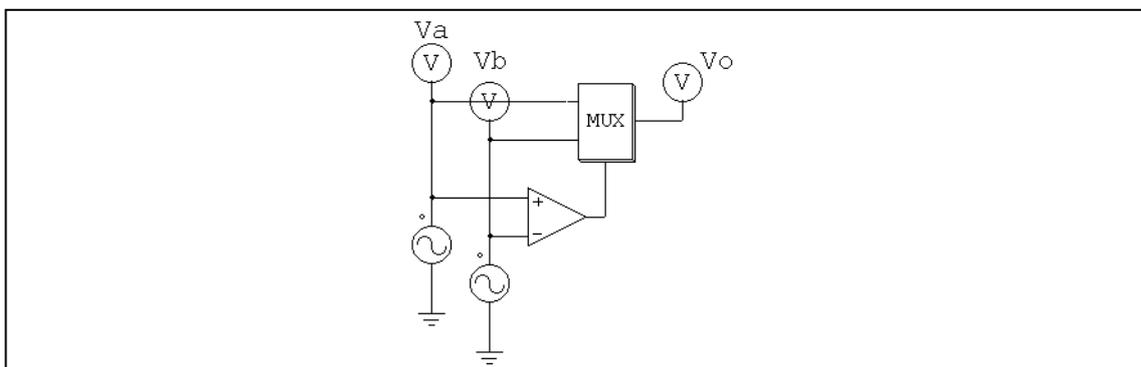
4入力 MUX		
s1	s0	Y
0	0	d0
0	1	d1
1	0	d2
1	1	d3

8入力 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

データ入力はアナログ・デジタルとも使用可能です。

例.

以下の回路は2入力のうち最大の値を選択します。入力 V_a が V_b よりも大きいときは比較器の出力は1、および $V_o = V_a$ ；そうでない場合は、 $V_o = V_b$ となります。



3.3.9 高調波歪み (THD) ブロック

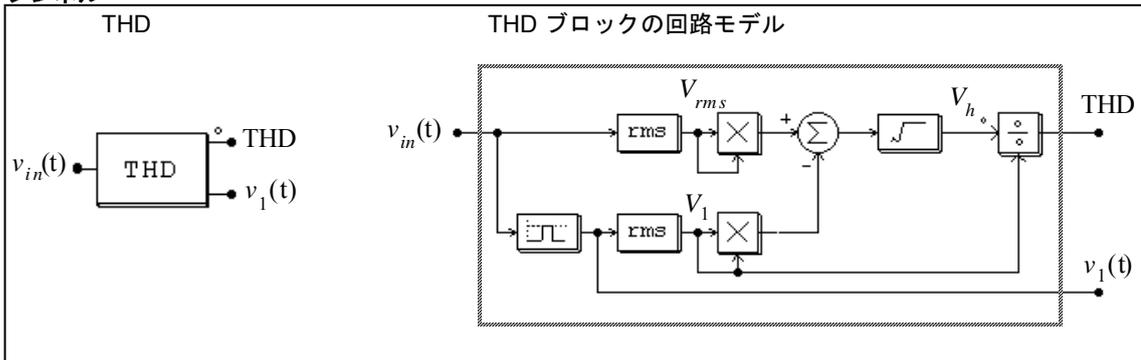
基本波成分と高調波成分を含む交流波形で、高調波ひずみは以下の式で計算できます。

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

ここで、 V_1 は基本波成分 (rms)、 V_h は高調波成分 (rms)、また V_{rms} は交流波形の全体の rms 値です。

PSIM では高調波ブロックは以下のようにモデル化されています。

シンボル



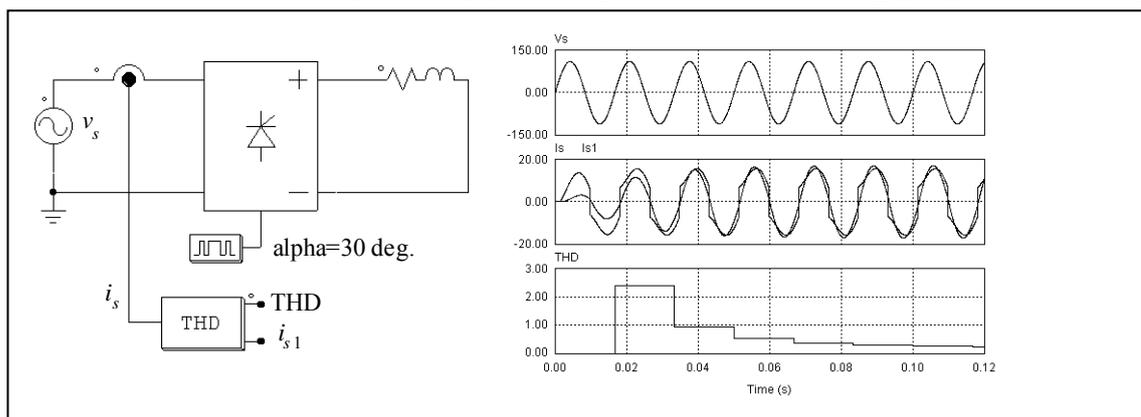
THD は基本波を取り出すのに 2 次のバンドパス・フィルタを使っているのので、中心周波数とバンド幅を指定する必要があります。

仕様

パラメータ	機能
Fundamental Frequency	入力の基本波周波数 (Hz)
Passing Band Freq.	バンドパス・フィルタのバンド幅 (Hz)

例.

以下に示す単相のサイリスタ回路では、高調波歪み (THD) ブロックを使って入力の高調波歪みを測定しています。サイリスタ・ブリッジの位相遅れは 30° とします。また、THD ブロックの基本周波数は 60Hz、バンドパス・フィルタのバンド幅は 20Hz としています。シミュレーション結果を回路図の右に示します。



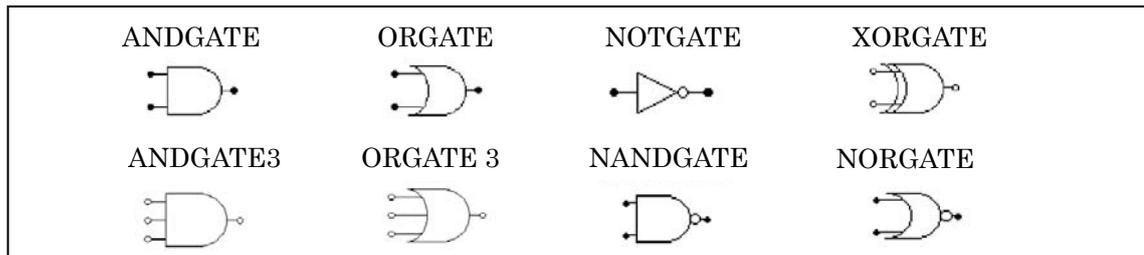
THD ブロックの出力の一方は入力電流の基本波成分 i_{s1} です。入力電圧 v_s と電流 i_{s1} の位相を比べることにより、入力の等価力率を計算することができます。この値を高調波の値 (THD) と組み合わせることにより、入力の力率を求めることができます。

3.4 デジタル素子

3.4.1 論理ゲート

基本論理ゲートには AND、OR、XORGATE (exclusiveOR) NOT、NAND および NOR があります。

シンボル



3.4.2 セット・リセット・フリップフロップ

セット・リセット・フリップフロップにはエッジ・トリガとレベル・トリガの2種類があります。

仕様

パラメータ	機能
Trigger Flag	トリガ・フラグ (0: エッジ・トリガ; 1: レベル・トリガ)

エッジ・トリガ・フリップフロップは立ち上がり時のみ状態が変わります。エッジ・トリガ・フリップフロップの真理値表を以下に示します。

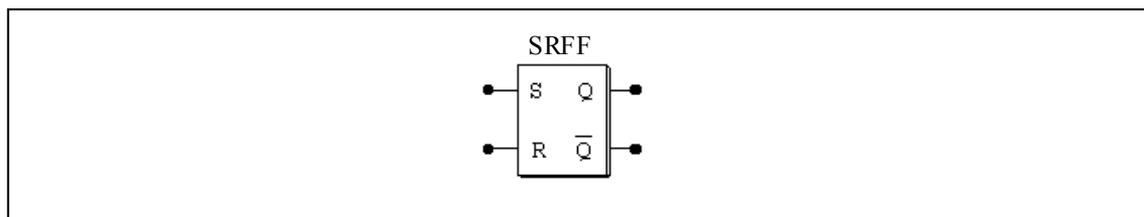
S	R	Q	\overline{Qn}
0	0	不変	
0	↑	0	1
↑	0	1	0
↑	↑	不使用	

一方、レベル・トリガ・フリップフロップは入力レベルにより状態が変わります。レベル・トリガ・フリップフロップの真理値表を以下に示します。

セット・リセット・フリップフロップの真理値表を以下に示します。

S	R	Q	\overline{Qn}
0	0	不変	
0	1	0	1
1	0	1	0
1	1	不使用	

シンボル



3.4.3 J-K フリップフロップ

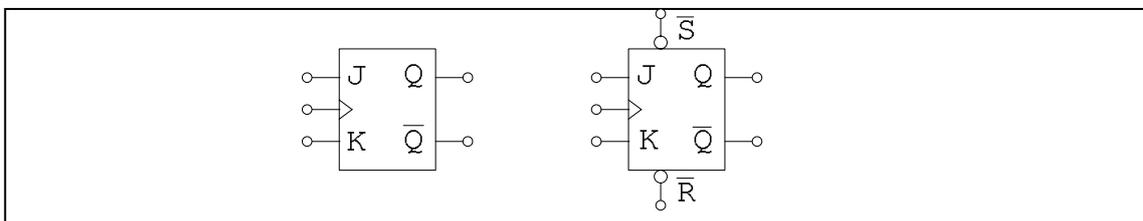
J-K フリップフロップはクロック入力の立ち上がりでトリガされます。

J-K フリップフロップの真理値表を以下に示します。

S	R	J	K	Clock	Q	\overline{Qn}
0	1	x	x	x	1	0
1	0	x	x	x	0	1
0	0	x	x	x	0	0
0	0	0	0	↑	不変	
0	1	0	1	↑	0	1
1	0	1	0	↑	1	0
1	1	1	1	↑	トグル	

X:

シンボル



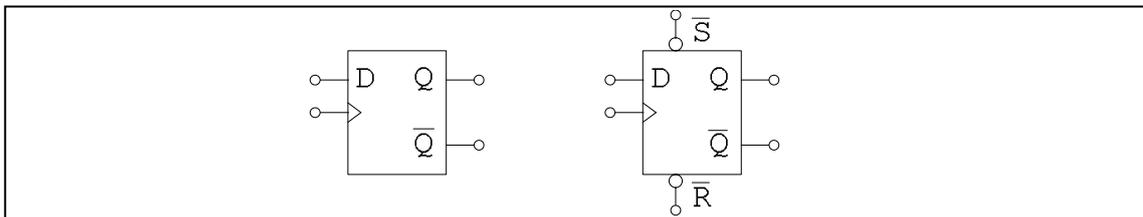
3.4.4 D フリップフロップ

D フリップフロップはクロック入力の立ち上がりでトリガされます。

D フリップフロップの真理値表を以下に示します。

S	R	D	クロック	Q	Qn
0	1	x	X	1	0
1	0	x	x	0	1
0	0	x	x	0	0
1	1	0	↑	0	1
1	1	1	↑	1	0

シンボル

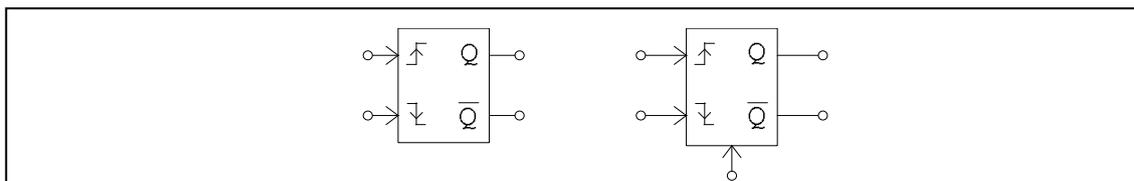


3.4.5 単安定マルチバイブレータ

単安定マルチバイブレータは入力信号の立ち上がり（または立ち下がり）でトリガされます。マルチバイブレータにより、指定された幅のパルスを生じることができます。

出力パルス幅は固定、または外部入力により可変です。可変出力マルチバイブレータは制御付きマルチバイブレータ (MONOC) と呼び、固定出力型と区別します。パルス幅(sec)は制御入力により決まります。

シンボル



仕様

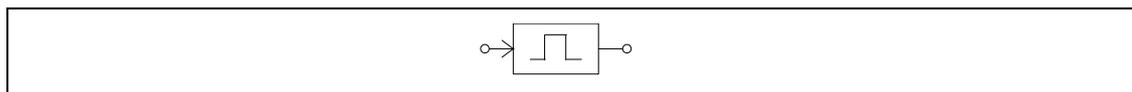
パラメータ	機能
Pulse Width	オン時間パルス幅(sec)

制御付きマルチバイブレータではシンボルの下側からの入力が制御信号になります。

3.4.6 パルス幅カウンタ

パルス幅カウンタはパルスの幅を測ります。カウンタは入力の立ち上がり信号により起動します。入力の立ち下がりを検出して、パルスの幅(sec)を測ります。次の立ち下がりを検出するまで、カウンタは前回の値を保持します。

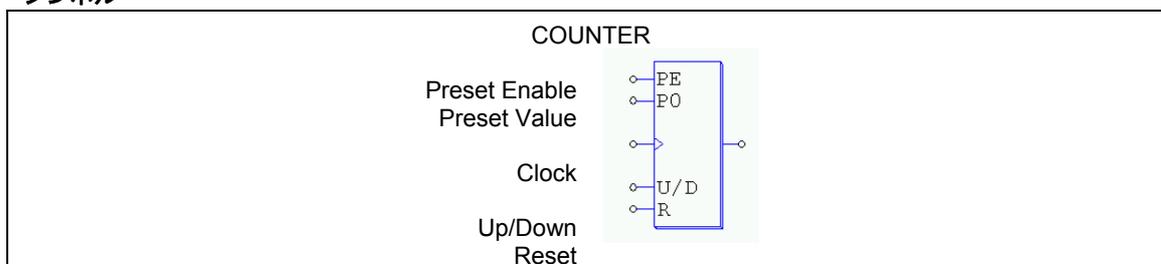
シンボル



3.4.7 Up/Down カウンタ

Up/down カウンタはクロックの立ち上がりエッジのたびに1を増大するか減少します。

シンボル



仕様

パラメータ	機能
No. of Bits	ビット N の数字

Up/Down の入力が0のとき、カウンタは減少します。入力が1の時、カウンタは増大します。
 Reset 入力は High(1)のときカウンタを0にリセットします。
 Preset Enable 入力は High のときカウンタを予め設定された値にセットします。

以下はカウンタの真理表です。

Up/Down	Preset Enable	Reset	Clock	動作
x	0	0	x	カウントしない
1	0	0	↑	カウントアップ
0	0	0	↑	カウントダウン

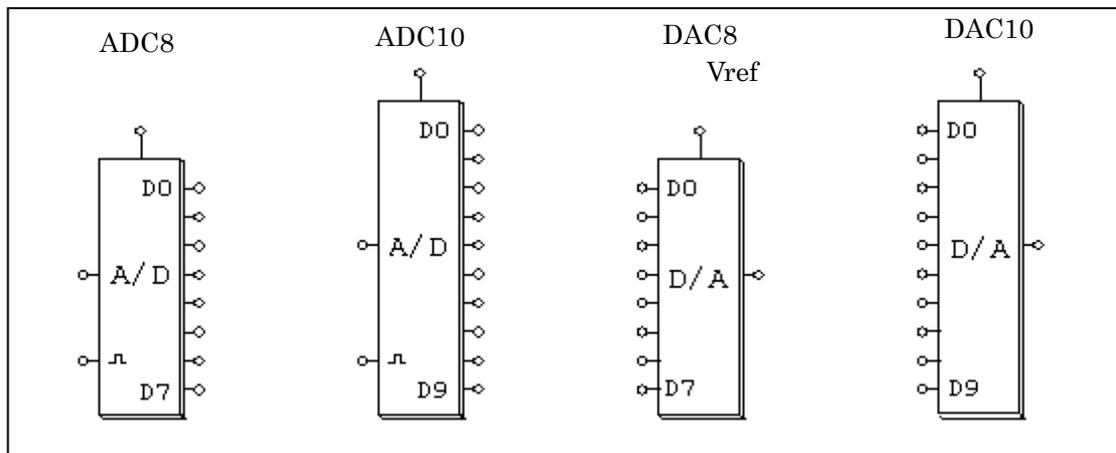
x	1	0	x	プリセット
x	x	1	x	リセット

x: Do not care

3.4.8 A/D および D/A 変換器

A/D 変換器はアナログ入力信号をデジタルへ、また D/A 変換器はデジタル信号をアナログへ変換します。8ビットと10ビットの変換器があります。

シンボル



ビット数を N とすると A/D 変換器の入力は次の式で計算されます。

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

たとえば、 $V_{ref} = 5V$ 、 $V_{in} = 3.2V$ 、 $N = 8\text{bits}$ とすると、 $V_o = 256/5 \cdot 3.2 = 163.84 = 10100011$ (バイナリ) となります。また、D/A 変換器の出力は以下の式で計算されます。

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

たとえば、 $V_{ref} = 5V$ 、 $V_{in} = 10100011$ (バイナリ)、 $N = 8\text{bits}$ とすると、 $V_o = 163/256 \cdot 5 = 3.1836V$ となります。

3.5 デジタル制御モジュール

デジタル制御モジュールを標準の PSIM プログラムにアドオンとして追加することで、ゼロ次ホールド、 z 領域伝達関数、デジタル・フィルタなどといったデジタル制御機能が利用可能になります。連続時間の s 領域回路に比べて z 領域の回路は離散型です。したがって、計算は離散サンプル点でのみ行われ、サンプリングの間は計算が実行されません。

3.5.1 ゼロ次ホールド

ゼロ次ホールドは入力を指定された周期の時点でサンプルし、次のサンプル点があるまで出力を保持します。

シンボル



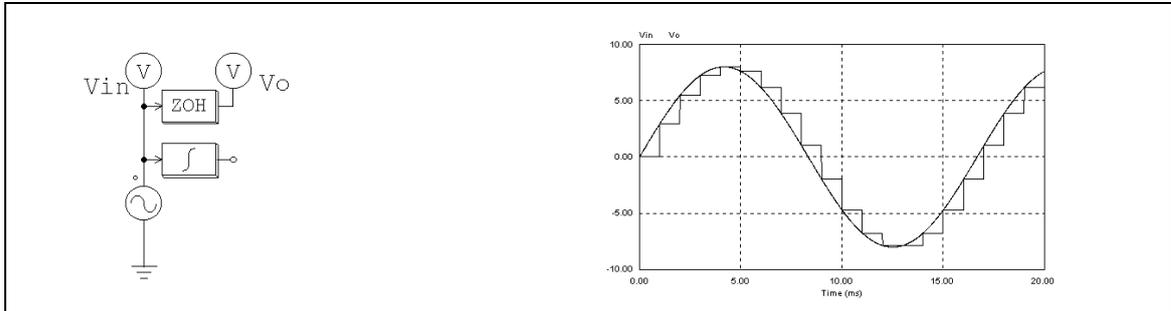
仕様

パラメータ	機能
Sampling Frequency	サンプリング周波数 (Hz)

他のデジタル素子同様、ゼロ次ホールドは内部のタイマーでサンプル時刻を決めます。したがって、サンプル時刻はシミュレーションの開始時刻と同期しています。たとえば、サンプリング周波数を 1000 Hz と指定すると、サンプリングは 0, 1 msec, 2 msec, ... という具合に実行されます。

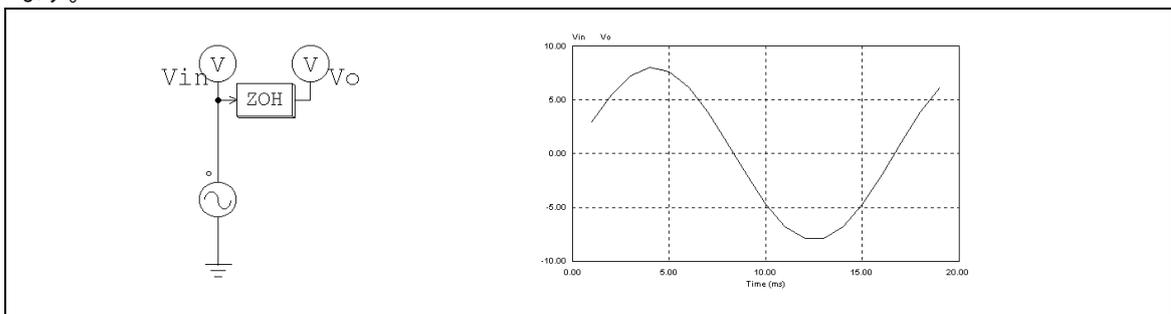
例.

以下の回路ではサンプリング周波数を 1000 Hz としています。右の図に入力と出力の波形を示します。



上の回路では入力の正弦波に連続型の積分器が接続されています。このため、この回路は連続・離散混合回路となり、連続系のシミュレーションタイムステップにより計算が行われます。こうしたシミュレーションタイムステップにより、ゼロ次ホールド素子の出力は上に示すように階段状となります。

一方、この積分器がないと回路は離散型となります。計算はサンプル時刻でのみ実行されるので、シミュレーションタイムステップはサンプル周期と同じになり、計算結果はサンプル時刻でのみ得られます。下の波形は一見、連続波形のように見えますが実は離散値で、サンプル点を結んだ線のみで構成されています。



3.5.2 z 領域伝達関数ブロック

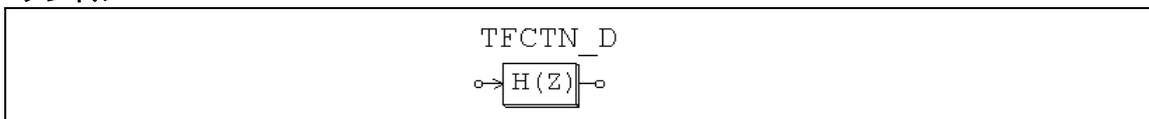
z 領域伝達関数ブロックは以下の多項式で示されます。

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

ここで、 $a_0 = 1$ のとき、 $Y(z) = H(z) \cdot U(z)$ は別の式で表現できます。

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

シンボル



仕様

パラメータ	機能
Order N	伝達関数の次元 N
Coeff. b_0, \dots, b_N	分子の多項式の係数
Coeff. a_0, \dots, a_N	分母の多項式の係数
Sampling Frequency	サンプリング周波数 (Hz)

例.

以下は 2 次の伝達関数です。

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

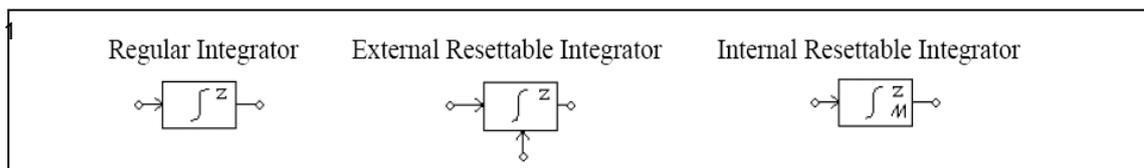
サンプリング周波数は 3kHz とします。PSIM では、以下のように指定します。

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

3.5.2.1 積分器 (離散型)

積分器 (Integrator) には 3 種類あります：通常の積分器、リセット機能付き外部積分器、リセット機能付き内部積分器です。

シンボル



リセット機能付き積分器の場合、シンボルの下の入力が制御信号になります。

仕様

パラメータ	機能
Algorithm Flag	積分アルゴリズムの切替フラグ 0: 台形法 1: 後退オイラー法 2: 前進オイラー法
Initial Output Value	出力の初期値
Reset Flag	リセットフラグ (0:エッジリセット; 1:レベルリセット) (リセット付き外部積分器のみ)
Lower Output Limit	出力の下限値(リセット付き内部積分器のみ)
Upper Output Limit	出力の上限値(リセット付き内部積分器のみ)
Sampling Frequency	サンプリング周波数 (Hz)

リセット付き外部積分器の出力は制御信号によりリセットすることができます。エッジリセット(リセットフラグ=0)の場合、制御信号の立ち上がりエッジで、積分器の出力は零にリセットされます。レベルリセット(リセットフラグ=1)の場合、制御信号を High(1)にすると、積分器の出力は零にリセットされます。

リセット付き内部積分器の出力は出力によりリセットされます。出力が下限値か上限値のどちらかに達するとき、リセット付き内部積分器の出力は 0 にリセットされます。これは、エッジリセット付き外部積分器と同じ動作をしますがこの場合外部のリセット回路を設定する必要はありません。

入力を $u(t)$ 、出力を $y(t)$ 、サンプリング周期を T 、離散型伝達関数を $H(z)$ とすると、積分器の入出力関係は以下のように表現できます。

台形法：

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

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

後退オイラー法：

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

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

前進オイラー法

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

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

3.5.2.2 微分器（離散型）

離散型微分器 (Differentiator) の伝達関数を以下に示します。

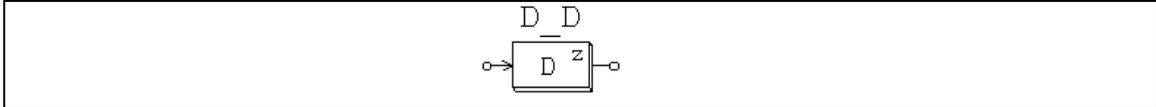
$$H(z) = \frac{1}{T} \cdot \frac{z-1}{z}$$

ここで、 T はサンプリング周期です。

入出力関係は差分方程式を使って、次のように表すこともできます。

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

シンボル



仕様

パラメータ	機能
Sampling Frequency	サンプリング周波数 (Hz)

3.5.2.3 デジタル・フィルタ

一般型 (FILTER_D) と有限インパルス応答型 (FIR) の2種類のデジタル・フィルタが利用可能です。一般型デジタル・フィルタの伝達関数を以下の多項式で示します。

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

$a_0=1$ のときは、出力 y と入力 u は以下の差分形式を使って表わすこともできます。

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

伝達関数の分母の係数が零でない場合、このタイプのフィルタは無限インパルス応答 (Infinite Impulse Response = IIR) フィルタと呼ばれます。

一方、FIR フィルタの伝達関数は以下の多項式で表わすことができます。

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

$a_0=1$ のときは、出力 y と入力 u は以下の差分形式を使って次のように表わせます。

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

フィルタの係数は直接指定するか、外部ファイルを使って指定します。以下にデジタル・フィルタ素子のシンボルとパラメータ (直接指定する場合) を示します。

シンボル

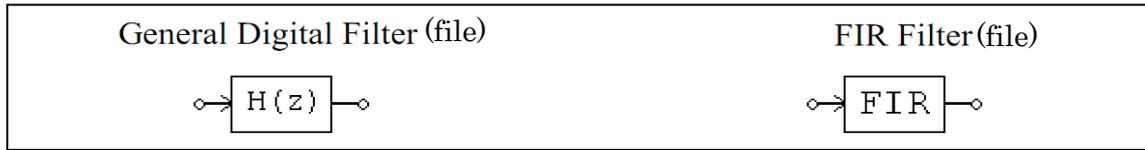


仕様

パラメータ	機能
Order N	伝達関数の次元 N
Coeff. b_0, \dots, b_N	分子の多項式の係数
Coeff. a_0, \dots, a_N	分母の多項式の係数
Sampling Frequency	サンプリング周波数 (Hz)

次に示すのはフィルタの係数を外部ファイルで指定する場合です。

シンボル



仕様

パラメータ	機能
File for Coefficients	係数格納ファイル名
Sampling Frequency	サンプリング周波数 (Hz)

係数格納ファイルの仕様は以下のとおりです。

Filter_FIR1 の場合

N
 b_0
 b_1

 b_N

FILTER_D1 の場合

N
 b_0
 b_1

 b_N
 a_0
 a_1

 a_N

または

N
 b_0, a_0
 b_1, a_1

 b_N, a_N

例.

ここでは例として 2 次の Butterworth 型ローパスデジタル・フィルタを設計します。
 カット・オフ周波数 $f_c = 1\text{kHz}$ 、サンプリング周波数 $f_s = 10\text{kHz}$ として、MATLAB* で計算すると
 Nyquist 周波数は $f_n = f_s/2 = 5\text{kHz}$
 正規化カット・オフ周波数は $f_c^* = f_c/f_n = 0.2$
 $[B,A] = \text{butter}(2, f_c^*)$
 となり、

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

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

を得ます。

したがって、伝達関数は、

$$H(z) = \frac{0.0201 + 0.0402 \cdot z^{-1} + 0.0201 \cdot z^{-2}}{1 - 1.561 \cdot z^{-1} + 0.6414 \cdot z^{-2}}$$

入出力差分式は、

$$y(n) = 0.0201 \cdot u(n) + 0.0402 \cdot u(n-1) + 1.561 \cdot y(n-1) - 0.6414 \cdot y(n-2)$$

PSIM では、パラメータを以下のように指定します。

Order N	2
Coeff. b_0, \dots, b_N	0.0201 0.0402 0.0201
Coeff. a_0, \dots, a_N	1. -1.561 0.6414
サンプリング周波数	10000.

係数がファイルに保存されている場合、そのファイルの内容は以下のようになります。

2
0.0201
0.0402
0.0201
1.
-1.561
0.6414

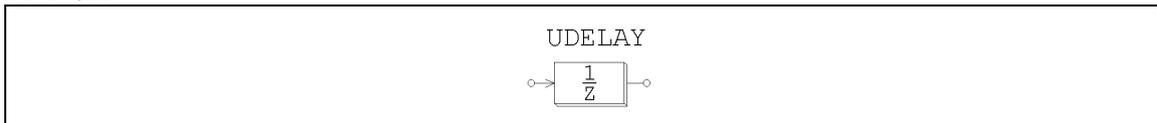
また、以下のような内容の場合もあります。

2
0.0201, 1
0.0402, -1.561
0.0201, 0.6414

3.5.3 単位遅れブロック

単位遅れブロック (Unit delay) は入力信号を 1 サンプル周期だけ遅らせます。

シンボル



仕様

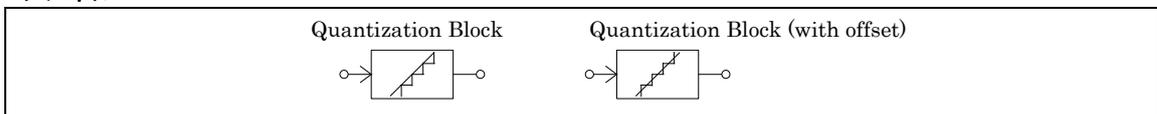
パラメータ	機能
Sampling Frequency	サンプリング周波数 (Hz)

離散素子の単位遅れブロックが標準化信号を 1 サンプル時刻だけ遅らせるのに対して、連続素子の時間遅れブロック (TDELAY) は波形全体を指定時間だけ遅らせます。

3.5.4 離散化ブロック

離散化ブロック (Quantization block) は A/D 変換プロセスの離散化誤差を模擬するのに使います。1LSB (least significant bit、最下位ビット) の離散化誤差をもったものと、0.5LSB の入力オフセットをもったものと、計 2 種類の離散化ブロックがあります。

シンボル

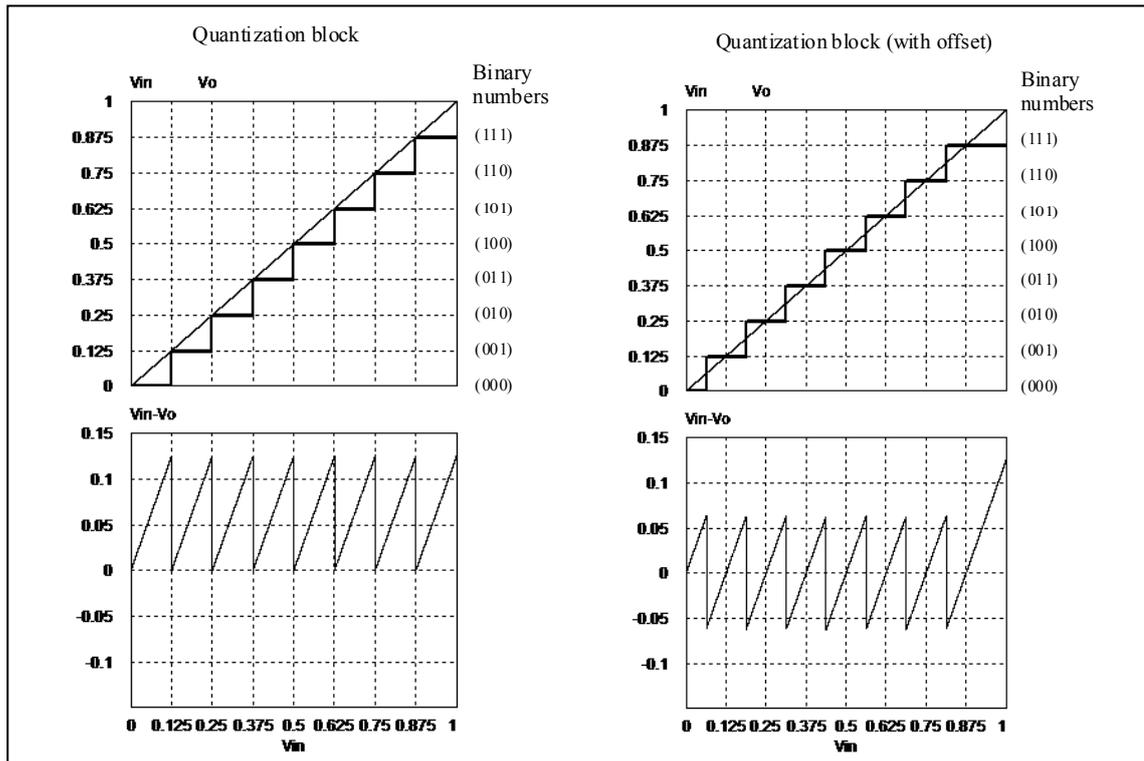


仕様

パラメータ	機能
No. of Bits	ビット数 N
Vin_min	入力値の下限 $V_{in,min}$
Vin_max	入力値の上限 $V_{in,max}$
Vo_min	出力値の下限 $V_{o,min}$
Vo_max	出力値の上限 $V_{o,max}$
Sampling Frequency	サンプリング周波数 (Hz)

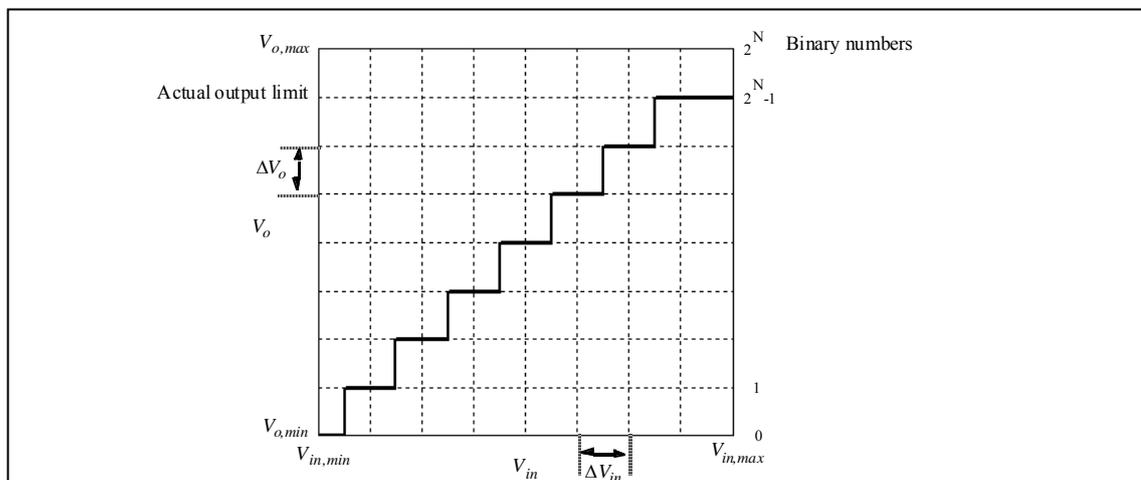
離散化ブロックの動作を理解するために、次のような例 ($N=3$, $V_{in,min}=0$, $V_{in,max}=1$) を考えます。この入力値は 2^3 (または8つ) のステージに分割されます。入力レベルに応じて、出力は3-bitの離散値 000, 001, 010, 011, 100, 101, 110, 111 のいずれかを取ります。出力は 2^3 (または8) のレベルと 2^3-1 (または7) のステップをもちます。2進数の000が10進数の0に、2進数の111が10進数の0.875に一致し、 $V_{in,min}=0$, $V_{in,max}=1$ のとき、入力と出力の波形をプロットすることができます。また、2つの離散化ブロックの離散化誤差 $V_{in}-V_o$ は以下のようにになります。

左の波形はオフセットなしの離散化ブロックのもので、右の波形はオフセット付きの離散化ブロックの波形です。



左の波形に示すように、離散化ブロックの離散化誤差は $0 \sim 1\text{LSB}$ 、または $1/2^3$ (または 0.125) となります。オフセット付きの離散化ブロックでは、離散化される前に $0.5 \cdot \text{LSB}$ の値が入力に付加されます。これは右の波形に見られるように、離散化誤差を -0.5LSB から $+0.5\text{LSB}$ まで減らします (入力が最大値に近い場合を除く)。

以下の図は一般のオフセット付き離散化ブロックの入出力関係を示しています。



ビット数は出力の解像度を決定します。入力幅 $V_{in,max}-V_{in,min}$ は 2^N ステージに分割されます。各ステージの幅は以下のように表されます：

$$\Delta V_{in} = \frac{V_{in,max} - V_{in,min}}{2^N}$$

ただし、1つ目のステージの幅は $0.5 \cdot \Delta V_{in}$ となり、最後のステージは $1.5 \cdot \Delta V_{in}$ となります。オフセットなしの離散化ブロックでは、1つ目のステージと最後のステージでその幅が ΔV_{in} となることにご注意ください。

入力が k 番目のステージで減少した場合、出力は次のように計算されます。

$$V_o = V_{o,min} + (k - 1) \cdot \Delta V_o$$

ここで k は 1 から 2^N の値をとり、出力ステップは以下の通り計算されます。

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

この値 $V_{o,max}$ は入力が $V_{in,max}$ の時に出力される値と一致します。しかし離散化によって、出力は N レベルで 0 から 2^N-1 の間の値で表されます。結果として、出力の実際の上限は上図で表される通り $V_{o,max}$ ではなく $V_{o,max} \cdot \Delta V_o$ となります。

例.

オフセット付きの離散化ブロックで $N=3$, $V_{in,min}=0$, $V_{in,max}=1$, $V_{o,min}=0$, $V_{o,max}=1$ とすると、 $\Delta V_{in}=\Delta V_o=1/8$ となります。

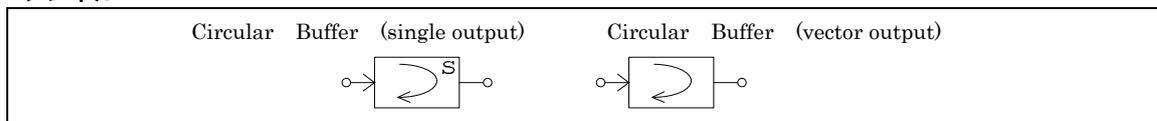
よって $V_{in}=0.25$ のとき、3 番目のステージ ($k=3$) で、 $V_o=0+(3-1) \cdot 1/8=0.25$ となります。

また、 $V_{in}=0.6$ のとき、6 番目のステージ ($k=6$) で、 $V_o=0+(6-1) \cdot 1/8=0.625$ となります。

3.5.5 循環バッファ

循環バッファ (Circular Buffer) はデータ配列を保存するメモリです。

シンボル



仕様

パラメータ	機能
Buffer Length	バッファの長さ
Sampling Frequency	サンプリング周波数 (Hz)

循環バッファはデータをバッファに格納し、ポインタがバッファの最後まで達したときには始めに戻ります。単出力(single output)の循環バッファは first-in-first-out のメモリ格納デバイスで、出力は押し出される値と等しい値を取ります。ベクトル出力の循環バッファはバッファの長さと同じ長さのベクトル配列を出力します。個別のメモリの内容を参照するには、メモリ読み出しブロック MEMREAD を使って下さい。

例.

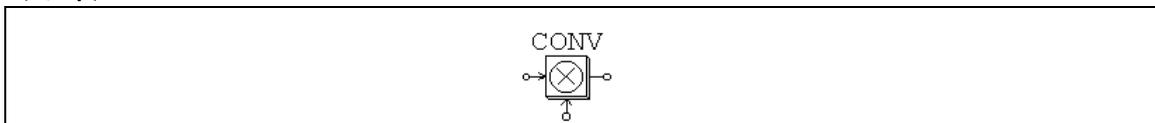
長さ 4 とサンプリング周波数 10Hz の循環バッファを考えましょう。サンプリング時刻によって、バッファの内容は以下のように変化します。

時刻	入力	メモリ位置による値				出力(単出力バッファ)
		1	2	3	4	
0	0.11	0.11	0	0	0	0
0.1	0.22	0.11	0.22	0	0	0
0.2	0.33	0.11	0.22	0.33	0	0
0.3	0.44	0.11	0.22	0.33	0.44	0
0.4	0.55	0.55	0.22	0.33	0.44	0.11
0.5	0.66	0.55	0.66	0.33	0.44	0.22
⋮	⋮	⋮	⋮	⋮	⋮	⋮

3.5.6 畳込みブロック

畳込みブロックは2つの入力ベクトルの畳込み積分を実行します。出力もベクトル値です。

シンボル



2つの入力ベクトルを

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

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

とすると A と B の畳込み積分は

$$C = A \otimes B$$

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

ここで、

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

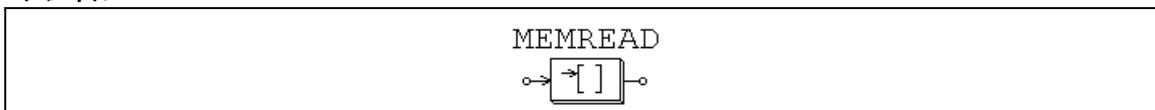
例.

入力を $A = [1 \ 2 \ 3]$ および $B = [4 \ 5]$ とすると、 $m=3, n=2$ で、出力は $C = [4 \ 13 \ 22 \ 15]$ となります。

3.5.7 メモリ読み出しブロック

メモリ読み出しブロックは指定された位置のメモリの内容（ベクトル要素）を読みだします。

シンボル



仕様

パラメータ	機能
Memory Index Offset	メモリの先頭位置からのオフセット距離

このブロックにより、ユーザは畳込みブロック、ベクトル配列、循環バッファなどの要素をメモリ位置によりアクセスできます。オフセット指標はメモリの先頭位置からの距離を示します。

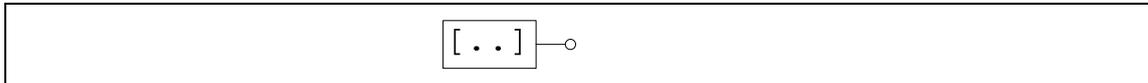
例.

ベクトル $A = [2 \ 4 \ 6 \ 8]$ を考えましょう。オフセットが0のとき、メモリ読み出しブロックの出力は2。オフセットが2のときは出力は6になります。

3.5.8 データ配列

これは1次元配列です。出力はベクトルとなります。データ入力は直接入力するもの (Array) とファイルにより指定するもの (Array(file)) があります。

シンボル



仕様

パラメータ	機能
Array Length	データ配列 N の長さ (Array のみ)
Values	配列要素の値 (Array のみ)
File for Coefficients	データ配列を格納しているファイル名 (Array (file) のみ)

あるファイルから配列が読み込まれる場合、そのファイルは以下のような形式です。

```

N
A1
... ..
AN
    
```

ここで、 N は配列の長さで、 $A_1 \dots A_N$ は配列要素の値です。

例.

配列 $A = [2\ 4\ 6\ 8]$ を定義するには、Array Length = 4; Values = 2 4 6 8 と指定します。あるファイルから配列が読み込まれる場合、そのファイルは以下のような形式です。

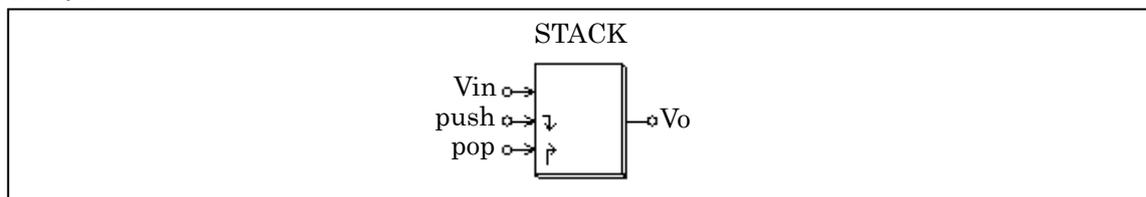
```

4
2.
4.
6.
8.
    
```

3.5.9 スタック

スタックは first-in-last-out のレジスタです。

シンボル



仕様

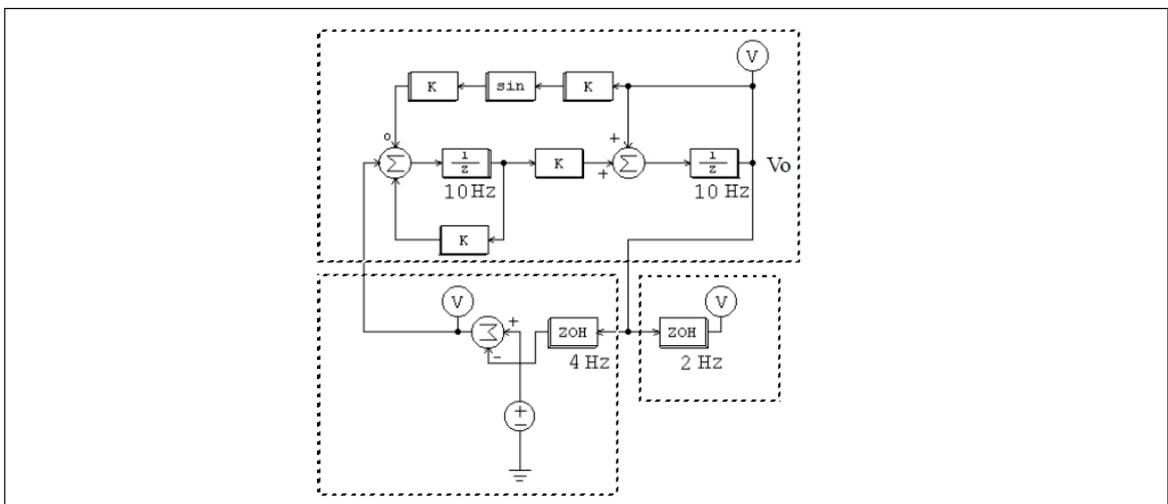
パラメータ	機能
Stack Depth	スタックの深さ

push または pop の操作を行うには立ち上がり信号を与えます。スタックが空のときに pop 操作をおこなった場合は、出力は不変です。スタックが一杯のときに push 操作をおこなうと、スタックの一番下にあるデータはレジスタから押し出されてなくなります。

3.5.10 多重サンプリングシステム

PSIM の離散系はサンプリング周期を複数持つことができます。以下に例を示します。

次のシステムは3つの部分に別れています。最初の部分システムはサンプリング周波数が 10Hz です。その出力 V_o はシステムのフィードバック信号として使っています。次の部分システムはサンプリング周波数が 4Hz です。3 番目の部分システムは同じ信号を周波数 2Hz でサンプリングして表示しています。



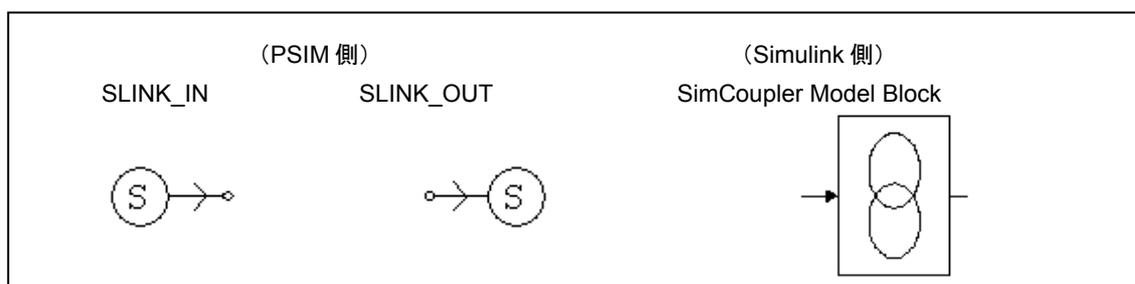
2つのサンプリング周波数が異なる要素の間にはゼロ次ホールドを使わなければなりません。

3.6 SimCoupler モジュール

SimCoupler は PSIM のアドオン・オプションで、PSIM と MATLAB/Simulink とのあいだのインタフェースを提供します。SimCoupler を使うことで、シミュレーションの一部は PSIM で、残りは Simulink で実行することができます。PSIM のパワー回路のシミュレーション機能をフルに活用しながら、MATLAB/Simulink の制御シミュレーション機能を補助的に使う、ということが可能です。

SimCoupler は PSIM の Link ノードと Simulink の SimCoupler ブロックからなります。以下にシンボルを示します。

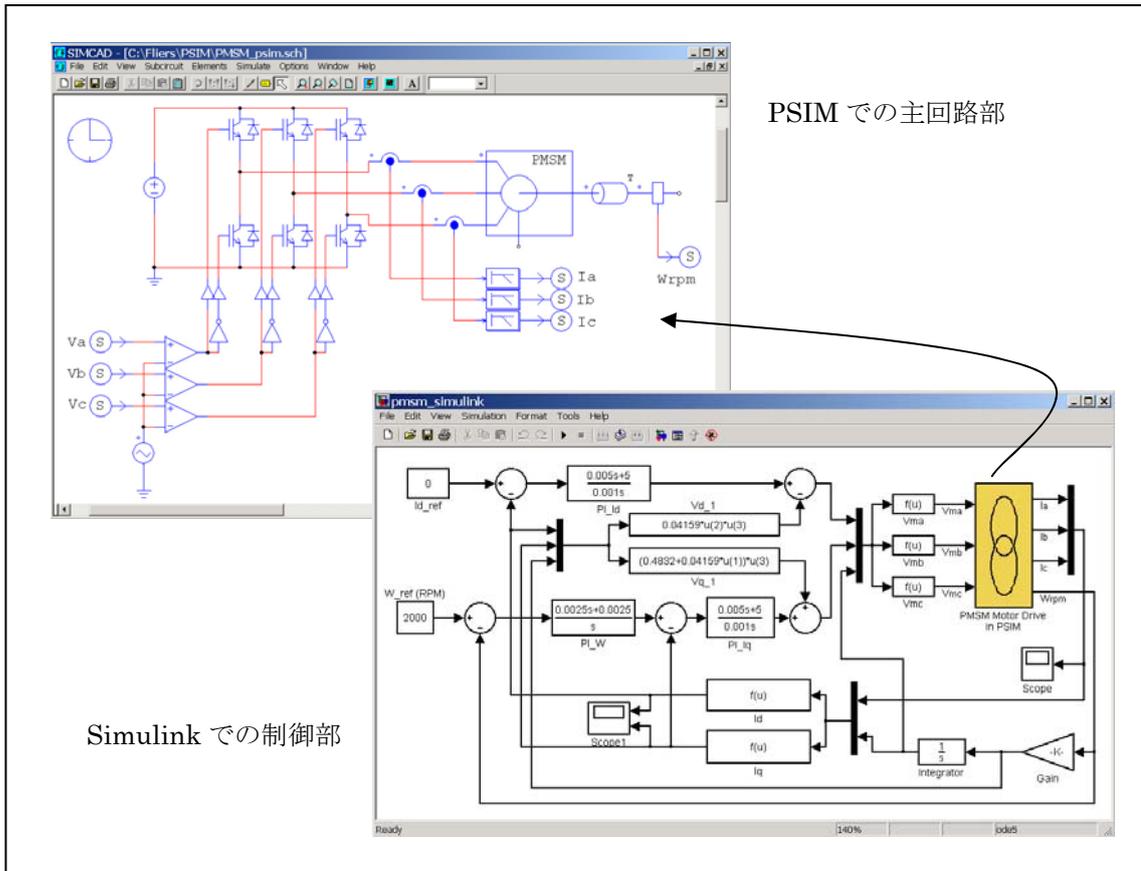
シンボル



PSIM 内では、SLINK_IN のノードが Simulink からの信号を受け取ります。また SLINK_OUT のノードから Simulink へ信号を送ります。これらはいずれも制御素子ですから、PSIM の制御回路内でのみ使用可能です。Simulink 内では、SimCoupler ブロックの入出力端子を經由して他の Simulink 内のシステムと接続することができます。

3.6.1 PSIM と Simulink での設定

SimCoupler の使用法は簡単かつ明瞭です。以下の例は、永久磁石同期機（PMSM）の駆動システムで、パワー回路には PSIM を使い、制御回路は Simulink で構成しています。



PSIM での主回路部

Simulink での制御部

以下にこの例題にもとづいて SimCoupler を使って PSIM と MATLAB/Simulink の連成シミュレーションを構成する手順を説明します。

SimulinkライブラリへSimcouplerブロックを追加：

HASPキーを備えたPSIMバージョンを使用している場合は、このステップをスキップしてください。プログラム「SetSimPath.exe」は、PSIMインストール中に既に自動的に実行されます。

PSIMデモバージョンあるいはPSIMトライアルバージョン(期日限定)を使用している場合は、PSIMディレクトリから、プログラム「SetSimPath.exe」を実行してください。これは、MATLAB/Simulinkを備えたSimCouplerモジュールをセットアップし、SimulinkライブラリブラウザへSimCouplerブロックを加えます。

このステップが必要であることに注意してください。そうでなければ、SimulinkはPSIMを見つけることができません。これで、MATLABパスに手動でPSIMフォルダを加えることは必要ではありません。

PSIMフォルダかMATLABフォルダが変わった場合のみ、再び「SetSimPath.exe」を実行する必要があります。

PSIM 側の操作：

- 1) パワー回路を構成したあと、A,B,C相の電流をローパス・フィルタを介して3つの SLINK_OUT のノードに接続し、端子名を Ia, Ib, Ic と指定します。また速度センサの出力を同様に別の SLINK_OUT ノードに接続し、Wrpm と名前を付けます。

2) 3つの SLINK_IN の入力ノードを比較器の正の入力に接続し、それぞれ端子名を Va, Vb, Vc と指定します。

3) **Simulate** メニューから **Arrange SLINK Nodes** を選びます。ダイアログウィンドウが表示されるので、SLINK_IN と SLINK_OUT の端子の並びが、Simulink の SimCoupler ブロックでの入出力の順番（上から下）と同じになるように並び替えてください。この例では、SLINK_IN のノードは Va, Vb, Vc の順で、SLINK_OUT のノードは Ia, Ib, Ic, Wrpm の順になります。

4) 回路図ファイルを保存します。この例では、ファイルは「C:\PSIM\pmsm_psim.sch」に保存されます。

Simulink 側の操作：

1) MATLAB を起動します。

2) Simulink を起動し、既存のファイルを開くか、新規ファイルを作成します。システムを作成した後、Simulink のライブラリブラウザのメニュー「S-function SimCoupler」に行き、SimCoupler のブロックを選択し回路図に置いてください。

3) PMSMファイルでは、SimCouplerブロックをダブルクリックしてから、**Browser**ボタンをクリックして「C:\PSIM\pmsm_psim.sch」というPSIMの回路図ファイルを選択してください。**Apply**をクリックすると、SimCouplerブロックの入出力の数はPSIMで作成したものと自動的に一致します。この例では3入力・4出力になります。

4) **Simulation** メニューから **Simulation Parameters** を選びます。**Solver Options** の項で、**Type** を Fixed Step に設定します。**Fixed step size** は PSIM のタイムステップと同じか、できる限り近い値に設定してください。この例ではタイムステップは 0.1ms です。Solver Option とタイムステップの設定についての詳細は次節で述べます。

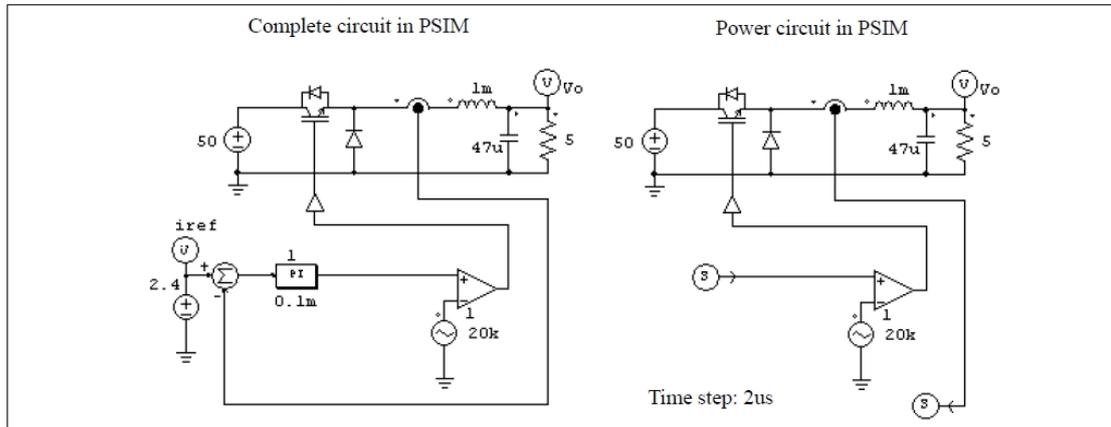
5) 設定が完了しました。Simulink でシミュレーションを開始します。

また、SimCoupler ブロックがフィードバックループの一部になるようなシステムでは、SimCoupler ブロックが代数ループ (algebraic Loop) の一部になってしまうことがあります (代数ループについてのより詳細な情報は MATLAB のヘルプをご覧ください)。MATLAB/Simulink のバージョンによっては代数ループが存在するシステムのシミュレーションができないことがあり、シミュレーションができる場合にもシミュレーション速度が著しく低下することがあります。代数ループを「Break」するためには、SimCoupler ブロックの各出力端子に Memory ブロックを接続します。Memory ブロックは 1 積分ステップ分の時間遅れを挿入します。

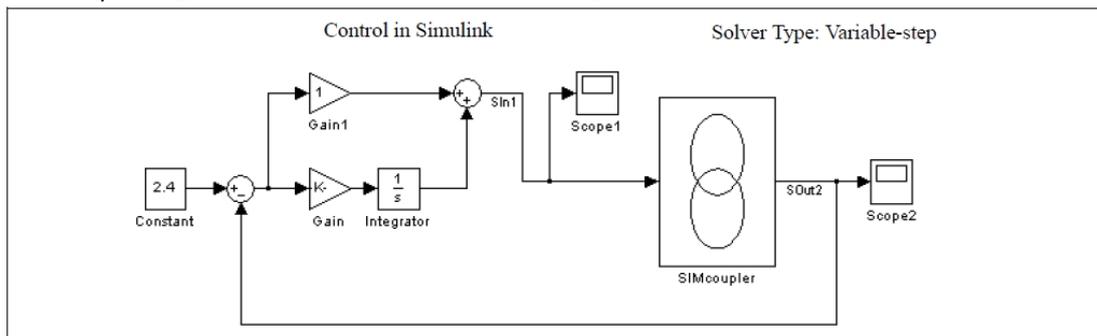
3.6.2 Simulink における Solver Type とタイムステップの設定

PSIM と MATLAB/Simulink で同時にシミュレーションをおこなう場合、Simulink では Solver Type とタイムステップについていくつかの制約があります。このことを以下に示すチョップ回路を例題に説明します。

左側の回路はすべて PSIM で構成し、右側の回路はパワー回路を PSIM で、制御回路を Simulink で構成しています。いずれの回路も PSIM のシミュレーションタイムステップは $2\mu\text{s}$ です。

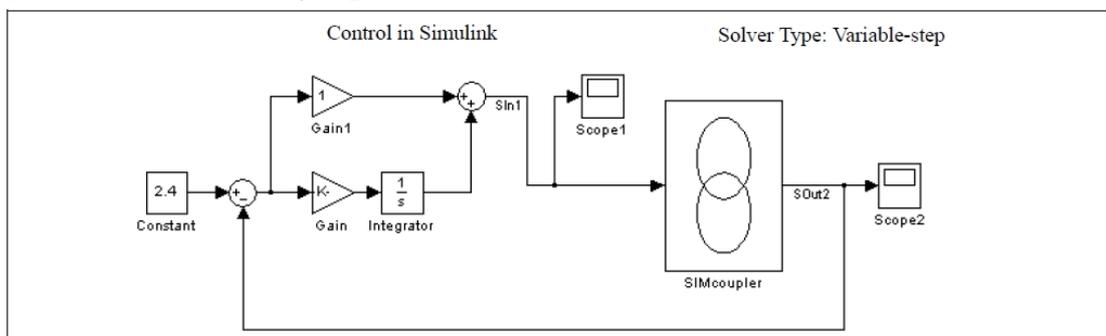


Simulink の設定はいくつか異なる方法があります。推奨の方法は Solver Type を **Fixed-step** に設定して、Fixed step size を PSIM のタイムステップと同じに指定することです。

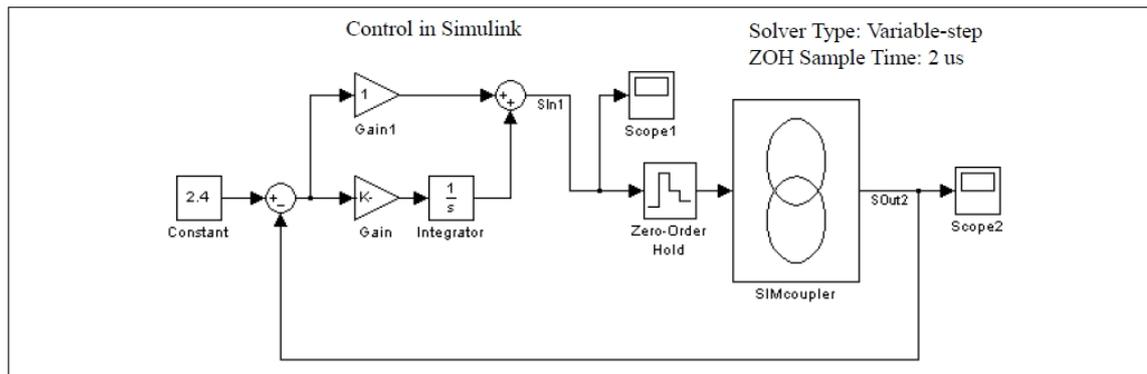


Simulink のタイムステップは PSIM と同一であることを推奨しますが、Simulink の側のタイムステップがわずかに PSIM のタイムステップよりも長い場合にも、十分な精度でシミュレーション結果が得られることがわかっています。上の例では、Simulink のタイムステップは PSIM よりも 10 倍も長い $20\mu\text{s}$ になっています。

しかし、Simulink の Solver Type を **Variable-step** に設定すると、シミュレーション結果は正しくなりません。下の図はこのような場合を示します。



Simulink の Solver Type を **Variable-step** に設定した場合は、正しい結果を得るには SimCoupler の入力にゼロ次ホールドを使う必要があります。さらに、ゼロ次ホールドのサンプル時間は PSIM のタイムステップと同じに設定する必要があります。下の図にこのような構成を示します。



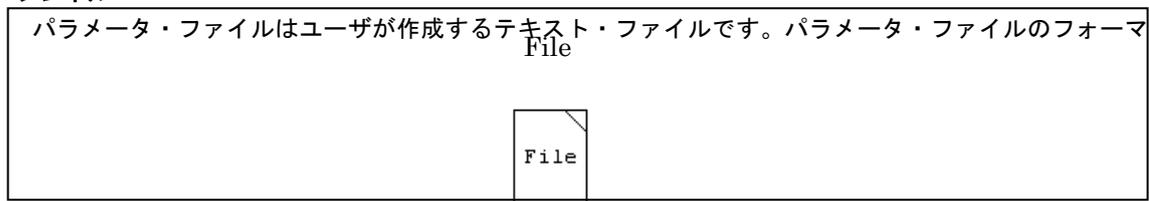
まとめとして、PSIM と MATLAB/Simulink で同時シミュレーションをおこなう場合は、Simulink の Solver Type は **Fixed-step** に設定して、タイムステップを PSIM と同じか、近い値に設定してください。また、Solver Type を **Variable-step** にした場合はゼロ次ホールドを使って、サンプル時間を PSIM のタイムステップと同じに設定するようにしてください。

第4章 その他の素子

4.1 パラメータ・ファイル

パラメータ・ファイル素子 (.FILE) は素子のパラメータやリミッタの設定を保存するファイルの名前を指定します。たとえば、抵抗の抵抗値を変数 R1 と指定して、パラメータ・ファイルの中で実際の値を記述することができます。

シンボル



ットは以下のとおりです。

```
<変数名> = <値>           %ここにコメントを入力
(grobal)<変数名> = <値>    %SimCoder でのみ"(grobal)"定義を行う
<変数名> <値>             //ここにコメントを入力
LIMIT <変数名> <下限値> <上限値>
% コメント行
//コメント行
```

<値> には数値 (例. R1=12.3) または数式 (例. R3 = R1 + R2/2.) を指定します。変数名と数値は等号 (例. R1 =12.3) または半角スペース (R1 12.3) で対応させます。行の最初に % または // の記号を置くと、その行はコメントとして扱われます。(例. %R3 は負荷抵抗)

"(grobal)" 定義は SimCoder で自動コード生成を行う際にのみ使用します。この使い方に関しては SimCoder のマニュアルをご参照ください。PSIM でのシミュレーションの際には、この定義は無視されます。よって、パラメータを "(grobal)Kp=1.2" と定義した場合、シミュレーション時は "Kp=1.2" と同様に扱われます。

例として、パラメータ・ファイルの内容は次のようになります ([] 内は説明です) 。

```
R1=12.3           [R1 を 12.3 に設定]
R2 23.4Ohm       [等号はスペースでも代用可]
% R3 は負荷抵抗 [コメント行]
R3=R1+R2/2.     [数式も指定可]
L1=3m            [10 の累乗も使用可 ; この場合 : m (ミリ) ; L1=0.003]
C1=100uF        [マイクロ]
LIMIT R3 5. 25. [R3 は 5. と 25. の間の値]
```

PSIM がパラメータ・ファイルを含んだ回路を保存する際、パラメータ・ファイルの内容を保存すると同時に相対パスと絶対パスの両方の情報を保存します。

PSIM がパラメータ・ファイルを含んだ回路を読み込む際、以下の順序でパラメータ・ファイルを検索します。

1. 相対パス (回路が保存されているフォルダとの相対位置)
2. 絶対パス
3. ローカルパス (回路が保存されているフォルダ)

1~3 内にパラメータ・ファイルが存在しない場合は、回路が保存されているフォルダに自動的に作成されます。

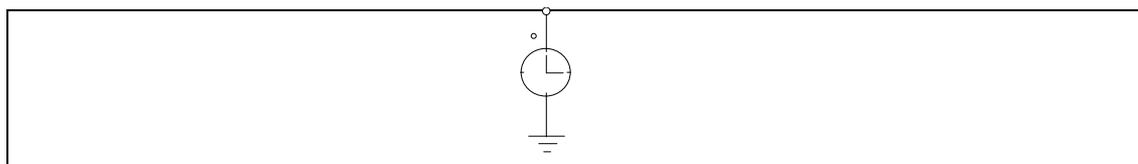
4.2 電圧源・電流源

PSIM では、いくつかの電圧源・電流源が用意されています。電流源の方向は次のように定義します。電流は高電位のノードから出て、外部回路を流れ、低電位のノードに戻ります。電流源はタイプにかかわらずパワー回路でのみ使用できます。

4.2.1 時間

時間要素は区分線形電圧源の特殊な場合です。これは接地された電圧源として扱われ、シミュレーション時刻(sec)と同じ値を出力します。

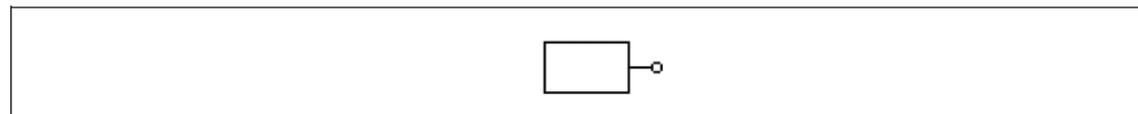
シンボル



4.2.2 定数

定数素子で、ユーザは定数を定義できます。これは、接地した電圧源として作用します。

シンボル



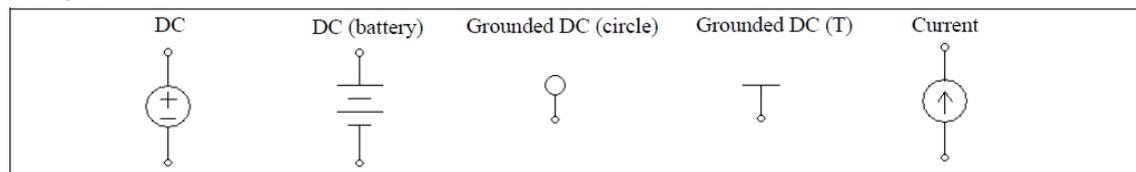
仕様

パラメータ	機能
Value	定数の値

4.2.3 直流電源

直流源は出力一定です。また、Grounded DC の一方の端子は接地されています。

シンボル



仕様

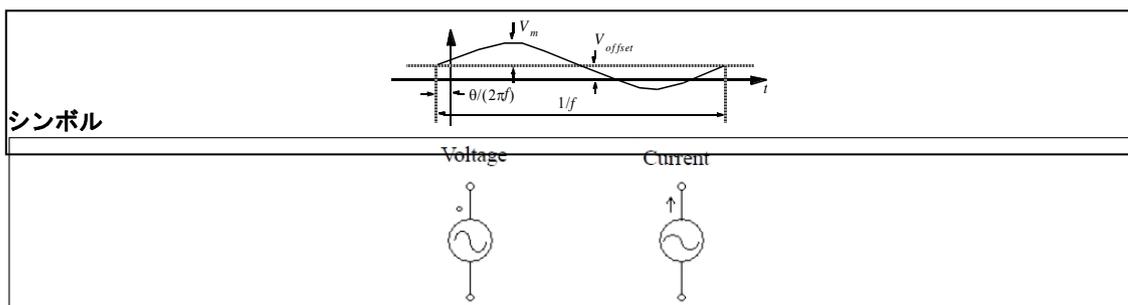
パラメータ	機能
Amplitude	電源の出力値

4.2.4 正弦波電源 (Sinusoidal Source)

正弦波電源は以下のように定義されています。

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

下の図に正弦波電源の仕様を示します。

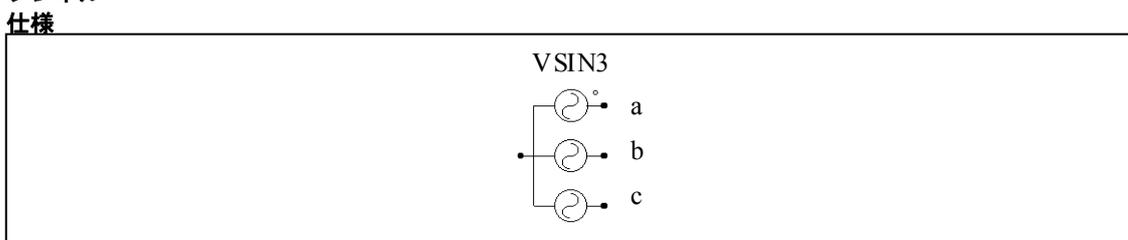


仕様

パラメータ	機能
Peak Amplitude	正弦波の最大値 V_m
Frequency	周波数 f (Hz)
Phase Angle	初期位相角 θ (度)
DC Offset	直流オフセット V_{offset}
Tstart	開始時刻；この時刻以前は電源は 0 になります。

3 相回路の構成に便利なように対象 3 相 Y 結線正弦波電圧源が用意されています。ドットのついた相が a 相です。

シンボル

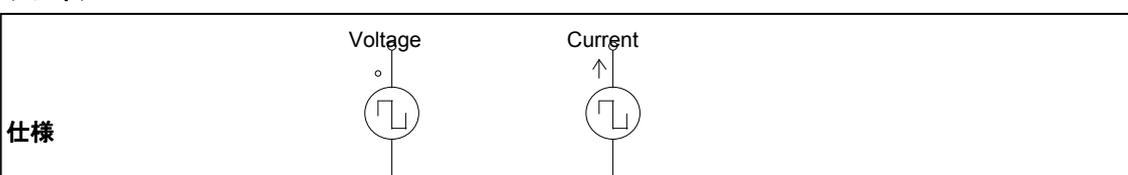


パラメータ	機能
V(line-line, V_{rms})	線間の実効値 (V)
Frequency	周波数 f (Hz)
Init.Angle(Phase a)	a 相の初期位相角 θ (度)

4.2.5 方形波電源 (Square-Wave Source)

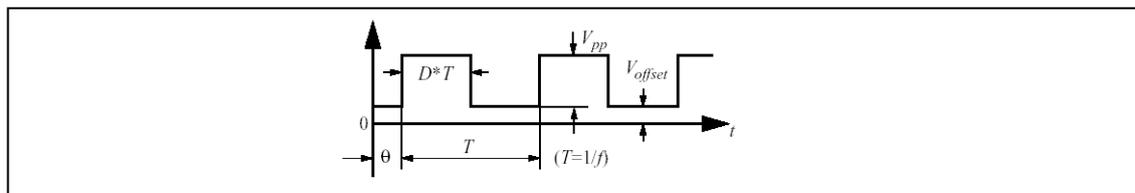
方形波電圧源および電流源は peak-to-peak の振幅、周波数、デューティ比および DC オフセットにより指定できます。通電周期は 1 周期に対する高電位が出力される時間の比で定義されます。

シンボル



パラメータ	機能
V _{peak-peak}	ピーク間の振幅 V _{pp}
Frequency	周波数 f (Hz)
Duty Cycle	デューティ比 D
DC Offset	直流オフセット V _{offset}
Phase Delay	波形の位相遅れ θ (度)

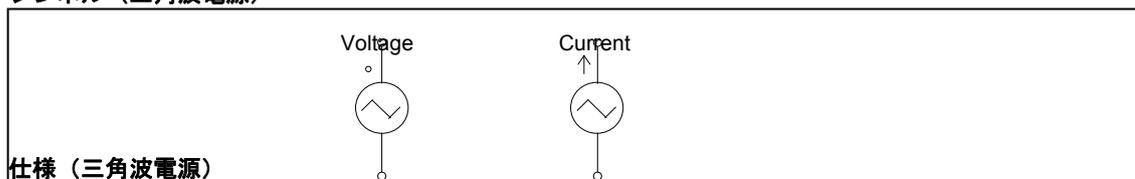
下の図に方形波電源の仕様を示します。



4.2.6 三角波電源／のこぎり波電源

三角波電圧源および電流源はピーク間振幅、周波数、デューティ比および DC オフセットにより指定できます。デューティ比は 1 周期に対する立ち上がり時間の比で定義されます。

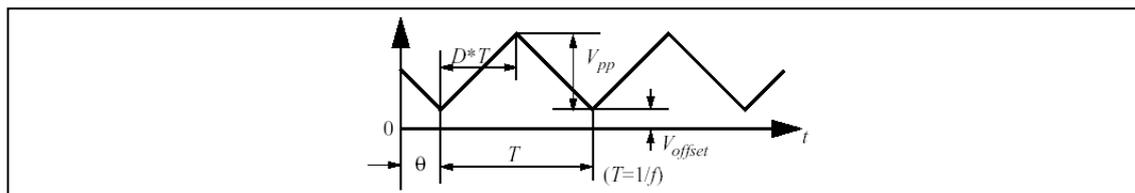
シンボル (三角波電源)



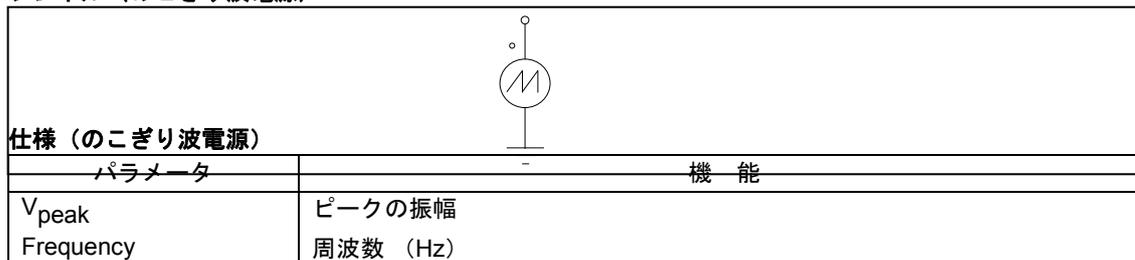
仕様 (三角波電源)

パラメータ	機能
V _{peak-peak}	ピーク間の振幅 V _{pp}
Frequency	周波数 f (Hz)
Duty Cycle	デューティ比 D
DC Offset	直流オフセット V _{offset}
Phase Delay	波形の位相遅れ θ (度)

下の図に三角波電源の仕様を示します。



シンボル (のこぎり波電源)



仕様 (のこぎり波電源)

パラメータ	機能
V _{peak}	ピークの振幅
Frequency	周波数 (Hz)

4.2.7 ステップ電源

ステップ電圧源および電流源は指定された時間に出力がある設定値から別の設定値に変化します。2種類のステップ電源があります：1つは0からある値まで変化するタイプ（Step）で、もう1つはある値から別の値へ変化するタイプ（Step(2-level)）です。

シンボル



仕様

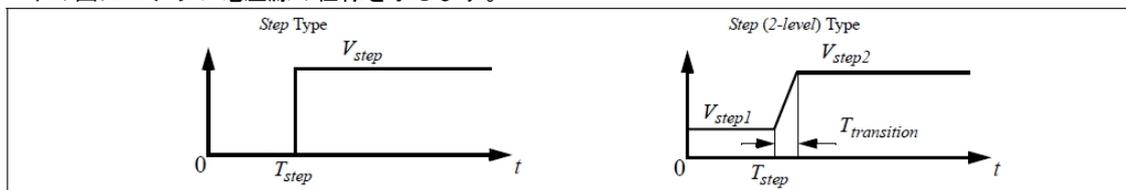
Step type の場合：

パラメータ	機能
V_{step}	ステップ変化の後の出力値 V_{step}
T_{step}	ステップ変化を発生させる時刻 T_{step}

Step (2-level) type の場合：

パラメータ	機能
V_{step1}	ステップ変化の前の値 V_{step1}
V_{step2}	ステップ変化の後の値 V_{step2}
T_{step}	ステップ変化を発生させる時刻 T_{step}
$T_{transition}$	V_{step1} から V_{step2} までの遷移時間 $T_{transition}$

下の図にステップ電圧源の仕様を示します。



4.2.8 区分線形電源（Piecewise Linear）

区分線形電源は多くの線分で波形を構成します。線分は接点の数、電圧・電流値、およびそのときの時刻(sec)で指定できます。値や時刻は独立して、またはペアで入力することができます。

シンボル



仕様

値と時刻を独立で定義するモデル（Piecewise linear）の場合：

パラメータ	機能
-------	----

Frequency	周波数 f (Hz)
No. of Points	接点の数
Values V_1, \dots, V_n	各接点における電圧または電流値
Time T_1, \dots, T_n	各接点の時刻(sec)

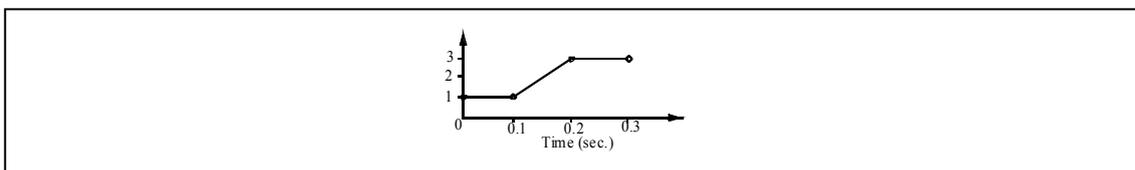
値と時刻をペアで定義するモデル (*Piecewise linear (in pair)*) の場合 :

パラメータ	機能
Frequency	周波数 f (Hz)
Time, Values (t_1, v_1) ...	各接点の時刻と値

時刻と値のペアは必ずカッコで囲んでください。カッコ内の時刻と値はカンマで区切るか (例 : 1.2m,5.5) 、スペースで区切るか (例 : 1.2m 5.5) 、あるいはその両方を同時に使用できます (例 : 1.2m, 5.5) 。

例. 区分線形電源の設定

以下に非周期的区分線形電源の例を示します。区分数は 3 で、図に示すとおり 4 つの接点により定義することができます。



Piecewise linear の場合は、以下のように指定します。

周波数	0.
接点数 n	4
値 V_1, \dots, V_n	1. 1. 3. 3.
時刻 T_1, \dots, T_n	0. 0.1 0.2 0.3,

Piecewise linear (in pair) の場合は、以下のように指定します。

周波数	0.
時刻, 値 (t_1, v_1)	(0., 1) (0.1, 1) (0.2, 3) (0.3, 3)

4.2.9 ランダム電源

ランダム電圧源 (VRAND) と電流源 (IRAND) の出力はシミュレーションの各時刻においてランダムに決まります。ランダム電圧源は以下のように定義されています。

$$v_o = V_m \cdot n + V_{offset}$$

ここで V_m は電源のピーク間振幅、 n は 0 と 1 のあいだのランダムな数、 V_{offset} は dc オフセットです。

シンボル



仕様

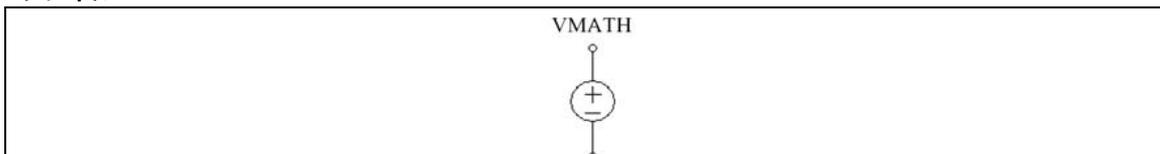
パラメータ	機能
Peak-peak Amplitude	ピーク間の振幅

DC Offset	直流オフセット
-----------	---------

4.2.10 数式関数電源

数式関数電源はユーザが任意の関数を数式で指定できる電源です。

シンボル



仕様

パラメータ	機能
Expression	関数の数式表現
Tstart	電源の動作開始時刻

数式のなかで T または t は時刻を示します。たとえば、正弦波関数の電源を指定するには、数式表現は $\sin(2*3.14159*60*t+2.09)$ などとなります。

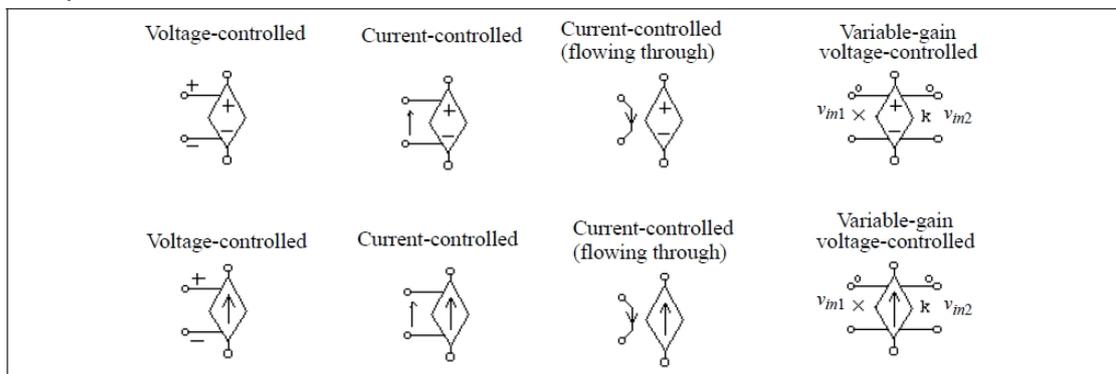
4.2.11 制御付き電圧源・電流源

PSIM では次の 6 種類の制御付き電源が使用可能です。

- 電圧制御電圧源 (Voltage controlled voltage source)
- 電流制御電圧源 (Current controlled voltage source)
- 電圧制御電流源 (Voltage controlled current source)
- 電流制御電流源 (Current controlled current source)
- 可変ゲイン電圧制御電圧源 (Variable-gain voltage controlled voltage source)
- 可変ゲイン電圧制御電流源 (Variable-gain voltage controlled current source)

電流制御の場合、RLC ブランチの電流を制御電流とする必要があります。また、電流源の場合は、制御電圧または電流は独立の電源から取ることはできません。制御付き電圧・電流源はパワー回路でのみ使用可能です。

シンボル



仕様

パラメータ	機能
Gain	電源のゲイン

電圧制御の電源において電圧は正(+)のノードから負(-)のノードに向かいます。一方、電流制御の電源では制御端子をRLC ブランチに直列に接続し、電流の方向はシンボル図の矢印に示すようにします。もう一組の電流制御の電源においては、制御電流は一方の端子から入り、他方の端子から出ます。制御電流を検出するために $10\mu\Omega$ の抵抗を使っています。

ゲインが連続で制御可能な電圧・電流源は Input1 を乗算記号の側に、また、Input2 は k の文字がある側に加えます。制御付き電圧・電流源では、出力は制御入力の電圧・電流をゲイン倍したものになります。一方、可変ゲイン電圧・電流源では出力は以下の式で決まります。

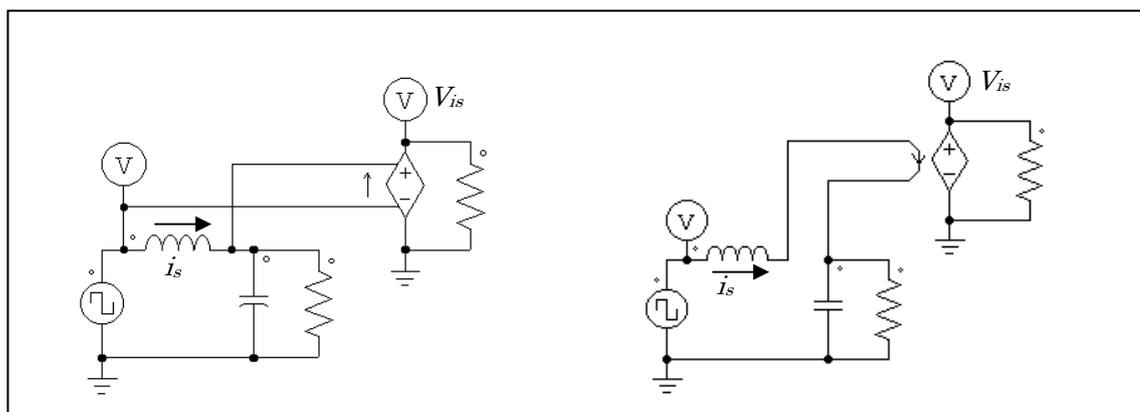
$$v_o = (k \cdot v_{in2}) \cdot v_{in1}$$

$$i_o = (k \cdot v_{in2}) \cdot v_{in1}$$

次節に述べる非線形電源と可変ゲイン電源の違いは、非線形電源の出力は現在時刻の v_{in1} および v_{in2} を使って計算し、各反復で更新されるという点です。一方、可変ゲイン電源では隣り合った時刻での v_{in2} の変化は小さいと見積もって、前の時刻での v_{in2} を現在時刻の計算に使います。この仮定は v_{in2} が v_{in1} に比べてかなりゆっくりと変化し、 v_{in2} の変化に比べてタイムステップが充分小さい場合に成り立ちます。可変ゲイン電源は非線形電源では収束の問題があるような場合に使えます。

例.

下の図に電流制御電圧源の回路を示します。左の回路では、電流制御電圧源をインダクタ電流 i_s によって制御しています。このように電流制御電圧源を使うことにより、電流値を電圧値に変換することができます。右の回路では、左の回路と異なる電流制御型電圧源を使っています。



4.2.12 非線形電圧制御電源

非線形の電圧制御電源 (Nonlinear Voltage-Controlled Source) は、入力の乗算・除算・平方根のいずれかを出力します。次に出力の式による定義を示します。

非線形 (乗算) 出力 $v_o = k \cdot v_{in1} \cdot v_{in2}$ または $i_o = k \cdot v_{in1} \cdot v_{in2}$

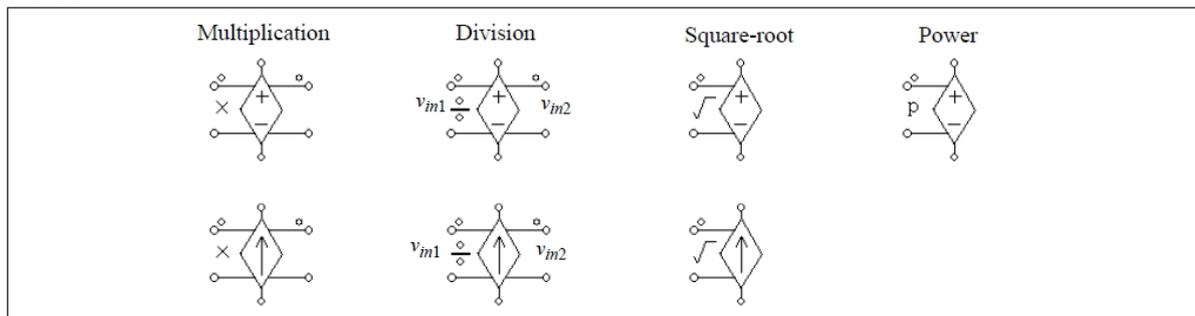
非線形 (除算) 出力 $v_o = k \cdot \frac{v_{in1}}{v_{in2}}$ または $i_o = k \cdot \frac{v_{in1}}{v_{in2}}$

非線形 (平方根) 出力 $v_o = k \cdot \sqrt{v_{in1}}$ または $i_o = k \cdot \sqrt{v_{in1}}$

非線形 (電力) 出力 $v_o = \text{sign}(v_{in}) \cdot k \cdot (k_1 \cdot v_{in})^{k_2}$

非線形電源（以下の図で *Power*）では、 $sign(v_{in})$ は、 v_{in} が正のときは 1、負のときは -1 となります。
非線形電圧制御電源はパワー回路でのみ使用できます。

シンボル



仕様

Power を除くすべての電源 :

パラメータ	機能
Gain	電源のゲイン k

Power :

パラメータ	機能
Gain	電源のゲイン k
Coefficient k_1	係数 k_1
Coefficient k_2	係数 k_2

非線形（除算）電源では、Input1 は除算記号の側に加えます。

4.3 電圧・電流センサ

電圧・電流センサはパワー回路の電圧・電流を計測し、制御信号に計測値を送ります。電流センサは $1\mu\Omega$ の内部抵抗を考えています。

シンボル



仕様

パラメータ	機能
Gain	センサのゲイン

4.4 プローブとメーター

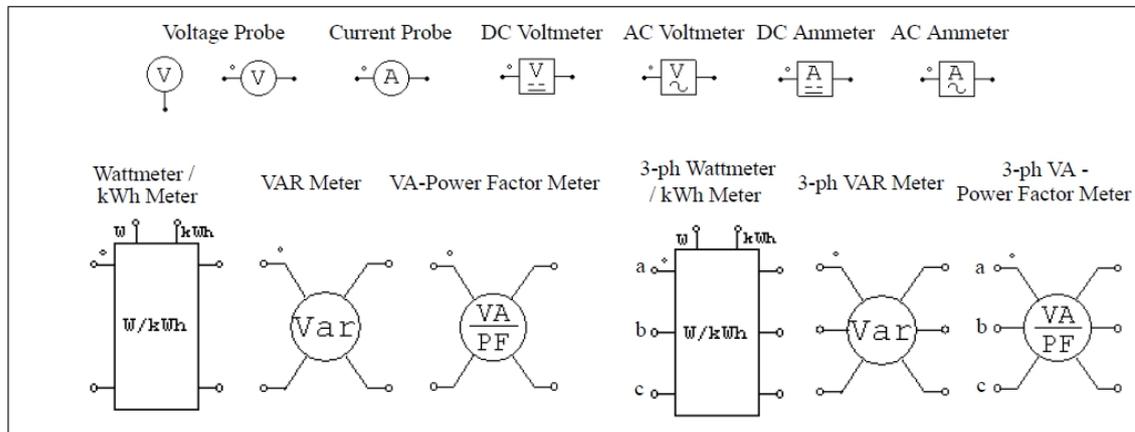
PSIM では電圧・電流等を測定するのに、プローブとメーターを使います。

以下の図に、電圧プローブ、電流プローブ、DC 電圧系、AC 電圧系、DC 電流計、AC 電流計、単相及び三相のワットメーター、kWh メーター、VAR メーター、VA 力率メーターを示しています。

電圧プローブは接続された端子と基準電位（GND）の間の電圧を測ります。2 端子間の電圧を測るには、2 端子用のプローブを使います。電流プローブ/メーターは、ドットの付いた側から流入する電流を測定します。同様に、パワーメーターはドットの付いた側から流入する電力を測定します。グラウンド接地型のプローブを除き、すべてのプローブとメーターはパワー回路でのみ使うことができます。

プローブが電圧や電流を瞬時値で測定するのに対し、メーターは直流または交流の電圧・電流、あるいは有効・無効電力を実際のメーターと同じように計測します。すなわち、直流は平均値を、交流は実効値を示します。電流プローブには $1\mu\Omega$ の小さな内部抵抗があります。

シンボル



仕様

パラメータ	機能
Operating Frequency	交流メーターの動作周波数または基本波周波数 (Hz)
Cut-off Frequency	ローパス・フィルタまたはハイパス・フィルタのカット・オフ周波数
kWh Meter Start Time	kWh メーターが測定を開始する時刻(sec)※ワット/kWh メーターのみ
kWh Meter Stop Time	kWh メーターが測定を停止する時刻(sec)※ワット/kWh メーターのみ
VA Display Flag	皮相電力の表示フラグ (0: 表示しない; 1: 表示する) ※VA 力率メーターのみ
PF Display Flag	力率の表示フラグ (0: 表示しない; 1: 表示する) ※VA 力率メーターのみ
DPF Display Flag	等価力率の表示フラグ (0: 表示しない; 1: 表示する) ※VA 力率メーターのみ

単相／三相ワットメーターおよび kWh メーターの図で、“W”と表示されたノードは有効電力の出力(W)で、“kWh”のノードは kWh 出力 (kWh) になります。

電力計と直流メーターには高周波成分を除くためローパス・フィルタを使っています。一方、交流メーターと無効電力計には直流分を除くためハイパス・フィルタを使います。カット・オフ周波数により各フィルタの過渡応答が決まります。

電圧・電流プローブを除いて、メーターの指示値は定常状態でのみ有効です。

また、VA/PF メーター素子のパラメータについて、Operating Frequency は測定対象信号の基本波周波数を設定してください。測定対象信号に高調波が含まれる場合、Cut-off Frequency の設定値に注意して下さい。

単相の回路では、有効電力 P 、無効電力 Q 、皮相電力 S 、総合力率 PF および等価力率 DFP (displacement power factor) は以下のように定義されます。

電圧・電流が高調波を含むと仮定すると、

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

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

ここで、 ω_1 は基本波の周波数、その他の ω はすべて高調波の周波数です。電圧と電流の実効値は、

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

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

有効電力（または平均電力） P は次のように定義します。

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

ここで T は基本波の周期です。

無効電力 Q は以下のように定義します。

$$Q = V_1 \cdot I_1 \cdot \sin(\phi_1 - \theta_1)$$

無効電力は基本波成分のみであることに注意して下さい。

皮相電力 S は以下で定義できます。

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

すると、総合力率 PF および等価力率 DPF は以下のようになります。

$$PF = \frac{P}{S}$$

$$DPF = \cos(\phi_1 - \theta_1)$$

3相回路についても同様の定義になります。ただし、3相 VA 力率メーターは3相3線回路用です。従って、3相電圧または3相電流の総和は零に等しくなければなりません。すなわち、

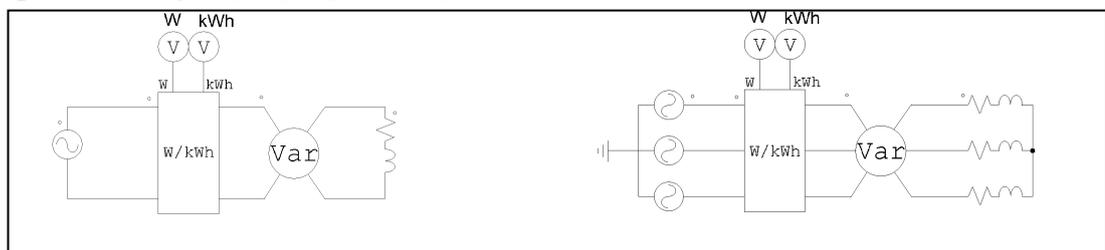
$$v_a + v_b + v_c = 0$$

$$i_a + i_b + i_c = 0$$

単相・三相のメーターの使用方法を以下に示します。

例

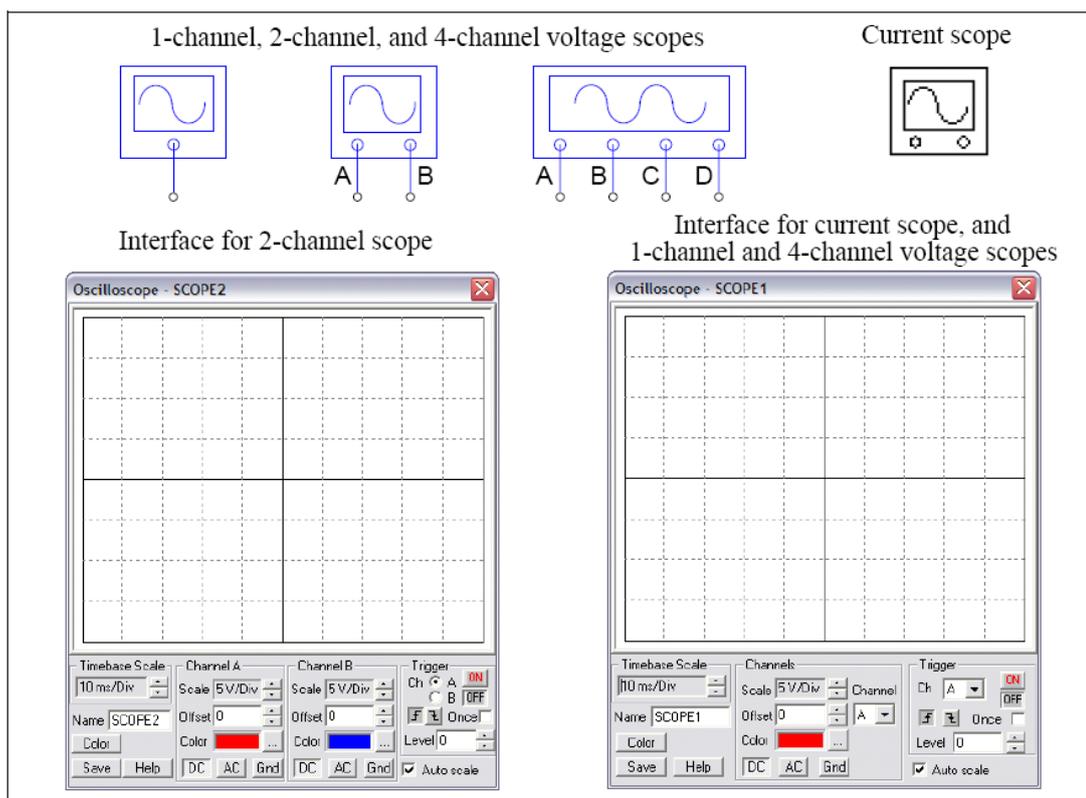
左の回路は単相有効、無効電力メーターの使用方を示しています、そして、右の回路は三相有効電力、無効電力メーターの使用方を示しています。



4.5 電圧/電流スコープ

電圧/電流のプロープとメーターはシミュレーション終了後の波形表示のために、シミュレーション結果を蓄えるだけでしたが、電圧/電流のスコープを用いて、ユーザはシミュレーション実行途中でシミュレーション波形を見ることができます。PSIMには次の4つのスコープがあります：1チャンネル電圧スコープ、2チャンネル電圧スコープ、4チャンネル電圧スコープおよび電流スコープ。

下記にこれらの電圧、電流スコープの要素、および波形表示ウィンドウを示します。



1チャンネル電圧スコープ、4チャンネル電圧スコープと電流スコープの波形表示ウィンドウの下に表示される操作パネルは同一です。実際のおシロスコープと同じように操作できるようになっており、次の3つの主な機能があります: *Timebase*機能、*Channel*機能、及び*Trigger*機能。

*Timebase*機能は時間(x軸)のスケールを定義することができます。

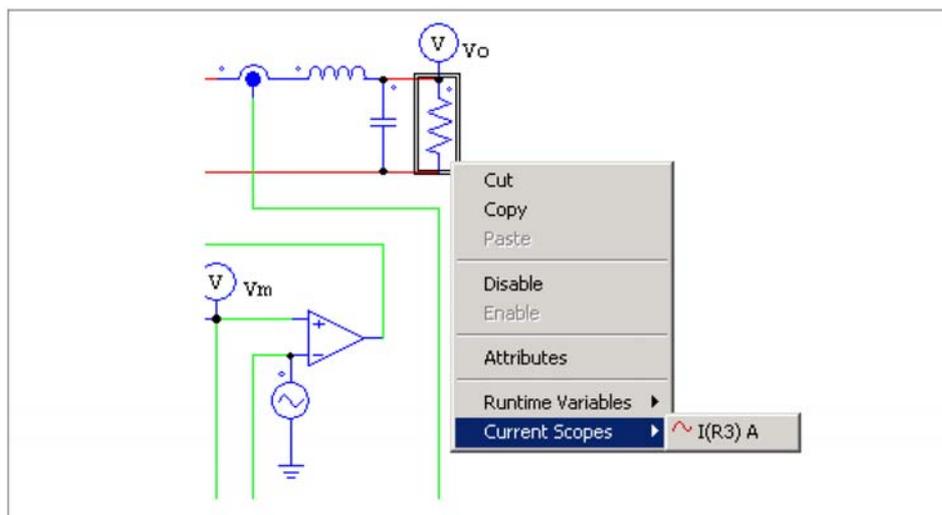
*Channel*機能は、波形のY軸のスケール、オフセット、および色を定義することができます。チャンネル表示モードはDC、AC、またはGNDのいずれかに設定できます。表示モードがDCのとき、全体の波形が表示されます。表示モードをACにすると、波形の交流部分だけが表示されます。表示モードがGNDにある時、波形は0を示します。

*Trigger*機能は、トリガ状態を定義することができます。ONまたはOFFのどちらかにトリガを設定します。トリガがオフである場合、波形は勝手に更新されます。トリガがオンである場合、トリガ条件が満たされた時だけ波形の表示更新を行い安定した波形表示ができます。

トリガモードは次の3つがあります。立ち上がりエッジ・トリガ、立ち下がりエッジ・トリガとワンショット・トリガ。onceチェックボックスをチェックするとワンショット・トリガが選択されます。ワンショット・トリガは一回だけ波形を取り込みます。過渡現象を捕らえる場合に有効です。トリガ・レベルはトリガ検出するレベルを設定します。例えば、チャンネルAは、立ち上がりエッジ・トリガで選択されており、トリガ・レベルを0Vとします。チャンネルAの入力が、マイナスから0を超える時にトリガが発生します。そして波形表示はその瞬間から開始されます。

電力回路ノードまたは制御回路ノードのどちらでも電圧スコープをつなげることができます。スコープはこれらのノードにおけるGNDに対する電圧を表示します。

電流スコープは、電流フラグのパラメータを持っている要素の電流を表示することができます。電流スコープには接続端子がありません。電流を表示させたい要素の上にマウスポインタを置き、右ボタンをクリックしてください。そして *Current Scopes* のブランチ電流を選択することによって、以下に示されるように電流スコープが有効になります。ブランチ電流が選択された後に、チェックマークはブランチ電流名の正面に現れます。



要素がCurrent Scopesメニューの複数の電流フラグを持っていると、複数のブランチ電流が表示されます。それぞれのブランチはそれぞれの電流のフラグに対応するようになります。

例えば、3相抵抗R1に関して、Current Scopesメニューで3つのブランチ電流が表示されます。

I(R1) A
I(R1) B
I(R1) C

文字「A」、「B」、および「C」はそれぞれチャンネルA、B、およびCについて言及します。例えば「I(R1)A」、「I(R1)B」及び、「I(R1) C」はすべて選択されたら、電流スコープでは、ChannelセクションのChannelプルダウンメニューに行って、チャンネルの一つを表示のために使うことができます。Channel Aが選択されると、スコープはA相のブランチ電流I(R1)を示します。

4.6 初期値

この素子を利用してパワー系、制御系回路の初期ノード電圧の定義ができます。

シンボル

仕様	
パラメータ	機能
Initial Value	ノード電圧の初期値

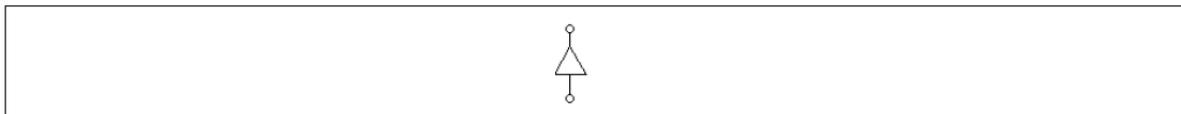
4.7 スイッチ制御器

スイッチ制御器は実際のゲートあるいはベース駆動回路と同じ機能を実現します。スイッチ制御器は制御回路から信号を受け取り、パワー回路のスイッチを制御します。ひとつのスイッチ制御器で、複数のスイッチを同時に制御できます。

4.7.1 オンオフ制御器

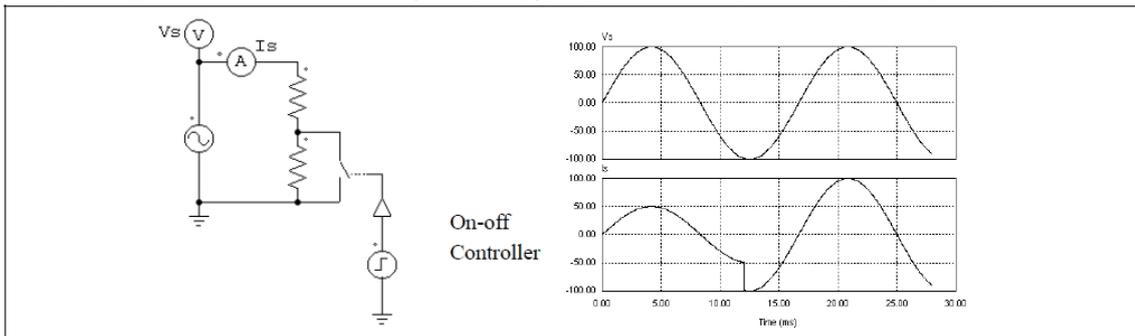
オンオフ制御器は制御回路のゲート信号とパワー回路のスイッチ動作とのインターフェースとして使います。制御回路からの入力は0か1の論理信号で、出力をパワー回路にゲート信号として加えることによりスイッチ素子の開閉を制御します。

シンボル



例.

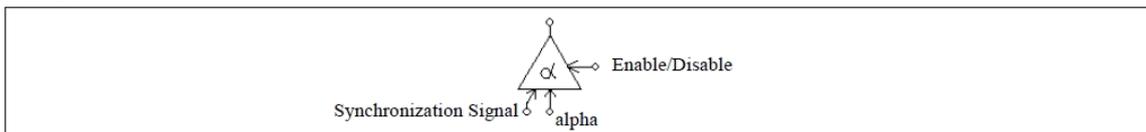
下に示すのは負荷のステップ変化を実現する回路です。オンオフ制御器が双方向スイッチを開閉するのに使われています。制御器の入力に加えたステップ電圧源が時刻 12ms で 0 から 1 に変化すると、スイッチ素子が閉となり、スイッチに並列に接続された抵抗をショートするので電流が増加します。



4.7.2 点弧角制御器

点弧角制御器 (α -controller) はサイリスタ・スイッチまたはブリッジの点弧遅れ角を制御するのに使います。この制御器への入力は、点弧角 (α) の値、同期信号、ゲート使用・不使用 (enable/disable) 信号の 3 種類です。同期信号が Low(0)から High(1)に変わるときに点弧角が零になるように同期をおこないます。点弧角制御器は α 度の遅れ角を調整して、サイリスタにゲート信号を送ります。点弧角 α の値は瞬時に更新されます。

シンボル



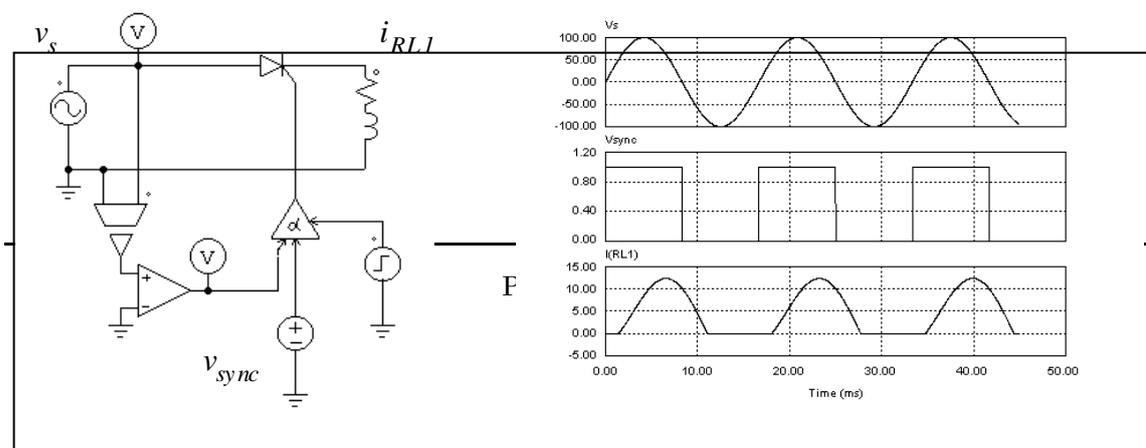
仕様

パラメータ	機能
Frequency	制御対象のスイッチの動作周波数 (Hz)
Pulse Width	スイッチのオン時間のパルス幅 (度)

遅れ角の入力値は度で与えます。

例.

下の図に遅れ角制御を使ったサイリスタ回路を示します。Vs のゼロクロス時点を同期に使っています。この時点でサイリスタは自然点弧するはずですが、ここでは遅れ角を 30°に設定しているため、ゲート信

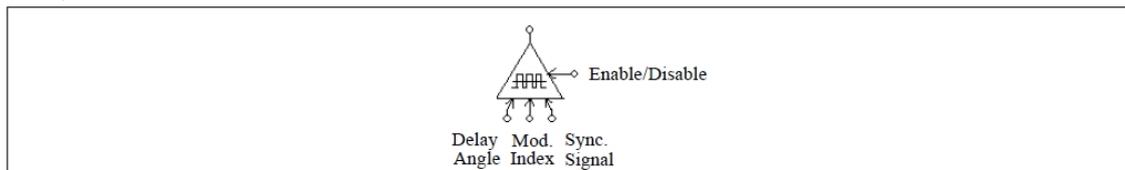


号は同期信号の立ち上がりから 30°遅れます。

4.7.3 PWM ルックアップテーブル制御器

PWM ルックアップテーブル制御器の入力は、変調指標、遅れ角、同期信号、およびゲート使用・不使用の 4 種類です。変調指標によりゲート・パターンが変わります。同期信号はゲート・パターンの同期に使います。同期信号が低から高に変わるとき、ゲート・パターンが更新されます。遅れ角はゲート・パターンと同期信号の相対的な角度を与えます。たとえば、遅れ角が 10°のとき、ゲート・パターンは同期信号から 10°進んでいることとなります。

シンボル



仕様

パラメータ	機能
Frequency	基本波周波数 (Hz)
Update Angle	ゲート信号を内部的に更新する角度 (度)。この角度が 360°のとき、ゲート信号は毎サイクルごとに更新されます。また、60°に設定すると、60°ごとの更新となります。
File Name	PWM ゲート・パターンを格納するファイル名

ゲート・パターンを示すルックアップテーブルは外部ファイルに格納されます。フォーマットは以下のとおりです。

$$n, m_1, m_2, \dots, m_n$$

$$k_1$$

$$G_{1,1}, G_{1,2}, \dots, G_{1,k_1},$$

$$\dots \dots \dots$$

$$k_n$$

$$G_{n,1}, G_{n,2}, \dots, G_{n,k_n},$$

ここで、 n はゲート・パターンの数、 m_i はパターン i に対応した変調指数、 k_i はパターン i のスイッチ点数です。変調指数の配列 $m_1 \dots m_n$ は単調増加となるように指定してください。入力が m_i 以下であると、出力は自動的に i 番目のパターンになります。入力が m_i を越えると、最後のパターンが選ばれます。

次の表に PWM パターン・ファイルの例を示します。変調指数は 5 で、スイッチング点数は 14 です。

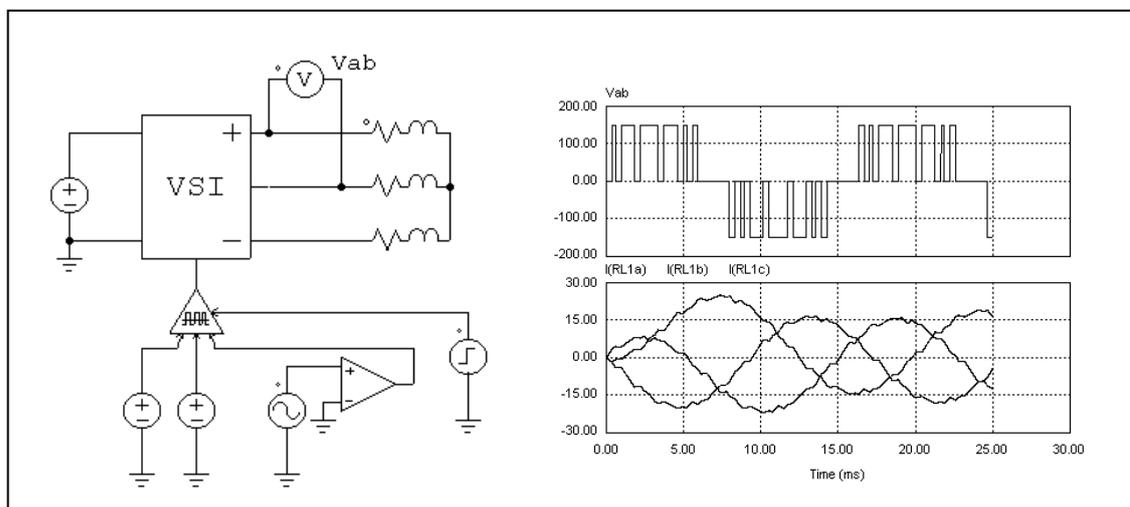
```

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.
    
```

このルックアップテーブルの例で、変調指数の入力が 0.8 のとき、最初のゲート・パターンが選ばれます。変調指数が 0.915 であると、3 番目のパターンが選ばれます。

例.

以下の回路図に3相電圧源インバータ(ファイル: vsi3pwm.sch に格納)を示します。コンバータ用のPWMは高調波消去をおこないます。上記のゲート・パターンが vsi3pwm.tbl にあらかじめ格納してあり、変調指標の値に応じたゲート・パターンが選ばれます。線間電圧と3相負荷電流の波形を下に示します。

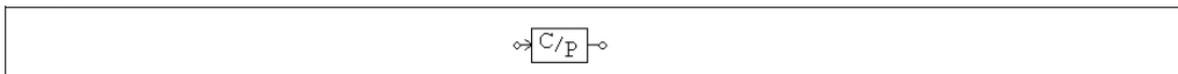


4.8 ファンクションブロック

4.8.1 制御・パワー変換ブロック

制御・パワー変換ブロックは制御回路の値をパワー回路に伝えます。このブロックは制御回路とパワー回路のバッファとして使います。PSIMがパワー回路を解析するとき、変換ブロックの出力は定電圧源として扱います。このブロックを使うことで制御回路でのみ得られる機能をパワー回路で使うことができます。

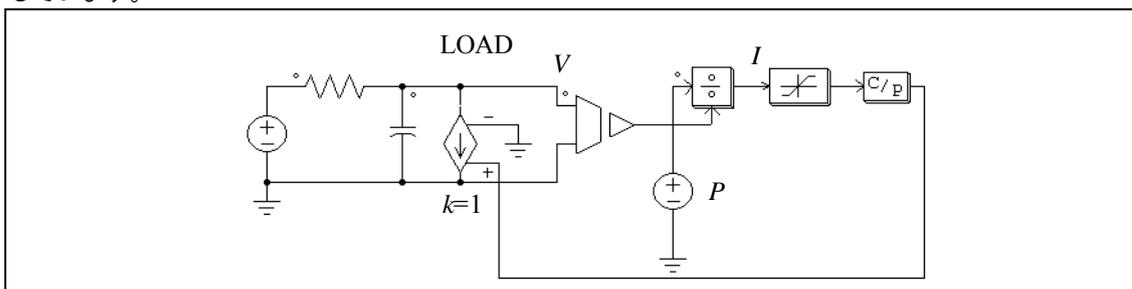
シンボル



例. 定電力負荷モデル

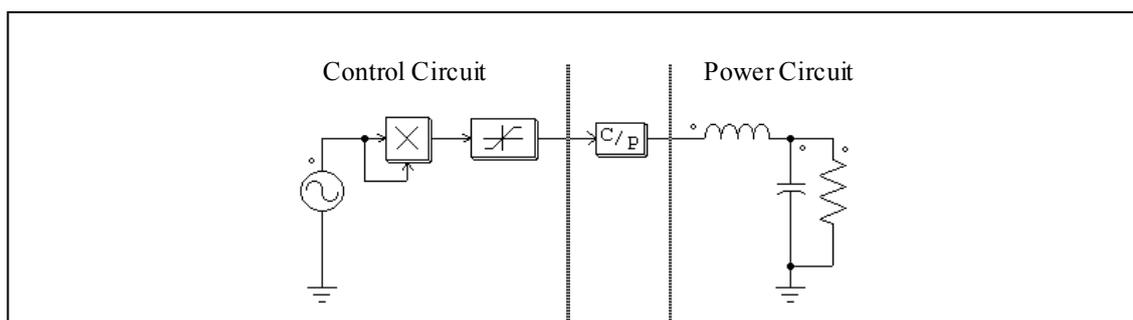
定電力の直流負荷では、電圧 V 、電流 I 、および電力 P のあいだに $P = V \cdot I$ の関係が成り立ちます。したがって、電力と電圧が決まれば、電流は $I = P/V$ から計算できます。この特性は次のような回路で実現できます。

負荷の両端の電圧を電圧センサで測り、除算器に入力すると、除算器の出力は電流 I を示します。初期状態では電圧が零または非常に低い値になる可能性があるため、電流の値をリミッタでおさえています。この値を制御・パワー変換ブロックを使って、負荷の電流として電圧制御電流源を通じてパワー回路に戻しています。



例.

以下に示す回路は制御回路の信号をパワー回路に伝える例です。パワー回路の構成からわかるように、制御・パワー変換ブロック (CTOP) は接地された電圧源として働きます。



4.8.2 変換ブロック

PSIM は以下の変換ブロックを提供します :

- abc-dq0 変換ブロック
- abc-αβ 変換ブロック
- αβ-dq 変換ブロック
- 直交座標系一極座標系変換ブロック

これらのブロックは制御回路だけでなく、パワー回路でも使用することができます。

4.8.2.1 abc-dqo 変換ブロック

ABC2DQ0 および DQ02ABC は 3 相電圧の座標系を abc 表現から dqo 表現に軸変換および逆変換します。これらの関数ブロックはパワー回路でも制御回路でも使うことができます。しかし、パワー回路では、軸変換の前に電流値は電圧値に（電流制御電圧源を使って）変換しなければなりません。使用しない入力端子は、グラウンドに接続してください。

本ブロックによる変換結果は、abc- $\alpha\beta$ 変換ブロック $\alpha\beta$ -dq 変換ブロックを組み合わせることで変換を行った結果とは変換スケールおよび回転方向（q 軸の方向）が異なります。詳細は下記の数式を参照して下さい。

abc 座標系から dqo 座標系への変換公式は以下のとおりです。

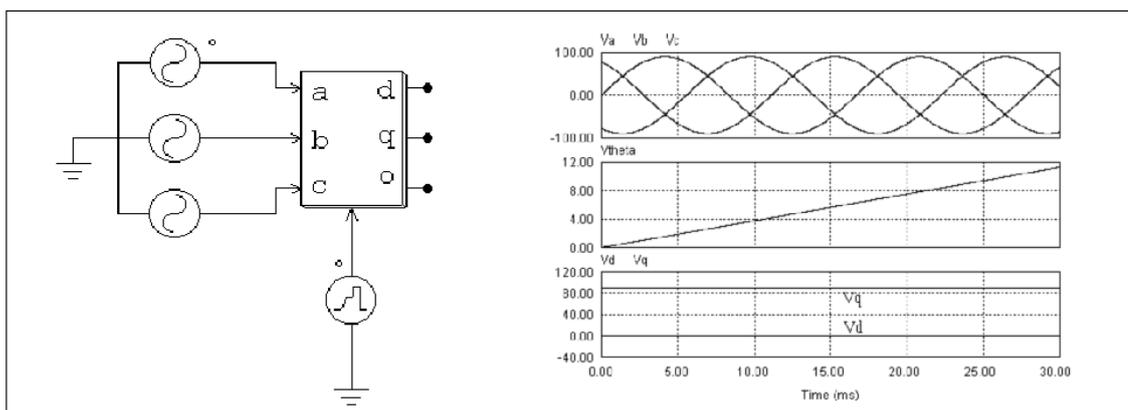
$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta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

また、dqo 座標系から abc 座標系への変換公式は以下のとおりです。

シンボル



例.

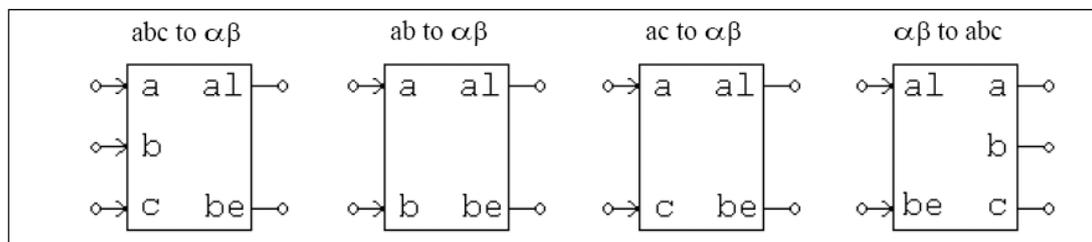


この例では、対称 3 相の電圧波形を dqo 系の値に変換しています。角度 θ は $\omega = 2\pi \cdot 60$ を使って $\theta = \omega t$ と定義しています。角度は時間に対して線形に変化するので、 θ を表現するのにランプ波形を使った区分線形電圧を使うことができます。シミュレーション波形は、3 相交流（上）、角度 θ （中）、dqo 出力（下）を示しています。この例では q 軸成分が定数で、d 軸と 0 軸の成分はどちらも零になっています。

4.8.2.2 abc- $\alpha\beta$ 変換ブロック

abc- $\alpha\beta$ 変換ブロックは abc 表現から $\alpha\beta$ 表現に軸変換及び逆変換します。

シンボル



イメージでは、文字「a|」は α を、「be」は β を示します。

$\alpha\beta$ -dq変換ブロックと本ブロックを組み合わせて変換を行った結果は、abc-dq変換ブロックの結果とは変換スケールおよび回転方向（q軸の方向）が異なります。詳細は下記の数式を参照してください。

abc座標系から $\alpha\beta$ 座標系への変換公式は以下の通りです：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}$$

ab座標系から $\alpha\beta$ 座標系への変換公式は以下の通りです：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ \frac{\sqrt{2}}{2} & \sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \end{bmatrix}$$

ac座標系から $\alpha\beta$ 座標系への変換公式は以下の通りです：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ -\frac{\sqrt{2}}{2} & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_c \end{bmatrix}$$

$\alpha\beta$ 座標系からabc座標系への変換公式は以下の通りです：

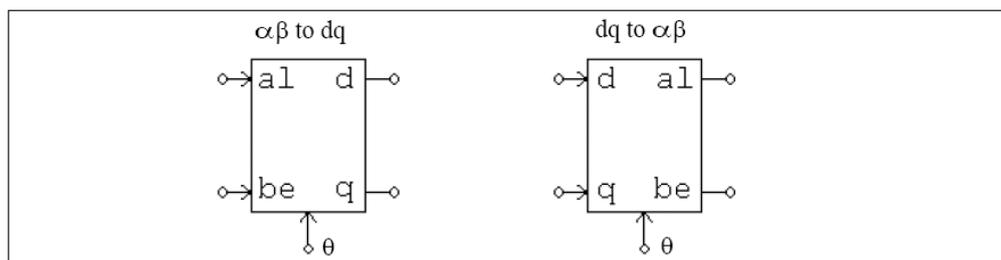
$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

ab- $\alpha\beta$ 及びac- $\alpha\beta$ 座標変換の時、 $v_a+v_b+v_c=0$ が仮定されています。

4.8.2.3 $\alpha\beta$ -dq変換ブロック

$\alpha\beta$ -dq変換ブロックは $\alpha\beta$ 表現からdq表現に軸変換及び逆変換します。

シンボル



イメージでは、文字「a1」はαを、「be」はβを示します。
 abc-αβ変換ブロックと本ブロックを組み合わせると変換を行った結果は、abc-dqo変換ブロックの結果とは変換スケールおよび回転方向（q軸の方向）が異なります。詳細は下記の数式を参照してください。

αβ座標系からdq座標系への変換公式は以下の通りです：

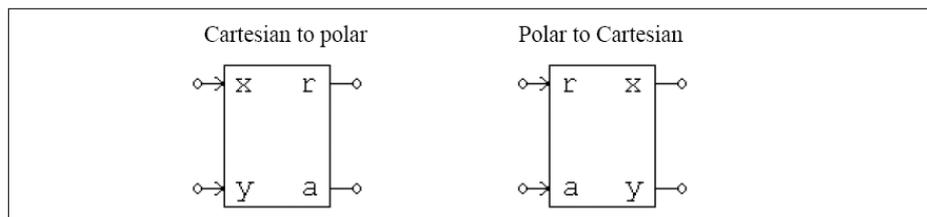
$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

dq座標系からαβ座標系への変換公式は以下の通りです：

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \end{bmatrix}$$

4.8.2.4 直交座標-極座標変換ブロック

直交座標-極座標変換ブロックは直交座標表現から極座標表現に軸変換及び逆変換します。



イメージでは、文字「r」は振幅を表します。また、文字「a」は位相角θ(単位、rad)を表します。

直交座標系から極座標系への変換公式は以下の通りです：

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \text{atan}\left(\frac{y}{x}\right)$$

極座標系から直交座標系への変換公式は以下の通りです：

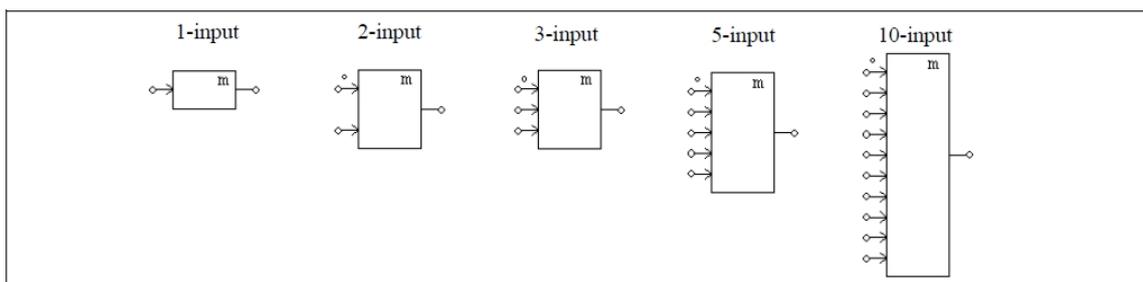
$$x = r \cdot \cos\theta$$

$$y = r \cdot \sin\theta$$

4.8.3 数式関数ブロック

数式関数ブロックの出力は入力変数の関数として表現されます。このブロックを使うことにより、複雑かつ非線形の入出力関係を簡単に実現できます。入力数が1,2,3,5,10のブロックが用意されています。

シンボル



仕様

パラメータ	機能
Expression $f(x_1, x_2, \dots, x_n)$	入出力関係を与える数式表現 ; n は入力の数です。
Expression df/dx_i	関数 f の i 番目の入力 x_i に関する微分の数式表現

微分はゼロに設定してもかまいません。

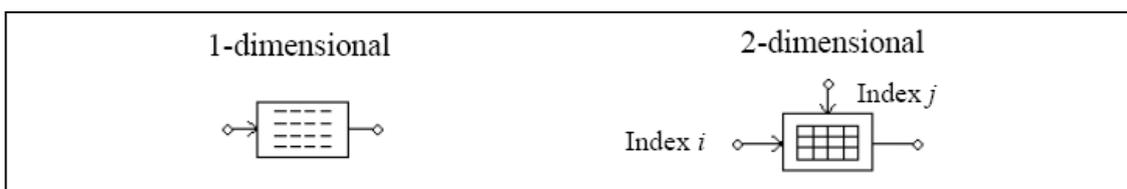
関数のなかで使える変数は、時刻 T または t および入力変数 x_i (i は 1 から n まで) です。ここで変数 x_i は i 番目の入力を示します。たとえば、3 入力の数式関数ブロックを使用した場合、変数は T, t, x_1, x_2, x_3 となります。1 入力の数式関数ブロックでは、入力 x も変数とみなされます。

4.8.4 ルックアップテーブル (Look-up Table)

二種類のルックアップテーブルが存在します。一つは 1 次元ルックアップテーブル(LKUP)ともう一つは 2 次元ルックアップテーブルです。さらに、二種類の二次元ルックアップテーブルが存在し、それぞれが整数入力(LKUP2D)タイプと浮動小数入力(LKUP2D_1)タイプになっています。

三種類のルックアップテーブルはともにパワー回路と制御回路のいずれにも適用できます。ただし、Lookup テーブルファイル(.tbl)はテキスト編集ソフト (Notepad など) で作成後、回路図 (.sch)と同じフォルダに保存する必要があります。

シンボル



仕様

パラメータ	機能
File Name	ルックアップテーブル保存ファイルのファイル名

2 次元のルックアップテーブルブロックの左側の端子は行のインデックス入力用で、上部にある端子は列のインデックス入力用です。

1 次元のルックアップテーブルでは入力と出力が一つずつ付いています。入力と出力に対応する二つのデータの配列がファイルのルックアップテーブル内に保存されています。そのフォーマットは以下の通りです。

$V_{in}(1), V_{o}(1)$
 $V_{in}(2), V_{o}(2)$
 ...
 $V_{in}(n), V_{o}(n)$

左のカラムの V_{in} は単調に増加する必要があります。入力が2つのデータの間の値の場合、出力値は線形補完された値になります。また、入力が $V_{in}(1)$ より小さい場合および $V_{in}(n)$ より大きい場合は、出力は $V_o(1)$ または $V_o(n)$ にクランプされます。

整数入力の二次元ルックアップテーブルには二つの入力があります。一方、出力データは 2 次元の行列の形で保存されています。二つの入力は、行列の行と列を指しています。例えば、行のインデックスが 3 で、列のインデックスが 4 の場合、出力は $A(3,4)$ になります。ただし、 A は行列のデータを意味します。ルックアップテーブルのデータはファイルに保存され、以下のフォーマットになります。

```
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)
```

ただし、 m と n がそれぞれ行と列の番号です。行または列のインデックスは必然的に整数である必要がありますので、入力値は自動的に整数に変換されます。行または列のインデックスのいずれかが有効範囲から外れた（例えば行のインデックスが 1 より小さいまたは m より大きい）場合、出力はゼロになります。

浮動小数入力用二次元ルックアップテーブル(LKUP2D_1)は整数入力用二次元ルックアップテーブル(LKUP2D)に似た機能を持ちます。両者の違いとしては、入力が浮動小数であるケースです。そのとき、出力の計算に補間が使われます。

ルックアップテーブルのデータはファイルに保存され、以下のフォーマットを持ちます：

```
m, n
Vr,1 Vr,2 ... Vr,m
Vc,1 Vc,2 ... Vc,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)
```

ただし、 m は行の番号、 n は列の番号、 V_r は行のベクトル、 V_c は列のベクトル、 $A(i,j)$ は i 行目と j 列目における出力の値です。ベクトル V_r と V_c は単純増加になるように指定してください。

入力が二つのポイントの区間内にある場合、値の計算に補間が使われます。入力が最小値より小さいかまたは最大値より大きい場合、入力是最小値ないし最大値にセットされます。

例：

以下では 1 次元ルックアップテーブルを示します。

```
1., 10.
2., 30.
3., 20.
4., 60.
5., 50.
```

入力が 0.99 であるとき、出力は 10 になります。また、入力が 1.5 である場合、出力は

$$10 + \frac{(1.5-1) \times (30-10)}{2-1} = 20 \text{ になります。}$$

以下に整数入力用二次元ルックアップテーブルを示します。

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

行のインデックスが2で、列のインデックスが4の場合、出力は8になります。行のインデックスが5の場合、列のインデックスに関係なく出力が0になります。

以下では浮動小数入力用二次元ルックアップテーブルを示します。

3, 4
 1.1 2.2 3.3
 1.2 2.3 3.4 4.5
 1., -2., 4., 1.
 2., 3., 5., 8.
 3., 8., -2., 9.

行の入力が2.0で、列の入力が3.0である場合、入力値を囲む4つのポイントのデータを利用して、以下のように補完計算が行われ、出力は3.826になります。

		Column		
		2.3	3.0	3.4
Row	1.1	-2		4
	2.0	2.091	3.826	4.818
	2.2	3		5

4.8.5 Cブロック

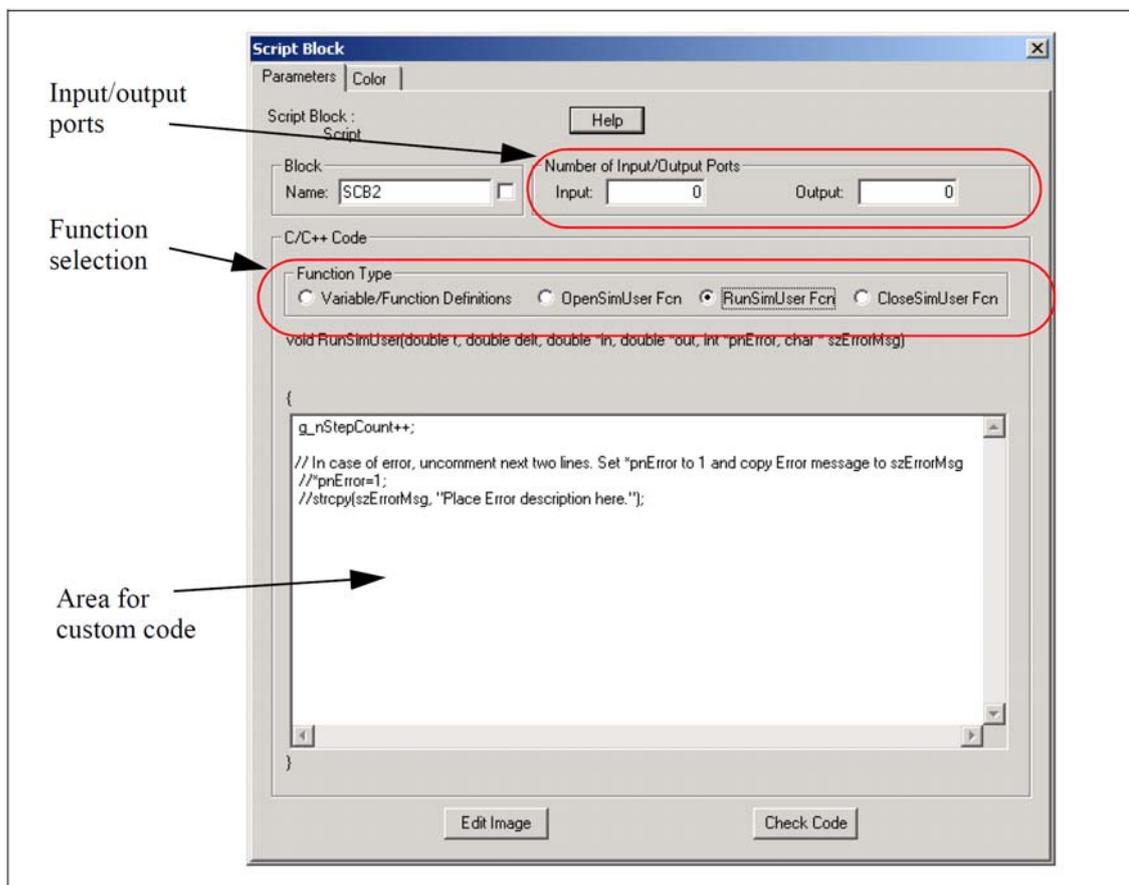
Cブロック (C Block) は、ユーザのCコードを直接入れることができます。Cインタプリタにより、Cコンパイラがなくても、Cコードを記述することができます。組み込みソフトウェアデバイスのシミュレーションやブロック関数の定義ができます。CブロックのCコードは、シミュレーション実行時にPSIM内蔵のCインタプリタによって解釈され実行されます。

Cブロックダイアログウィンドウのインターフェースを以下に示します。Number of Input/Output Portsセクションでは、入力ポート及び出力ポートの数を定義します。ポートの数を変えた後、回路図でのブロック・図はそれによって変化します。

Function Typeでは、下記4つの選択肢があります:

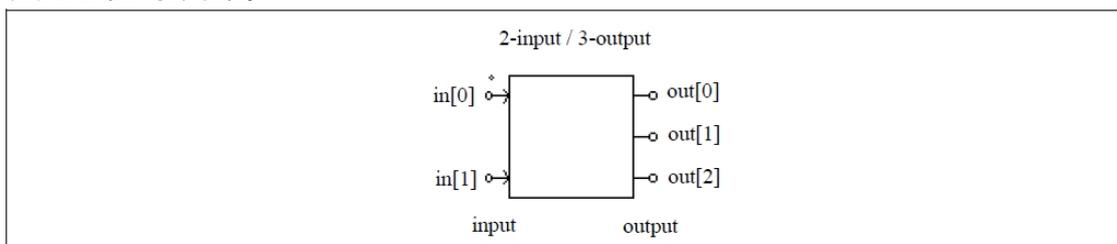
- Variable/Function Definitions: インクルードファイルとグローバル変数の定義を行います。
- OpenSimUser Fcn: 初期化のためのシミュレーションの初めに一度だけ呼ばれる関数を記述します。
- RunSimUser Fcn: 各シミュレーションステップに呼ばれる関数を記述します。
- CloseSimUser Fcn: 終了のためのシミュレーションの終わりに一度だけ呼ばれる関数を記述します。

上記Typeの中の1つが選択される時、その下部領域には対応するコードが示されます。全Cコードは、4タイプすべての記述を結合したコードから成ります。



記述したCコードにコンパイラエラーがあるかどうかチェックするために**Check Code**ボタンをクリックしてください。Cブロックの図をカスタマイズするために**Edit Image**ボタンをクリックしてください。

例として、2入力、3出力のCブロックを考えます。入出力のポート数が決まったらCブロックの素子画像は以下ようになります。



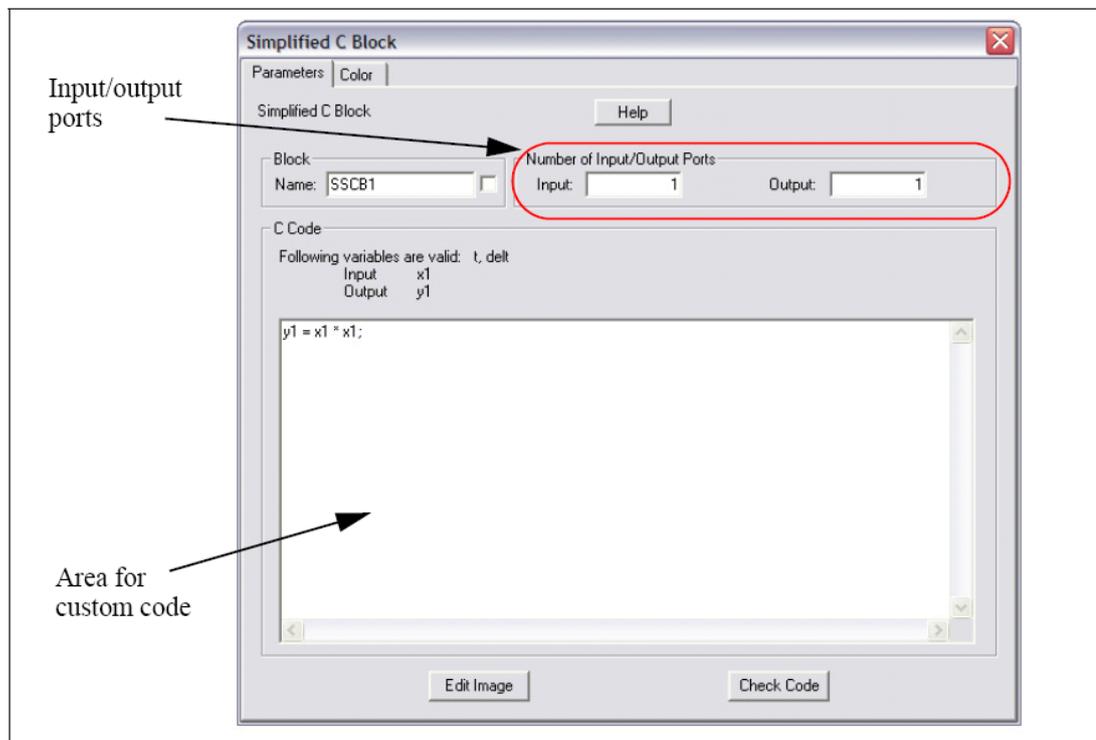
ノードは上から下へ番号が振られます。

Cコードでは、PSIMからCブロックへ値を受け渡すためin配列が使われ、Cブロックから値を返すためout配列が使われます。この例では、最初の入力[0]が左上に位置し、最初出力[0]が右上に位置します。Cブロックと外部DLLブロックの違いは、Cブロックは簡単に使えるものの、カスタムコードをデバッグすることができないという点にあります。外部DLLブロックでは、デバッグするためにブレークポイントをセットすることが可能です。

4.8.6 シンプル C ブロック

シンプル C ブロックは C ブロックの簡易版です。C ブロックが変数定義領域などを持っているのに対し、シンプル C ブロックは C コードだけを含んでおり、C コードはタイムステップごとに呼んで実行されます。

シンプル C ブロックダイアログウィンドウのインターフェースは以下の通りです：



Number of Input/Output Ports セクションでは、入出力ポートの数を定義します。ポートの数を変えると、シンプル C ブロックのイメージはそれに従って変わります。

以下の変数をコードで使用することが可能です：

t:	PSIM から渡される時間
delt:	PSIM から渡される刻み
x1, x2,:	入力 1, 2,
y1, y2, ...:	出力 1, 2,

C ブロックとは異なり、シンプル C ブロックは自動コード生成に使用することができます。

4.8.7 外部 DLL ブロック

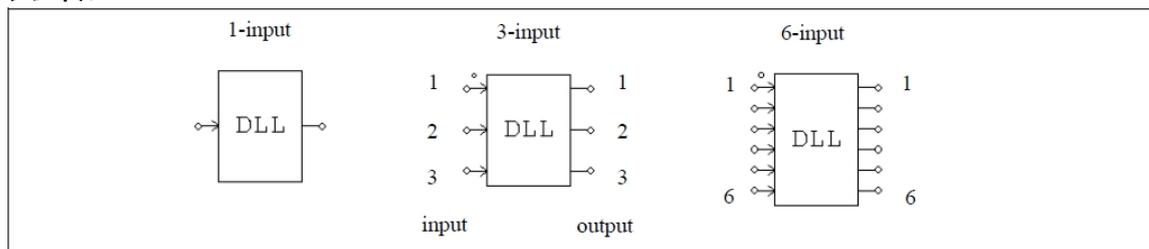
外部 DLL (Dynamic Link Library) ブロックを使うことでユーザ自作の C 言語または Fortran のプログラムを追加することができます。それには Microsoft C/C++、Borland C++、または Digital Visual Fortran を使い、プログラムを DLL としてコンパイルし、PSIM にリンクして使います。これらのブロックはパワー回路または制御回路として使うことができます。

DLL ブロックは PSIM から入力値を受け取り、計算処理後、出力を PSIM に返します。PSIM は DLL をシミュレーションタイムステップごとに呼び出します。ただし、DLL ブロックの入力端子が離散素子（ゼロ次ホールド、単位時間遅れ、離散積分器、離散微分器、z-領域伝達関数およびデジタル・フィルタのいずれか）に接続されているときは、DLL は離散的なサンプル時刻にのみ呼び出されます。

DLL ブロックでは Simple DLL ブロックと General DLL ブロックの二つタイプが提供されます。Simple DLL ブロックは入出力端子の本数が固定されていて、ブロックで設定する必要があるのは DLL ファイル名のみです。これに対し、General DLL ではユーザが入出力端子の本数を自由に設定することが可能で、またユーザが設定した任意の追加パラメータを使用することが可能です。また General DLL ブロックはシンボルをカスタマイズすることも可能です。

一般的には Simple DLL ブロックが簡単なプログラミングで利用できて便利です。
Simple DLL ブロックのモデルと仕様は下のようになります。

シンボル



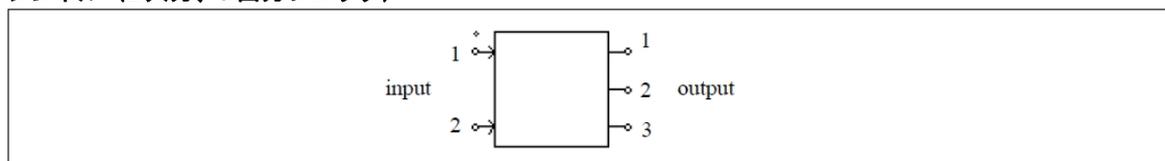
仕様

パラメータ	機能
File Name	DLL ファイルのファイル名

ドットの付いた端子には最初の入力 (in[0]) を加えます。入出力端子の順番は上から下です。

General DLL ブロックのモデルと仕様は下のようになります。

シンボル (2 入力、3 出力ブロック)



仕様

パラメータ	機能
DLL File	DLL ファイル名
Input Data File	DLL が読む入力データファイル名 (オプション)
Number of Input Nodes	入力端子の数 (オプション)
Number of Output Nodes	出力端子の数 (オプション)
In Nodes	入力端子のリスト (オプション)
Out Nodes	出力端子のリスト (オプション)
Parameter 1	PSIM から DLL に渡すパラメータ (オプション)
Parameter 2	PSIM から DLL に渡すパラメータ (オプション)
Edit Image (button)	DLL ブロックモデルを編集、作成します。
Display File (button)	入力データファイルの内容を示します。(オプション)
Read File (button)	入力データファイルが修正されたとき、このボタンを押すとデータファイルを再読み込みします。(オプション)

ドットの付いた端子には最初の入力 (in[0]) を加えます。入出力端子の順番は上から下です。

デフォルトの設定ダイアログでは、ユーザは入出力の数を定義します。入出力の数と端子名、パラメータの数とパラメータ名は、全て DLL の中で定義できます。

General DLL ブロックは強力なカスタマイズが可能な反面、C 言語および Windows プログラミングに関する高度な知識を要求されます。必要性に合わせて通常の DLL ブロックと使い分けることをお勧めします。本機能に関する詳細な解説は、PSIM をインストールしたフォルダの doc フォルダにある「Help General DLL Block.pdf (英文)」およびサンプルファイルをご覧ください。

DLL のファイル名は任意で、以下の 3 種類のフォルダのいずれかに置くことができます。DLL ファイルは以下の順序で検索されます。

1. PSIM をインストールしたフォルダ (PSIM.exe が存在するフォルダ)
2. 回路図ファイル (.sch ファイル) が存在するフォルダ
3. **Options -> Set Path** で設定したフォルダ

PSIM ではサンプル DLL ファイルが提供されて、ユーザがこれらをテンプレートとして用い、作成することができます。DLL をコンパイルする方法と PSIM にリンクさせる方法を以下に説明します。サンプルファイルの使い方を示します。

このプログラムはマイクロソフト C/C++5.0 または 6.0 のためのサンプル C プログラムです。また、PSIM にリンクさせる DLL 作成法です。

DLL としてプログラムをコンパイルするために、「msvc_dll.dsw」というワークスペースファイルを開きます。または、以下の手順に従ってプロジェクトを作ります。

- ① 「C:\msvc_dll」というディレクトリを作成し、PSIM と共に使用する「msvc_dll.c」を「C:\msvc_dll」の中にコピーして下さい。
- ② Visual C++を起動し、ファイルメニューから、「New」を選び、「プロジェクト」ページで、「Win32 Dynamic-Link Library」を選択してください。そして、Project name を「msvc_dll」として、場所を「C:\msvc_dll」と設定してください。そして、「Create new workspace」を選択し、「Platform」として「Win32」を選択してください。
- ③ 「What kind of DLL would you like to create?」と尋ねられますので、バージョン 6.0 では、「An empty DLL project.」を選択してください。
- ④ 「プロジェクト」メニューから、「Add to Project"/"Files...」を選択し、「msvc_dll.c」を選択してください。
- ⑤ 必要に応じてコードを加えてください。
- ⑥ 「Build」メニューから、「Set Active Configurations...」、そして、「Win32 Release」を選択してください。「Build」メニューから、「Rebuild All」を選ぶと、「msvc_dll.dll」という DLL ファイルが生成されます。DLL ファイルは「C:\msvc_dll\release」というディレクトリに保存されます。
- ⑦ DLL ファイルに名前をつけてください。例えば、回路図ファイル(.sch ファイル)が「test_msvc_dll.sch」ならば、「test_msvc_dll.dll」となります。
- ⑧ 回路図ファイル(.sch ファイル)と同じディレクトリに DLL ファイルをコピーしてください。回路中で、外部の DLL ブロックファイルを生成した DLL ファイル(例えば、この場合、「test_msvc_dll.dll」と)指定してください。これで、自作した DLL ファイルを使用して PSIM を実行する準備ができました。

このサンプルプログラムは、60Hz の入力 in[0]の実行値について計算して、out[0]に出力を記憶します。

ファイルが C++ファイル(「msvc_dll.cpp」)、extern "C"であるなら以下の系列を利用可能です。
このプログラムの変数名を変える(「t」から「Time」のように)ことができますが、関数名、変数の数、変数の種類、およびシーケンスを変えないでください。

・ 変数:

t: Time PSIM から値が入力されます。
delt: Time step PSIM から値が入力されます。
in: input array PSIM から指令として入力されます。
out: output array 計算後、PSIM に値が送り返されます。(注意:out[*]の値は PSIM で修正されることもあります。)

・ 入出力配列の“in”と“out”の最大の長さは 30 です。

・ 警告: simuser 関数(t,delt,in,out)上のグローバル変数は許容されていません。

```
#include <math.h>
__declspec(dllexport) void simuser (t, delt, in, out)

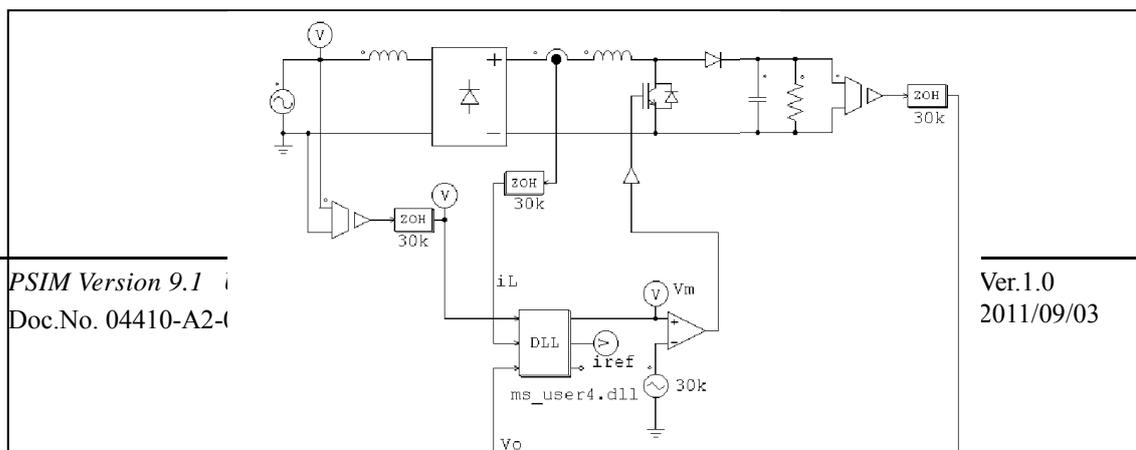
// すべての変数を「double」と定義しなければならないのに注意してください。
double t, delt;
double *in, *out;

{
// ここから本文を書きます。
// 値を保持するために「sum」を「static」と定義してください。
    static double nsum=0., sum=0., rms;
    double Tperiod;

    Tperiod=1./60.;
    if (t >= nsum*Tperiod)
    {
        nsum=nsum+1.;
        rms = sqrt(sum*delt/Tperiod);
        sum=0.;
    }
    out[0] = rms;
    sum=sum+in[0]*in[0];
// ここで本文が終わりです。
}
```

例.

次の例はインダクタ電流と負荷電圧フィードバックを使った力率補正回路です。入力電圧は電流指令値を発生させるのに使っています。制御回路はデジタルで、サンプリング周波数は 30kHz です。この制御回路は外部の C プログラムで作られ、DLL によりパワー回路と組み合わせています。



DLL ブロックの入力は入力電圧のサンプル値、インダクタの電流、および出力電圧です。DLL ブロックの出力のひとつは変調波形 V_m で、これをキャリア波形と比較して PWM のゲート信号を発生させています。もうひとつの出力はインダクタ電流の指令値で、モニタ用に使っています。

ソースコード (pfc_vi_dll.c) をご覧ください。内側の電流ループと外側の電圧ループはともに PI 制御器を使っています。制御器を離散型にするために、台形ルールを使っています。後退オイラー法による離散化手順もありますが、ここではコメントにしています。

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

4.8.8 組み込みソフトウェアブロック

組み込みソフトウェアブロックは独自モジュール（ユーザが独自に作成を行うモジュール）内の DLL ブロックの一つで、機能を拡張した General DLL ブロックであると考えて下さい。組み込みソフトウェアブロックは制御回路素子で、制御回路中でのみ使うことができます。以下の条件を満たすものを使用したい場合に使用して下さい。

- ・制御プログラムを使う。
- ・マイクロコントローラや DSP などのように、各ポートが入力ポートであるか出力ポートであるかを自由に変えたい場合。
- ・各パラメータの変化に時間を考慮したい場合。

仕様

パラメータ	機能
DLL File	DLL ファイル名
Number of Nodes	入出力端子の合計数

組み込みソフトウェアブロックについての詳しい説明は PSIM インストールフォルダの Doc フォルダにあります「Help Embedded Software Block.pdf（英文）」を参照して下さい。

第5章 解析

5.1 過渡解析

PSIM の Simulate メニューから Simulation Control を選ぶことにより、以下のシミュレーション設定を変更することができます。

シミュレーション設定パラメータ	
Time Step	シミュレーションタイムステップ(sec)
Total Time	シミュレーション総時間(sec)
Free Run checkbox	Free Runチェックボックスをチェックしない時、シミュレーションは、総時間で止まります。Free Runをチェックすると、シミュレーションはフリーランモードで動き、手動で停止するまで動き続けます。 フリーランモードで、シミュレーション中の電圧と電流をモニタするのに電圧/電流のスコープを使用できます。
Print Time	シミュレーション結果の出力ファイルへの保存を始める時刻。この時刻より前のシミュレーション結果は保存されません。
Print Step	結果保存の周期：Print Step = 1 のとき、すべてのデータが出力ファイルに保存されます；Print Step = 10 とすると、10 回に 1 度だけ結果が保存されます。この設定により、出力ファイルの大きさを制限することができます。
Load Flag	LOAD 機能のフラグ：Load Flag = 1 のとき、前回のシミュレーション結果をファイル（拡張子：.ssf）から初期状態として読み込みます。
Save Flag	SAVE 機能のフラグ：Save Flag = 1 のとき、現在のシミュレーションが終了した時点の数値をファイル（拡張子：.ssf）に保存します。
Hardware Target	これはSimCoderを使用して自動コード生成するためのハードウェアターゲットを指定するものです。 <ul style="list-style-type: none"> - None: ハードウェアターゲットの指定なし - TI F28335: TI F28335 ハードウェアターゲット - PE-Pro/F28335: PE-Pro/F28335ハードウェアターゲット - PE_Expert3: PE-Expert3 ハードウェアターゲット - General_Hardware: 汎用DSP PE_Expert3を選択した場合、ドロップダウンでPE-Viewのバージョンの指定ができます。詳細はSimCoderマニュアルをご参照ください。

SAVE および LOAD 機能を使うことにより、回路の電圧・電流その他の値をひとつのシミュレーションが終了した時点で保存し、次のシミュレーションを実行するときに初期状態として読み込むことができます。この機能により、長時間のシミュレーションをいくつかの短いシミュレーションに分け、タイムステップとパラメータを変えて実行できます。素子の変数およびパラメータはシミュレーションごとに設定を変えることができますが、回路構成は同じでなければなりません。

PSIM では、シミュレーションの実行中のタイムステップは不変です。正確な結果を得るために、タイムステップを適切に選ぶ必要があります。適切なタイムステップは、スイッチの動作時間、パルス・波形の

幅、過渡現象の長さ、などにより決まります。タイムステップは少なくともこれらの現象のうち最も短いものよりも一桁小さく設定することをお奨めします。

Version6.0 では、補完機能により、スイッチングのタイミングに正確に合わせて計算を行うことができるようになりました。このため、スイッチ時刻と離散時間のシミュレーションが同期していない場合の誤差が格段に減少しました。長いタイムステップでシミュレーションを実行した場合も精度よい結果が得られます。

許容できる刻み時間の最大値は PSIM が自動的に計算します。これをユーザが設定した刻み時間と比較し、シミュレーションではどちらか小さな値を使います。もし、PSIM がユーザが設定したタイムステップよりも小さいタイムステップを使う場合は、message.doc のファイルにコメントを書き込みます。

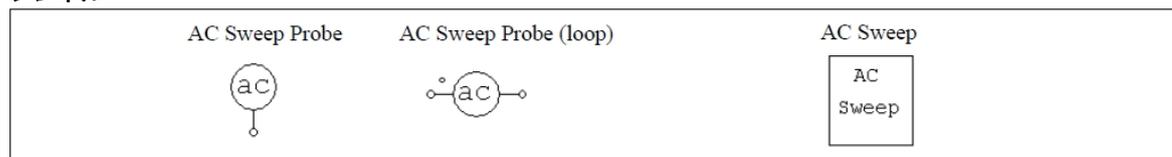
5.2 周波数特性解析

周波数特性解析により主回路もしくは制御ループの周波数応答を求めることができます。PSIM を使った周波数特性解析の特徴は、回路がスイッチ素子を含む場合も、平均値による解析に頼ることなく、実際のスイッチ動作を忠実に模擬できることです。

PSIM で周波数特性解析を行う際の手順を以下に示します。

- 交流スイープの励起電源に使う正弦波電源 (VSIN) を同定する。
 - 出力を観測したい位置にスイープ用のプローブ (ACSWEEP_OUT) を置く。閉ループの制御応答を観測するには、ノード間プローブ (ACSWEEP_OUT2) を使います。
 - ACSWEEP 素子を回路に置き、パラメータを設定する。
 - PSIM のシミュレーションを実行する。
- 以下に交流スイープ用素子のシンボルと仕様を示します。

シンボル



仕様

パラメータ	機能
Start Frequency	交流スイープの開始周波数 (Hz)
End Frequency	交流スイープの終了周波数 (Hz)
No. of Points	データ数
Flag for Points	データ点のフラグ： Flag=0 のとき、周波数は LOG10 のスケールで等間隔に変化します。 Flag=1 のとき、周波数は線形のスケールで等間隔に変化します。
Source Name	スイープ対象の励起電源の名称
Start Amplitude	励起電源の開始周波数における振幅大きさ
End Amplitude	励起電源の終了周波数における振幅大きさ
Freq. for extra Points	追加データ点の周波数。周波数特性がある区間で急激に変化する場合は、この区間にデータ点を追加することにより、より詳細な解像度が得られます。 ※使用しない場合は空白にしてください。0 を入れると計算が終わらなくなる可能性があります。

周波数特性解析の原理は交流励起信号を回路に抽入して、出力の観測点で同じ周波数の信号を取り出すことにあります。周波数特性解析の結果を正確にするには、励起電源の振幅を適切に設定する必要があります。応答が線形領域にとどまるように振幅は小さく設定することが必要です。一方、出力の精度が数値誤差の影響を受けないように、振幅は充分大きくする必要があります。

一般に物理系は周波数の低い領域では減衰が低く、周波数の高い領域では高い傾向があります。したがって、励起電源は開始周波数では振幅が小さく、終了周波数では大きな設定とするのが良いでしょう。

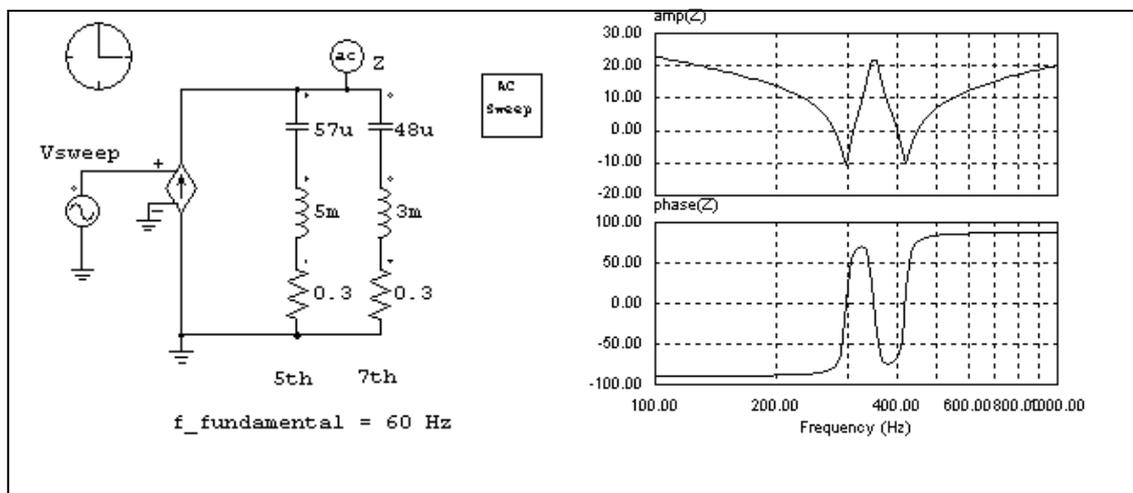
場合によって、周波数特性解析の終了後に次の警告メッセージが表示されることがあります：

Warning: The program did not reach the steady state after 60 cycles. See File "message.doc" for more details.

この警告メッセージは、交流スイープを開始して 60 サイクル後になっても定常状態を検出できなかったときにが表示されます。この問題を解決するには、回路の制動を増す（たとえば抵抗を追加する）か、励起電源の振幅を調整する、あるいはシミュレーションのタイムステップを短くする、などの対策が考えられます。“message.txt”というファイルには、この現象が起こった周波数、また相対誤差などといった情報が書き込まれています。相対誤差はデータ点が定常状態からどれだけ離れているかを示しています。

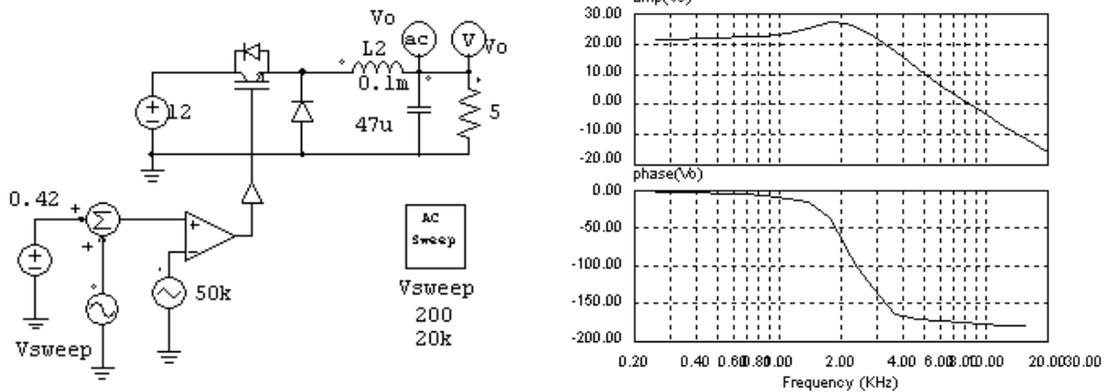
例. 並列フィルタのインピーダンス

以下に示す回路は基本波周波数 60Hz に対して 5 次と 7 次の高調波フィルタを構成しています。励起電源として電流を抽入し電圧を測定することで、このフィルタのインピーダンス特性を得ることができます。図の左に示す周波数特性解析の波形はあきらかに 300Hz と 420Hz に谷があることを示しています。



例. コンバータの開ループ応答

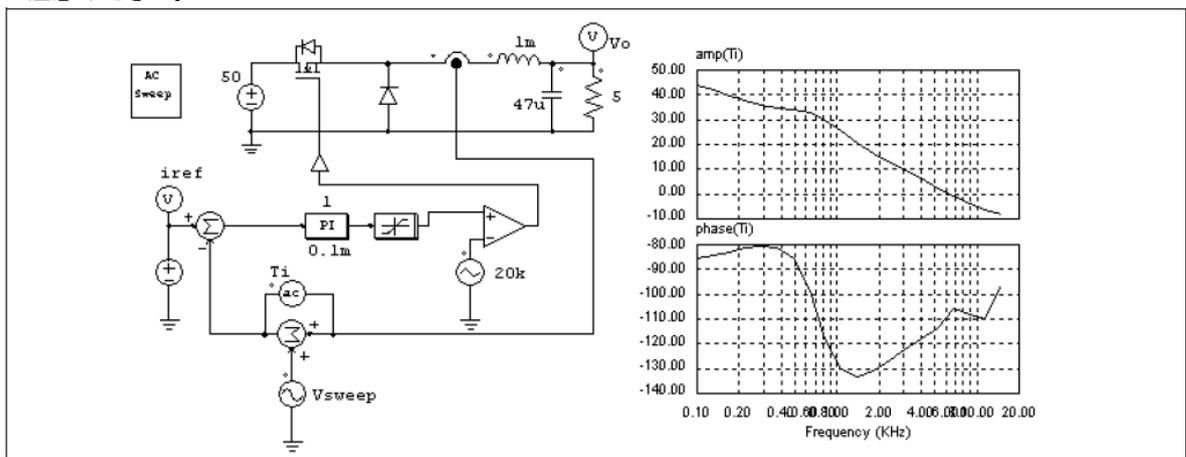
以下に示す回路は直流チョップ回路で、変調率信号に周波数特性解析のための正弦波電圧を加算し、出力電圧を測定しています。右に示すのは周波数特性解析の結果で、変調率入力に対する出力電圧の周波数応答を示しています。



例. 閉ループ回路の一巡伝達関数

周波数特性解析はまた、閉ループ回路の応答を調べるのに使うことができます。以下に示す回路は平均電流制御付きのコンバータ回路です。電流フィードバック回路に励起電源を挿入し、ノード間プローブ (ACSWEEP_OUT2) により観測することにより、このループの一巡伝達関数を直接得ることができます。こうして得られた一巡伝達関数から、ユーザはさらにバンド幅と位相余裕を求めることができます。

交流スイープ用のプローブはドットの付いた側が励起電源の挿入点よりも後に接続されているところにご注意ください。



例. スイッチ・モード電源の伝達関数

PWM 用 IC を制御に使ったスイッチ・モード電源の伝達関数も同様にして求めることができます。以下の回路ではコンバータの制御に Unitrode UC3842 を使っています。励起電源はフィードバック回路の中でオペアンプの出力の前に接続しています。

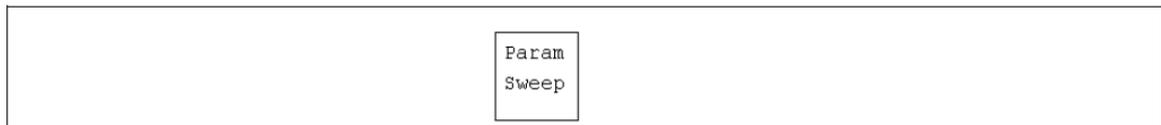
5.3 パラメータのスイープ

PSIM では次のパラメータに関してスイープ解析をおこなうことができます：

- RLC ブランチ回路の抵抗、インダクタンス、キャパシタンス
- 比例制御ブロック (P) のゲイン
- 積分器 (I) の時定数
- PI (比例積分) 制御器のゲインと時定数
- 2 次のローパスまたはハイパスフィルタ (FILTER_LP2、FILTER_HP2) のゲイン、カットオフ周波数、およびダンピング比
- 2 次のバンドパスまたはバンド阻止フィルタ (FILTER_BP2、FILTER_BS2) のゲイン、中心周波数、およびバンド幅

パラメータスイープ用素子のシンボルと仕様を以下に示します。

シンボル



仕様

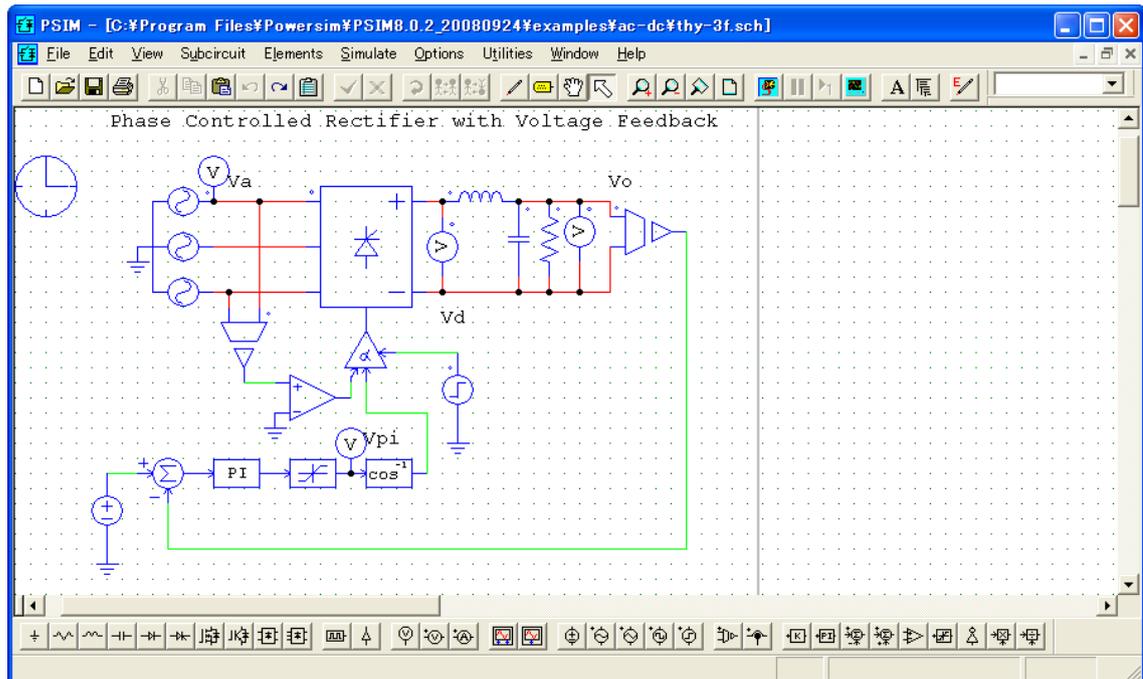
パラメータ	機能
Start Value	パラメータスイープの開始値
End Value	パラメータスイープの終了値
Increment Step	刻み幅
Parameter Name	スイープするパラメータの名称

たとえば、抵抗器の抵抗を R_o とします。この抵抗器を 2 Ohm から 10 Ohm までスイープするには以下のように設定します。

Start Value	2
End Value	10
Increment Step	2
Parameter Name	R_o

第6章 PSIM による回路図入力

PSIM には、回路図設計用の対話的で使いやすいインターフェースが備わっています。次の図に PSIM 環境での整流器回路を示します。

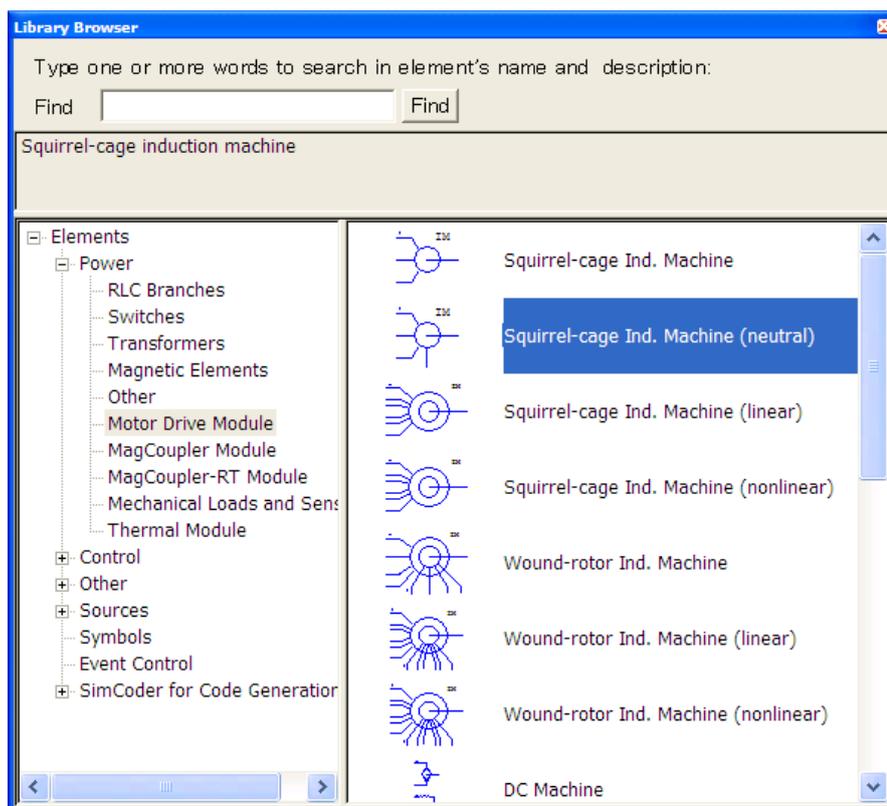


全ての PSIM の構成要素は Elements というメニューの下にまとめられています。要素は 4 つのグループに分かれます：パワー回路（Power；パワー回路用素子）、制御回路（Control；制御回路用素子）、その他（Other；スイッチ制御器、センサ、プローブ、インターフェース機器、その他パワーと制御に共通する要素）、および電源（Sources；電圧源と電流源）です。

6.1 回路の作成

PSIM では回路作成のために以下のような機能が備わっています。

- Get Element** 要素ライブラリから要素を取り出すために、Elements メニューをクリックします。サブメニューを選択し、選択する要素をハイライトしてください。例えば直流電圧源を得るには、Elements → Sources → Voltages をクリックして Vdc をハイライトしてください。
- 更に、**View->Library Browser** をクリックすることで、ライブラリブラウザを利用して要素を選択することができます。



- Place** メニューから要素を選択すると、要素のシンボルが画面上に現れ、マウスとともに移動します。要素の位置を決めるにはマウスの左ボタンをクリックしてください。要素を配置する前に回転する場合は、要素を選択した後、右クリックすると回転します。配置後に回転する場合は  をクリックします。
- Rotate**
- Wire** 2つのノードを接続するために、 をクリックしてください。ペンのシンボルが現れます。結線を描くには、マウスの左ボタンを押し続けてドラッグしてください。結線は常にグリッドの交点に始まり交点に終わります。視認性を良くするために接続されていないノードは○で表され、接続されたノードは●で表されます。
- Label** 複数のノードが同一のラベルに接続している場合、それらは結合され、ワイヤによる接続と同等のものになります。ラベルを使うことにより、クロスワイヤリング（混線）が減少し、回路図のレイアウトが改善されます。
ラベルのテキストは移動可能です。テキストを選択するには、ラベルを左クリックして、Tab キーを押します。
- Assign** 要素のパラメータを割り当てるために、要素をダブルクリックしてください。ダイアログ・ボックスが現れます。値を明記してリターンキーを押すか OK をクリックしてください。
- Move** 要素や回路のブロックを移動するには、要素、回路ブロックを選択して、マウスの左ボタンを押し続けながらドラッグしてください。
- Move Schematic** 回路全体を選択してマウスの左ボタンを押し続けながらドラッグすると、選択した範囲を移動することができます。

6.2 ファイルメニュー

ファイルメニューには以下の機能が提供されています：

New	新しい回路図を作成します。
Open	すでに作成した回路図ファイルを開きます。
Close	回路図ファイルを閉じます。
Close All	すべての回路図ファイルを閉じます。
Save	現在の回路図ファイルを保存します。
Save As...	現在の回路図ファイルを別の名前で保存します。
Save All	すべての回路図ファイルを保存します。
Save with Password	パスワードで回路図ファイルを保存します。パスワードで保護されたファイルを利用してシミュレーションは可能ですが、回路図の内容を見るために正しいパスワードを入力する必要があります。
Save in Package File	回路図とすべての関連ファイルを 1 個のパッケージに保存するには、この機能を選択してください。複数のサブ回路含む回路図を誰かに送る必要がある場合、この機能は役に立ちます。
Save as Older Versions	バージョン 8.0 か 7.1 フォーマットでファイルを保存します。ファイルがバージョン 9.0 特有な要素を使用すると、これらの要素が省略されることに注意してください。
Print	回路図を印刷します。
Print Preview	プリントアウトをプレビューします。
Print Selected	回路図の選択した部分だけ印刷します。
Print Selected Preview	回路図の選択した部分だけプレビューします。
Print Page Setup	印刷ページの位置を調整と印刷ページの伝説を設定します。
Printer Setup	プリンタを設定します。
Exit	PSIM 回路図プログラムを終了します。

6.3 回路の編集 (Edit)

要素の上にマウスカーソルを置いてマウスの左ボタンをダブルクリック、または右ボタンをクリックすると、要素のプロパティウィンドウが表示されます。また、マウスの右ボタンを押しながらドラッグすると、画面をスクロールすることができます。

回路の編集用に Edit メニューと View メニューには以下のような機能が備わっています。

Undo	以前の変更を取消することができます。
Redo	以前の変更をやり直すことができます。
Cut	要素、回路のブロック、結線を消去するためには、対象を選択して、Cut を選んで下さい。Cut を使用すると最後に消去された図形をペーストできます。要素または回路の一部を削除するには<Delete>キーを押してください。
Copy	選択した要素や回路のブロックをコピーすることができます。
Paste	コピーした要素や回路を貼りつけることができます。
Select All	回路全体を選択できます。回路の一部を選択するために、マウスの左ボタンをクリックしてドラッグしてください。
Copy to Clipboard	回路図イメージを別のソフトウェアでペーストするときこの機能を選択してください。3 種類のフォーマットを選ぶことができます: メタファイルフォーマット、カラービットマップ、白黒 ビットマップ。メタファイルフォーマットはベクトルベースであり、イメージがスケーリングされているとき、きれいに表示されます。黒白ビットマップの場合はカラービットマップより小さいイメージファイルのサイズで起因します。
Drawing	展示目的で回路図にイメージを追加するためにこれを選択してください。次のイメージが使用できます: ライン、長円、長方形および半円。
Text	テキストを画面の上に置くには Text を選んでください。ダイアログ・ボックスにテキストを入力し、マウスの左ボタンをクリックして位置を決めてください。
Wire	2つのノードを接続するために、Wire をクリックして下さい。ペンのシンボルが現

	れます。
Label	複数のノードが同一のラベルに接続している場合、それらは結合されてワイヤによる接続と同等のものになります。ラベルを使うことにより、クロスワイヤリング(混線)が減少し、回路図のレイアウトが改善されます。
Attributes	選択した素子のプロパティダイアログウィンドウを使用します。
Add/Remove Current Scope	電流フラグ付き要素の電流スコープを追加と削除するためにこの機能を使用します。機能が選ばれた後、電流スコープを表示するために要素の上にクリックしてブランチ電流名を選んでください。電流スコープを削除するために、またブランチ電流名を選択してください。
Show/Hide Runtime Variables	シミュレーション中に素子のパラメータを変更する方法の一つです。これを選択後、素子をクリックするとその素子の可変パラメータの値が現われます。この値をダブルクリックして現れる入力フィールドに書き込むことで素子パラメータの変更ができます。
Disable	要素または回路の一部を「無効」にします。要素または回路が無効状態のとき、このオプションはグレー表示になり、現在のシミュレーションが関係している限り使用できなくなります。この機能は、要素または回路を「削除」ではなく「除外」する必要がある場合に便利です。
Enable	無効になっている要素または回路を「有効」にします。
Rotate	要素または回路の一部を時計回り 90°回転させるために Rotate をクリックします。
Flip Left/Right	要素を水平にフリップします。
Flip Top/Bottom	要素を垂直にフリップします。
Find	タイプと名前に基づく特定の要素を見つけるために選択してください。
Find Next	以前の Find 操作を振り替えすために選択して下さい。
Edit Library	PSIM イメージライブラリを編集するために Edit Library を選択して下さい。
Image Editor	イメージエディタを起動します。詳細は 6.10 を参照してください。
Escape	編集を中止します。

6.4 View メニュー

回路の編集用に View メニューには以下のような機能が備わっています。

Status Bar	ステータス・バーの表示/非表示を切替えます。
Toolbar	ツールバーの表示/非表示を切替えます。
Element Bar	要素バーの表示/非表示を切替えます。
Recently Used Element List	最近ライブラリから使用された要素をドロップダウンリストボックスに示します。
Library Browser	ライブラリブラウザを起動します。
Zoom In	回路図を拡大します。
Zoom Out	回路図を縮小します。
Fit to Page	回路全体はスクリーンに合うようにズームを調節します。
Zoom In Selected	選択したブロック図に拡大します。
Element List	回路図の部品リストを作成します。
Element Count	回路図の要素をカウントします。電圧・電流プローブ及び計測器は要素カウントに含まれていません。
Refresh	画面表示をリフレッシュします。

6.5 サブ回路 (Subcircuit)

サブ回路の編集および操作用に以下のような機能が備わっています。

以下の機能は主回路 (サブ回路外) に適用されます：

New Subcircuit	新しいサブ回路を作成します。
Load Subcircuit	既存のサブ回路をロードします。サブ回路は画面上にブロックとして現れます。

Edit Subcircuit	サブ回路のサイズとファイル名を編集します。
Display Subcircuit Name	サブ回路の名前を表示します。
Show Subcircuit Ports	メイン回路でサブ回路の端子を表示します。
Hide Subcircuit Ports	メイン回路でサブ回路の端子を隠します。
Subcircuit List	メイン回路とサブ回路のファイル名のリストを見るために使います。

以下の機能はサブ回路内に適用されます：

Set Size	サブ回路のサイズを設定します。
Place Bi-directional Port	サブ回路に双方向接続ポートを置きます。
Place Input Signal Port	サブ回路に入力信号ポートを置きます。
Place Output Signal Port	サブ回路に出力信号ポートを置きます。
Display Ports	サブ回路の接続端子を表示します。
Edit Default Variable List	サブ回路用のデフォルト変数を編集します。
Edit Image	サブ回路のシンボルを作成・編集します。
One Page up	メイン回路に戻ります。サブ回路は自動的に保存されます。
Top Page	低レベルのサブ回路から最上位のメイン回路にジャンプするために使います。これはサブ回路が多層になっている場合に有効です。

Set Size, Display Port, Edit Default Variable List, Edit Imageの設定を主回路で行った場合、主回路側でも適用されます。

主回路とサブ回路を接続するために3タイプのサブ回路ポートがあります：パワー回路、機械システム、そして制御回路の入出力用の双方向ポートです。

双方向ポートは制御回路で使用できますが、明快にするために制御回路の入力か出力信号ポートを使用することをお勧めします。また、サブ回路を利用してコード生成するときは入力か出力信号ポートしか使用できません。

サブ回路ブロックの上をマウスの右ボタンをクリックして**Attributes**を選択するとサブ回路のプロパティダイアログウィンドウを表示します。3つタブがあります：*Subcircuit Info*、*Subcircuit Variables*及び*Color*です。

Subcircuit Infoタブ:

このタブでは、サブ回路の名前を編集することができます。**Change Subcircuit File** ボタンをクリックして他のサブ回路を選択することができます。

Subcircuit Variables タブ:

このタブでは、サブ回路の変数の編集は可能です。例えば、1mΩ「Rparasitic」という名前の抵抗のラベルを「Parasitic Resistance」ある場合、以下のように記入できます：

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

「Parasitic Resistance」ラベルの次のチェックボックスをチェックすると、この変数は主回路に以下のように表示されます：

Parasitic Resistance = 1m

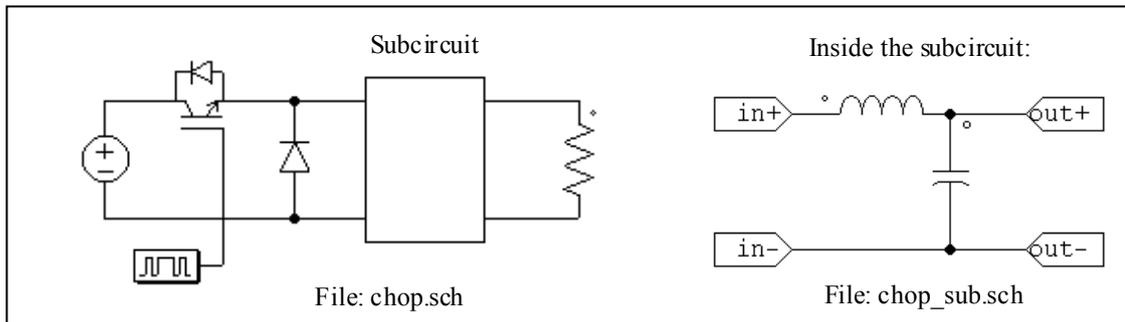
サブ回路の変数のリストは編集できますので、現在の変数のリストはデフォルト変数のリストと異なる可能性があります。**Set As Default Variables**ボタンをクリックして現在の変数リストをデフォルト変数リストとして設定することができます。または、**Reload Default Variables**ボタンをクリックしてデフォルトの変数のリストを変更になっている場合、デフォルト変数リストをリロード可能です。

SimCoderを利用してサブ回路の自動コードを生成することができます。サブ回路のコードを生成するために、**Generate Code**ボタンをクリックします。**Replace subcircuit with generated code for simulation**のチェックボックスをチェックすると、サブ回路の回路図の代わりに生成されたコードを取り替えられます。

Color Tab:

このタブでは、サブ回路の色を変えることができます。

以下のチョップ回路でサブ回路の利用の例を示します。



6.5.1 サブ回路作成 — メイン回路から

メイン回路でサブ回路を作成するステップは以下の通りです。

- Subcircuit メニューから、New Subcircuit を選んでください。
- サブ回路ブロック（長方形）が画面上に現れます。ここにサブ回路を置いてください。
- サブ回路のサイズとファイル名を指定するために Subcircuit メニューの Edit Subcircuit を選んでください。この例では、ファイル名は"chop_sub.sch"でサイズは4*7（幅4目盛、高さ7目盛）です。サブ回路は適当な大きさに結線が容易になるようにサイズを選択しなければなりません。

サブ回路を置いたら、サブ回路の周辺へ結線してください。サブ回路ブロックの4つの角のノードは結線には使用できません。

6.5.2 サブ回路作成 — サブ回路内部から

サブ回路に入るためには、サブ回路をダブルクリックしてください。いったんサブ回路の中へ入れば、メイン回路と同様に回路を作成、編集できます。

サブ回路が完成すれば、サブ回路の接続端子は、メイン回路の端子に対応するサブ回路の端子に接続するように指定しなければなりません。Subcircuit メニューにある Place Port を選ぶと、端子のシンボルが現れます。回路上に端子が置かれた後、ポップアップウィンドウが（左下に示すように）現れます。

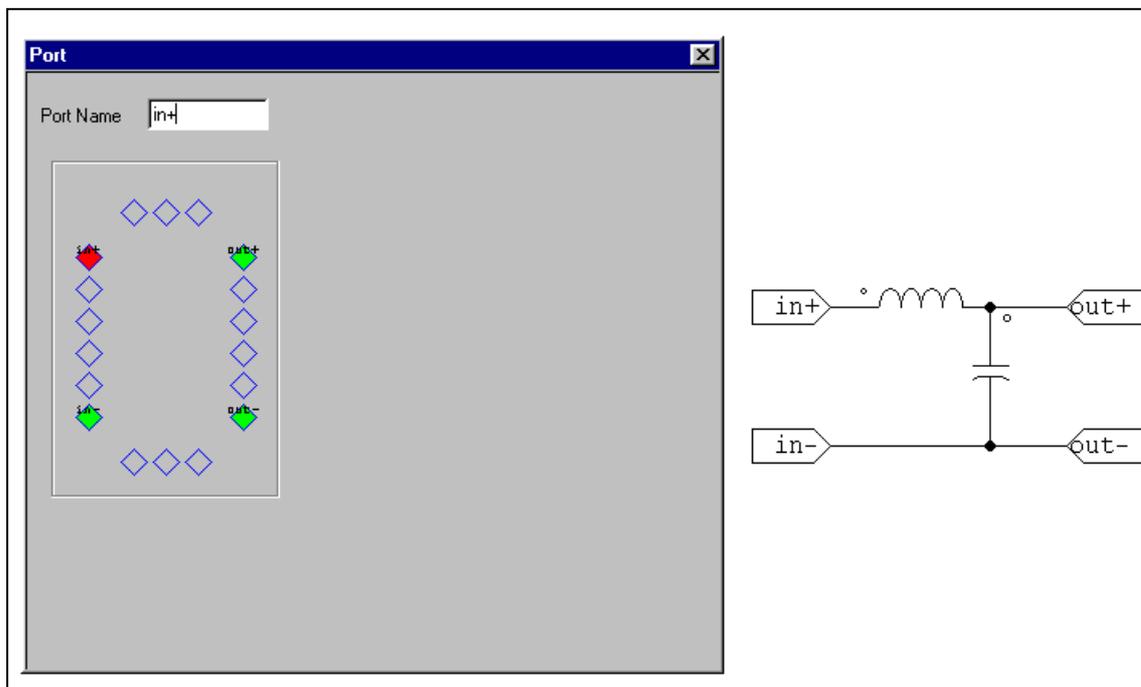
Subcircuit port assignments

右側のサ
ま
せん。
菱形が選
は、か
す。

右側に2つの、全4つの接続端子で、下部2つの端子は下端より1目

左上部の接続端子を指定するためには、左一番上の菱形をクリックして“in+”とタイプしてください。テキスト“in+”が菱形の内部に入り、端子に“in+”とラベルが付きます。端子を左上部の端子に接続してください。端子“in-”、“out+”、“out-”を接続するには同様の手続きを繰り返してください。

4つの端子が置かれた後、端子の割り当てとサブ回路がPSIMに以下のように現れます。



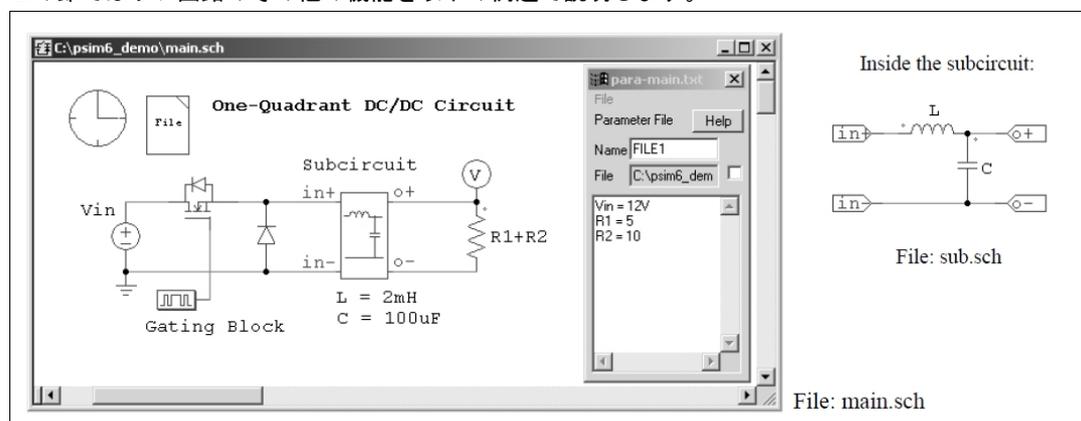
6.5.3 サブ回路の接続 — メイン回路で

サブ回路を作成し、接続端子を設定したら、サブ回路ブロックをメイン回路に接続します。

- メイン回路上でサブ回路ブロックの接続端子が境界上で白抜きの丸として現れます。
- サブ回路ブロックを指定し、Subcircuitメニューにある Show Subcircuit Ports を選ぶと、端子名がサブ回路内で設定したとおりに表示されます。
- 接続端子にメイン回路を接続します。

6.5.4 サブ回路のその他の機能

この節ではサブ回路のその他の機能を以下の例題で説明します。



6.5.4.1 メイン回路からサブ回路への変数値の受け渡し

この例題では、メイン回路 main.sch がサブ回路 sub.sch を使います。サブ回路内では、インダクタンスは L、キャパシタンスは C と定義されています。L と C のデフォルト値は **Subcircuit** メニューから **Set Default Variable List** を選ぶことで設定できます。この例題の場合、L は 5mH に、C は 100uF に設定しています。

サブ回路が最初にメイン回路に組み込まれたとき、メイン回路 (main.sch) から **Subcircuit** メニューの **Edit Subcircuit** を選ぶと、**Subcircuit Variables** のタブにデフォルト値のリストが表示されます。ここで新しい変数を追加したり、値を変更したりすることができます。この例題では、L は 2mH に変更し、C はデフォルトのままにしています。

変数とその値は netlist ファイルに保存され、シミュレーションに使われます。サブ回路内のデフォルト値は netlist ファイルには保存されません。

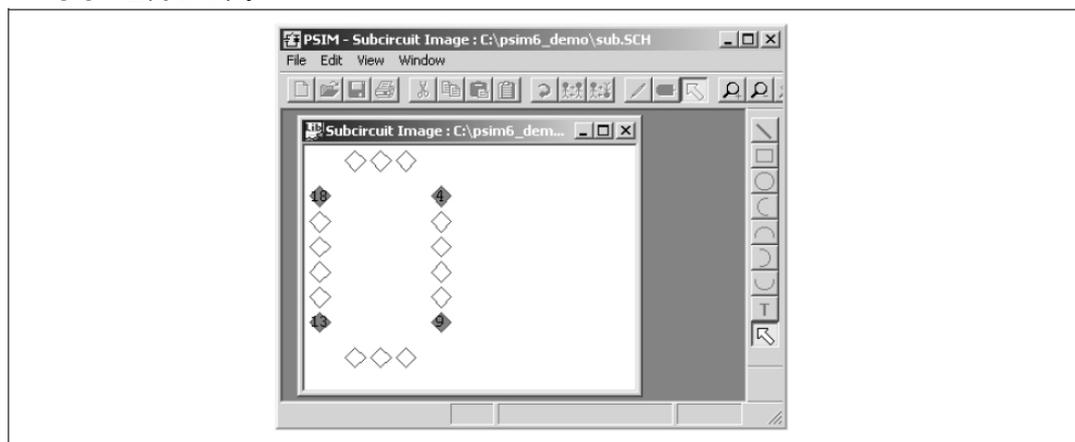
この機能を使えば、サブ回路のパラメータをメイン回路から設定できます。同じサブ回路をひとつのメイン回路で何度も使う場合は、同じ変数に違う値を設定することができます。たとえば、上の例題ではサブ回路 sub.sch を 2 度使っていますが、L は 1 回目は 3mH、次は 1mH として使っています。

また、この例題ではパラメータは変数 (たとえば直流電圧源の Vin) あるいは数式 (負荷抵抗の $R1+R2$) として使っています。変数 Vin、R1、R2 はパラメータ・ファイル para-main.txt にて設定しています。パラメータ・ファイルの詳細は 4.1 を参照してください。

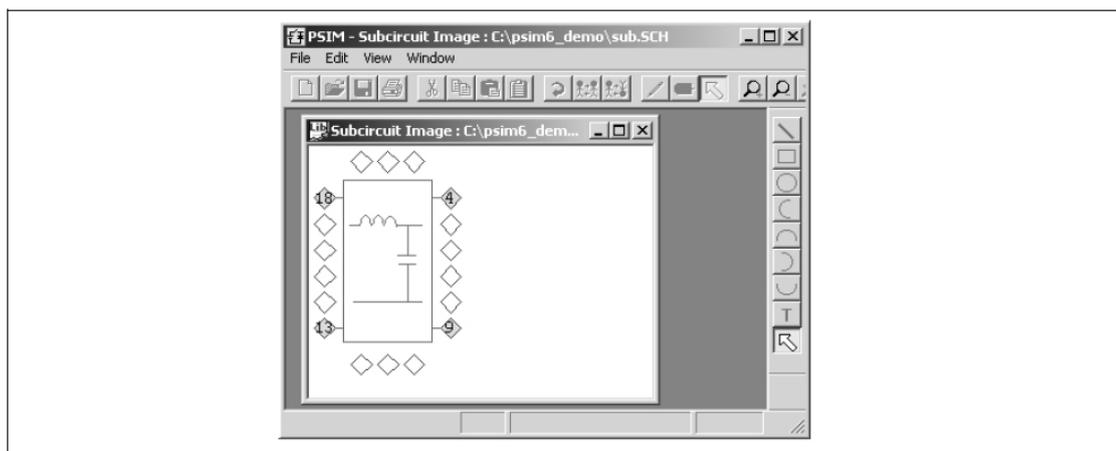
6.5.4.2 サブ回路シンボルの作成

以下にサブ回路 sub.sch のシンボルを作成する手順を示します。

- **Subcircuit** のメニューから **Edit Image** を選びます。すると下のようなウィンドウが開きます。ウィンドウのなかで、赤のダイヤモンドはサブ回路の接続端子を示します。位置はメイン回路に表示されるものと同じです。



- 描画ツールを使ってサブ回路のシンボルを作成・編集します。描画ツールがまだ表示されていない場合は、**View** メニューから **Drawing Tools** を選んでください。Zoom In または Zoom Out のアイコンをクリックすることで、シンボル作成領域の大きさを調整することができます。シンボルを作成すると、以下のようなウィンドウが開きます。



サブ回路のウィンドウ (sub.sch) に戻ってサブ回路を保存します。作成されたサブ回路のシンボルがメイン回路に表示されます。

6.5.4.3 サブ回路を PSIM の素子リストに追加する

PSIM のディレクトリ内に、User Defined という名前でディレクトリを作り、このディレクトリにサブ回路ファイルを入れると、サブ回路が他の PSIM の素子と同じように **Elements** メニューから **User Defined** を選ぶことで表示できます。さらに、**User Defined** ディレクトリの中にサブ・ディレクトリを作成して、その中にサブ回路ファイルを入れることができます。たとえば、**Elements** のメニューは以下のように整理できます。

- Power
- Control
- Other
- Sources
- Symbols
- User Defined
 - Subcircuit 1
 - Project A
 - Subcircuit 2
 - Subcircuit 3
 - Project B
 - Subcircuit 4

このように、共通に使うサブ回路はグループに分けて保存しておくくと便利です。

6.6 Simulate メニュー

シミュレーションを実行するために、Simulate メニューには以下の機能を提供します：

Simulation Control	タイムステップ、合計時間、プリントタイム等シミュレーションパラメータを設定します。これを選択するとカーソルのイメージは時計のイメージに変わります。時計をダブルクリックすると、シミュレーション制御パラメータを指定できます。
Run Simulation	シミュレーションを実行します。
Cancel Simulation	現在進行中であるシミュレーションを中止します。
Pause Simulation	現在進行中であるシミュレーションを一時停止(ポーズ)します。
Restart Simulation	ポーズされたシミュレーションを再開します。

Simulate Next Time Step	次のタイムステップまでシミュレーション実行して停止します。
Run SIMVIEW	波形表示プログラム SIMVIEW を実行します。
Generate Netlist File	回路図の Netlist ファイルを生成します。
View Netlist File	生成された Netlist ファイルを表示します。
Show Warning	シミュレーションから警告メッセージがあれば表示します。
Arrange SLINK Nodes	SLINK ノードを再配列します。SimCoupler モジュールを利用して PSIM と MATLAB/Simulink との共同シミュレーションを行う際この機能を利用します。その他の詳細についてセクション 3.6 を参考にしてください。
Generate Code	回路図のコードを生成します。この機能は SimCoder モジュールを利用して自動コード生成のためです。その他の詳細について SimCoder のユーザマニュアルを参考にしてください。
Runtime Graphs	ユーザはシミュレーションの実行時に必要な波形だけを選んで表示することができます。この機能はシミュレーション結果を実行中に確認し、途中でシミュレーションを中止するときなどに便利です。実行中に表示される波形は、回路で指定された出力のリストから選ばれます。

シミュレーションを実行するためには、Simulate メニューから Run Simulation を選んでください。PSIM シミュレータが起動します。シミュレーションの実行形態として、Total time を指定するワнтаイムモードと、指定しないフリーランモードがあります。フリーランモードで実行する場合、Simulation Control のウィンドウで Free Run チェックボックスをチェックします。シミュレーション終了後、結果を見るには、Simulate メニューから Run SIMVIEW を選んでください。

シミュレーション途中で結果を見るためには次の2つの方法があります。1) Simulate -> Runtime Graphs に行き波形を選択してください（複数選択可能）。または 2) 見たい波形のノードに電圧/電流のオシロスコープ(Elements->Other->Probes の)を接続してください。電圧/電流のスコープは、ワнтаイムモードとフリーランモードにおいて使用することができます。ランタイムグラフはワнтаイムモードにおいて使用できます。Runtime Graphs (ランタイム・グラフ) で表示できる変数は、SIMVIEW で表示するために接続した電圧プローブ、電流プローブ等で定義した変数です。電圧スコープは、どんなノードにも接続することができます。ノードと接地間の電圧波形を表示します。電流スコープは電流のフラグ(R-L-C ブランチやスイッチなどの)を持っている要素に利用できます。

フリーランモードで、ランタイムの間、シミュレーションの途中で素子のパラメータ値を変えることができます。この機能を用いて、電圧/電流スコープで波形を観測しながら、所望の性能が得られるように調整することができます。

フリーランモードでのシミュレーション例:

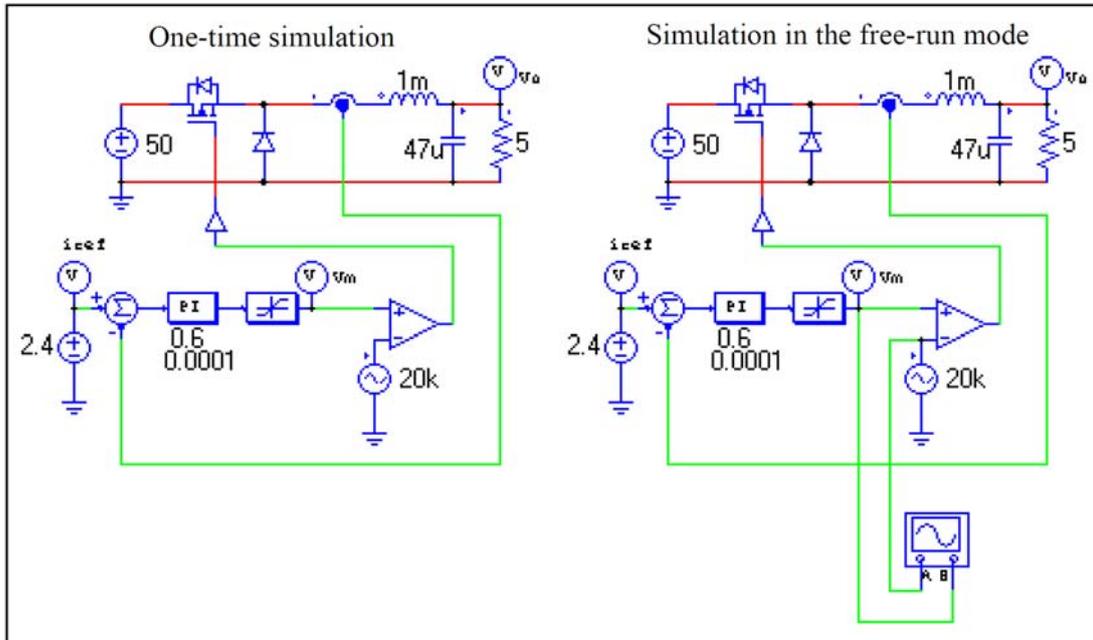
降圧コンバータ回路を用いてフリーランモードでのシミュレーションを示します。

下記、左側の回路ではワнтаイムモードのシミュレーションを行うため Simulation Control の Free Run チェックボックスがチェックしません。Total time(シミュレーション時間)が有効になります。シミュレーション結果はシミュレーション終了後しか見ることはできません。

下記、右側の回路では、フリーランモードのシミュレーションを実行するため Simulation Control の Free Run チェックボックスをチェックします。シミュレーション途中で結果を見るためには次のようになります。

- 1) Elements -> Other ->Scopesに行き、2チャンネルの電圧スコープを選択してください。
- 2) 右側の回路のようにスコープをつなげます。
- 3) スコープをダブルクリックすると、スコープウィンドウが現れます。

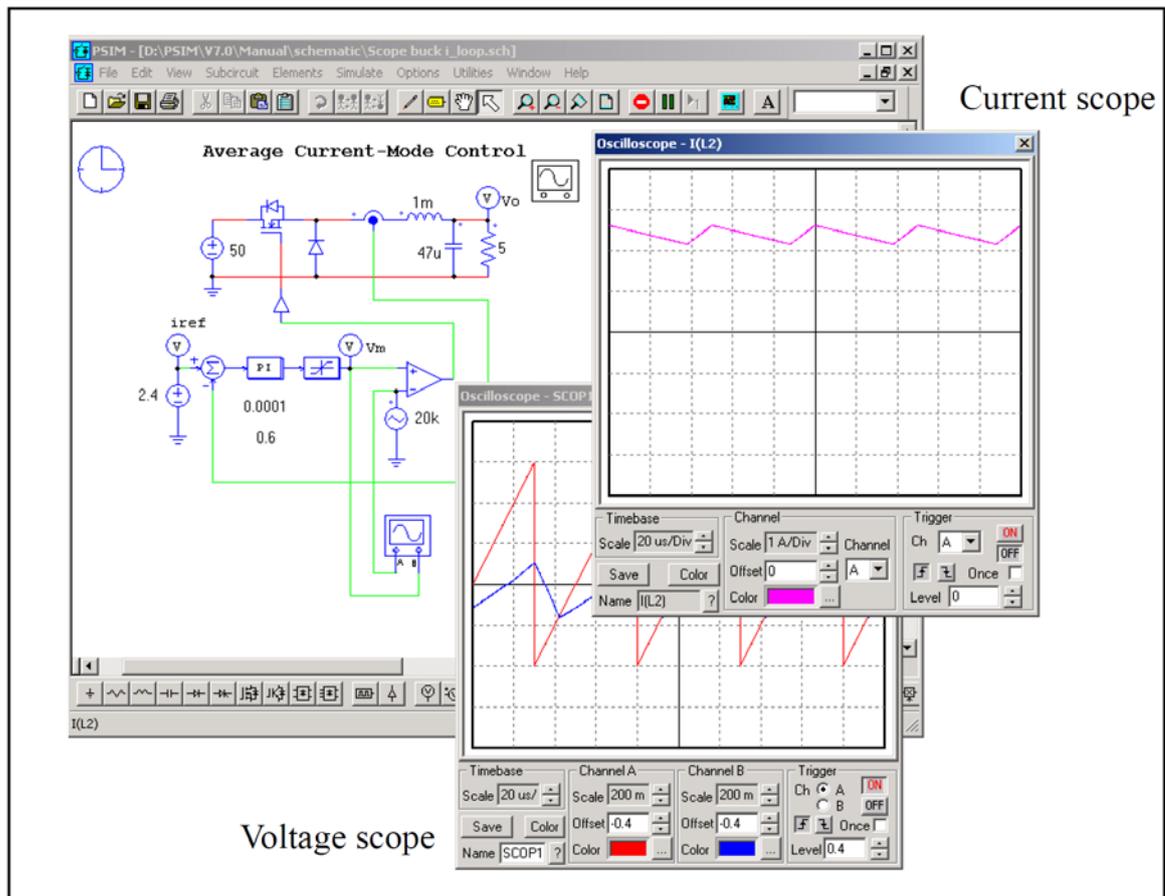
- 4) シミュレーションを開始します。スコープにシミュレーション結果としての波形が表示され、連続的に更新されます。
- 5) 波形が見やすいように時間軸と垂直軸のscale, offset設定を変えます。



フリーランモードでは、ブランチ電流も表示することができます。例えば、インダクタ電流を表示するためには、次の要します。下図を参照ください。

- 1) インダクタの上にマウスの右ボタンをクリックしてください。メニューが現われます。
- 2) Current Scopes→観測したいブランチ電流名を選択してください。
- 3) 電流スコープのブロックが現れます。電圧オシロスコープブロックと同様の図ですが、接続端子がありません。
- 4) インダクタ電流波形を見るためには、スコープをダブルクリックするとオシロスコープの運動が開きます。

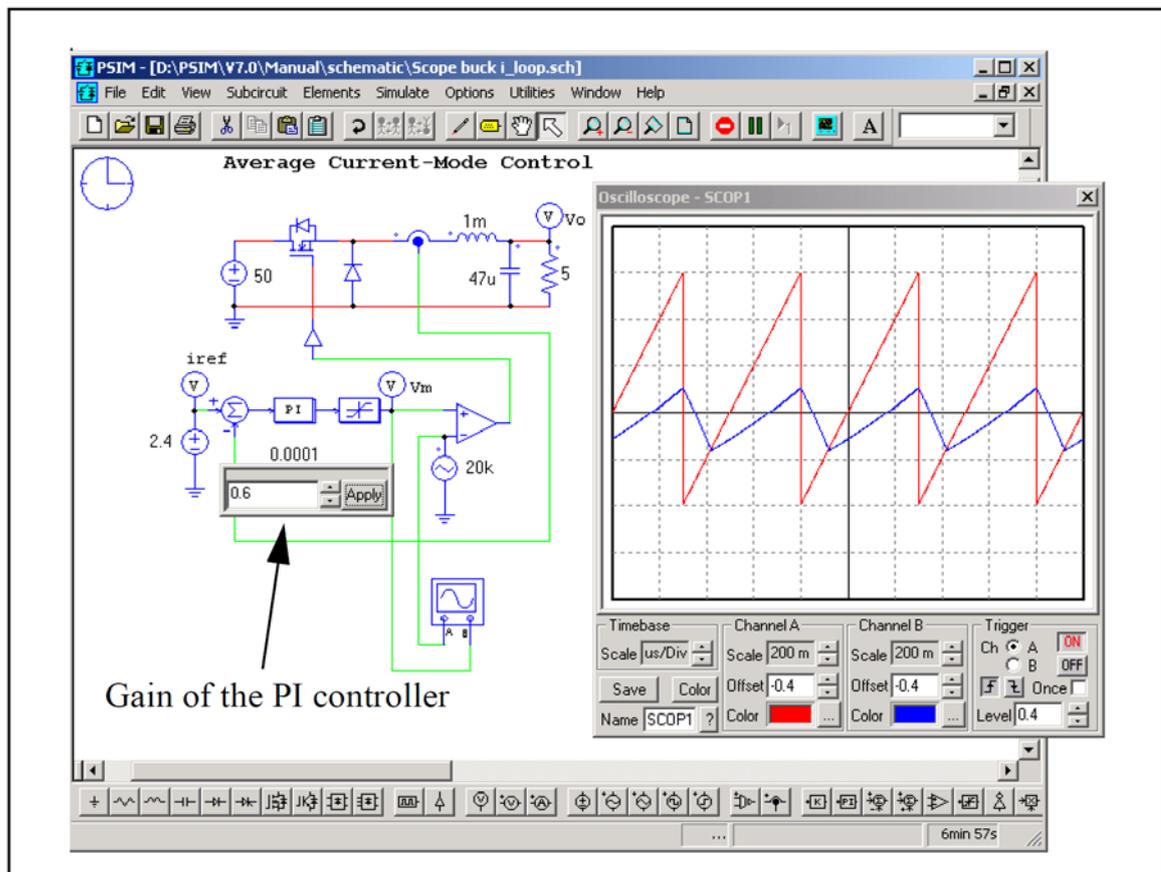
同様の方法で、他のブランチ電流、例えば、コンデンサ電流、負荷電流、ダイオード電流、またはMOSFETスイッチ電流を表示することができます。



シミュレーションの途中で要素パラメータを調整することができます。例えば、PIコントローラのゲインを調整するためには次のようにします。下図を参照して下さい。

- 1) PIブロックを右クリックすると、メニューが表示されます。
- 2) Runtime Variables-> Gain を選んでください。
- 3) ゲインの Display チェックボックスが自動的にチェックされ、現在設定されているゲイン「0.6」が表示されます。
- 4) 表示された数字「0.6」をクリックすると小さいダイアログウィンドウが現れます。
- 5) このダイアログウィンドウに直接キーボードから数字を入力するか、またはアップ/ダウン矢印キーを押すことにより数字を変更することができます。
- 6) Apply をクリックすることにより、シミュレーションに設定されたゲインが反映されます。
- 7) ゲインの変更後の波形の変化はオシロスコープのブロックで確認できます。

同様の方法で、他のパラメータ、例えば、電流指令、直流入力電圧、インダクタンス、キャパシタンス及び負荷抵抗を変更することができます。



コマンドラインオプションでシミュレーションの実行:

プログラムPsimCmd.exeの実行によりコマンドラインオプションでシミュレーションも起動することができます。例えば、フォルダ「c:\psim\examples」に格納される「chop.sch」回路をシミュレートするためには、PSIMフォルダに行き、下記コマンドを実行します:

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

コマンドラインのフォーマットは以下の通りです。

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

パラメータ値のまわりの引用が存在するに違いないことに注意してください。コマンドラインパラメータは以下の通りです。

- i: 入力回路図のファイル名
- o: 出力ファイル名(.txt か .smv エクステンション)
- v: 変数名及び値。このパラメータは複数の回、使用することができます。例えば、抵抗R1を1.5とインダクタンスL1を0.001として定義するためには :


```
-v "R1=1.5" -v "L1=0.001"
```
- t: シミュレーション総時間
- s: シミュレーションタイムステップ
- g: シミュレーションが完全になった後にSIMVIEWを実行します。

コマンドラインオプションで、バッチランで自動的にいくつかの回路を動かすことができます。

6.7 Options メニュー

Optionsメニューを利用して色々な設定が可能になります：

Settings...	回路図のテキストフォント設定、SIMVIEW 波形表示フォーマットの選択、バッファサイズの設定などの変更をします。
Auto-run SIMVIEW	これをチェックするとシミュレーション完了後 SIMVIEW 画面が自動的に立ち上がります。
Set Path ...	熱モジュールのデバイスファイル、MagCoupler モジュールのパスなどを設定します。
Enter Password	パスワードで保護された回路図を表示するためにパスワードを入力します。
Disable Password	パスワードで保護された回路図の保護を無効にします。
Customize Toolbars	カスタマイズツールバーを作成します。
Customize Keyboard	キーボードをカスタマイズします。要素またはコマンドにショートカットキーを追加することができます。
Save Custom Settings...	カスタマイズした設定のうち、保存する項目を選択します。
Load Custom Settings...	カスタマイズした設定を保存ファイルから読み込みます。
Deactivate	PSIM ライセンスを無効にします。この機能はソフトキー(Softkey)バージョンのみに有効です。
Change Password	PSIM ソフトキーバージョンのログインのパスワードを変更します。この機能はソフトキー(Softkey)バージョンのみに有効です。
Check for Software Updates	Powersim 社のサーバへアップデート可能なソフトウェアがあるかどうかチェックします。新しいアップデートがあれば、このアップデート情報をダウンロードしてインストールができます。(サポート契約に加入している方のみ)

OptionsメニューのSettings...機能では以下の設定ができます：

編集に関して:

Display grid	PSIM 回路図にグリッドを表示します。
Enable rubber band	これをチェックして要素か回路の一部を動かすと、回路図と接続されます。
Show print page borders	チェックすると、プリントアウトの境界が表示されます。
Zoom factor	回路図の拡大と縮小ファクターを定義します。

テキストフォントに関して:

Default Text Font	回路図のデフォルトテキストフォントを設定します。
Runtime Graph Font	ランタイムグラフのテキストフォントを設定します。

プリントに関して:

Line thickness	プリントアウトの時のラインの太さを設定します。ラインの太さは 1(最小)から 4(最大)まで設定することができます。 ラインの太さはプリントアウトした時のみ影響し、スクリーンには反映されません。
-----------------------	--

シミュレーションに関して:

Save simulation result in:	シミュレーション結果の保存形式をバイナリフォーマット(デフォルト)かテキストフォーマットか選択できます。バイナリフォーマットの場合、ファイルサイズが小さく、ロード時間が速くなります。
Limit output buffer size to ** MB	このオプションをチェックすると、シミュレーションデータはセグメントで結果ファイルに書かれます。例えば、バッファサイズを20MBに設定する場合、シミュレーションデータは最初にバッファに保存され、そして、20MBに達すると、全体の20MBのデータは結果ファイルに書かれます。 ランタイムグラフはバッファのデータを利用してグラフをプロットします。したがって、古いデータがファイルに保存されて、新しいデータがバッファに記録されます。更に、ランタイムグラフは新しいデータの波形だけを示します。 ランタイムグラフですべての波形を保有するためには、バッファサイズを増加させるか、またはこのオプションのチェックを外します。 しかしながら、このオプションを選択しないと、PSIM は最初に必要なメモリバッファを割り当てます。シミュレーション刻みが小さく、総合時間が長く、多くの出力データがある場合、非常に大きいメモリが必要になります。この場合、十分なメモリ容量がないと、シミュレーションが中断されます。 このチェックボックスをチェックすると警告メッセージが表示されなくなります。
Disable simulation warning messages	
Save all voltages and current	このオプションがチェックされていると、回路のすべての電圧と電流は、表示のために保存されます。電圧や電流を表示させるためには、シミュレーションが完了し、結果が SIMVIEW にロードされた後に [View] -> [Display Voltage/Current]と選択します(または対応するアイコンをクリック)。カーソルがノードまたはブランチの上にあるとき、電圧プローブまたは電流クランププローブの画像に変わります。マウスの左ボタンをクリックすると、対応する電圧または電流が SIMVIEW で表示されます。差動電圧を表示するには、[View] -> [Display Differential Voltage]と選択します。その後、最初のノード、2番目のノードの順にクリックします。差動電圧の波形が SIMVIEW で表示されます。
Maximum number of points for oscilloscope	オシロスコープの最大ポイント数を設定します。10,000 から 100,000,000 まで設定できます。

アップデートに関して:

Check for Software Updates every month	ソフトウェアのサポート契約に加入した場合、Powersim 社のサーバから最新のアップデートファイルをダウンロードすることが可能になります。ここをチェックすると、毎月アップデート情報を自動的に確認します。
---	--

バックアップに関して:

Create backup files every ** minutes	設定した時間ごとにファイルのバックアップが行われます。
---	-----------------------------

アイドル時間に関して (ネットワーク版のみ) :

Release license if PSIM is idle for ** minutes	このオプションをチェックすると、プログラムが指定された時間に使用されていなかった後にPSIMはタイムアウトされます。このオプションは、PSIMネットワークバージョンのみに有効です。 PSIMがタイムアウトされる場合、そのライセンスが解放され他のユーザはこのライセンスを使用することができます。 ユーザがライセンスを無意識にホールドすることを防ぎます。
---	---

コード生成に関して:

Show image next to elements that	このオプションは SimCoder モジュールがある場合にのみ有効です。このオプションをチェックすると、イメージはコード生成に使用することができる要素の横に現
---	---

can be used for code れます。これは、コード生成に使用できる要素と使用できない要素を差別化するのに使用されます。

PSIM Helpファイルに関して:

Use alternate PSIM help file path デフォルトでは、PSIM は PSIM フォルダからヘルプファイル psim.chm を読み込みます。このオプションをチェックすると、PSIM は指定の場所からヘルプファイルを読み込みます。

Simviewファイルの自動保存に関して:

Delete Simview files on exit ここをチェックすると、Simview の出力ファイルをユーザが保存しない状態で PSIM を Exit した場合、出力ファイルを自動的に削除します。

カラーに関して:

Colors グリッド、要素、サブ回路、ポート、ワイヤ、テキスト、ノード、ラベルのカラーを設定します。

Wire color **Wire Color** タブでは、ワイヤのカラーはデフォルトか回路タイプで設定することができます。

Options メニューの **Set Path** 機能では、外部 DLL ファイルを読み込む際の検索パスを設定することができます。例えば、回路に DLL ブロックが含まれていて、そこで使用する DLL ファイルが、.sch ファイルの存在するフォルダおよび PSIM フォルダ以外のフォルダに置かれているときに、DLL ファイルが存在するフォルダを **Set Path** 機能で設定しておくことで正しくシミュレーションすることができるようになります。

PSIM は以下の順序で外部 DLL ファイルを検索します。

1. PSIM をインストールしたフォルダ (PSIM.exe が存在するフォルダ)
2. 回路図ファイル (.sch ファイル) が存在するフォルダ
3. **Options** -> **Set Path** で設定したフォルダ

例えば、PSIM をインストールしたフォルダが C:\PSIM で、シミュレーションする回路図ファイルが C:\TEMP にあり、**Set Path** 機能で C:\TEMPDLL フォルダが設定されているとき、DLL ファイルは以下の3つのフォルダのいずれかに置くことができます。また、DLL ファイルは以下の順序で検索されます。

1. C:\PSIM
2. C:\TEMP
3. C:\TEMPDLL

PSIM は以下の順序で Thermal Module のデバイスファイルを検索します。

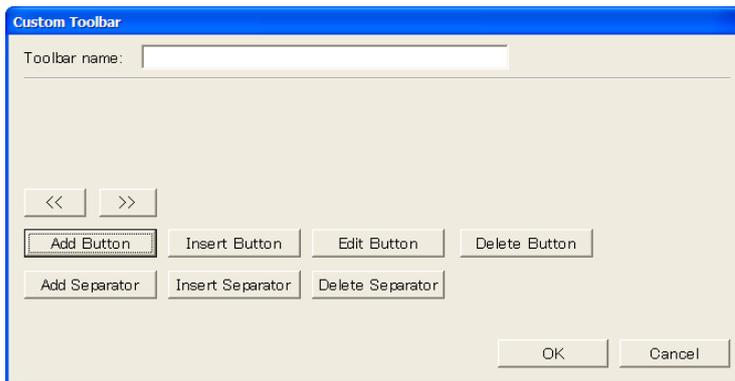
1. PSIM ディレクトリの "device" サブフォルダ
2. **Device File Path** で設定されたディレクトリ

PSIM のデバイスサブフォルダおよび **Device File Path** 内のすべてのデバイスファイルが読み込まれます。

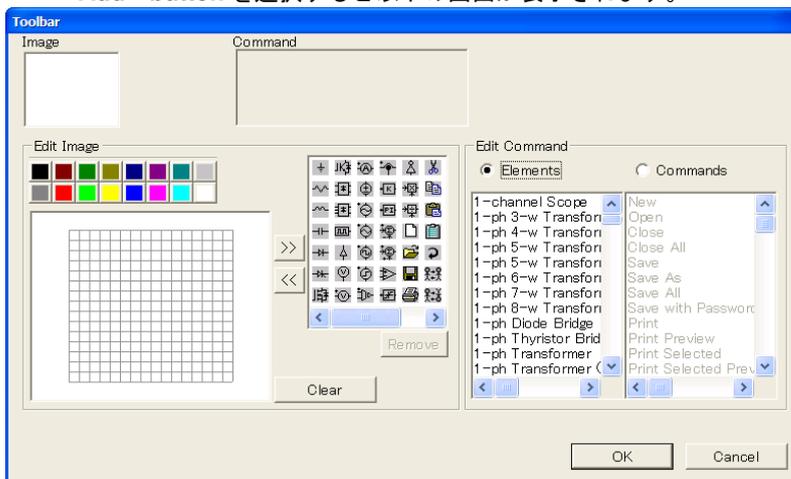
ツールバーのカスタマイズ:

AND Gate をカスタマイズツールバーで作成する方法を説明します:

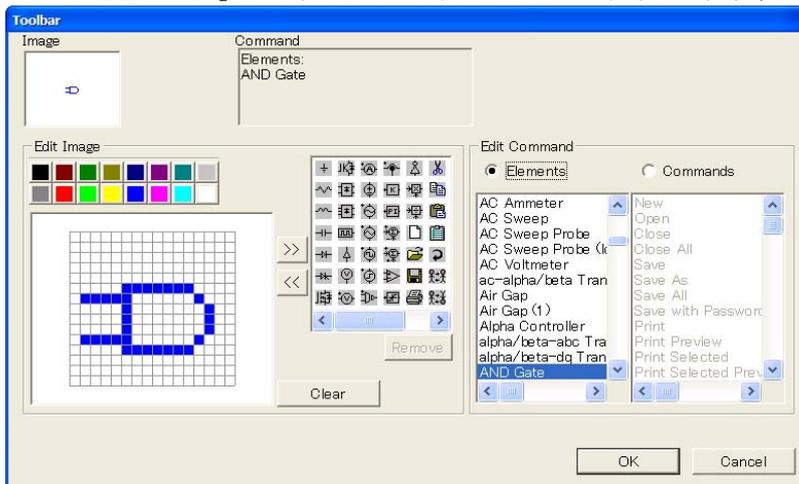
- **Options** -> **Custom Toolbars** を選択します。 **New Toolbar** を選択するとカスタムツールバーダイアログウィンドウが起動します。



- **Add button** を選択すると以下の画面が表示されます。



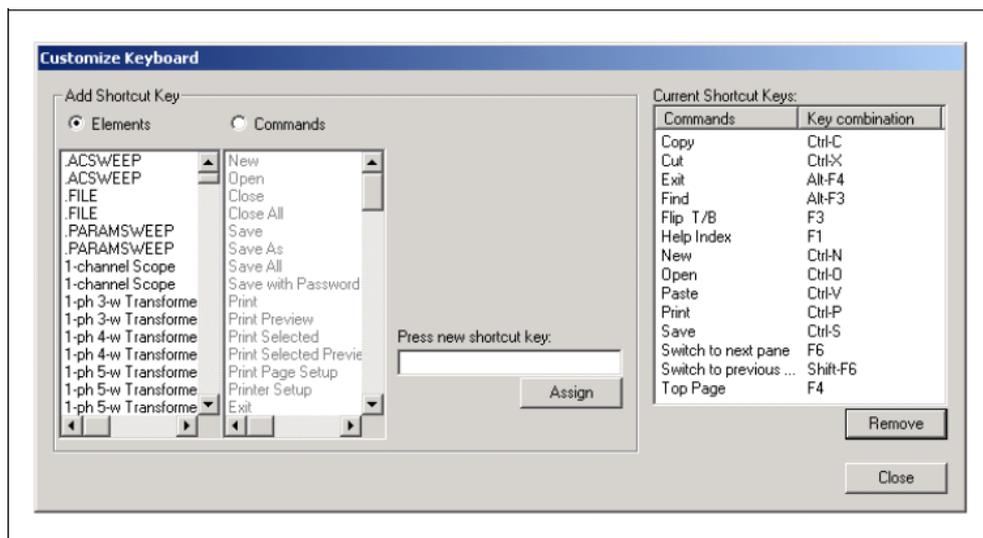
- アイコン編集エリアの「AND gate」のアイコンを描くか、Elements から選択します。
- Elements から選択する場合、Edit Command セクションの Elements オプションを選択して「AND Gate」をハイライトします。次に **OK** ボタンを押してカスタムツールバーウィンドウに戻ります。「AND Gate」のアイコンがカスタムツールバーダイアログウィンドウに表示されます。



キーボードのカスタマイズ:

抵抗要素をキーボード「r」キーで選択する例を説明します:

- **Options -> Custom Keyboard** を選択します。カスタムキーボードダイアログウィンドウを表示します。



- **Add Shortcut Key** では、**Elements** オプションを選択します。次に、抵抗要素「Resistor」を発見してハイライトします。
- **Press new shortcut key** の入力フィールドにカーソルを移動してキーボードの「r」キーを押します。次に Assign をクリックします。
- 「r」キーが要素「Resistor」にアサインされ、定義は **Current Shortcut List** に表示されます。

6.8 Utilities メニュー

UtilitiesメニューからはPSIMの便利な補助ツールを使用することができます：

- | | |
|------------------------------------|--|
| s2z Converter | このプログラムは s 領域の関数を z 領域の関数に変換します。デジタル制御モジュールのライセンスがある場合だけ、このプログラムは有効になります。 |
| Device Database Editor | 熱モジュールの PcdEditor.exe というデバイスデータベースエディタを起動します。 |
| B-H Curve | このプログラムは可飽和コア要素の B-H カーブプロットを起動します。 |
| Solar Cell (physical model) | このプログラムは太陽電池ブロックの i-v カーブをプロットします。この機能は Renewable Energy Models に含まれています。 |
| SimCoupler Setup | PSIM と MATLAB/Simulink との連成シミュレーションのセットアップを行うためにプログラム SetSimPath.exe を起動します。 |
| DSP Oscilloscope | DSP オシロスコープを起動します。この機能は、TI F28335 Target の一部として利用可能です。リアルタイムで DSP の波形を表示するには、SCI 機能と一緒に使用されません。
DSP オシロスコープを使用する方法の詳細については、「Tutorial -Using SCI for Real-Time Monitoring in TI F28335 Target.pdf」を参照してください。 |
| Launch/Export to SmartCtrl | SmartCtrl ソフトウェアを起動します。また、ac スイープの結果があるときにはこの結果を SmartCtrl へ出力します。SmartCtrl は制御ループ設計のための独立したソフトウェアです。詳細については、SmartCtrl のマニュアルを参照して下さい。 |
| Unit Converter Calculator | このプログラムは長さ、面積、重さ、および温度における単位変換を実行します。PSIMの電卓を起動します。使い方の詳細は 7.7章をご参照ください。 |

6.9 PSIM ライブラリの管理

PSIMライブラリは2種類のデータからなります。イメージライブラリ(psimimage.lib)とnetlistライブラリ(psim.lib)です。このうち、netlist ライブラリはユーザ自身で編集することはできませんが、イメージライブラリは編集または新規追加ができます。

イメージライブラリを新規作成または更新するには、Edit -> Edit Library -> Edit Library Filesを選んで、画面の指示に従ってください。PSIMディレクトリのどんなイメージライブラリでも、PSIMに自動的にロードされます。標準の画像ライブラリはpsimimage.libで与えられています。ユーザはこれを編集することはできませんが、**Edit->Edit Library->Edit library files**で画像を表示してコピーすることができます。

ユーザは**Edit->Edit Library->Edit library files**で**New library**を選択することで新しい画像ライブラリを作ることができます。その後ライブラリ名を決定するとその名前がPSIMのElementsメニューに表示されます。このライブラリファイルはPSIMフォルダ内に作成されます。

PSIMライブラリリストにカスタムモデルを追加するためには、2つ方法があります。一つの方法は、モデルをサブ回路の形をして、次に、PSIMディレクトリのUser definedと呼ばれるフォルダあるいはUser definedのサブフォルダのうち、1つにこの回路図のファイルをおくことです。User definedフォルダのどんな回路図ファイルおよびサブフォルダも、PSIMライブラリリストに現れます。

もう一つの方法は、イメージライブラリにカスタムモデルを直接加えることです。この方法の利点は、カスタム用素子が標準PSIM素子と同じ仕様を持たせることが可能であることです。さらに、カスタムモデルには、ヘルプファイルに関連付けることも可能です。

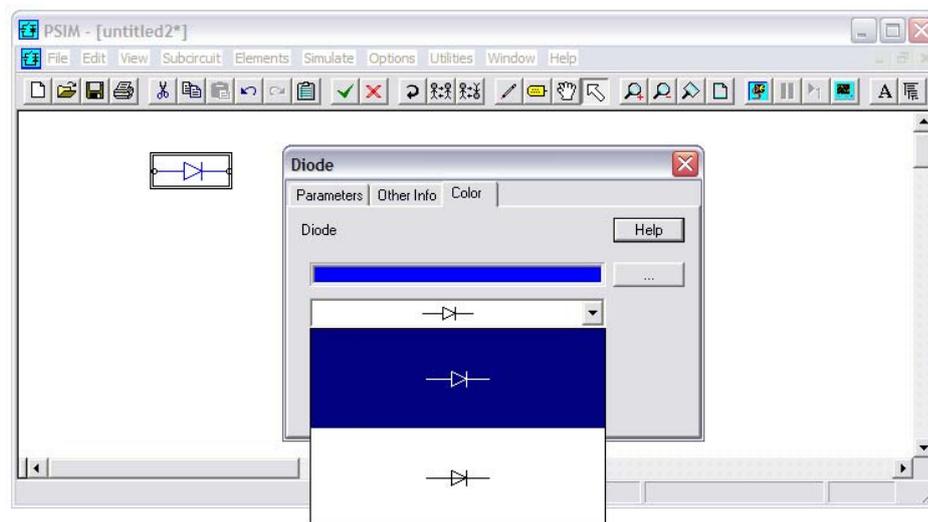
6.9.1 2番目のイメージ図作成

1つ以上のイメージ図をPSIM要素に関連づけることができます。新しい要素イメージは標準イメージライブラリあるいはユーザのカスタムイメージライブラリで作成することができます。新しいイメージは2番目のイメージライブラリに.lib2拡張子で保存されます。

標準イメージライブラリ「mylib.lib2」へ「Diode」の2番目のイメージを作成する方法を以下に説明します：

- **Edit -> Edit Library -> Edit secondary image library files**へ行き**New library**をクリックしてください。ダイアログウィンドウでは、2番目のイメージライブラリファイル名は「mylib.lib2」として定義してください。
- 次に「mylib.lib2」を選択して**Edit selected library**をクリックしてください。二次イメージライブラリの編集用のダイアログウィンドウが表示されます。
- **Add**ボタンをクリックしてください。PSIMライブラリリストの**Power ->Switches**へ行き**Diode**選択してください。要素Diodeは2番目のイメージリストに表示されます。「Diode」テキストの前のテキスト「PSIMIMAGE」は標準イメージライブラリpsimimage.libのダイオード要素であることを示します。
- 「Diode」をハイライトしてEditボタンをクリックしてイメージを作成してください。
- このイメージをこの要素のデフォルトイメージとして使用したい場合は「**Set As Default Image**」をクリックしてください。

2番目のイメージ図が生成された後、このイメージはPSIMの回路図で利用できます。例えば、ダイオードを回路図に追加してダブルクリックすると、ダイアログウィンドウが表示されます。このウィンドウの**Color**タブを選択してプルダウンを選択すると2つイメージが表示されます。標準イメージライブラリとカスタムイメージライブラリmylib.libそれぞれ1つです。



2番目のイメージ図を選択すると、同じ要素のすべてのイメージが自動的にこのイメージ図に変わります。選択されたイメージ図はデフォルトイメージとして回路図を作成する時、あるいは別の回路図を開く時、設定されます。

6.9.2 ライブラリへ新しいサブ回路要素の追加

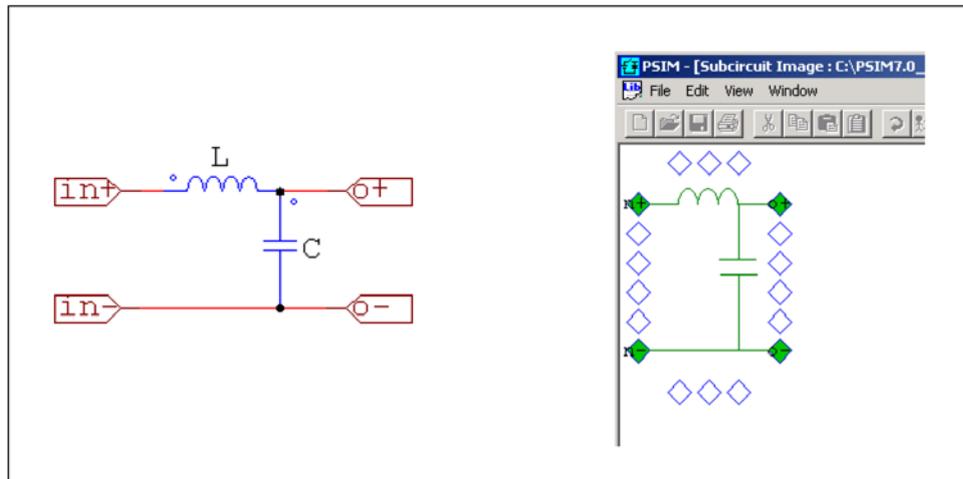
サブ回路でモデル化された新しい要素をPSIMライブラリに追加するために、主な3ステップがあります:

- 新しい要素のサブ回路モデルを作成
- PSIMライブラリへこの要素を追加
- この新しい要素用のオンライン・ヘルプファイルを作成

このプロセスを例証するために、LCフィルタエレメントは例として使用されます。

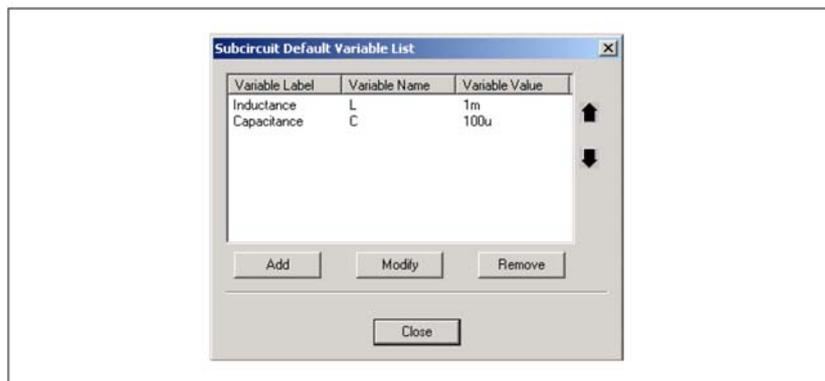
サブ回路作成:

第一歩は、新しい要素のサブ回路を作成することです。まるでサブ回路がもう一つの回路で呼ばれることになっているように、サブ回路は同様につくられます。あたかもサブ回路が別の回路によって呼ばれることになっているかのように、同じ方法で新しい要素のサブ回路を作成することです。例えば、「LC_filter.sch」と呼ばれる2次LCフィルタのサブ回路およびその図を以下に示します。



この場合、インダクタンスとキャパシタンスの値はインタフェースを通して定義され、新しいLCフィルタエレメントのプロパティウィンドウに現われる必要があります。したがって、インダクタンスに対するパラメータ値はこの場合変数Lとして定義される必要があります。同様に、キャパシタンスに対するパラメータはCとして定義される。

次に、Subcircuit->Edit Default Variable Listから、デフォルト変数として変数LおよびCを加えてください。新しい要素がデフォルト変数リストからパラメータ情報を得るとともに、このステップは必要です。デフォルト変数リストウィンドウが以下の通りに見えるはずですが。



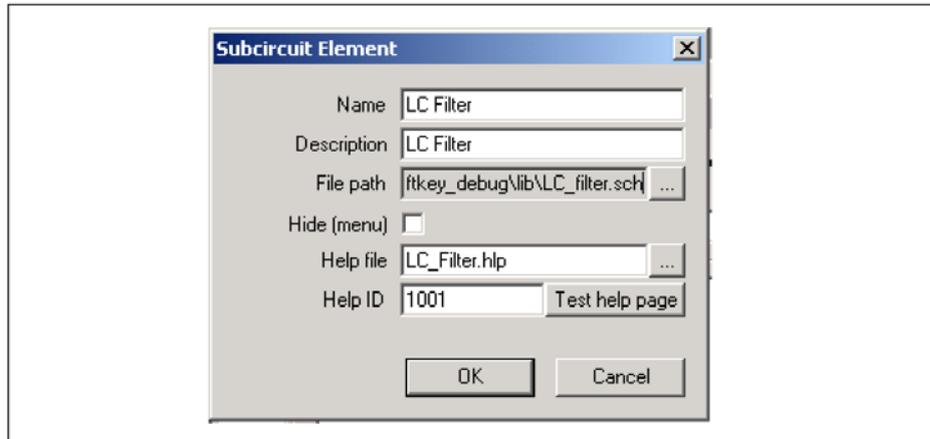
ここに、Variable Labelはパラメータについて記述するテキストで、Variable Nameはサブ回路の中でパラメータ値として使用され、Variable Valueはパラメータのデフォルト値です。例えば、インダクタンスLについては、Variable LabelはInductanceで、Variable NameはLで、Variable Valueは1mです。

PSIMライブラリへ新しい素子の追加:

PSIMライブラリへサブ回路要素を加えるためには、これらのステップに従ってください。

- Edit -> Edit Library -> Edit Library Filesに行き、新しい素子のために、ライブラリを選んでください。新しいイメージライブラリを作るために、New Libraryをクリックしてくださいあるいは、既存のライブラリを選んでください。そして、Edit Selected Libraryをクリックしてください。

-ライブラリエディタでは、New Subcircuitボタンをクリックしてください。以下に示されるようなダイアログウィンドウへの情報を入力してください。



各フィールドの説明は以下の通りです。

Name: PSIMライブラリに示される新しい要素の名前

Description: 新しい要素の説明。

File Path: サブ回路の回路図のファイル「LC_filter.sch」の位置。サブ回路ファイルをPSIMディレクトリのlibサブフォルダに置かなければなりません。

Hide (menu): このボックスはチェックしないでください。このボックスがチェックされると、この要素はライブラリには示されません。

Help File: この要素と関連付けたオンラインヘルプファイル。

Help ID: 指定されたヘルプページをリンクするためにヘルプマップファイルで使用されるID。

- Save Image Library及びUpdate Menuボタンをクリックしてください。新しい要素はライブラリに現れて使用に準備ができます。

新しい要素用のオンラインヘルプファイルを作成:

新しい要素のためにオンラインヘルプファイルを作成することができます。プロパティダイアログウィンドウのヘルプボタンがクリックされる時、対応するヘルプページが表示されます。マイクロソフトヘルプWorkshopを使用することでヘルプファイルを作成する手順は以下のようになります。

- マイクロソフト・ワードを使用して.rtfファイル「LC_Filter.rtf」で、ヘルプコンテンツを準備してください。

ファイルの脚注が以下のように見るべきであることに注意してください。

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

ここで、#ラベルは、PSIMでヘルプページをLCフィルタ要素にリンクするのに使用されます。kラベルはヘルプ・ファイルのIndexに現れます。そして、\$ラベルが使われません。

-ヘルプマップファイル「HelpMap.txt」を作ってください。それはテキストファイルで、次の内容があります。

```
LC_FILTER=1001
```

ここでは、テキスト「LC_FILTER」は「LC_Filter.rtf」ファイルの#ラベルと同じでなければなりません。そして、数「1001」はLCフィルタ要素のために定められるヘルプIDと同じでなければなりません。

-ヘルプ・ワークショップ・プロジェクトでは、「LC_Filter.hpj」ファイルは、次の設定を持っています。

- Filesボタンをクリックして、「LC_Filter.rtf」を追加してください。

- OptionsボタンおよびFilesタブをクリックして、LC_Filter.hlpとしてヘルプ・ファイルを設定してください。

- Mapボタンをクリックし、次に、Includeをクリックして、マップファイルHelpMap.txtを含んでください。

-ヘルプファイル「LC_Filter.hlp」を作成するために、Save And Compileをクリックしてください。PSIMディレクトリへこのファイルを置いてください。

6.9.3 ライブラリへ新規 DLL 要素の追加

DLLでモデル化された新しい要素をPSIMライブラリに追加するために、主な3つステップがあります。

- DLLファイルのモデルを作成
- PSIMライブラリへこの要素を追加。
- この新規要素用のオンラインヘルプファイルを作成。

このプロセスを例証するために、インダクタは例として使用されます。

DLL作成:

第一歩はDLLのインダクタンスモデルを作成することです。カスタムDLLを作成する方法について、関連セクションを参照してください。

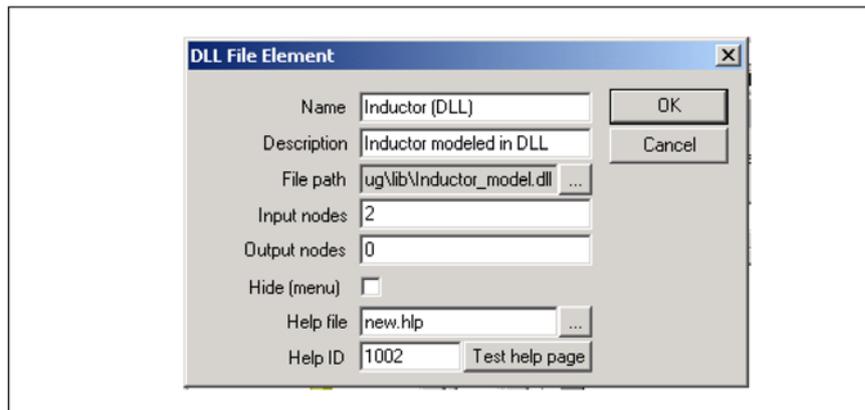
ここで、「インダクタ_model.dll」というDLLファイルがパワーモデリングブロックを使用することで既に作成されたと思われます。それには、「インダクタンス」と呼ばれる1つのパラメータ、および2つの接続ノードがあります。

PSIMライブラリへ新規要素の追加:

PSIMライブラリへDLL要素を加えるためには、これらのステップに従ってください。

- Edit -> Edit Library -> Edit Library Filesに行き、新しい要素のために、ライブラリを選んでください。新しいイメージライブラリを作るために、New Libraryをクリックしてくださいあるいは、既存のライブラリを選んでください。そして、Edit Selected Libraryをクリックしてください。

-ライブラリエディタでは、New DLL Fileボタンをクリックしてください。以下に示されるようなダイアログウィンドウへの情報を入力してください。



各フィールドの説明は以下の通りです。

- Name: PSIMライブラリに示される新しいインダクタ要素の名前
- Description: 新しいインダクタ要素の説明
- File Path: 新規インダクタ要素をモデル化するDLLファイル「inductor_model.dll」の位置。
DLLファイルは、PSIMディレクトリのlibサブフォルダに置かなければなりません。
- Input Nodes: 新規要素の入力端子の数。パワー・モデル・ブロックでは、この値は端子の合計の数で、DLLファイルから自動的に読まれます。
- Output Nodes: 新規要素の出力端子の数。パワーモデリングブロックでは、この値は0です。
- Hide (menu): このボックスはチェックしないでください。このボックスがチェックされると、この要素はライブラリには示されません。
- Help File: この要素と関連付けたオンラインヘルプファイル。
- Help ID: 指定されたヘルプページをリンクするためにヘルプマップファイルで使用されるID。

- 次のダイアログウィンドウで、以下として新しい要素サイズを設定してください:幅=5及び高さ=2。次に、この要素のために図を作成するか、あるいはデフォルト・図を受理してください。

- Save Image Library及びUpdate Menuボタンをクリックしてください。新しい素子がライブラリに現われて、使用する準備ができます。

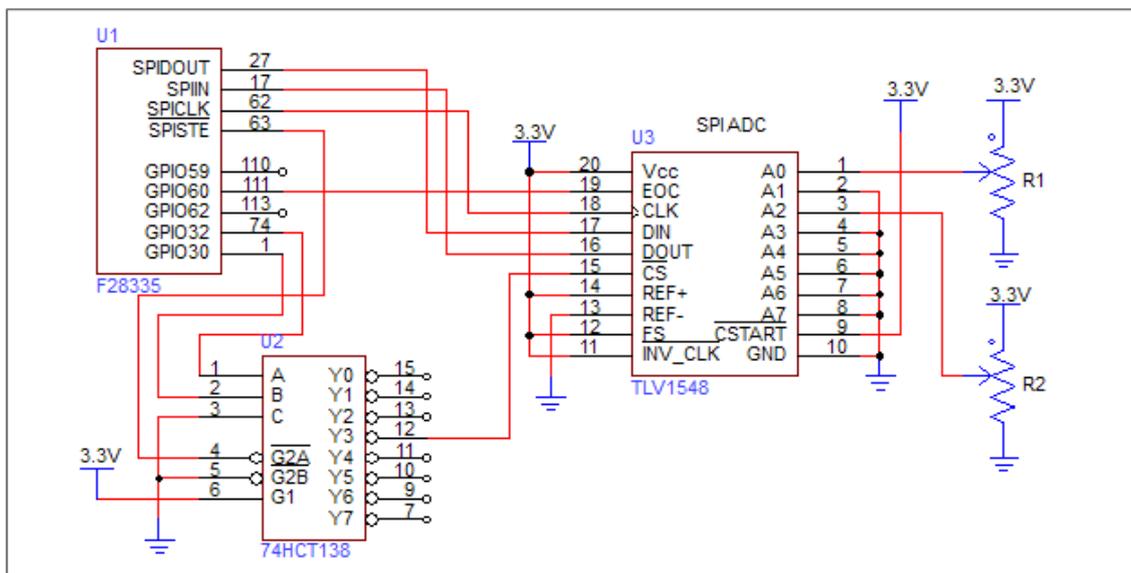
パラメータの数および新しいインダクタ要素のためのパラメータ記述に関する情報は、DLLファイルから自動的に得られます。この場合、新しい要素に「インダクタンス」として1つのパラメータがあります。

オンラインヘルプファイルを作成するプロセスは前のセクションで記述されている通りです。

6.10 シンボルライブラリの作成

[Edit]メニューの[Image Editor]を使うと、良い部品の画像を容易に、また非常に迅速に作成することができます。これらの画像は、PSIMライブラリ要素の第2の画像、またはサブキットの画像として使用することができます。回路の配線図の目的のためにこれらのイメージをシンボルライブラリに保存することができます。そのような回路図の表示が目的であり、シミュレートできないことに注意してください。

下図は、SPI、A/D 変換器からなるハードウェア回路図の例を示します。

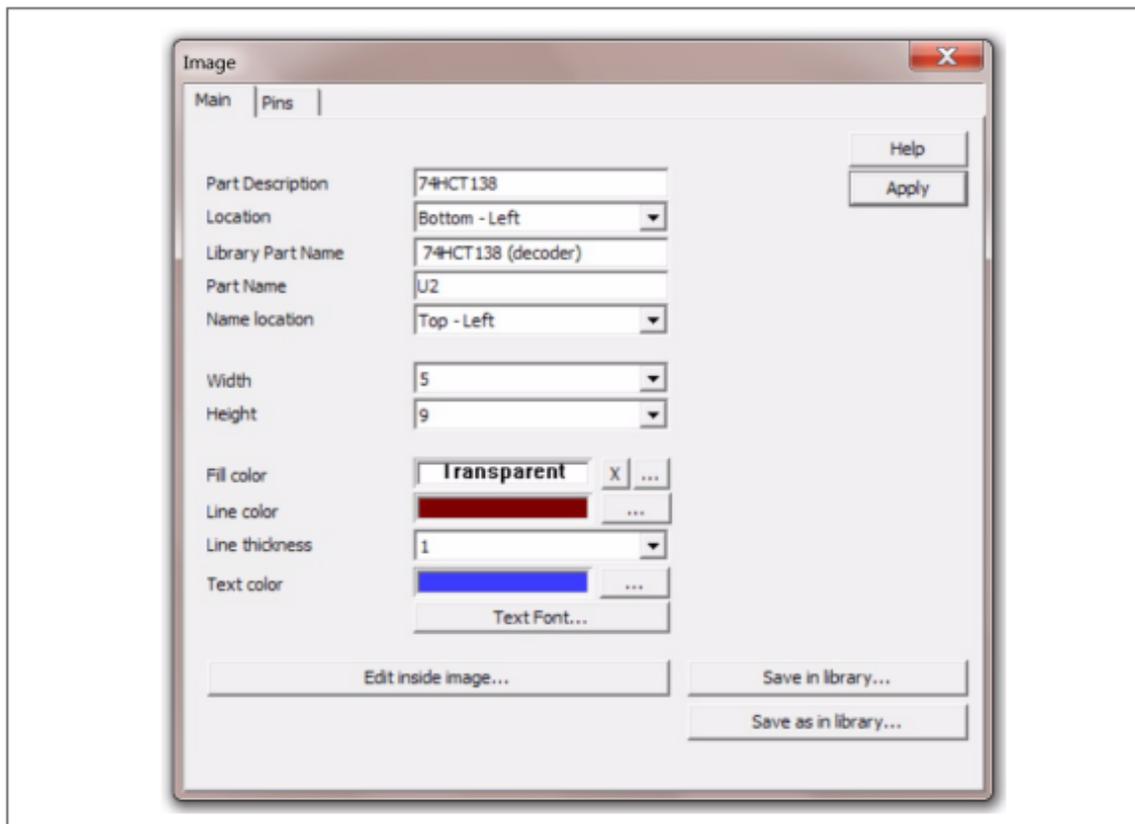


F28335、74HCT138、および TLV1548 はすべてイメージエディタによって作成されています。

イメージエディタを使用してイメージを作成し、カスタムライブラリにシンボルとしてのイメージを保存する方法を、以下に示すようにデコーダ 74HCT138 用イメージを作成して説明します。

- PSIM にて新しい回路図ウィンドウを作成します。[Edit] -> [Image Editor]を選択します。長方形が表示されます。回路図上に長方形を配置し、脱出するために Esc キーを押してください。

- 長方形をダブルクリックすると、ダイアログウィンドウが表示されます。[Main]タブで、下の画面で示すようにデータフィールドを入力します。[Library Part Name]はライブラリに表示される要素の名前です。ブロックのサイズは、リードの長さを考慮せずに幅 5、高さ 9 にします。

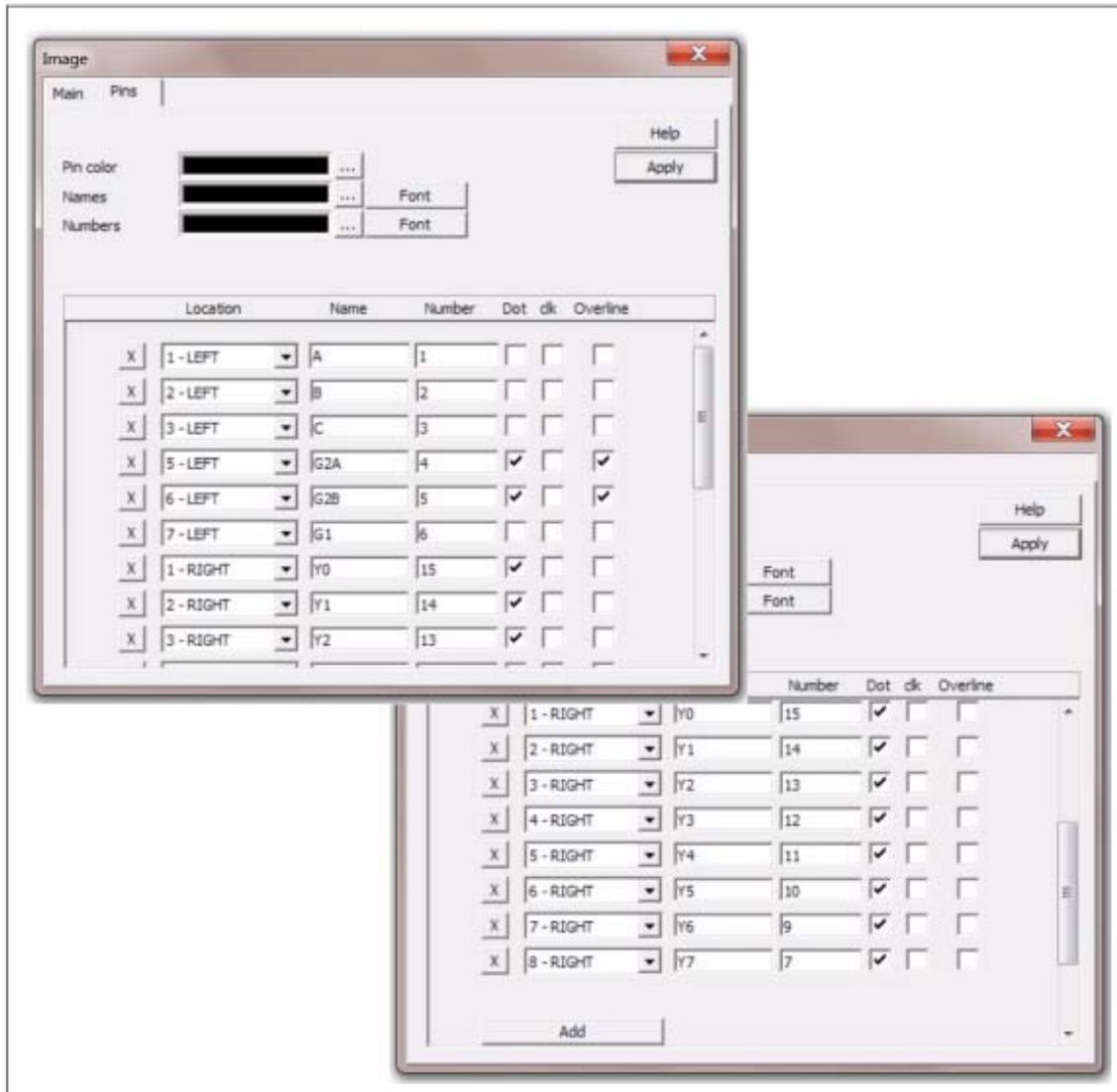


- [Pins]タブをクリックします。このタブでは、接続ピンが定義されます。[Add]ボタンをクリックして、[Location]を"1-Left"、[Name]を"A"、[Number]を"1"と定義します。[Dot]、[CLK]のチェックボックスをそのままにし、[Overline]のチェックを外します。

[Dot]がチェックされている場合、接続ピンとブロックの間に円があります。[clk]がチェックされていれば、クロックであることを示すために">"がブロックの内部に追加されます;そして、[Overline]にチェックが入っていれば、名前に上線が追加されます。

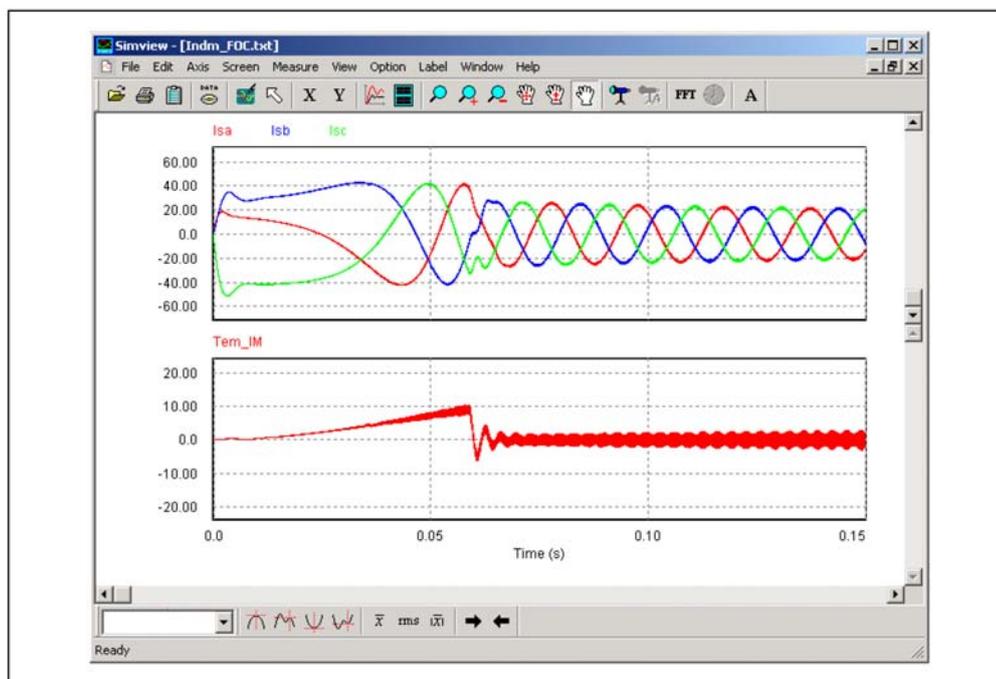
- [Add]ボタンをクリックし、[Location]を"2-Left"、[Name]を"B"、[Number]を"2"と定義します。残りのピンの設定のために同じプロセスを繰り返します。以下ダイアログウィンドウはすべての接続ピンの定義を示しています。

- 一度、すべてのピンを追加してから、メインタブに戻ります。[Save in library]をクリックし、[New Library]を選択し、ライブラリ名を"My Symbol"と入力し、ライブラリのファイル名を"My Symbol"と指定します。"My Symbol"ライブラリは Elements メニューの下に表示され、このライブラリには新しく作成された要素 74HCT138 が含まれます。



第7章 SIMVIEW による波形処理

SIMVIEW は波形の表示とシミュレーション実行後のデータ処理をおこなうプログラムです。下の図は SIMVIEW により、波形を表示した例です。



SIMVIEW はデータを ASCII テキストフォーマットで読み込みます。以下はデータファイルの例です。

```

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

```

各メニューから選べる機能を次に説明します。

7.1 File メニュー

機能	説明
Open	ASCII テキスト形式 (拡張子.txt) か SIMVIEW バイナリ形式(拡張子.smv)のデータをファイルから読み込みます。
Merge	現在表示用に使っているファイルに他のファイルを結合させます。
Re-Load Data	現在表示しているファイルの元のデータを再度読み込みます。
Save Settings	現在の波形表示環境の設定を保存します。
Save As	バイナリ形式かテキスト形式、または.csv ファイル (Tab 区切り、カンマ区切り) の中から選択して波形を保存します。

Print	バイナリ形式で保存する場合、現在の設定も保存されます。
Print Setup	FFT 表示中は、FFT の結果をテキスト形式でユーザが指定した名前のファイルに保存します。
Print Page Setup	波形を印刷します。
Print Preview	プリンタの設定をおこないます。
Exit	ページ設定をおこないます。
	印刷画面で確認します。
	SIMVIEW を終了します。

「Merge」

また、「Merge」機能を使うことにより、複数のファイルから読み込んだデータをひとつの波形として表示できます。たとえば、あるファイルが I1 と I2 の波形データを保存し、また別のファイルが V1 と V2 のデータを保存しているとき、4 つの波形を合成してひとつの画面に表示することができます。後から読み込むファイルに同じ I1 という波形データがあった場合は、この波形は I1_1 と自動的に名称が変更されます。

「Re-load Data」

ファイルから読み込んだデータを波形として表示中のとき、（新しいシミュレーションにより）同じファイル名で内容が更新されたデータが得られた場合は、「Re-Load Data」を選択することで新しいデータに基づいて波形を書き直します。

「Save settings」

現在の波形表示、編集の設定を保存することができます。この設定情報は「(SIMVIEWファイル名).ini」というファイルに記録され、そのSIMVIEWファイルを立ち上げる度に自動的に読み込まれます。

7.2 Edit メニュー

機能	説明
Undo	以前の X と Y 軸設定に戻ります。
Copy to Clipboard	スクリーンをクリップボードにコピーします。
View Data Points	データをテキスト形式で表示します。

「Copy to Clipboard」

ウィンドウ内のスクリーンを他の場所へコピーできます。

選択しているウィンドウ内のスクリーンを、ウィンドウの縁を除いてクリップボードへ写します。

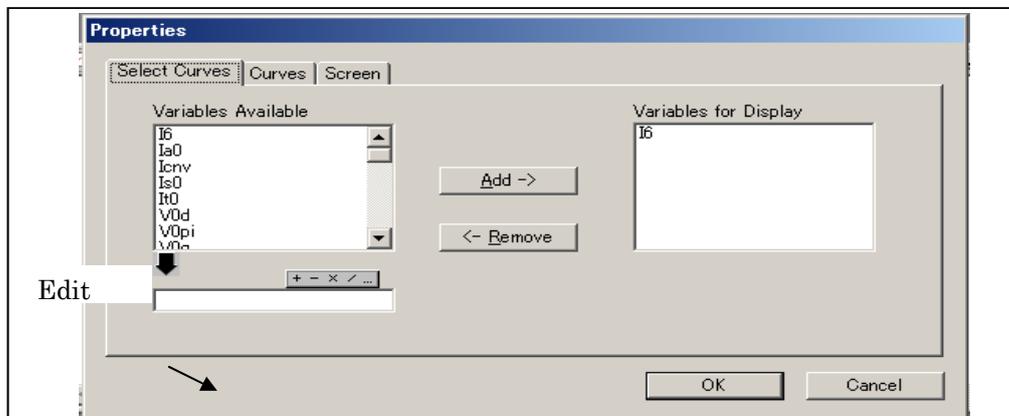
7.3 Axis メニュー

機能	説明
X Axis	X 軸の設定を変更します。
Y Axis	Y 軸の設定を変更します。
Choose X-Axis Variable	X 軸として任意のデータ列を選定することができます。 デフォルトではデータの最初の列、時間 time が X 軸として選定されます。

7.4 Screen メニュー

機能	説明
Add/Delete Curves	選択された画面から波形を追加または削除します。
Add Screen	新しい画面を追加します。
Delete Screen	選択された画面を削除します。

画面を選択するには、マウスを選択したい画面の上において左ボタンを押します。
「Add/Delete Curves」のダイアログ・ボックスを下に示します。



「Select Curves」タブにおいて、表示できるすべての変数は「Variables Available」のボックスに一覧表示されます。現在表示中の変数は「Variables for Display」のボックスに示されています。「Variables Available」のボックスで変数をハイライトしてから、「Add」のボタンを押すと「Variables for Display」のボックスにこの変数が追加されます。同様に、「Variables for Display」の変数をハイライトして「Remove」のボタンを押すと、その変数を削除できます。

「Edit Box」では、次の数式表現を使うことができます。

+	加算	
-	減算	
*	乗算	
/	除算	
^	べき乗	[例 : 2^3=2*2*2]
SQRT	平方根	
SIN	サイン	
COS	コサイン	
TAN	タンジェント	
ATAN	アークタンジェント	
EXP	指数	[例 : EXP(x) = e ^x]
LOG	自然対数	[LOG(x)=ln(x)]
LOG10	常用対数	(基底 10)
ABS	絶対値	
SIGN	符号関数	[例 : SIGN(1.2)=1; SIGN(-1.2)=-1]

これらの数式を使うには、数式を「Edit Box」に打ち込み、「Add」のボタンを押します。数式を変更する場合には数式をハイライトし、「Remove」のボタンを押してください。すると数式が「Edit Box」に表示されるので、数式を変更することができます。

「Curve」タブでは、色、線の太さおよびマーカー・シンボルなどの曲線特性を定義することができます。

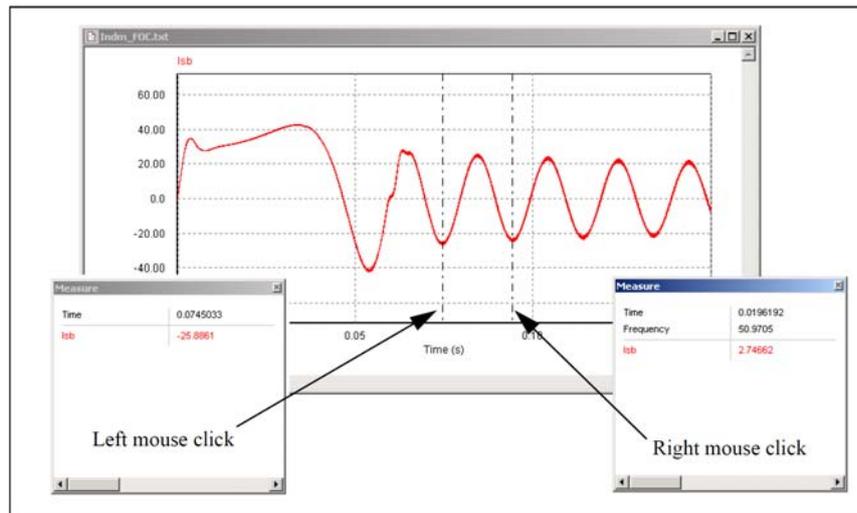
「Screen」タブでは、前景/背景色、グリッド・カラーおよびフォント・サイズ/タイプなどなスクリーン特性を定義することができます。

7.5 Measure メニュー

機能	説明
Measure	波形の値を表示します。
Mark Data Point	選択された点の x 軸、y 軸の値を表示します。

Max	選択された波形の全領域での最大値を求めます。
Min	選択された波形の全領域での最小値を求めます。
Next Max	選択された波形の近傍で次の最大値を求めます。
Next Min	選択された波形の近傍で次の最小値を求めます。

波形の領域を選択するには、マウスの左ボタンを押しながらドラッグしてください。「Measure」機能により、波形の値を測ることができます。「Measure」を選ぶと、「Measurement」ウィンドウが表示されるので、マウスの左ボタンを押すことで波形の上に線が引かれて、その位置での値が表示されます。



7.6 Analysis メニュー

機能	説明
Perform FFT	時間領域で表示中の波形の FFT 結果を表示します。
Display in Time Domain	周波数領域で表示中の波形を時間領域の波形で表示します。
Avg	選択された波形の平均を計算します。
Avg(x)	選択された波形の絶対値の平均を計算します。
RMS	選択された波形の実効値を計算します。
PF (power factor)	スクリーンに表示された 2 つの波形の力率を計算します。このスクリーンには 2 つの波形のみを表示させておく必要があります。力率は $\cos \theta$ で表され、 θ は 2 つの波形の位相差となります。
P (real power)	スクリーンに表示された 2 つの波形の有効電力を計算します。このスクリーンには 2 つの波形のみを表示させておく必要があります。
S (apparent power)	スクリーンに表示された 2 つの波形の無効電力を計算します。このスクリーンには 2 つの波形のみを表示させておく必要があります。
THD	THD を計算します。

これらのすべての機能は、現在スクリーンに表示されている時間範囲(x 軸範囲)に適用されます。

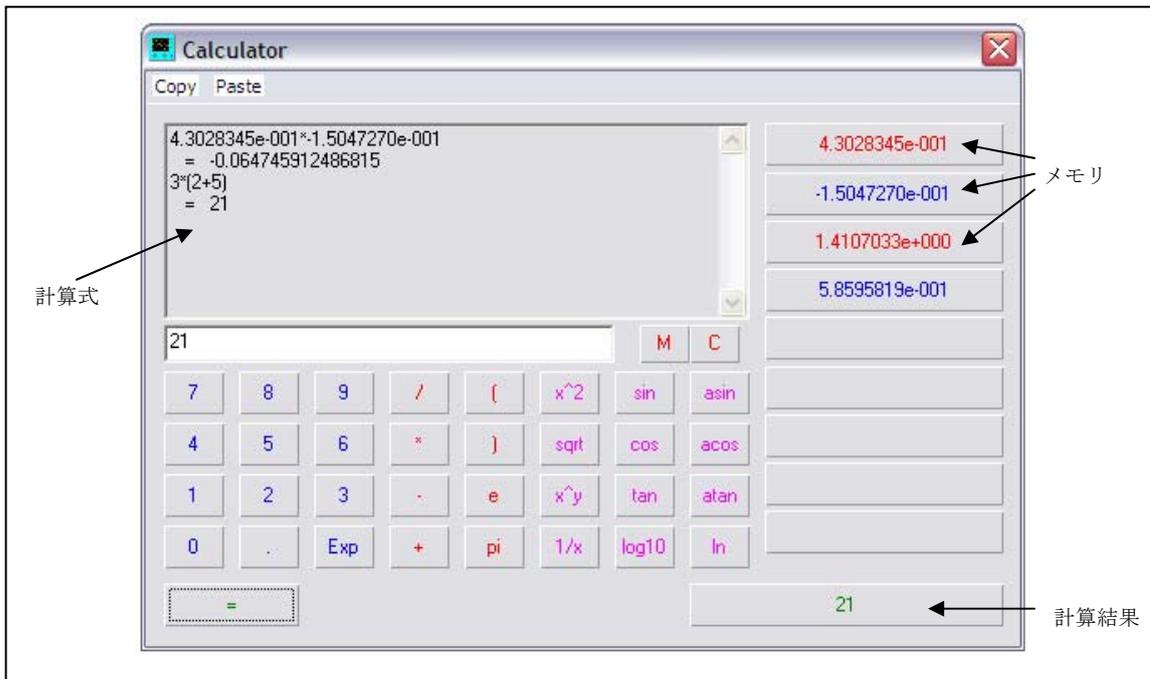
Perform FFT を選ぶことにより、時間領域の波形に含まれる高調波の振幅と位相を計算して表示します。FFT 結果の位相を表示するには、スクリーン上の任意の場所をダブルクリックするか **Add/Delete** アイコン  をクリックして下さい。**Select Curves** タブで **Angle** タブをクリックし、位相を選択します。位相の名称は、単位が度(deg)の場合は $Angle(D)_{\{name\}}$ 、ラジアンの場合は $Angle(R)_{\{name\}}$ となります。またここでの $\{name\}$ には時間領域の波形の名称が入ります。

正確な FFT 結果を得るには、シミュレーションが定常状態に収束して、データが基本周期の整数倍に収まるように調整してから「Perform FFT」を実行してください（X-Axis の機能を使ってレンジを調整してください）。

7.7 View メニュー

機能	説明
Zoom	選択された範囲を拡大表示します
Re-Draw	波形を再表示します。倍率は自動で調整されます。
Escape	Zoom または Measure モードから抜けます。
Standard Toolbar	Standard ツールバーの表示を有効・無効にします。
Measure Toolbar	Measure ツールバーの表示を有効・無効にします。
Status Bar	ステータス・バーの表示を有効・無効にします。
Calculator	Simview の電卓を起動します。

電卓のインタフェース画面は以下の通りです。



7.8 Options メニュー

機能	説明
Options...	Redraw x-axis when loading new data をチェックすると、データを再読み込みした際に x 軸の範囲設定を新しいデータのものを反映します。
Color	Right mouse action では右クリックしたときの動作を選択できます。波形の色を多色 (Color; デフォルト)、黒 (Black)、または白 (White) に切り替えます。
Grid	グリッドを表示・消去します。

7.9 Label メニュー

機能	説明
Text	文字を画像に追加します。
Line	線を引きます。
Dotted Line	破線を引きます。
Arrow	矢印のついた線を引きます。

線を引くには、まず、「Label」メニューから「Line」を選んでください。線の始点でマウスの左ボタンを押し、左ボタンを押したままマウスをドラッグします。破線や矢印も同様にして描けます。

「Zoom」または「Measure」のモードにいるときに文字またはラベルを編集するには、まず「View」メニューから「Escape」を選んで「Zoom」または「Measure」モードを抜けてください。

7.10 Settings メニュー

機能	説明
Re-Load Settings	.ini ファイルから設定を再読み込みし、現在の表示に適用します。
Save Settings	現在の設定を.ini ファイルに保存します。
Save Temporary Settings	現在の設定を一時的に保存します。一時設定は他のファイルには適用されず、ファイルを閉じると無効になります。
Load Temporary Settings	一時設定を読み込み、現在の表示に適用します。
Add to Favorites	現在の設定をお気に入り(favorite)に登録します。お気に入りに登録するとき、線の色や太さ、テキストフォント、Log/dB/FFT 表示設定、x 軸と y 軸の幅の保存が選択できます。
Manage Favorites	お気に入りを管理します。

Simview がデータファイル (.txt または .smv ファイル) を読み込むとき、対応する.ini ファイルが存在する場合はこの設定を自動的に読み込みます。

Load Temporary Settings と Save Temporary Settings の機能は設定を一時的に保存したい場合に使います。例として、一つの波形を他の波形と比較したい場合、ユーザは最初の波形を表示している時に“一時保存”を選択します。その後、2つ目の波形を表示させる際、一時設定を適用します。

Favorite (お気に入り) は、グラフ設定を後で使用するために保存しておくのに便利な方法です。例として、Simview が2つのスクリーンを表示する場合を考えます。上のスクリーンでは V1 を赤色で、特定の x 軸、y 軸幅で表示させていて、下のスクリーンでは V2 を青色で特定の y 軸幅で表示させています。この設定を再度使いたい場合は、この設定をお気に入りに登録し、後で使用することができます。

現在の表示をお気に入りに登録する際は、Setting メニューの“favorite”を選択して下さい。お気に入りに登録するときは、現在表示しているスクリーンの番号とお気に入りのスクリーン番号を一致させる必要があることに注意して下さい。

7.11 データの書き出し (Export)

「FFT」の結果はテキスト・ファイルに保存できます。シミュレーション結果 (*.txt) と「FFT」の結果 (*.fft) が両方ともテキスト形式で保存できるので、テキスト・エディタもしくは他のソフトウェア (Microsoft Excel など) で読み込むことができます。たとえば、Excel では、データファイルを開くことにより、データは自動的に表形式に変換されます。

7.12 実際の SIMVIEW 使用の流れ

ここではSIMVIEWの基本操作手順の流れを説明していきます。

7.12.1 SIMVIEW の立ち上げ

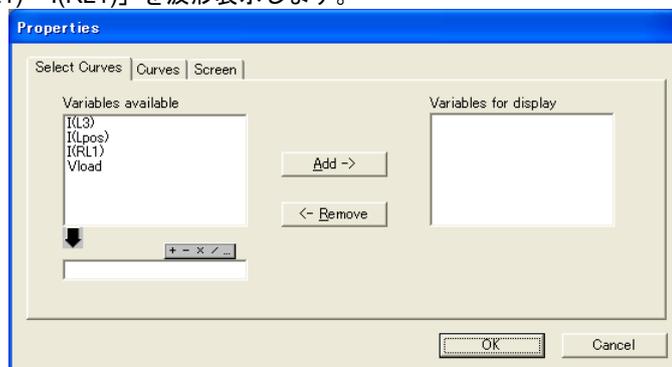
ここでは、SIMVIEW起動時の操作について説明します。

SIMVIEWの起動方法は大きく分けて二つあります。

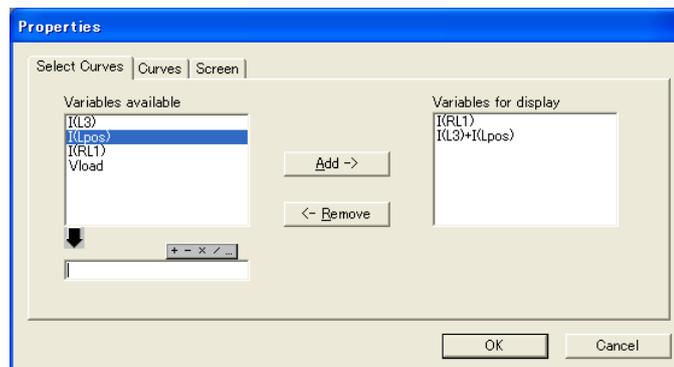
- ・ 「simview.exe」 から起動する
- ・ PSIM内の「Options」の「Auto-run SIMVIEW」にチェックを入れた状態で、「Run Simulation」を行う。

今回は後者の方法でSIMVIEWを起動させます。

- 「波形表示する変数指定」
SIMVIEWが立ち上がり、下に示す「Properties」ウィンドウが現れます。ここで、どの変数を波形表示するのかを決めます。（今回は過渡解析結果の状態になっていますので、自動的にx軸はTimeに設定されています。）「Variables Available」からy軸に設定したい変数を選び、「Add」をクリックすると、その変数が「Variables for display」の中に移動します。
- 「計算」
また、変数を計算処理してから波形を作ることもできます。ウィンドウ内左下（「Variables Available」の下）の欄に数式を記入して「Add」を押すと、その数式を「Variables for display」の中に移動させることもできます。今回は下図（Properties 変数決定）のように「I(RL1)」、「I(RL1) × I(RL1)」を波形表示します。

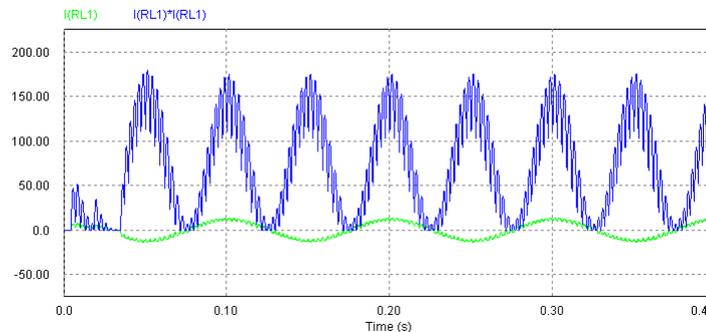


SIMVIEW Properties



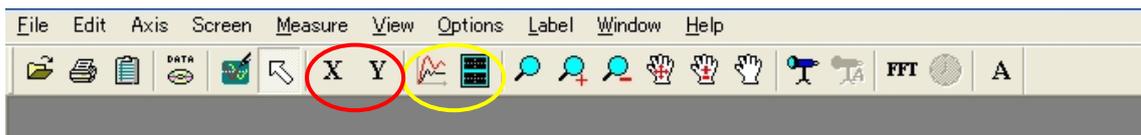
SIMVIEW Properties 変数決定

- 「OK」を押すと下図のようにスクリーン上の波形が表示されます。



SIMVIEW I(RL1)及び I(RL1)×I(RL1))波形

7.12.2 波形表示



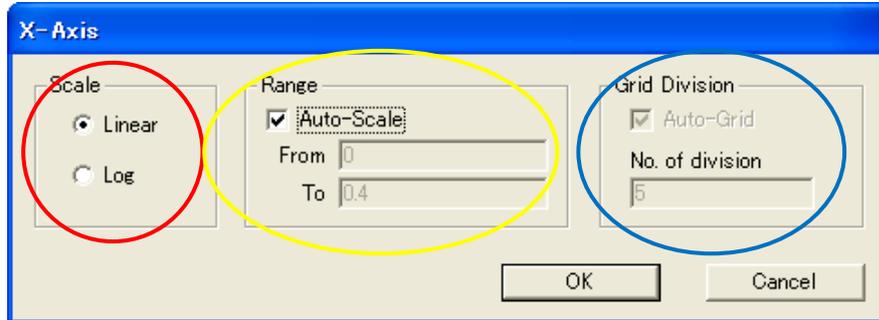
SIMVIEW 上部ツールバー1

次に、スクリーンまたは波形の追加、軸の設定変更を行います。

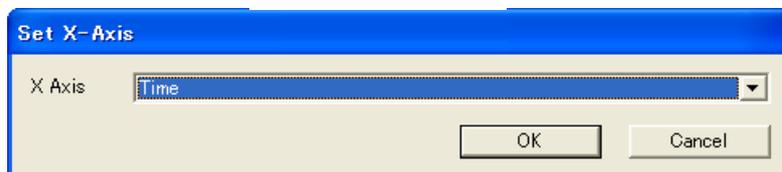
- 「スクリーンの追加」
 ウィンドウ上側の「Add one Screen」（上図（上部ツールバー1）の黄丸内右側）をクリックすると、再び「Properties」のウィンドウが現れますので、波形表示したい変数を「Add」して「OK」をクリックします。すると、ウィンドウの中に新しいスクリーンが作成されます。
- 「スクリーン内に波形を追加」
 「Add one Screen」の左横の「Add or Delete Curves to selected Screen」をクリックすると、再び「Properties」ウィンドウが現れますので、「Add」と「Remove」を用いて波形の追加や削除をします。
- 「軸の設定」
 上図（上部ツールバー1）の赤丸内の「X」をクリックすると、下図の「X-Axis」ウィンドウが現れ、x軸の設定を行うことができます。
 図「X-Axis」内の赤丸内で、x軸の目盛をLinear（線形）もしくはLog（対数）に変更できます。
 図「X-Axis」内の黄丸内で、表示するx軸の目盛の幅を決めることができます。
 図「X-Axis」内の青丸内で、目盛を表示するグリッドの数の設定ができます。例えば、図中では0から0.4までを5個の目盛で分割するという設定になっていますので、上図「I(RL1)及びI(RL1)×I(RL1))波形」では、「0.0」、「0.10」、「0.20」、「0.30」、「0.40」に目盛が表示されています。
 y軸も「Y」をクリックすることで、同様の設定変更が可能です。

- 「x軸の変更」

「Axis」 → 「Choose X-Axis Variable」 を選びますと、下図「Set X-Axis」のようなウィンドウがでてきます。ここでx軸に設定する変数を変更することができます。



SIMVIEW X-Axis

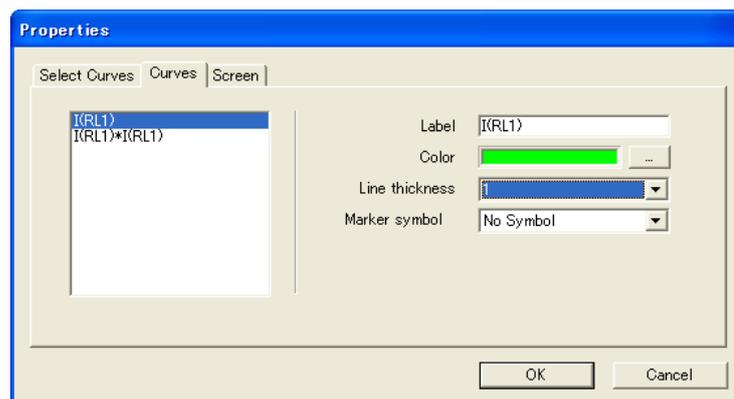


SIMVIEW Set X-Axis

- 「波形編集」

「Add or Delete Curves to selected Screen」 をクリックすると、「Properties」ウィンドウが現れますので、上側のタブの内「Curves」をクリックすると、下図「Properties Curves」のような設定画面が現れます。

「Label」で、波形の表示名を変更できます。また、「Color」で波形の色、「Line Thickness」で波形の太さ、そして「Marker Symbol」で波形の軌跡を描いている点の形を選ぶことができます。

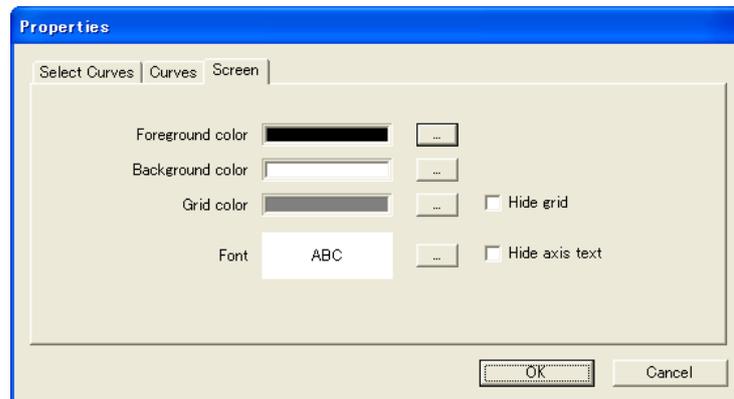


SIMVIEW Properties Curves

- 「スクリーン編集」

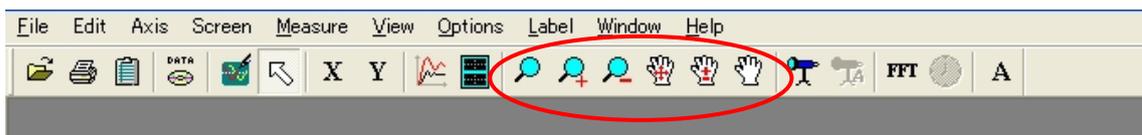
同じウィンドウ「Properties」の、上側のタブの内「Screen」をクリックすると下図「Properties Screen」のような設定画面が現れます。

「Foreground color」で、スクリーンの外枠の色を編集できます。また、「Background color」で背景の色、「Grid color」でグリッドの色、そして「Font」でスクリーン内の文字のフォントを編集できます。



SIMVIEW Properties Screen

7.12.3 波形観測



SIMVIEW 上部ツールバー2

次に、波形の観測を行います。

- 「Zoom Box」

上図「上部ツールバー2」の赤丸内の左の「Zoom Box」を使うと、画面内の任意の位置をズームインして観測することができます。波形の特定の箇所を拡大して見たいときに使います。

「Zoom Box」をクリックすると、マウスカーソルの形が変わりますので、この状態で、画面上でズームインしたい場所にマウスを持っていき、左クリックをドラッグしながら四角を描きます。そしてマウスの左クリックを戻すと、描いた四角内部がズームインされます。

- 「Zoom InとZoom Out」

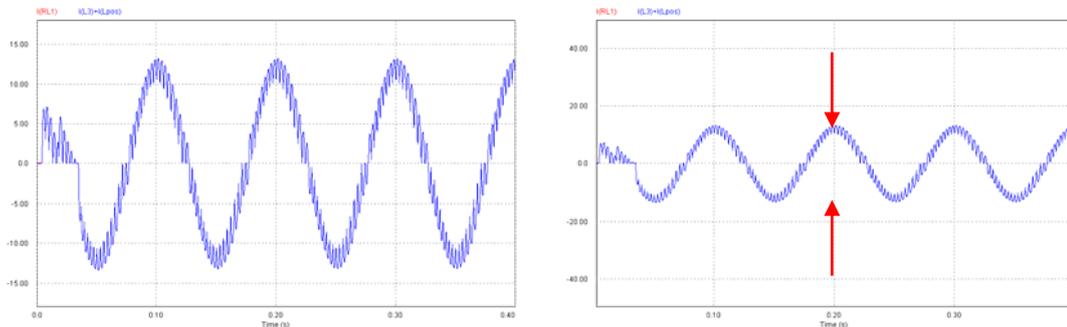
左から二番目と三番目はそれぞれ「Zoom In」と「Zoom Out」です。これらは、クリックする度に少しずつ画面をズームイン、またはズームアウトさせることができます。波形表示を少しずつ変化させて観測したいときに使います。（画面の中央をズームする仕様ですので、ズームする場所を変えたい場合は、図8の赤丸内の一番右の「Pan」を使って波形の中心位置を変えて下さい。）

- 「Zoom」

次の左から四番目と五番目は、互いに「Zoom」と名付けられていますが、使用方法が少し違います。

四番目の「Zoom」は、マウスカーソル位置に対して、画面の上下または左右が均等にズームされます。ボタンをクリックすると、マウスカーソルの形が変わります。この状態で画面中央にカーソルを持っていきます。

例として、左クリックでドラッグしながらカーソルを上にもっていくと、下図「Zoom1」のようにマウスをドラッグした位置に対して、y軸方向上下対称にズームアウトされていきます。（マウスを上:y軸方向をズームアウト、マウスを下:y軸方向をズームアップ、マウスを右:x軸方向をズームアウト、マウスを左:x軸方向をズームアップなど。）



(a)

(b)

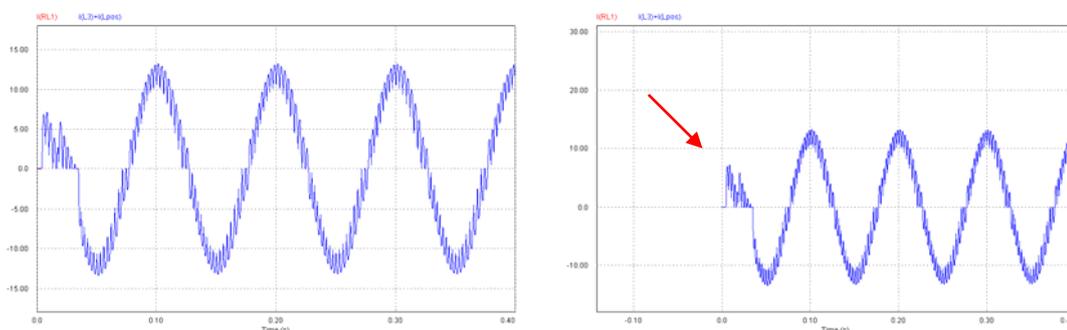
SIMVIEW Zoom1 (a)Zoom 前、(b)Zoom 後

五番目の「Zoom」は、マウスカーソル位置に対して、スクリーンの上下または左右一方向にスクリーンをズームします。ボタンをクリックすると、マウスカーソルの形が変わります。

例として、スクリーン中央より左上にカーソルを持って行って、左クリックでドラッグしながらカーソルを右下にもっていくと、下図「Zoom2」のようにマウスをドラッグした位置に対してスクリーンが右下に縮小されていき、ズームアウトされていきます。（例えば、マウスを右上からさらに右上へドラッグしながら持っていくと左下方向がズームアップされる形になります。）

- 「Pan」

一番右側の「Pan」は、ここを選択した状態で、スクリーン上の任意の位置をドラッグすると、ドラッグしながらマウスの動きに合わせてスクリーンを自由に縦横に移動させることができます。

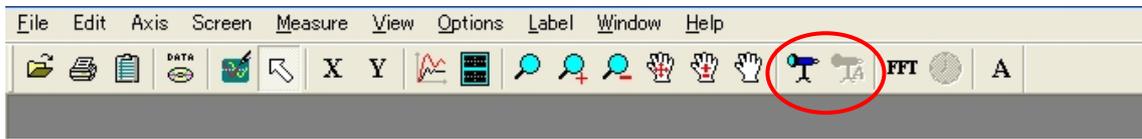


(A)

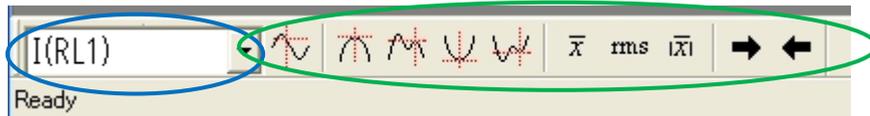
(b)

SIMVIEW Zoom2 (A)Zoom 前、(b)Zoom 後

7.12.4 測定



SIMVIEW 上部ツールバー3



SIMVIEW 下部ツールバー

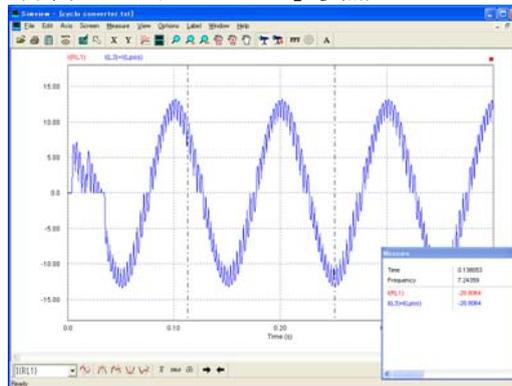
ここでは、波形から値の測定をします。

- 「メジャーカーソル1」

SIMVIEWウィンドウ上側の上图「上部ツールバー3」の赤丸内の「Show values of X and Y At the selected curves」をクリックすると、x軸に垂直なメジャーカーソルが現れます。左クリックでドラッグしながらマウスを動かすと、それに合わせてメジャーカーソルも動き、その点での波形のxy値が右下のウィンドウに表示されます。（また、赤丸内の「Place a text[X, Y] at the selected curves」をクリックすると、その点でのxy値をスクリーン上にテキストボックスとして作ることができます。）

- 「メジャーカーソル2」

ここで、スクリーン上で右クリックをすると新しいメジャーカーソルが現れます。右クリックでドラッグをしながらマウスを動かすと、それに合わせてそのメジャーカーソルも動きます。このとき、「Measureウィンドウ」には二つのメジャーカーソル間の差の値が表示されます。（下图「メジャーカーソル」参照）



SIMVIEW メジャーカーソル

- 「水平方向メジャーカーソル」

上图「下部ツールバー」緑丸内の一番左をクリックすると、メジャーカーソルが垂直方向だけでなく、水平方向も現れます。もう一度クリックすると垂直方向だけに戻ります。

- 「最大値」

上图「下部ツールバー」緑丸内の左から二番目の「Find global maximum value」をクリックすると、波形のスクリーン表示範囲内で最も大きなy値を持つ場所にメジャーカーソルが移動します。波形の最大値を求めたい時などに使用します。

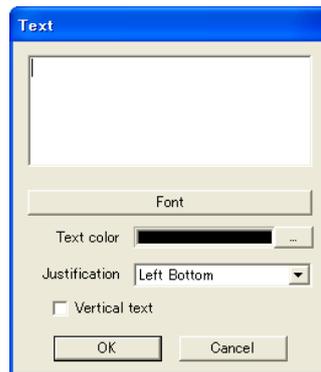
- 「次の最大値」
その右隣りの「Find next maximum value」は、現在値から近傍で最も大きな値にカーソルが移動します。
- 「最小値」
上図「下部ツールバー」緑丸内の右から二番目の「Find global minimum value」をクリックすると、波形のスクリーン表示範囲内で最も小さなy値を持つ場所にメジャーカーソルが移動します。波形の最小値を求めたい時などに使用します。
- 「次の最小値」
その右隣りの「Find next minimum value」は、現在値から近傍で最も小さな値にカーソルが移動します。
- 「y値の平均値、RMS、絶対値の平均値」
次の三つはそれぞれ、波形のスクリーン表示範囲内での、y値の平均値、RMS、そして絶対値の平均値を計算し、「Measureウィンドウ」に表示してくれます。
- 「メジャーカーソルの微小ステップ移動」
右から一、二番目はそれぞれ「go to next Point」と「Go to previous point」です。クリックするごとに、それぞれ微小ステップでx軸の増加方向、または減少方向にメジャーカーソルを移動させることができます。少しずつx値を変えて波形上をスイープして値を測定したいときなどに使います。
- 「測定する波形の切り替え」
最後に、一つのスクリーン内に複数の波形がある場合は、上図「下部ツールバー」青丸内によって波形の切り替えが行えます。

7.12.5 描画

最後に、スクリーン上への描画機能を使用してみます。

描画機能はSIMVIEWウィンドウ上部のツールバーの「Label」内に納まっています。

- 「Text」
スクリーン内に文字をテキストボックスとして配置することができる機能です。
「Label」内の「Text」を選ぶと、下図「Text」のようなウィンドウが立ち上がります。空欄部に配置したい文字を書き込んで下さい（日本語も可能です。）
「Font」では、文字のフォント、スタイル、そして文字の大きさを変更可能です。
「Text color」では、配置する文字の色の設定ができます。
「Justification」では、行端の揃え方を指定することができます。
「Vertical text」にチェックを入れると、配置するテキストを90度回転させることができます。
「OK」をクリックするとウィンドウが消えますので、文字を配置したい場所にマウスカーソルを持っていき、左クリックで文字を配置できます。



SIMVIEW Text

- 「Line」、「Dotted Line」、「Arrow」
スクリーン内に線を引く機能です。
「Label」内から引きたい線を選び、スクリーン内にマウスカーソルを持ってきます。
線の始点としたいところで左クリック、ドラッグしたまま、線の終点としたいところへマウスを持っていき、左クリックを離すことで線が引けます。
一度、線を引いた後でも、線の両端をドラッグすれば、再度、線の始点、終点の変更が可能です。

第8章 エラー・警告と全般に関する注意

8.1 全般に関する注意

8.1.1 タイムステップ

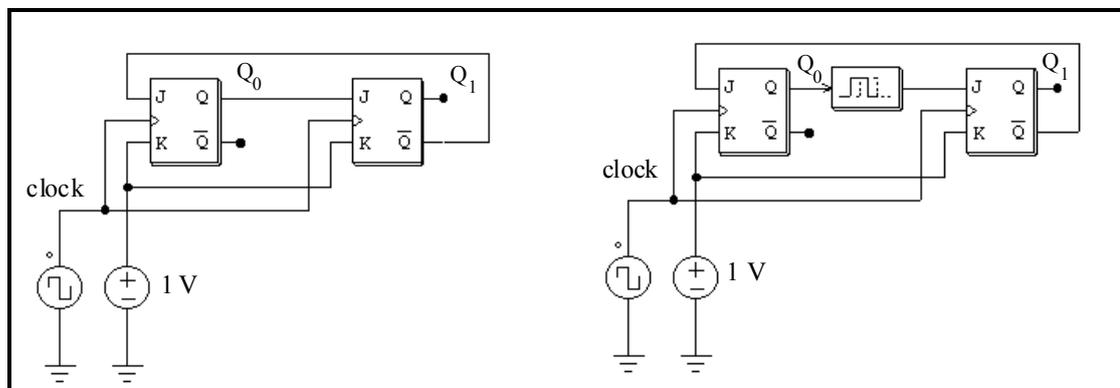
PSIM ではシミュレーションのタイムステップは固定です。正確な結果を得るためには、タイムステップを適切に選ぶ必要があります。適切なタイムステップは、たとえば、スイッチ切り換え期間、パルスや方形波の幅、変化の速い過渡現象の継続時間などにより決まります。タイムステップはこれらの現象のうち最も応答時間が短いものに較べてさらに1桁以上小さくなるように設定してください。

8.1.2 論理回路の伝搬遅れ

PSIM 内部の論理回路は理想回路で伝搬遅れがありません。伝搬遅れを動作に含む論理回路をシミュレーションするには、PSIM では時間遅れ (TDELAY) ブロックと呼ばれる関数ブロックを使うことで伝搬遅れを実現できます。

この例として、2ビット・カウンタ回路を以下に示します。

上の回路では、初期値 Q_0 と Q_1 はいずれも零と仮定しています。クロックの立ち上がりにおいて、 Q_0 は



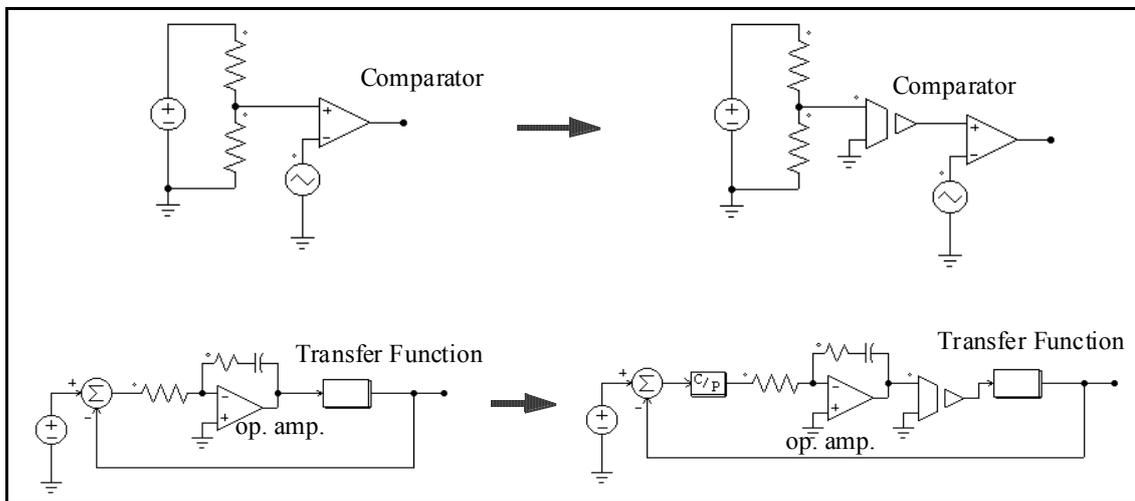
0 から 1 に状態変化します。伝搬遅れなしとすると、 Q_1 は Q_0 と同時に 1 に切り替わってしまいます。これを防ぐために、右の回路では 1 タイムステップの時間遅れを持つ時間遅れ要素を Q_0 と次のフリップフロップの入力 (J) の間に追加しています。

8.1.3 パワー回路と制御回路のインターフェース

PSIM ではパワー回路は個別の回路素子で表現され、制御回路は伝達関数ブロック線図で表現されています。パワー回路の要素、たとえば、RLC ブランチ、スイッチ、変圧器、相互結合インダクタ、電流源、接地されていない電圧源、そしてすべての制御付き電源は、制御回路では使うことができません。同様に、制御回路の要素、たとえば、論理ゲート、PI 制御器などの関数ブロックはパワー回路では使えません。

パワー回路と制御回路の入力を直接つなげた場合は、PSIM は電圧センサを自動的に挿入します。同様に、制御回路の出力を直接、パワー回路につなげた場合、制御・パワー変換ブロック (CTOP) が自動的に挿入されます。これらの様子を下の図に示します。

PSIM では、パワー回路と制御回路は別々に解かれることにご注意ください。パワー回路と制御回路の間には、1 タイムステップ分の時間遅れがあります。



8.1.4 FFT による高調波解析

FFT を使って高調波解析をする場合、次の条件が満たされることを確認してください。

- 波形が定常状態に達したこと；
- FFT 解析に指定したデータの範囲が基本波の周期の整数倍であること。

たとえば、60Hz の波形では、データの長さは 16.67msec かその倍数に設定する必要があります。これを守らないと FFT の結果が正しくありません。データ範囲の設定は、SIMVIEW で「X Axis」をクリックし、「Auto-scale」のチェックボックスをクリアした上で、From と To に適切な値を設定することで行うことができます。FFT 解析は画面に表示されている範囲に対してのみ実行されます。

FFT 解析結果は離散的であることに注意してください。FFT 解析結果は、データの時間間隔 Δt と解析するデータの長さ T_{length} に依存します (Δt はシミュレーションタイムステップの print step 倍です)。FFT 解析の基本波周波数は $1/T_{\text{length}}$ となり、 $\Delta f=1/T_{\text{length}}$ 毎に解析結果が得られます。また、最大周波数は $f_{\text{max}}=1/(2*\Delta t)$ となります。

例えば、1kHz の方形波を $10\mu\text{s}$ 刻みで 1ms 表示させて FFT 解析した場合、 $T_{\text{length}}=1\text{ms}$ 、 $\Delta t=10\mu\text{s}$ となります。従って、 $\Delta f=1/T_{\text{length}}=1\text{kHz}$ となり、解析結果の最高周波数は $f_{\text{max}}=1/(2*\Delta t)=50\text{kHz}$ となります。

8.2 エラーおよび警告メッセージ

次にエラーおよび警告メッセージについて解説します。

E-1 Input format errors occurred in the simulation.

(入力データの形式が正しくありません)

このエラーは次のいずれかの原因によります。

- データ設定が正しくないかまたは不足している
- 整数および文字列の入力が正しくない。

PSIM ライブラリが変更されていないか、またはお使いの PSIM が最新バージョンであることを確かめてください。

回路接続情報ファイルでは、文字列はアポストロフィで囲みます (例: 'test')。また、整数の変数に整数を指定していることを確認してください。整数のパラメータに実数を入れると (たとえば、3 の代わりに 3.) このエラー・メッセージが出ます。

E-2 Error message: The node of an element is floating.

(端子が接続されていません)

このメッセージは PSIM で作成した回路の接続が悪いときに表示されます。2つの端子を結線するときは、端子が素子に接続されているか確かめてください。

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

(要素の数が多すぎます)

このメッセージはある要素の全体数がプログラムの上限を越えた時に出力されます。この問題を解決するには、PSIM プログラムの配列を大きくして再コンパイルする必要があります。Myway プラスにご相談ください。

W-1 "Warning!!! The program failed to converge after 10 iterations when determining switch position. The computation continues with the following switch positions: ..."

(10 回の収束計算でもスイッチの状態が決まりませんでした。スイッチの状態は以下のように決めてシミュレーションを続行します)

このメッセージはプログラムがスイッチ状態を決める過程で収束しなかったときに表示されます。シミュレーションは 10 回目の計算を終えた時点のスイッチ状態を使って続行します。シミュレーション結果は正確ではないかもしれませんが、結果を解析するときには充分ご注意ください。

この問題はいろいろな場合に生じる可能性があります。次の対策により、原因を特定し修正することができます。

- 回路の接続が正しいか確認してください；
- スwitchのゲート信号を調べてください；
- スwitchと電圧源に直列に小さな抵抗またはインダクタを接続してください。

W-2 "Warning!!! The program did not reach the steady state after 60cycles when performing the ac sweep."

(周波数解析を行いました。60 回反復しても計算が収束しませんでした。)

このメッセージは周波数解析を行った際に、計算が 60 回反復しても収束しない問題が起きた場合に現れます。この問題の原因として、システムが、特定の周波数で著しく減衰を引き起こすものであるか、もしくは信号振幅が小さすぎるか、が考えられます。

以下の方法で問題を解決して下さい。

- 問題が起きた周波数の信号源で時間領域のシミュレーションを走らせ、時間領域の波形が発振していないかどうかを調べて下さい。
- 信号レベルを上げるために入力の電圧源の振幅を大きくして下さい。
- または適切な時間区切りになるように時間ステップを減らして下さい。

8.3 デバッグ

シミュレーションで問題が発生する例について、対策方法を説明します。

問題：

インダクタ電流とキャパシタ電圧のシミュレーション結果が急に変化する（不連続がある）。

対策：

この障害はインダクタの電流経路を開いたか、キャパシタを含むループ回路を短絡したことが原因です。スイッチのゲート信号を調べてください。必要ならば、パルスを重ねたり、間隔を開いたりして開路や短絡をふせいでください。

インダクタの初期電流が指定されているとき、スイッチの初期状態はそのような電流が流れるように閉状態としてください。そうでなければ、電流は零から始まることになります。

問題：

波形が正しくない、または正確でない。あるいは、分解能が悪い。

対策：

この原因は2つあります。ひとつは刻み時間の設定です。PSIMはシミュレーション全体で一定のタイムステップを使うので、タイムステップは十分に小さく設定するようにしてください。おおよその目安として、タイムステップはスイッチ周期の数十分の1にしてください。

もうひとつの原因は波形の表示のしかたです。表示刻み `lprint` が大きすぎないように設定してください。すべてのデータ点を表示するには `lprint = 1` とします。

PSIM Version 9.1 User's Guide

発行 : Myway プラス株式会社

〒220-0022

横浜市西区花咲町 6-145 横浜花咲ビル

TEL : 045-548-8836

FAX : 045-548-8832

ホームページ : <http://www.myway.co.jp>

Eメール : sales@myway.co.jp
