

# RT Box Tutorial

## Model Optimizations

モデルの最適化

Tutorial Version 1.1

# 1 はじめに

リアルタイムシミュレーションモデルの複雑さが増すにつれて、望ましいパフォーマンスを実現するため、さらなる最適化手順が必要になる場合があります。このチュートリアルでは、ハードウェアインザループ(Hardware-In-the-Loop: HIL)アプリケーションに重点を置いた RT Boxでのリアルタイムシミュレーションのための、さらなる最適化手順について説明します。

このチュートリアルの最後には、PLECSモデル内のスイッチングの組み合わせ数を減らす方法を理解することができます。これにより、モデルの実行時間を短縮し、生成するコードのサイズを小さくできることがわかります。

**始める前に** この演習では、以前のRT Boxチュートリアルに関する知識が必要です。このチュートリアルには、RT BoxとPLECS Coderのライセンスが必要です。

## 2 最適化の概念

リアルタイムシミュレーションモデルが複雑になると、特にHILアプリケーションでは、モデルを記述する一連の方程式を計算するために必要になる時間も当然長くなります。スイッチング電源の場合、モデル内のスイッチの数が増加すると、潜在的なスイッチングの組み合わせの総数が指数関数的に増加します。スイッチングの組み合わせが増えると、コンバータの物理モデルを記述するコードも大きくなります。通常、コードサイズが大きくなるにつれて、モデルをリアルタイムで実行するのにかかる時間も長くなります。

ある時点で、モデルの実行時間が必要な離散化時間ステップを超える場合があります。あるいは、生成したコードのサイズがリアルタイムハードウェアのストレージ容量を超える可能性があります。これらの制限に対処するために使用する一般的な方法は3つあります:

- モデルの分割
- スイッチの削減手法とサブサイクル平均化モデル
- 数値の最適化

これらの手法については、次のセクション以降で詳しく説明します。

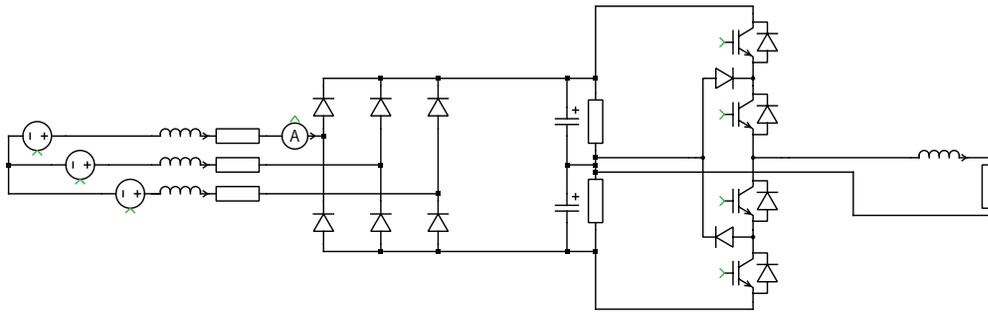
### 2.1 モデルの分割

大規模なシステムの場合、重要な最適化手法は、モデルを数値的に2つ以上の状態空間方程式のセットに分割することです。電気ドメインでは、分割は一般にキャパシタやインダクタで行います。可変電流源をモデルの片側に配置し、可変電圧源をもう一方に配置します。両方の電源は、モデルの他の部分におけるそれぞれの測定値によって制御します。これにより、モデルの各部分における状態空間行列が数値的に分離されます。モデルを分割すると、スイッチングの組み合わせの総数が削減され、状態空間行列のサイズが縮小されます。

例として、[図1](#)に示すような、ダイオード(理想モデル)整流器と3レベルハーフブリッジ(NPC)インバータを備えた回路を考えます。PLECSは、各スイッチングの組み合わせの回路を記述する一意の状態空間行列を生成します。整流回路には、6つのダイオードそれぞれが導通または非導通できるため、 $2^6$ 通りのスイッチングの組み合わせが可能です。同様に、各IGBTダイオードペアを単一のスイッチと見なすと、インバータ回路には $2^6$ 通りのスイッチングの組み合わせが考えられます。

インバータと整流器がDCバスを介して直接結合されているため、各スイッチは回路全体の電流と電圧に直接影響を及ぼします。状態は6つ(インダクタ4個とキャパシタ2個)あるため、スイッチングの組み合わせごとに $6 \times 6$ の行列が必要です。これにより、合計 $2^{12}$ または4096通りのスイッチングの組み合わせと、4096通りの異なる $6 \times 6$ の行列が生成されます。

図1: モデルを分割していない整流器とインバータ回路

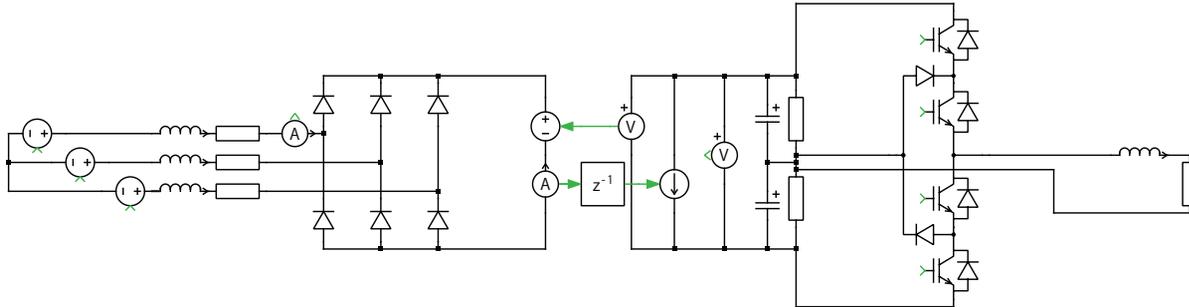


Rectifier:  $2^6$  combinations      VSI:  $2^6$  combinations

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 4096 \text{ switching combinations}$$

図2は同じ回路を示していますが、可変電圧源と可変電流源を介してDCバスで分割しています。可変電流源と可変電圧源の使用により、整流回路とインバータ回路の状態空間行列が電氣的に分離されます。ここで、整流器のスイッチング状態はインバータのスイッチング状態に直接結合せず、逆もまた同様です。切り替えの組み合わせの総数は $2^7$ または128に減少します。さらに、各スイッチングの組み合わせの状態空間行列のサイズは、 $6 \times 6$ の行列から2つの $3 \times 3$ の行列に減少します。つまり、行列サイズと関連する計算が2分の1に削減されます。

図2: 整流器とインバータ回路をジャイレータで分割



Rectifier:  $2^6$  combinations      VSI:  $2^6$  combinations

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = 128 \text{ switching combinations}$$

## 2.2 スイッチの削減手法とハイブリッドモデル

以前のチュートリアルで説明したハイブリッド電源モジュールは、デューティ比の有効範囲を拡大するだけでなく、多くのトポロジのスイッチングの組み合わせの数を大幅に削減することもできます。

たとえば、3-レベルハーフブリッジ(NPC)の従来の実装を図3に示します。3-レベルハーフブリッジ(NPC)は、各ダイオード内蔵IGBTのペアを1つのスイッチと見なした、6つのスイッチで構成されます。しかし、図4に示すこの回路のハイブリッド電源モジュールの実装では、実質的に2つのダイオードしか必要ありません[2]。

この概念を図5の3-レベルハーフブリッジ(NPC)インバータなどの多相回路に適用すると、スイッチの数とスイッチングの組み合わせが大幅に削減されます。従来のスイッチモデルを使用すると、3-レベルハーフブリッジ(NPC)には合計 $2^{18}$ 通りのスイッチングの組み合わせが存在します。ハイブリッドスイッチモデルを使用すると、同じ回路に対して $2^6$ 種類のスイッチングの組み合わせのみになります。モデル全体の複雑さが大幅に減少します。

図3: 従来のNPCの電気構成

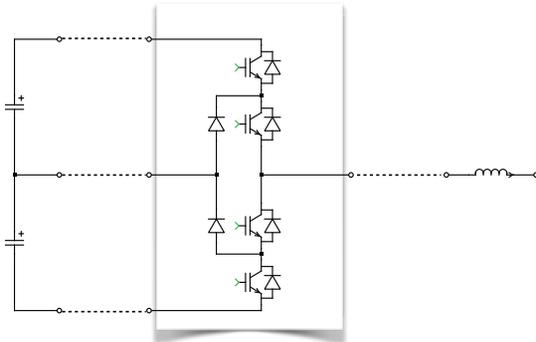


図4: ハイブリッド電源モジュールNPCの電気構成

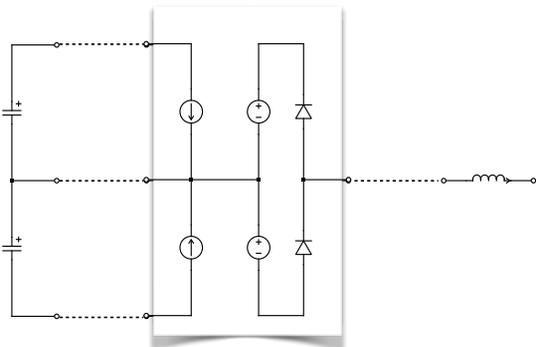
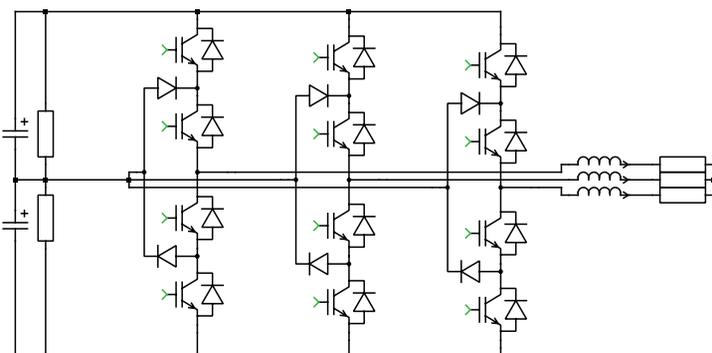


図5: ハイブリッド電源モジュールNPCの電気構成



## 2.3 数値の最適化

他の計算タスクよりも多くの計算時間を必要とする計算タスクがいくつかあります。一般的に、非線形計算、例えば、いくつかの三角関数などは特に時間がかかります。

リアルタイムシミュレーションモデルを開発する際には、計算負荷の高いタスクの総数を念頭に置き、不要な計算を削除し、これらのタスクのより効率的な数値実装を検討することが重要です。

コード生成ステップでは、一部の関数は自動的に個別の最適化を使用することに注意してください。たとえば、正弦波信号は、`math.h`ライブラリの`sin(x)`への関数呼び出しを行わず、より効率的な離散正弦波計算手法を使用します。

## 2.4 モデル設定ブロック

モデル設定ブロックを使用すると、ブロックの端子に接続されている特定の物理システム(電気、熱、回転、並進)のコード生成に影響するパラメータ設定を構成できます。モデル設定ブロックは、端子によって接続されている物理回路に影響します。物理モデルは、基礎となるモデル方程式が完全に分離されている場合、複数の状態空間システムに分割できます。PLECSが状態空間モデルをより小さな独立したモデルに分割できるようにするには、**シミュレーションメニュー -> シミュレーション・パラメータ...**からシミュレーションパラメータウィンドウを表示します。**オプション**タブをクリックし、**状態空間分割の有効化**パラメータを有効にします。同じタブで、**状態空間分割の表示**パラメータも有効にします。この設定では、PLECSは、分割後の個々の状態空間モデルを構成するコンポーネントを強調表示する診断メッセージを発行します。個々の状態空間システムごとに、最大1つのモデル設定ブロックを接続できます。

### スイッチングアルゴリズム

物理モデルのコードを生成する際に、PLECSでは、**反復**と**直接ルックアップ**の2つのアルゴリズムを選択できます。独立した状態空間モデルごとにアルゴリズムを個別に指定できます。デフォルトのスイッチングアルゴリズムは**直接ルックアップ**です。詳細については、PLECSユーザマニュアルの**物理システムのコード生成**の項を参照してください。

### 行列コード化スタイル

この設定では、物理システムの状態空間行列を保存するために使用する形式を指定できます。スパース行列に設定すると、ゼロ以外の行列エントリとその行と列のインデックスのみが保存されます。完全行列に設定すると、行列は完全な $m \times n$ 配列として保存されます。完全行列(インライン)に設定すると、行列はヘルパ関数に埋め込まれ、これによりコンパイラはコードサイズが大きくなる代わりに、行列ベクトル乗算をさらに最適化できる可能性があります。

## 3 誘導機駆動の最適化

この演習では、誘導機駆動の一連の最適化手順を検討します。最初は、スイッチの種類を変更して、スイッチの組み合わせの数を減らします。次に、DCリンクでモデルを電氣的に分割して、スイッチングの組み合わせの数をさらに減らします。両方の最適化手順に従って生成したコードの実行時間とサイズを比較します。



あなたのタスク:

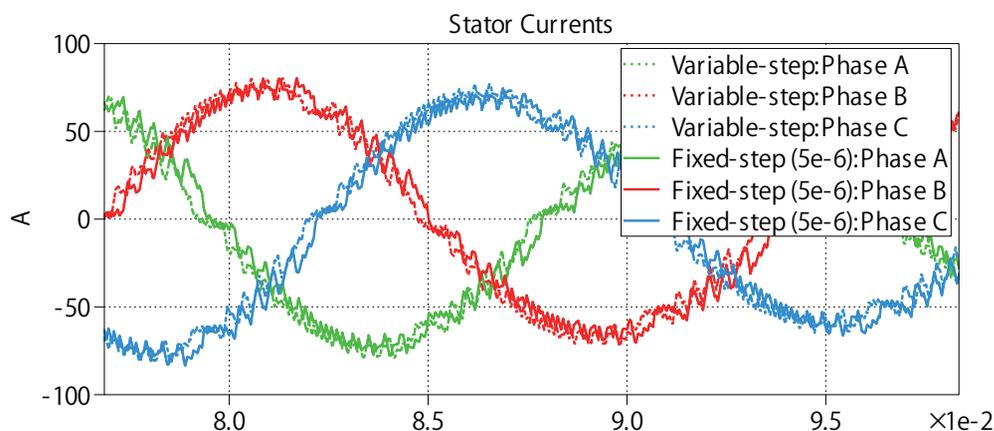
- 添付のdtc\_start.plecsファイルを開き、"Circuit"および"Direct Torque Control"サブシステムを含むモデルを開きます。モデルを確認した後、シミュレーション -> シミュレーション・パラメータドロップダウンメニューから可変ステップソルバを使用してシミュレーションを実行し、すべてのスコープトレースを保持します。
- 固定ステップソルバに変更し、シミュレーションタイムステップを選択します。結果を可変ステップソリューションと比較します。

**?** 妥当なタイムステップはいくつでしょうか？

**A** 通常、タイムステップはスイッチング周波数よりも少なくとも1桁小さくなります。ただし、このモデルで使用しているヒステリシスコントローラでは、スイッチング周波数は可変です。5e-6秒の固定ステップソリューションが妥当です。ステップサイズをさらに小さくするとシミュレーションの精度は向上しますが、さらなる計算オーバーヘッドが発生します。

図6は、5e-6秒の固定ステップと可変ステップの結果を示しています。固定ステップソルバとヒステリシスコントローラの時間分解能には制限があるため、2つの数値手法の間には常に何らかの誤差が生じます。

図6: 可変ステップソリューションと固定ステップソリューションの比較



- 選択した離散時間ステップを使用して、変更を加えずに"DTC"サブシステムのコードを生成します。プロセスに時間がかかりすぎると思われる場合は、コード生成をキャンセルできます。

**?** このシステムのコード生成に問題が生じるのはなぜですか？

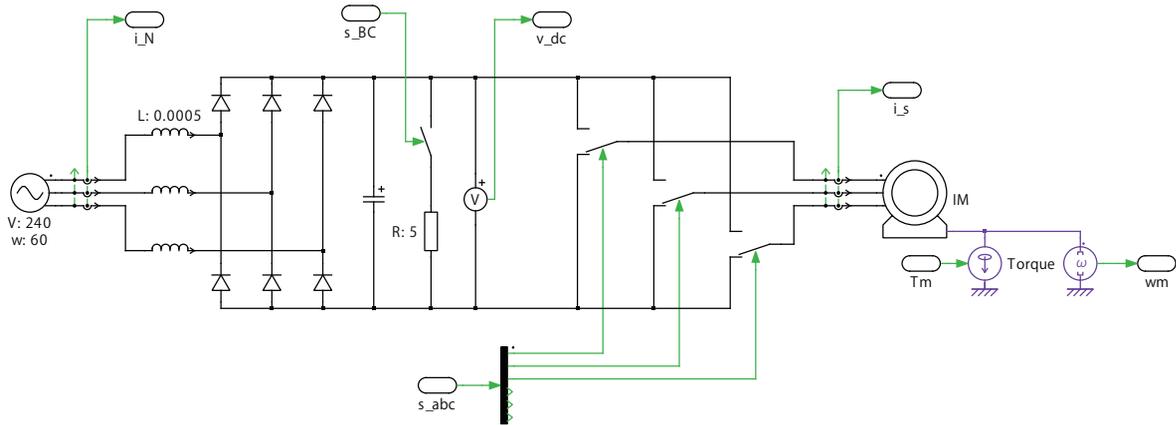
**A** 整流器入力に6個のダイオードがあり、モータードライブに6個のIGBTがある場合、Coderはデフォルトであらゆるスイッチングの組み合わせが可能であると想定しますが、 $2^{12}$ または4096通りのスイッチングの組み合わせが存在します。スイッチングの組み合わせ数が多いと、コードサイズとモデルの複雑さが増加します。

- 次に、PLECSライブラリのスイッチ(ダブル)コンポーネントを使用してダイオード内蔵IGBTを置き換え、インバータシステム内のスイッチングの組み合わせ数を削減します。図7を参照してください。

**?** スイッチ(ダブル)コンポーネントの使用上の制限は何ですか？

**A** スイッチ(ダブル)コンポーネントには、インバータのブランキング状態を表すダイオードが組み込まれていません。スイッチ(ダブル)コンポーネントはPWM Captureブロックと互換性がなく、ショートスルー状態を検出できません。ハイブリッド電源モジュールはこれらの制限を克服します。

図7: スイッチ(ダブル)に置き換えた誘導モータドライブ



5 コードを生成し、RT Boxでモデルを実行します。

**?** 実行時間はどれくらいですか？

**A** 実行時間は  $5.5\mu\text{s}$  近くなります。これは  $5\mu\text{s}$  の離散化ステップサイズを超えています。現在のサイクル時間は、実際には複数のシミュレーションタイムステップにわたるピークサイクル時間であることに注意してください。シミュレーションが中止される前に発生する可能性のある連続オーバーランの最大数は、**Coderオプション...** -> **ターゲットタブ**で定義します。

6 ファイルブラウザで、PLECSモデルを保存した場所に移動します。\_codegenフォルダを開き、DTC.cファイルを探します。このファイルには、"DTC"サブシステム用に生成されたコードが含まれています。

**?** DTC.cのファイルサイズは何メガバイト(MB)ですか？

**A** ファイルは15.2MBです。

この段階で、モデルはdtc\_1.plecsと同じになるはずですが。



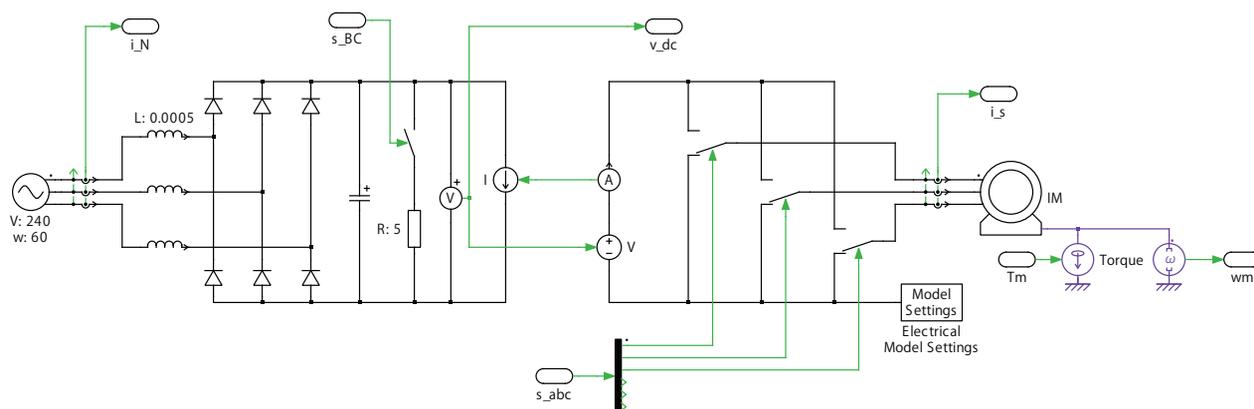
**あなたのタスク:**

1 図8に示すように、モデルを電氣的に分離した2つの部分に分割します。

**?** なぜここがモデルを電氣的に分割するのに適した場所なのでしょう？

**A** DCリンク容量が大きいため、キャパシタ電圧の変化はシミュレーションの時間ステップに比べて遅くなります；つまり、電圧は時間ステップごとに大きく変化しません。したがって、可変電源によって導入される遅延が電圧ダイナミクスに与える影響は無視できるため、キャパシタを可変電圧および可変電流源としてモデリングすることが適切です。

図8: スイッチ(ダブル)とDCバス分割を備えた誘導モータドライブ



**注意:** 図8の制御された可変電流源と可変電圧源の**離散化の挙動**は、ゼロ次ホールドに設定する必要があることに注意してください。この構成では、現在の状態変数を計算するために、前のシミュレーションステップの入力値のみが必要です。

2 コードを生成し、モデルを RT Boxにデプロイします。



モデルの実行時間はどれくらいですか? ステップ5の結果と比較するとどうでしょうか?



実行時間はステップ5の  $5.5\mu\text{s}$  から  $3.5\mu\text{s}$  に短縮されます。



インバータのブランキング状態はどのように含めることができますか?



スイッチ(ダブル)をPLECSライブラリの電気回路ブロック -> パワー素子モジュールにあるハーフブリッジコンポーネントに置き換えます。

3 DTC.cファイルのサイズを検査します。



DTC.cのファイルサイズはいくつですか? ステップ6の結果と比べてどうですか?



DTC.cファイルのサイズは0.660MBになりました。これはステップ6の結果の20分の1です。



この段階で、モデルはdtc\_2.plecsと同じになるはずですが。

4 添付しているdtc\_1.plecsおよびdtc\_2.plecsモデルを開きます。シミュレーション -> シミュレーションパラメータ... を選択してシミュレーションパラメータウィンドウを開きます。オプションタブで、**状態空間分割の有効化**オプションがチェックされていることを確認します。この設定では、この設定では、PLECS は、分割後の個々の状態空間モデルを構成するコンポーネントを強調表示する診断メッセージを発行します。

モデルを実行し、シミュレーション ウィンドウの右下隅にある感嘆符  アイコンをクリックして、PLECS診断ウィンドウを開きます。

 PLECS診断ウィンドウに示されているdtc\_1.plecsとdtc\_2.plecsの状態空間モデルの合計数はいくつですか？  
これは、前のステップに基づいて期待どおりですか？

 3つの状態空間システムを観察します。1つ目の状態空間システムは整流回路に関連付けられ、2つ目はモータと駆動システムを表し、3番目はモータシャフトを含む機械ドメインを表します。これにより、電気ネットワークが、可変電圧源と電流源を通じて数値的に結合された2つの小さな状態空間システムに正常に分割されたことが確認されます。

5  8に示すように、電気モデル設定ブロックを分割回路の右側の駆動システムに接続します。**スイッチングアルゴリズム**パラメータを直接ルックアップに設定し、**行列コード化スタイル**を完全行列(インライン)に設定します。コードを生成し、モデルをRT Boxにデプロイします。

 モデルの実行時間はどれくらいですか？ ステップ2の結果と比較するとどうでしょうか？

 実行時間はステップ2の3.5 $\mu$ sから3.1 $\mu$ sに短縮されます。

 この段階で、モデルはdtc\_3.plecsと同じになるはずですが。

## 4 まとめ

このチュートリアルでは、リアルタイムモデルのさまざまな最適化手法を学習し、実際の回路に適用しました。最適化の各ステップで実行時間とコードサイズを評価しました。PLECSモデル内のスイッチングの組み合わせの数を減らすことで、リアルタイムパフォーマンスがどのように向上するかを直接確認できました。

これらのトピックを理解することで、独自のリアルタイム モデルを最適化できるようになります。

## 5 参考文献

[1] Jost Allmeling, Niklaus Felderer “Sub-cycle average models with integrated diodes for real-time simulation of power converters”, 2017 IEEE Southern Power Electronics Conference (SPEC), pp. 1-6, 2017.

[2] Jost Allmeling, Niklaus Felderer, Min Luo, “High Fidelity Real-Time Simulation of Multi-Level Converters”, Power Electronics Conference (IPEC-Niigata 2018 –ECCE Asia) 2018 International, pp. 2199-2203, 2018.

改訂履歴:

Tutorial Version 1.0 初版  
Tutorial Version 1.1 モデル設定ブロックの説明を追加

**plexim**

☎ +41 44 533 51 00

+41 44 533 51 01

✉ Plexim GmbH

Technoparkstrasse 1

8005 Zurich

Switzerland

@ info@plexim.com

<http://www.plexim.com>

**Pleximへの連絡方法:**

Phone

Fax

Mail

Email

Web

**KESCO** KEISOKU ENGINEERING SYSTEM

計測エンジニアリングシステム株式会社

<https://kesco.co.jp>

*RT Box Tutorial*

© 2002–2022 by Plexim GmbH

このマニュアルで記載されているソフトウェアPLECSは、ライセンス契約に基づいて提供されています。ソフトウェアは、ライセンス契約の条件の下でのみ使用またはコピーできます。Plexim GmbHの事前の書面による同意なしに、このマニュアルのいかなる部分も、いかなる形式でもコピーまたは複製することはできません。

PLECSはPlexim GmbHの登録商標です。MATLAB、Simulink、およびSimulink Coderは、The MathWorks、Inc.の登録商標です。その他の製品名またはブランド名は、それぞれの所有者の商標または登録商標です。