

# PLECS *Tutorial*

## Introduction to PLECS Blockset

2つの簡単な電気回路を構築してPLECS Blocksetの学習を開始

Tutorial Version 1.0

*Advancing Automation*  
**AUTO  
ADVANCEMENT**

翻訳:

アドバン オートメーション株式会社

plecs\_adva@adv-auto.co.jp www.adv-auto.co.jp

# 1 はじめに

新しいプログラムに慣れる唯一の方法は、実際に使ってみることです。このチュートリアルでは、2つのサンプル回路を段階的にモデル化し、PLECSの基本機能を説明します。この演習では、次の内容を学習します:

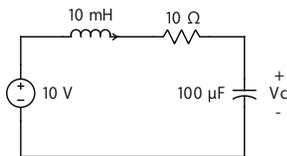
- PLECS Library browserのコンポーネントを使用して簡単なモデルを作成する方法
- PLECS回路図でコンポーネントを相互接続する方法
- コンポーネントをパラメータ化する方法
- サブシステムを作成する方法
- コントロールを使用してマルチドメインモデルを作成する方法
- Simulinkでコントロールを使用してマルチドメインモデルを構築する方法
- プローブブロックを使用する方法

**始める前に** 始める前に、PLECS Blockset(Matlab R2007a以降)がインストールされていることを確認し、演習の各段階で独自のモデルと比較できる参照ファイルがあることを確認してください。

## 2 シンプルなパッシブネットワークの構築

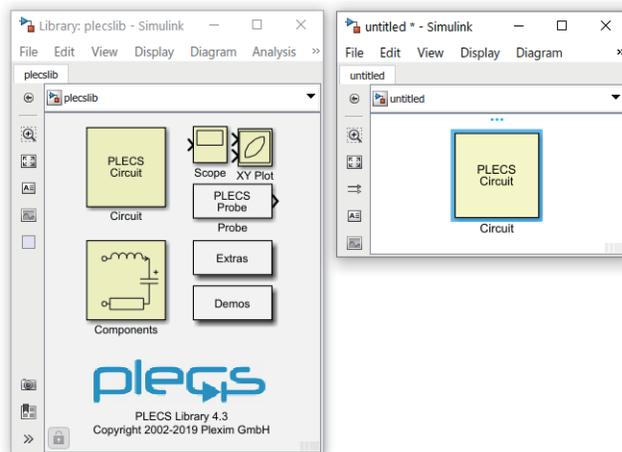
最初にモデル化する電気システムは、[図1](#)に示すような単純なRLCネットワークです。キャパシタはRL分岐を介してDC電圧源によって充電され、その電圧は電圧計で監視されます。

図1: 単純なRLCネットワーク



PLECS Blockset にアクセスするには、MATLABコマンドラインにplecslibと入力します。これにより、"Circuit"という名前の汎用PLECSブロックとさまざまなコンポーネントライブラリを含むSimulinkモデルが表示されます。このドキュメントでは、これをplecslibツールボックスと呼びます。このツールボックスには、[図2](#)に示すように、独自の回路を作成できる電気部品が含まれています。あるいは、Simulink Library browserで PLECS ツールボックスを開いてアクセスすることもできます。

図2: SimulinkのplecslibツールボックスとPLECS Circuit





### あなたのタスク:

- 1 MATLABコマンドラインにplecslibと入力します。
- 2 新しいSimulinkモデルを開きます。plecslibツールボックスからPLECS "Circuit"ブロックをドラッグアンドドロップします。
- 3 "Circuit"ブロックをダブルクリックして、PLECS回路図ウィンドウを開きます。

黄色い表示の回路図ウィンドウはモデリングインタフェースです。PLECS回路に必要なコンポーネントは、PLECSの"Components"ライブラリからこのウィンドウにコピーする必要があります。使用可能なライブラリを参照して、どのコンポーネントが使用可能かを確認できます。Simulinkと同様に、これはライブラリからドラッグして回路図にドロップすることによって行われます。PLECS回路図ウィンドウの**ウィンドウメニュー**から、または **Ctrl + L**を押してPLECS Library browserを表示することもできます (Macユーザの場合、以降のすべてのショートカットでは**Ctrl**ではなく**Cmd**を使用します)。

## コンポーネント

回路の構築に必要なコンポーネントは、PLECS Library Browserからモデルウィンドウにドラッグアンドドロップする必要があります。モデルウィンドウにすでに存在するコンポーネントを複製する場合は、**Ctrl**キーを押したまま、コンポーネントを新しい場所にドラッグアンドドロップするか、マウスの右ボタンを使用します (もちろん、通常どおりコンポーネントを選択して**コピーアンドペースト**することもできます)。



**注意:** 両方のプログラムが同じグラフィカルユーザインタフェースを共有していないため、SimulinkコンポーネントをPLECS回路図ウィンドウ内に配置することはできず、PLECSライブラリのコンポーネントをSimulinkレベルに配置することもできません。PLECS Scope, XY Plot, およびProbeブロックの2番目のバージョンは、[図2](#)に示すように、Simulink環境内で使用するために提供されています。

RLCネットワークを構築するために必要な電気コンポーネントは、ElectricalドメインライブラリのSources、Meters、Passive Componentsにあります。Scopeコンポーネントは、Systemライブラリにあります。ドメイン名の横にある矢印をクリックするか、コンポーネントを参照する代わりに、検索バーに必要なコンポーネントの最初の文字を入力して検索することもできます。たとえば、scと入力するとスコープが表示され、resと入力すると使用可能な抵抗器などがすべて表示されます。

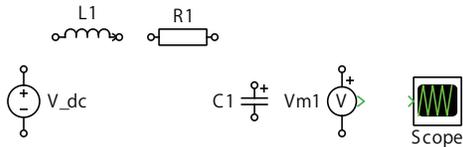
## 信号

PLECSは、電気部品を接続するために使用する電気接続(黒線)に加えて、単一方向信号も扱います。信号は緑色で表示し、方向を示す矢印が付いています。これらの信号は、測定値やスイッチのトリガパルスなどの非電気情報を伝送します。信号は計算に使用したり、スコープに表示することができます。電気接続をスコープに直接入力することはできません。必ず最初に電圧計または電流計を使用して電流量を信号に変換する必要があります。

**あなたのタスク:**

- 1 [図3](#)に示すように、ライブラリからVoltage Source DC、Inductor、Resistor、Capacitor、Voltmeter、Scopeの各コンポーネントを回路図ウィンドウにドラッグアンドドロップします。
- 2 必要なコンポーネントは、マウスの左ボタンで移動できます。選択したコンポーネントを回転するには、**Ctrl + R**キーを押します。水平方向に反転するには、**Ctrl + F**キーを押します。垂直方向に反転するには、**Ctrl + I**キーを押します。これらの機能はすべて、**Format**メニューから、またはコンポーネントを右クリックしてアクセスすることもできます。

図3: RLCネットワークのコンポーネント



すべてのコンポーネントを回路図ウィンドウにコピーすると、モデルは参照モデルrlc\_network\_1.mdlと同じになります。

**接続**

現在の回路図の各コンポーネントには、端子を表す小さな円または緑の矢印が付いています。マウスポインタをその端子に近づけると、ポインタの形状が矢印から十字に変わります。ここで、マウスの左ボタンを押したままにすると、接続を別のコンポーネントにドラッグできるようになります。別の端子または同じ接続タイプの既存の接続に近づくと、ポインタの形状が二重十字に変わります。マウスボタンを放すとすぐに、新しい接続が作成されます。

既存の接続から分岐接続を作成するには、マウスカーソルをその上に置き、**Ctrl**キーを押しながら、分岐を開始する方向にカーソルをクリックしてドラッグします。

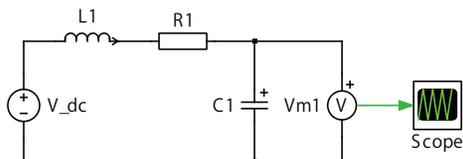


**あなたのタスク:** コンポーネントを相互接続してRLC回路を完成させます。次に、電圧計の出力端子をPLECS スコープの入力端子に接続します。



この段階では、[図4](#)のように、モデルは参照モデルrlc\_network\_2.mdlと同じになるはずですが。

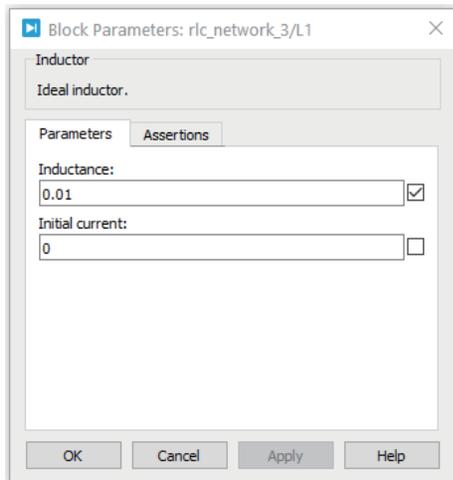
図4: スコープを備えたRLCネットワーク



## コンポーネントのパラメータ

各コンポーネントは、自動的に選択される一意の名前で識別されます。必要に応じて、回路図内のコンポーネントをダブルクリックしてこの名前を変更できます。これはドキュメント化のみを目的としており、シミュレーションには影響しません。一方、パラメータはより重要です。パラメータは、たとえばインダクタのインダクタンス、キャパシタの静電容量、またはDC電圧源の電圧を決定します。コンポーネントのアイコンをダブルクリックすると、[図5](#)に示すように、これらのパラメータを設定できるダイアログウィンドウが開きます。

図5: インダクタのパラメータダイアログウィンドウ



各パラメータフィールドの右側にチェックボックスがあることに注意してください。このチェックボックスをオンにすると、指定したパラメータ値が回路図上に表示されます。これにより、回路図から直接コンポーネントのパラメータを表示および編集できるようになります。回路図に不要なテキストを表示させず、コンポーネントの最も重要なパラメータのみを表示させることを推奨します。

## 単位

Simulinkと同様に、PLECSはカスタム変数として文字列が使用できるため、単位を認識しません。したがって、ユーザは変数のスケールが正しいことを確認する必要があります。パワーエレクトロニクスでは、SI単位系の使用が推奨されます。たとえば、インダクタンス値には10mHではなく、10e-3または0.001を使用します。電力システムでは、「単位あたり」の数量を扱う方が適切である場合があります。



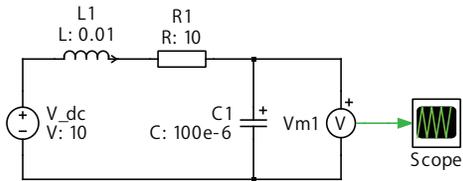
### あなたのタスク:

- 1 RLCネットワークの各コンポーネントをダブルクリックして、パラメータダイアログウィンドウを表示します。次に、**Voltage**を10、**Inductance**を0.01、**Resistance**を10、**Capacitance**を100e-6に設定します。
- 2 これらのパラメータの横にあるチェックボックスをオンにすると、回路図上に表示されます。
- 3 インダクタとキャパシタには、それぞれ**Initial current**や**Initial voltage**などの追加のパラメータがあります。これらの値は両方ともデフォルトのゼロのままにしておきます。



この段階では、[図6](#)のように、モデルは参照モデルrlc\_network\_3.mdlと同じになっているはずです。

図6: パラメータを表示させたRLCネットワーク



## シミュレーション

シミュレーション時間、ソルバのタイプ、ソルバオプション、モデル初期化コマンドなどのパラメータは、**シミュレーションメニュー**の**シミュレーションパラメータ**ダイアログウィンドウにあります。このダイアログは、キーボードショートカット**Ctrl + E**を使用して開くこともできます。



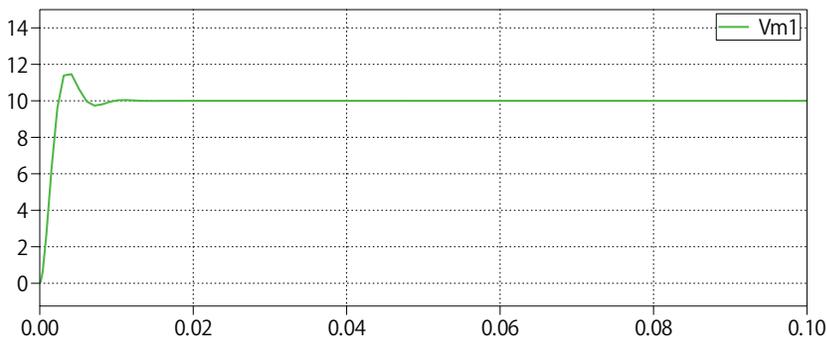
### あなたのタスク:

- 1 **Simulation**メニューから Simulinkの**Model Configuration Parameters**ダイアログ ウィンドウを開き、シミュレーションの**Stop time**を0:1に設定します。
- 2 次に、**Simulation**メニューから**Start**を選択するか、**Ctrl + T**を押してシミュレーションを開始します。



得られるシミュレーション結果は、[図6](#)と同じになるはずです。

図7: RLCネットワークキャパシタ電圧のシミュレーション結果



## 制御ブロックの追加

静的電気モデルを強化するために、動的な動作を追加することができます。パルス電圧を印加すると、この例のキャパシタがどのように充電および放電するかを観察してみましょう。



### あなたのタスク:

- 1 回路図で、Voltage Source DCコンポーネントを削除し、Voltage Source(Controlled)ブロックに置き換えます。このブロックは、制御器ブロックドメインからの任意の信号を入力として受け入れます。
- 2 "パルス発生器"コンポーネントをVoltage Source(Controlled)の左側にドラッグアンドドロップします。**High-state output**を10に、**Frequency**を25に設定します。他のすべてのパラメータはデフォルト値のままにします。
- 3 Pulse Generatorブロックの出力端子を制御電圧源の入力端子に接続します。
- 4 次に、Signal Multiplexerコンポーネントを電圧計ブロックとScopeブロック間にドラッグアンドドロップし、入力数を2に設定します。
- 5 Pulse GeneratorとVoltmeterの出力を、Signal Multiplexerを使用してScopeに接続します。
- 6 シミュレーションを開始します。



この段階では、モデルは図8のように参照モデル`r1c_network_final_plecs.mdl`と同じになっているはずです。得られるシミュレーション結果は、図9と同じになるはずです。

図8: パルス電圧源を備えたRLCネットワーク

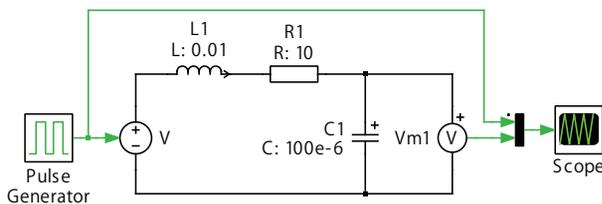
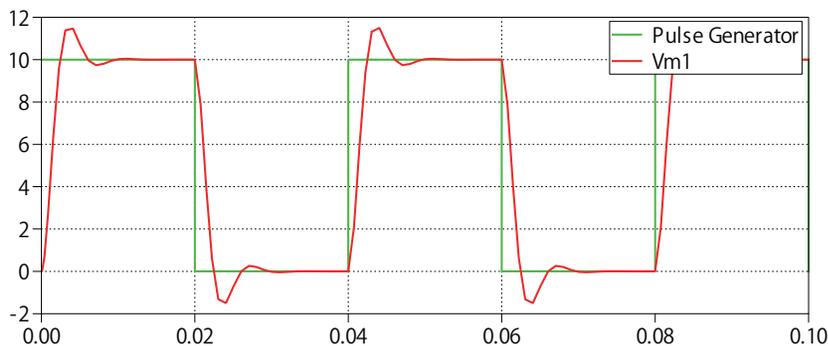


図9: 動的刺激を伴うRLCネットワークのシミュレーション結果



## PLECSとSimulinkの接続

ここまでは、PLECS回路にはSimulink環境との接続がありません。これはSimulinkのPLECS Circuitブロックに入力も出力もないことからわかります。入力と出力を追加するには、PLECSのSystemライブラリのポートブロックを使用する必要があります。

この場合、SimulinkでPLECSサブシステム内の電圧計によって測定された電圧にアクセスし、Simulinkから方形波信号を受信して、PLECSサブシステム内の電圧源を変調します。PLECS内の信号はSimulinkブロック間の接続に対応し、ポートブロックはPLECSとSimulink内のコンポーネント間で一方向の情報交換を提供します。これを実証するために、[図8](#)に示す制御ブロックをSimulinkレベルで再作成し、PLECSで既に作成した電気回路をシミュレーションしてみましょう。



### あなたのタスク:

- 1 rlc\_network\_final\_plecs.mdlモデルを開き、SystemライブラリからSignal InportブロックとSignal Outputブロックを1つずつPLECS回路図ウィンドウに追加します。
- 2 Pulse Generatorブロックを削除し、Signal Inportコンポーネントに置き換えます。
- 3 Scopeブロックを削除し、Signal Outputコンポーネントに置き換えます。

これでSimulinkレベルのPLECS Circuitブロックに入力端子と出力端子が表示されるはずです。



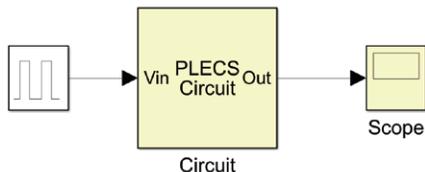
### あなたのタスク:

- 1 次に、SimulinkライブラリからPulse Generatorコンポーネントをドラッグ アンドドロップし、plecslibツールボックスからScopeコンポーネントをSimulink回路図ウィンドウにドロップして、これらのコンポーネントをPLECS回路に接続します。
- 2 前述のようにPulse Generatorに必要なパラメータを設定し、シミュレーションを開始します。



この段階では、モデルは[図10](#)のように参照モデルrlc\_network\_final\_simulink.mdlと同じになっています。得られるシミュレーション結果は、[図9](#)と同じになるはずです。

図10: Simulinkの制御ブロックとPLECSのScopeを備えたRLCネットワーク



**注意:** パルス発生器によって生成される信号は離散的であり、つまりその値は突然変化します。通常、PLECS Scopeは信号の種類を自動的に判別し、垂直方向の傾きを表示します。ただし、この場合、パルス発生器から出力される離散信号は、Scopeに到達する前に連続信号と多重化されるため、台形曲線が表示されることがあります。これを回避するには、[図11](#)に示すように、スコープのDataウィンドウで信号タイプを手動で**Discrete**に設定する必要があります。

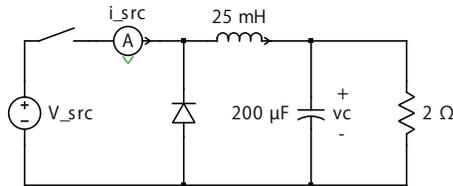
図11: PLECS ScopeのDataウィンドウ

Data		
Name	Cursor 1	Cursor 2
Time	0.0343148	0.066315
Plot 1		
Vin	0	0
Vm1	0.00556783	0.111828

## 3 降圧コンバータ

次に、このセクションでは、理想スイッチの概念を紹介します。図12に示す降圧コンバータと閉ループコントローラを構築します。

図12: 降圧コンバータの回路図



### スイッチ

図12に示す降圧コンバータの場合、トランジスタは、一方向にのみ電流を流すことができることに念頭に置き、制御可能なスイッチ、または理想的なIGBTやMOSFETコンポーネントを使用してモデル化できます。また、フリーホイールダイオードも必要です。ダイオードは、両端の電圧が正になると閉じ、ダイオードを流れる電流が負になると開くスイッチです。

ダイオードはElectrical/Power Semiconductorsライブラリにあり、スイッチはElectrical/Switchesライブラリにあります。これら2つのライブラリのすべてのコンポーネントは、オン抵抗がゼロ(オプションで追加可能)でオフ抵抗が無限大の理想スイッチに基づいています。スイッチは瞬時に開閉します。詳細については、ライブラリ内の任意のコンポーネントを右クリックし、**Help**を選択してください。

スイッチは外部信号によって制御されます。ゼロ以外の入力でスイッチは閉じ、信号がゼロに戻るとスイッチは開きます。



### あなたのタスク:

- 1 新しいSimulinkモデルを作成し、その上にPLECS回路ブロックをドラッグアンドドロップします。
- 2 PLECS ライブラリからSwitchおよびDiodeコンポーネントをPLECS回路図ウィンドウにドラッグアンドドロップします。
- 3 図13に示すように、降圧コンバータの残りの電気部品を組み立てます。

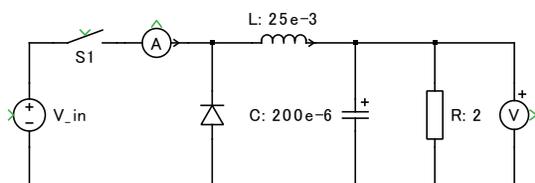


**注意:** 回路図上のコンポーネントの名前を非表示にするには、コンポーネントを右クリックし、**Format**メニューから**Show name**のチェックを外すか、キーボードショートカット**Ctrl + Shift + N**を使用します。



この段階では、モデルは参照モデルbuck\_converter\_1.mdlと同じになるはずですが。

図13: 降圧コンバータの電気部品



## サブシステム

すべてのモデルの特定の部分を互いに分離することができます(この場合は電気部分と制御部分など)。これはシミュレーション結果には影響しませんが、システム全体をより構造化します。サブシステムを使用すると、モデルを明確に表示でき、デバッグが容易になります。



### あなたのタスク:

- 1  図13に示す回路が完成したら、すべてのコンポーネントを選択します(回路図の左上隅の空白部分をクリックしてフレームを右下隅にドラッグするか、キーボードショートカット**Ctrl + A**を使用します)。
- 2 **Edit**メニューから**Create Subsystem**をクリック、キーボードショートカット**Ctrl + G**、または選択したコンポーネントの1つを右クリックして新しいサブシステムを作成します。これで、電気コンポーネントは"Sub"という新しいサブシステムに含まれるはずですが。
- 3 必要に応じて、サブシステムの名前は"Circuit"などに変更できます。



**注意:** サブシステムアイコンのサイズは、選択した角の1つをドラッグすることで変更できます。名前ラベルはクリックしてアイコンの境界または角にドラッグすることで、別の位置に移動することもできます。

## 信号ポート

Simulinkでリンクする場合、PLECSサブシステムを外部の回路図に接続するには、サブシステムにポートを配置する必要があります。



### あなたのタスク:

- 1 サブシステムをダブルクリックして開き、2つのSignal Inportと2つのSignal Outputportをサブシステム回路図にドラッグします。
- 2 Signal Inportをそれぞれ電圧源とスイッチに接続します。
- 3 Signal Outputportをそれぞれ電圧計と電流計に接続します。
- 4 必要に応じて、信号ポートの名前を変更します。



**注意:** サブシステム回路図にドラッグしたポートごとに、サブシステムアイコンに新しい端子が表示されます。これらの端子をアイコンのフレームに沿って移動するには、**Shift**キーを押しながらマウスでドラッグします。

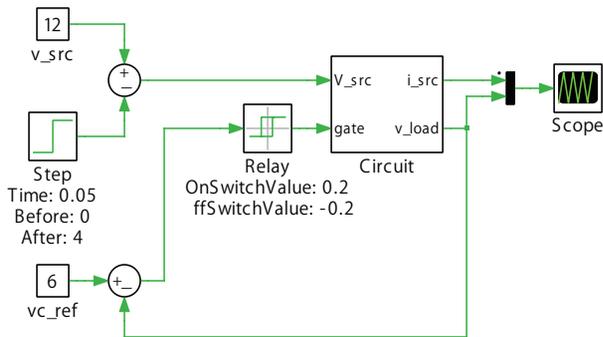


この段階では、モデルは参照モデルbuck\_converter\_2.mdlと同じになるはずですが。

## PLECSでの制御

降圧コンバータでは、キャパシタ電圧を約6Vの±0.2Vの範囲に維持するヒステリシス制御を実装します。これには、Constantブロックを基準設定点として使用し、測定した負荷電圧をフィードバックすることが含まれます。次に、設定値と測定電圧の差としてエラー信号が生成され、これがRelayブロックに供給され、その出力がスイッチへのゲート信号となります。ここでの制御方式は、負荷電圧が5.8Vを下回るとスイッチへのゲート信号をHighに設定し、電圧が6.2Vを超えるとゲート信号をLowに設定します。インダクタを流れる電流を調整しているだけで、キャパシタの充電状態も調整されています。

図14: ヒステリシス制御を備えた降圧コンバータ



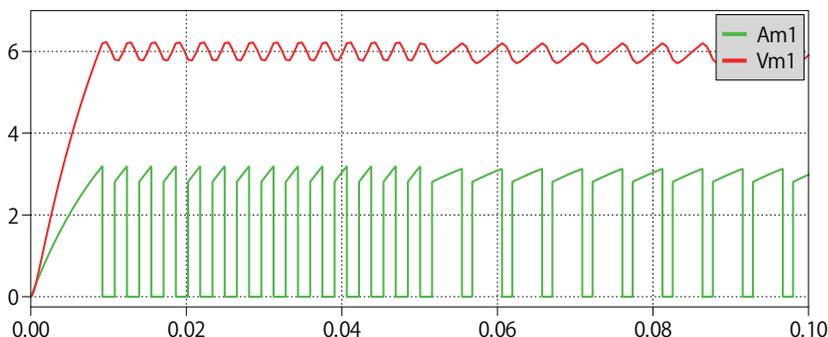
### あなたのタスク:

- 1 ControlドメインライブラリからRelay、Sum、Constantブロックを回路図ウィンドウにドラッグ アンド ドロップします。
- 2 図14に示すコントローラを構築します。
- 3 少し面白くするために、シミュレーション中に入力電圧に12Vから8Vへのステップ変化を適用します。
- 4 シミュレーションの**Stop time**を0.1秒に設定し、シミュレーションを開始します。



この段階では、モデルは参照モデルbuck\_converter\_3\_plecs.mdlと同じになるはずです。得られるシミュレーション結果は、図15と同じになるはずです。

図15: ヒステリシス制御を備えた降圧コンバータのシミュレーション結果



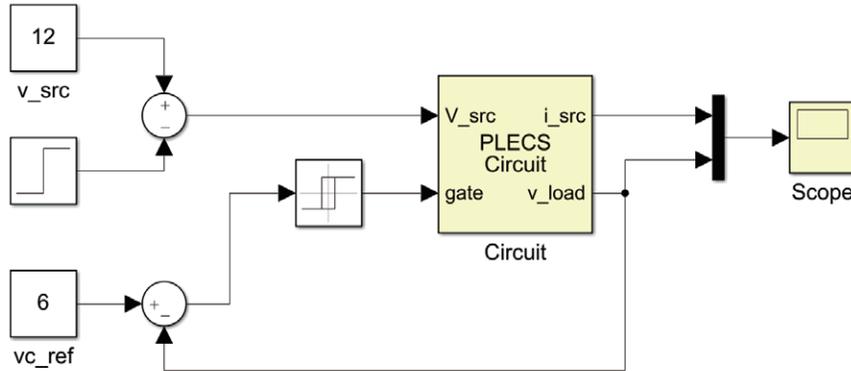
## Simulinkでの制御

また、コントローラをSimulinkレベルで構築することもできます。



**あなたのタスク:** 図16に示すように、Simulinkレベルでヒステリシス型コントローラを構築し、Simulinkレベルでの使用を目的としたPLECS Scopeを使用します。

図16: Simulinkのヒステリシス制御を備えた降圧コンバータ



この段階では、モデルは参照モデルbuck\_converter\_3\_plecs.mdlと同じになるはずですが、得られるシミュレーション結果は、図15と同じになるはずですが。

## プローブ

信号を測定するためにAmmeterやVoltmeterのコンポーネントを使用する代わりに、Probeブロックを使用して信号を測定することもできます。このコンポーネントは、不要な接続を減らすことでモデルを整理するのに役立ち、他の方法では測定が難しい量にアクセスできるようになります。

Probeブロックには2つのバージョンがあります。1つはPLECS 回路図内で使用するもので、PLECSライブラリブラウザのSystemカテゴリにあります。もう1つのバージョンはSimulinkレベル専用(ただし、Circuitブロック内のPLECSコンポーネントからの量のみを監視します)で、plecslibツールボックスウィンドウにあります。

PLECSでコンポーネントをProbeブロックに関連付けるには、回路図で対象のコンポーネントをその上にドラッグし、矢印が表示されたらマウスを放します。接続しているワイヤも移動しますが心配しないでください。これを正しく実行していれば、マウスを放すとすぐに、ドラッグしていたコンポーネントは元の位置に戻ります。Probeブロックをダブルクリックし、開いたProbe Editorウィンドウに対象のコンポーネントをドラッグすることもできます。

SimulinkレベルでPLECSコンポーネントをProbeブロックに関連付けるには、Probeブロックをダブルクリックし、開いたProbe Editorウィンドウに対象のコンポーネントをドラッグして矢印が表示されたらマウスを放します。



### あなたのタスク:

- 1 ProbeブロックをPLECS回路図またはSimulink回路図にドラッグ アンド ドロップします。
- 2 "Circuit"サブシステムのVoltage SourceコンポーネントとResistorコンポーネントを関連付け、それぞれ**Source current**と**Resistor voltage**コンポーネント信号を選択します。

- 3 プロットを右クリックして**Insert plot above**を選択し、スコープ ウィンドウに2つ目のプロット(プローブ ブロックと同じレベルのプロット)を追加します。
- 4 Probeの出力端子を、Scopeブロックの新しく作成された入力端子に接続します。
- 5 シミュレーションを開始します。上部と下部のプロットが同一であることがわかります。



この段階では、モデルは参照モデル**buck\_converter\_final\_plecs.mdl**、または**buck\_converter\_final\_simulink.mdl**と同じになっているはずです。同じになるはずはです。

## 4 デモモデル

PLECSで最初のモデルを構築したら、PLECSに付属するデモ モデルを確認することをお勧めします。PLECS Blockset内の**View**メニューまたは**plecslib**ツールボックスウィンドウにあるボタンから**Demo Models**を選択して、デモモデルブラウザを開きます。

改訂履歴:

Tutorial Version 1.0

初版



**Pleximへの連絡方法:**

☎ +41 44 533 51 00

Phone

+41 44 533 51 01

Fax

✉ Plexim GmbH

Mail

Technoparkstrasse 1

8005 Zurich

Switzerland

@ info@plexim.com

Email

<http://www.plexim.com>

Web



**アドバンオートメーションへの連絡方法:**

☎ +81 3 5282 7047

Phone

+81 3 5282 0808

Fax

✉ ADVAN AUTOMATION CO.,LTD Mail

1-9-5 Uchikanda, Chiyoda-ku

Tokyo, 101-0047

Japan

@ plecs\_adv@adv-auto.co.jp

Email

<https://adv-auto.co.jp/>

Web

*PLECS Tutorial*

© 2002–2021 by Plexim GmbH

このマニュアルに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。