Myway

POWERSIM

PSIM バッテリモデルおよび Ver.11.0 の新機能

PSIM

- **バッテリにおける OCV および内部抵抗の SOC 依存モデル（ver10.0.6 以降）**
- **SPICE エンジンを搭載した SPICE モジュール**
- **スクリプト機能によるシミュレーション実行や波形の後処理**
- **C ブロックの機能拡張**
- **ベクトル軌跡プロット**
- **PIL モジュールによる実 CPU との HW 協調シミュレーション**
- SmartCtrl のバージョンアップ
- SimCoder の Expert4 対応（4 月以降）

バッテリにおける OCV および内部抵抗の SOC 依存モデル

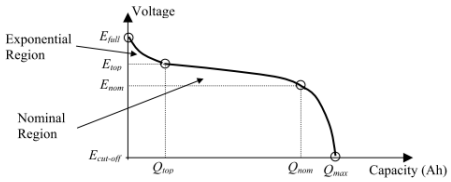
PSIM のリチウムイオンバッテリモデルは、再生可能エネルギーモジュールに含まれる素子のひとつです。直並列のセル数や電圧・電池容量・内部抵抗などの特性を入力してシミュレーションすることができます。最新版では OCV（開回路での端子間電圧）および内部抵抗の値を **SOC によるルックアップテーブルデータ**として入力することができるようになりました。た。

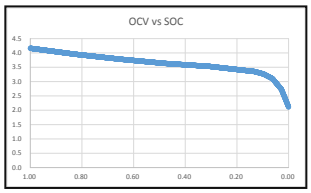
また、バッテリメーカが配布しているデータシートのグラフから簡単に**放電特性カーブをキャプチャ**することができるようになりました。実験データがなくてもデータシートがあれば放電特性をシミュレーションで再現させることができます。

従来のモデル（４点入力）

ルックアップテーブルモデル　（任意のポイント数）

OCV



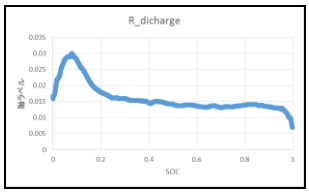


SOC

ルックアップテーブルモデル　（任意のポイント数）

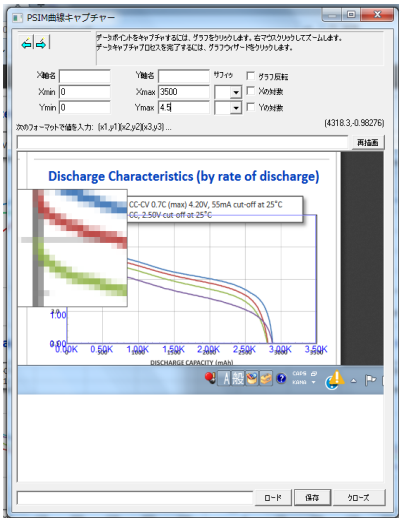
内部抵抗

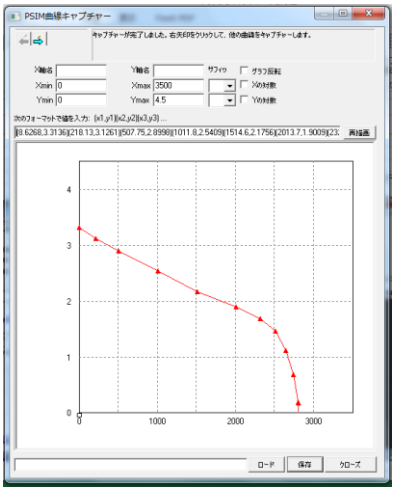
従来は固定値



SOC

キャプチャ機能

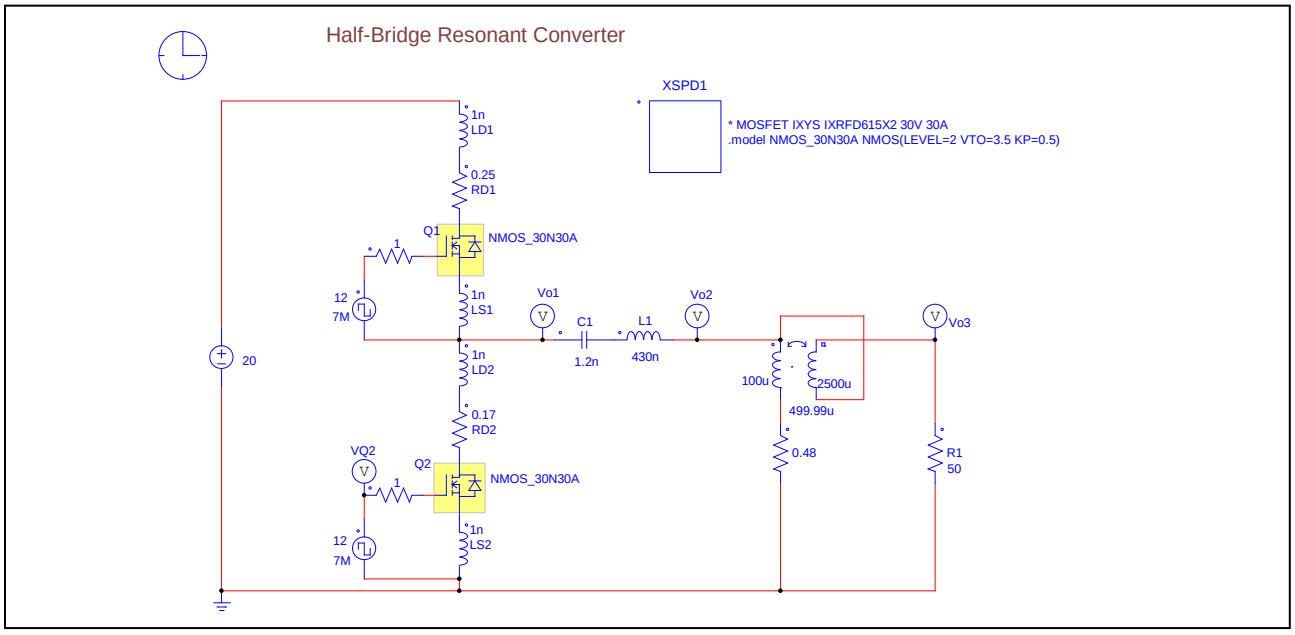




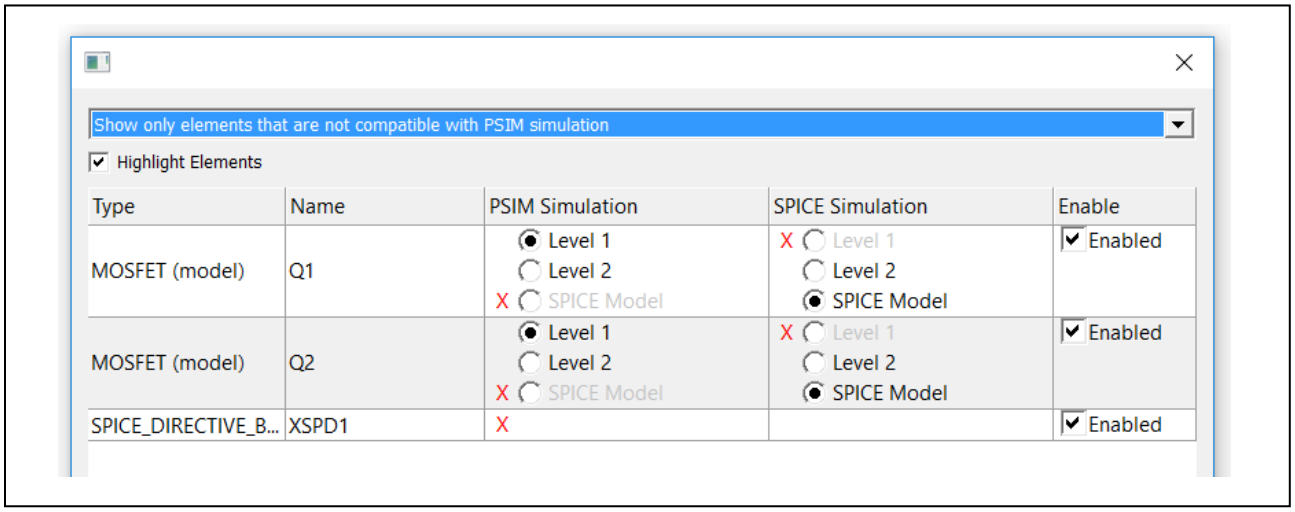
拡大してマウスでクリックしながらデータ取込み

SPICE エンジンを搭載した SPICE モジュール

新しく SPICE モジュールが追加されました。これにより、SPICE 対応素子や **SPICE ネットリスト**でモデリングされた回路を SPICE エンジンによりシミュレーションできるようになりました。従来の理想素子による PSIM シミュレーションに比べてより**詳細なモデリング**ができるようになります。また、半導体メーカからダウンロードした **SPICE モデル**をライブラリとして登録して使用することも可能です。下図は SPICE モデルを用いた MOSFET の共振コンバータ回路です。

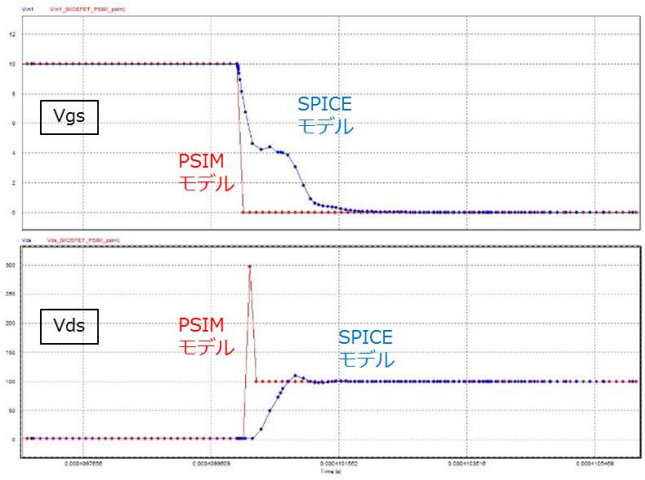


PSIM シミュレーション用のモデルと SPICE シミュレーション用のモデルをリストから簡単に切り替えることが可能です。



MOSFET ターンオフ

過渡解析波形



スクリプト機能によるシミュレーションの実行や波形の後処理

新しく追加されたスクリプト機能により、**シミュレーションの実行**や整数・浮動小数点数・複素数・文字列・配列などの**各種演算**や**条件分岐**、ボード線図やベクトル線図を含む**グラフのプロット**などが実行できるようになりました。

下記の例は、降圧コンバータ回路における出力インダクタの電流リップルの最大値を抽出するスクリプトです。

```
// The purpose of this script is to find out the inductor current ripple for the circuit "buck converter - script.psimsch"
Lind = 20e-6;
out = Array(0);
inc = 0;
ScriptOption("NoLog: *", "Log:iL_ripple,inc,L2");

while (Lind <= 60e-6)
{
        File1 = GetLocal("PARAMPATH") + "buck converter - script.psimsch";
        File2 = GetLocal("PARAMPATH") + "buck converter - script.txt";
        L2=Lind;
        Simulate (File1, File2, "", g1);   // run simulation and associate the result with object g1

        row = SizeOf(g1[0]);                    // read number of rows
        iL_max = -1000.;
        iL_min = 1000.;
        count = 0;
        To = g1[0];                             // obtain time column
        iL = g1[1];                             // obtain iL column
```

```
        while (count < row)
        {
                if (To[count] > 0.4e-3)                 // start checking after t = 0.4ms
                {
                        if (iL[count] > iL_max)
                        {
                        iL_max = iL[count]                      // save maximum
                        }
                        if (iL[count] < iL_min)
                        {
                        iL_min = iL[count]                      // save minimum
                        }
                }
                count++;
        }
        iL_ripple = iL_max - iL_min;                    // calculate ripple
        AddToArray(out, "iL ripple = " + string(iL_ripple) + " at iL = " + string(L2));    // save to out array
        inc++;
        Lind = Lind + 5e-6;
}
File3 = GetLocal("PARAMPATH") + "buck circuit - script - output.txt";
FileWrite(File3, out);                          // write result to a file
```

このスクリプトではインダクタンスを 20uH から 60uH まで 5uH ステップで変化させています。それぞれの値でシミュレーションの実行とリップル値の抽出が行われています。

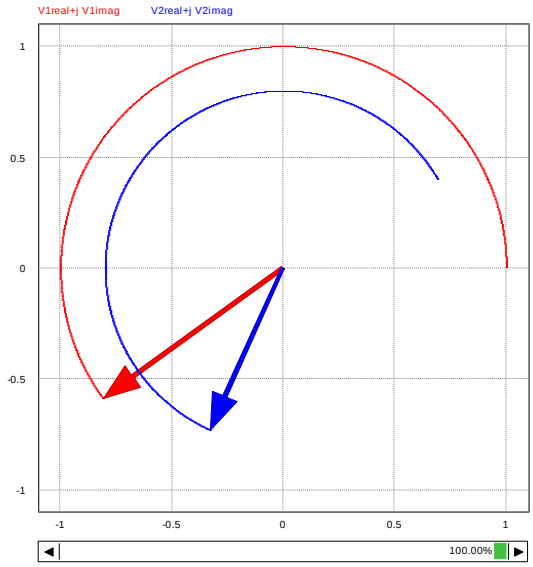
C ブロックの機能拡張

C ブロックの機能がいくつか改善されています。

- C ブロックから DLL ブロックへ変換するために必要な、Visual Studio 用のプロジェクトを生成することが可能です。
- 外部ファイルとリンクする機能が追加されました。これによりコピー・貼りつけの手間がなくなり、**好きなエディタで編集**した結果が即座に反映されます。

ベクトル軌跡プロット

ベクトル軌跡のプロット機能が追加されました。ベクトルは実部と虚部で定義されます。シミュレーション実行の開始から終了までの**任意の時間における軌跡**を表示することができます。



PIL モジュールによる実 CPU との HW 協調シミュレーション

新しく PIL モジュールが追加されました。これにより、制御部分を実 CPU に書き込んで、通信しながら **HW 協調シミュレーション**を行うことができます。プラントの実機がない状態で CPU のデバッグが開始できるため**開発期間を短縮**することができます。