



PLECS *Tutorial*

Introduction to the State Machine Block

PLECS ステートマシンブロックの使用

Implementation of an interleaved triangular current mode (TCM) control scheme for a single-phase PFC rectifier
インタリーブ電流三角波モード(TCM)単相PFC整流器の制御を実装

Tutorial Version 1.0

1 はじめに

ステートマシンは、システムの動作を制御、モデル化、予測するために使用できる強力なツールです。これは、時間の経過とともに進化するプロセスを表現するのに便利な方法でもあります。また、時間の経過とともに進化するプロセスを表現するのに便利な方法でもあります。ステートマシンを使用する代わりに、特定のタスク処理関数をコードの適切なループ下に分散させるという方法もありますが、これは非効率的で、混乱を招く可能性が高くなります。

"ステートマシン"ブロックは、PLECSユーザに入力信号に対するシステムの反応をグラフィカルにモデル化する方法を提供します。ステートマシンの実行中に実行される動作(開始、経過、終了など)は、Cプログラミング言語を使用して実装できます。PLECSは、サブステートがスーパーステート内にネストされた階層型ステートマシンもサポートします。互いのエッジが交差する状態は許可されず、シミュレーションの開始時にエラーが発生することに注意してください。

この演習では、次のことを学習します:

- PLECSでシンプルなステートマシンをモデル化し、シミュレーションに組み込む方法
- ステートマシンブロックを使用して変調器を実装する方法

始める前に 演習の各段階で独自のモデルと比較できるように、`finite_state_machine_start.plecs`、`finite_state_machine_1.plecs`、`finite_state_machine_2.plecs`ファイルが作業ディレクトリに配置されていることを確認してください。



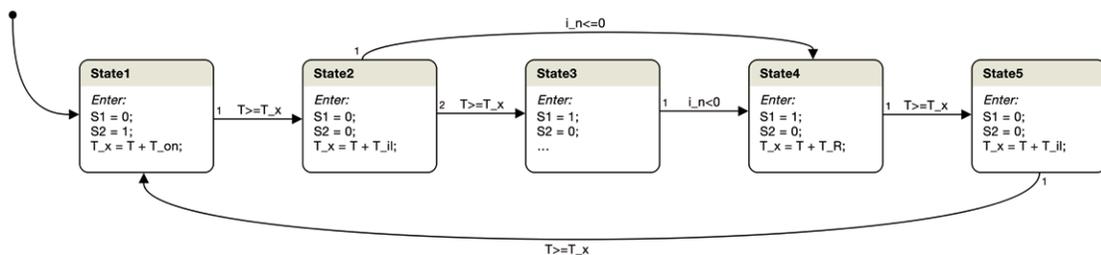
注意: このモデルには、以下からアクセスできるモデル初期化コマンドが含まれています:

PLECS Standalone: 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... -> 初期化タブ

PLECS Blockset: Simulink model windowで右クリック -> Model Properties -> Callbacks -> InitFcn*

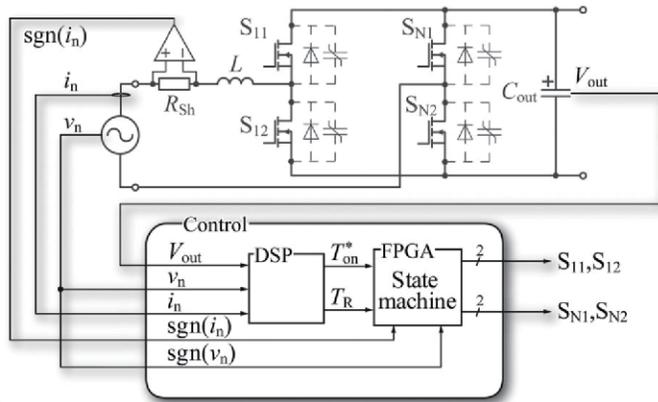
2 変調器ステートマシン

図1: 変調器ステートマシンモデル



このチュートリアルでは、上の図1に示すような単純な変調器モデルを構築します。このステートマシンは、元々、図2に示すように、超平坦インタリーブ電流三角波モード(TCM)単相PFC整流器でスイッチの入力を制御するために使用されていました。

図2: TCM単相PFC整流器を制御するために使用するステートマシン(図は参照[1]のFig. 8.(a)から引用)



あなたのタスク:

- 1 最初にfinite_state_machine_start.plecsファイルを開きます。これがこの演習の開始点になります。
"ライブラリブラウザ"から"ステートマシン"コンポーネントを回路図にドラッグします。これを行うには、ライブラリの"制御器ブロック"にある"ステートマシン"セクションに移動するか、検索ボックスを使用します。
- 2 "ステートマシン"ブロックをダブルクリックすると**エディタウィンドウ**が開き、ステートとステート間の遷移をグラフィカルにモデル化できます。このウィンドウ内でモデル化するものはすべて、Cコードファイルに変換され、"C言語入力"と同様に、シミュレーション中にコンパイルおよび実行されます。

2.1 パラメータの指定

まず、**エディタウィンドウ**の上部にある"ステートマシン設定"  ボタンをクリックして、このモデルのパラメータを指定します。ここで、入力、出力、内容、変数、サンプル時間、その他の設定を指定できます。これらは、ステートマシン内の状態と遷移のトリガ、条件、動作を定義するCコードで直接使用できます。"ステートマシン"ブロックは、それに応じて入力端子と出力端子を自動的に追加し、ラベルを付けます。

アニメーションモードを有効にするには、**アニメーションラベル**の横にあるチェックボックスをオンにします。有効な場合、ステートマシンはアクティブ状態と遷移を強調表示して、シミュレーション中の実行フローを表示します。アクティブ状態が変化するたびにシミュレーションが一時停止されるため、実行エラーを見つけるのに役立ちます。

最後に、このモデルはCコードでマクロCurrentTimeを使用します。この値には現在のシミュレーション時間が含まれており、遅延遷移の実装に役立ちます。



あなたのタスク: + ボタンと - ボタンを使用して、以下の**表1**にリストされている変数、内容、およびC言語宣言文を適切なパラメータタブに追加します。

表1: ステートマシンのパラメータ宣言

入力	入力変数	種類	説明
	i_n	連続	インダクタ電流
	T_on	連続	S2 on-time
	T_R	連続	逆導通時間
出力	出力変数		説明
	S1	-	スイッチ1の制御信号
	S2	-	スイッチ1の制御信号
内容	内部定数	値	説明
	T_il	400e-9	インターロック遅延定数
変数	内部変数	初期値	説明
	T_x	0	現在の時間位置
C言語宣言文	#define T CurrentTime		現在の時刻を含むマクロ



この段階では、モデルは参照モデルfinite_state_machine_1.plecsと同じになるはずですが。

2.2 ステート(状態)の追加と編集

新しいステートを追加するには、ツール ボタンをクリックし、ステートを配置するエディタ領域をクリックします。ステートは角丸のボックスで表されます。ステートの名前は、灰色の背景を持つタイトルバーのボックスの上に表示されます。名前を変更するにはダブルクリックします。ステートの四角形は、マウスで角をドラッグすることでサイズを変更できます。ステートをダブルクリックしてパラメータを変更します。"状態(パラメータ)"ダイアログには、ステートが実行できるアクションの種類ごとにタブが含まれています。**開始動作**は、ステートマシンの実行フローがステートに入るたびに実行します。**経過動作**は、実行フローがステートにある間、各タイムステップで実行します。**終了動作**は、実行フローがステートを離れるたびに実行されます。すべての動作はCコードを使用して実装します。

PLECSは階層ステートマシンをサポートします。ステートを他のステートの長方形の境界内にドラッグして階層を確立できます。別のステートによって(境界内に)包含されるステートは、"サブステート"と呼ばれます。包含状態は"スーパーステート"と呼ばれます。互いのエッジと交差する重複は許可されず、シミュレーションの開始時にエラーが発生します。



あなたのタスク:

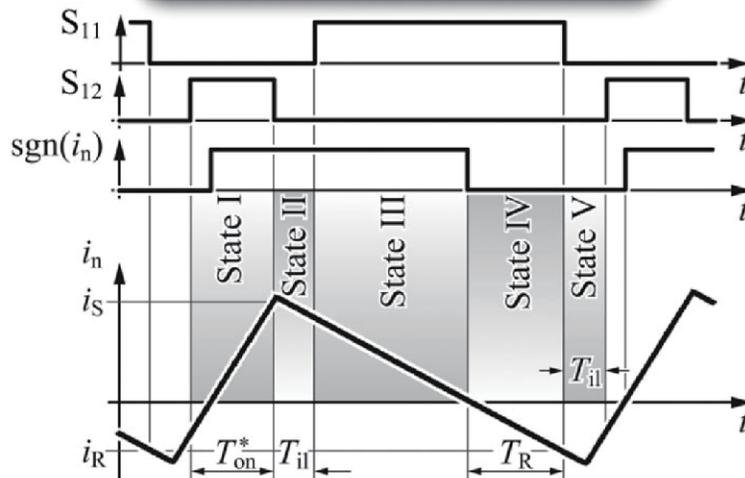
- 1 に示すように、左から右に5つのステートを追加し、"State1"から"State5"までの名前を付けます。
- 2 これらの5つのステートの**開始動作**内にに示すパラメータを入力します。

表2: "開始動作内"ステートのパラメータ宣言

State1	State2	State3	State4	State5
S1 = 0;	S1 = 0;	S1 = 1;	S1 = 1;	S1 = 0;
S2 = 1;	S2 = 0;	S2 = 0;	S2 = 0;	S2 = 0;
T_x = T + T_on;	T_x = T + T_il;	-	T_x = T + T_R;	T_x = T + T_il;

これらのパラメータ セットは、に示す遷移に対応します。

図3: ステートマシンの遷移(図は参照[1]のFig. 8.(b)から引用)



2.3 遷移の追加と編集

5つのステートができたので、まずデフォルト遷移を作成してシミュレーションの開始ステートを指定する必要があります。



あなたのタスク: 新規デフォルト遷移を追加するにはツールボタン(🔧)をクリックし、この遷移を開始するエディタ領域(State1の外側の回路図領域内)をクリックしてから、初期状態としてマークするステートのエッジ(この場合はState1のエッジ)をクリックします。この遷移矢印はさまざまな方法で移動できます。

デフォルト遷移をダブルクリックすると、パラメータダイアログが開き、実行フローがこの遷移を実行するたびに実行される一連のアクションを指定できます。他のすべてのステートおよび遷移パラメータと同様に、これはCコード式を使用して実装できます。この場合、この"動作"フィールドは空白のままにしておくことができます。



注意: このチュートリアルでは、複数の遷移を結合または分岐する分岐点として機能する円形の"ジャンクション"コンポーネント(Q)は使用しません。ジャンクションは、シミュレーションの開始ステートが常に同じではない場合に便利です。

ステート間の遷移を追加するには、ステートの端をクリックし(十字カーソルが表示されたら)、マウスを目的のステートの端までドラッグします(二重十字カーソルが表示されたらマウスボタンを放します)。遷移の終了点は、後でマウスをドラッグすることで変更できます(移動する終了点の近くに手のひらアイコンのカーソルが表示されます)。ハンドルをドラッグすることで、遷移の形状を変更できます。マウスを遷移の上に移動すると、利用可能なハンドルが灰色のドットとして表示されます。



あなたのタスク: 図1と表3に示すように遷移を追加します。

表3: ステートの相互接続

State1 -> State2	State4 -> State5
State2 -> State3	State5 -> State1
State3 -> State4	State2 -> State4

遷移をダブルクリックすると、遷移パラメータダイアログが開きます。**優先度**パラメータは1から始まり、遷移のソースステートから出力遷移の数までの値です。複数の送信遷移のトリガと条件(以下を参照)がtrueと評価されると、実行フローは優先度パラメータが最も小さい遷移を実行します。

"トリガ"と"条件"は各タイムステップで評価されます。遷移のトリガと条件の両方がtrueと評価された場合、遷移が実行されます。PLECSは、明示的なトリガ(外部入力信号)、タイムベーストリガ(遷移のソース状態に入ってから正確な遅延秒後に実行)、暗黙的なトリガ(式がtrueになったときにアクティブになる関係式)を区別します。一方、条件は、遷移を実行する必要があるかどうかを確認するために単純に評価されるブール式です。暗黙的なトリガ $x > 0$ と条件 $[x > 0]$ の基本的な違いに注意してください。前者は x 値がゼロより大きくなった瞬間にのみ評価されますが、後者は x がゼロより大きくなるたびに試行されます。トリガと条件は両方ともCコード式として入力されます(最後にセミコロンは付加しません)。

遷移の"動作"は遷移が行われるたびに、ステート終了アクションの後、ステート開始アクションの前に実行されます。このアクションはCコードを使用して実装できます。PLECSは、内部遷移、つまりソースステートのエッジからステートの内部に向かって移動し、サブステートのエッジで終了する遷移をサポートしています(階層状態の詳細については上記を参照してください)。ステートAからサブステートB(またはステートBからスーパーステートA)への内部遷移が行われると、ステートAの終了アクションと開始アクションは実行されません。



あなたのタスク:

- 1 表4に示す式を、ステート間のそれぞれの遷移のトリガとして指定します。すべての遷移の動作フィールドを空白のままにしておくこともできます。

表4: ステート間の遷移のトリガ

遷移	トリガ
State1 -> State2	$T \geq T_x$
State2 -> State3	$T \geq T_x$
State3 -> State4	$i_n < 0$
State4 -> State5	$T \geq T_x$
State5 -> State1	$T \geq T_x$
State2 -> State4	$i_n < 0$

- 2 コンバータモデルを含む回路図ウィンドウに戻り、ステートマシンブロックのすべての入力(i_n , T_{on} , および T_R)と出力($S1$ および $S2$)がCtrlブロックとGateブロックのそれぞれのポートに接続されていることを確認します。
- 3 ステートマシンをダブルクリックして回路図エディタに戻り、モデルをシミュレートします。"ステートマシン設定"で**アニメーション**チェックボックスをオンにした場合、現在のシミュレーション状態が点灯します。スペースバーを使用して状態を切り替え、ステートマシンとスコープの出力を確認します(ズームインする必要がある場合があります)。また、Ctrl + T(Macの場合はCommand + T)を押してシミュレーションを終了し、**アニメーション**チェックボックスをオフにしてシミュレーションを再実行し、最終結果を確認することもできます。



この段階では、モデルは参照モデルfinite_state_machine_2.plecsと同じになるはずですが。

3 まとめ

この演習では、PLECSを使用してステートマシンを作成するための手順を段階的に説明しました。ステートとステート間の遷移を作成する方法、およびステートのパラメータを操作する方法を学習しました。また、ステートマシンブロックをより大きなPLECSシミュレーションに結び付ける方法も学習しました。グラフィカルなエディタにより、複雑な制御アルゴリズムのモデリングが非常に直感的になり、コーナーケースの特定にも役立ちます。詳細については、ステートマシン ブロック内の ⓘ ボタンをクリックするか、弊社WebサイトにあるPLECSユーザマニュアルの該当するセクションを参照してください。

4 参考文献

- [1] Christoph Marxgut, Florian Krismer, Dominik Bortis, Johann W. Kolar, “Ultraflat Interleaved Triangular Current Mode (TCM) Single-Phase PFC Rectifier”, *IEEE Transactions on Power Electronics*, Vol. 29, No. 2, pp. 873-882, Feb. 2014.

改訂履歴:

Tutorial Version 1.0 初版

plexim

Pleximへの連絡方法:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	info@plexim.com	Email
	http://www.plexim.com	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>

PLECS 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。