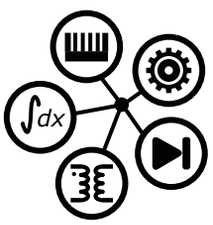


PLECS
DEMO MODEL

Buck Converter with Parameter Sweep

降圧コンバータのパラメータスイープ

PLECS Standaloneの並列シミュレーション機能の使用

Last updated in PLECS 5.0.2

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

1 概要

このデモは、PLECS demosライブラリの"Buck Converter with Voltage Controls"に基づいています。シミュレーションスクリプトでインダクタL_1の値を変更することにより、パラメータスイープを実行します。バージョン4.6.1以降のPLECS Standaloneでは、並列シミュレーションを実行する機能を備えており、このデモモデルでも実証されています。

1.1 要件および制限事項

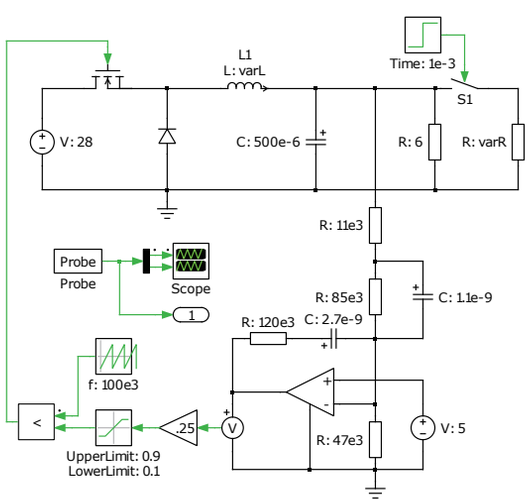
このデモモデルでは、さまざまなPLECSの様々な機能を紹介しています。PLECSのバージョン(BlocksetまたはStandalone)に応じた機能のみ使用できることにご注意ください。

- PLECS Blocksetモデルは、逐次パラメータスイープを備えたシミュレーションスクリプトを備えています。
- PLECS Standaloneモデルは、パラメータスイープの逐次および並列実装のシミュレーションスクリプトを備えています。
- RPCインタフェースを使用して外部シミュレーションスクリプトを実行するには、**PLECS設定** -> **基本設定**タブで**RPCインターフェイス**を有効にし、ポートを`1080`に設定する必要があります。

2 モデル

この回路図は、アナログ比例積分微分(PID)コントローラを備えた降圧コンバータを示しています。インダクタL_1は、図1の回路図でラベル付けされており、40から220μHから20μH刻みで、自動的に10回シミュレートします。PLECSプローブコンポーネントに接続された出力信号ポートは、シミュレーションスクリプトとRPCインタフェースにデータを渡します。

図1: アナログ制御を備えた降圧コンバータの回路図。PLECSプローブコンポーネントに接続された出力信号ポートは、シミュレーションスクリプトにデータを返すために使用します。



3 シミュレーション

シミュレーションは、コンバータの起動と1秒での負荷変動を示します。モデルの初期化時にL_1に40μHのインダクタンス値が割り当て、単一の過渡シミュレーションを実行できます。あるいは、**シミュレーション** → **シミュレーションスクリプト**で定義されたシミュレーションスクリプトを使用して、L_1のパラメータスイープを実行することもできます。各シミュレーションの結果は、PLECSスコープに新しいトレースとして表示されます。それぞれのトレースには、

対応するインダクタンス値がラベル付けされています。また、スクリプトはシミュレーション結果を解析し、ピーク電流値をMATLABまたはOctaveコンソールに出力します。

3.1 PLECS Standalone用Octaveスクリプト

降圧コンバータのインダクタ値のパラメータスイープを実装する2つのシミュレーションスクリプトがあります。1つは逐次パラメータスイープを実行し、もう1つは並列パラメータスイープを実行します。

Parameter Sweep (Sequential)

逐次パラメータスイープは、forループで複数の連続したシミュレーションを実行します。1つのシミュレーションが終了した場合のみ、次のシミュレーションを開始します。forループの各反復処理で、`ModelVars`構造体にインダクタ(変数`varL`)の新しい値が割り当てられます。`ModelVars`構造体は、`plecs('simulate')`コマンドのパラメータとしてPLECSに渡される`simStruct`の一部です。`SolverOpts`構造体には、特定の時点のシミュレーションデータのみを返す変数`OutputTimes`が含まれています。これにより、シミュレーションスクリプト内で処理されるデータの量を減らすことができます(セクション3.2を参照)。このデモモデルでは、デフォルトのソルバ設定と選択された`OutputTimes`ベクトルを使用して、シミュレーションデータの出力は約4300ポイントから501ポイントに削減されます。

```
% create simStruct with field 'ModelVars'
mdlVars = struct('varL', 50e-6);
simStruct = struct('ModelVars', mdlVars);
% clear all previous traces in scope 'Scope' in the current model
plecs('scope', './Scope', 'ClearTraces');
% parametric values to be swept
inductorValues = [40:20:220]; % in uH
for ix = 1:length(inductorValues)
 % set value for L1
 simStruct.ModelVars.varL = inductorValues(ix) * 1e-6;
 simStruct.SolverOpts.OutputTimes = 0.001:0.2e-5:0.002;
 % start simulation, return probed signal values in 'out'
 out = plecs('simulate', simStruct);
 % hold and label trace
 plecs('scope', './Scope', 'HoldTrace', ['L=' mat2str(inductorValues(ix)) 'uH']);
 % find maximum current value and index
 [maxv, maxidx] = max(out.Values(1,:));
 % Output maximum current values to Octave console
 printf('Max current for L=%duH: %fA at %fs\n',
 inductorValues(ix), maxv, out.Time(maxidx));
end
```

Parameter Sweep (Parallel)

並列シミュレーションを実行するには、`plecs('simulate')`コマンドのパラメータに複数のシミュレーション構造体をセル配列として指定する必要があります。各シミュレーション構造体には、`Name`と呼ばれる追加変数が含まれる場合があります。これにより、個々の並列シミュレーションにラベルを付け、シミュレーションデータやエラーを特定のシミュレーションに関連付けることができます。この例では、$L_1 = 120\mu H$ のときに人為的な短絡が作成されます(シミュレーション5)。シミュレーションが完了すると、OctaveコンソールとPLECS回路図の右下隅にエラー メッセージが表示されます。図2を参照してください。

図2: シミュレーション5における負荷部での短絡を示すエラーメッセージシンボル。

plecs('simulate')コマンドには、並列シミュレーション設定用の追加パラメータがあります。これは各シミュレーションの後実行されるコールバック関数です。この関数を使用して、シミュレーションデータ量を重要な数値のみに絞り込むことができます。これは、RPCインタフェースを介してシミュレーションを開始する場合、大量のデータ転送を避けたい場合に特に役立ちます。

```
% clear all previous traces in scope 'Scope' in the current model
plecs('clc');
plecs('scope', './Scope', 'ClearTraces');
% Evaluate simulation results in callback function
function result = callback(index, data)
  % hold and label trace
  name = ['L = ', mat2str(inductorValues(index)), 'uH'];
  plecs('scope', './Scope', 'HoldTrace', name);
  % Find maximum current values and index
  if isstruct(data)
    [maxi, maxidx] = max(data.Values(1,:));
    maxt = data.Time(maxidx);
    % Reducing simulation results by return value 'result'
    result = [maxi, maxt];
  else
    % Print error message to Octave console
    printf(' Error in Simulation %d for L=%duH: %s\n',
      index, inductorValues(index), data);
  end
end
% Set value for L1 to be swept
inductorValues = [40:20:220]; % in uH
% Initialize simStruct as cell array with all values for L1
for ix = 1:length(inductorValues)
  simStructs{ix}.ModelVars.varL = inductorValues(ix) * 1e-6;
  % Name of 'ModelVars' can be assigned for diagnostic purposes
  simStructs{ix}.Name = ['L=' mat2str(inductorValues(ix)) 'uH'];
end
% Create a shortcut in simulation 5
simStructs{5}.ModelVars.varR = 0;
% Start simulation, return result from callback function into 'out'
% Analysis will be moved to callback function to reduce simulation results
out = plecs('simulate', simStructs, @(index, data) callback(index, data));
for ix = 1:length(inductorValues)
  % Detect if errors occurred in parallel simulation
  if ischar(out{ix})
    printf(' Error for L=%duH: %s\n',
    inductorValues(ix), out{ix});
    % Output maximum current values to Octave console
  else
    printf(' Max current for L=%duH: %fA at %fs\n',
    inductorValues(ix), out{ix}(1), out{ix}(2));
  end
end
```

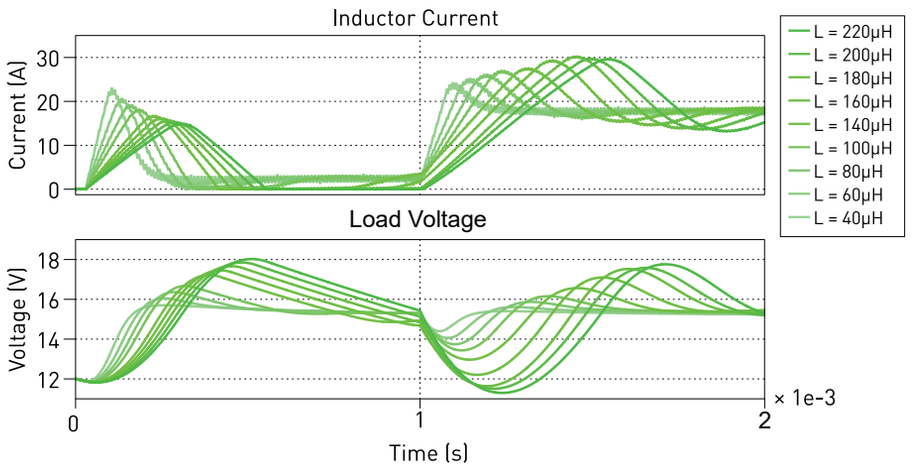
同様の並列シミュレーションは、このデモモデルのフォルダ内にあるparameter_sweep_script.py(Pythonスクリプト)、またはparameter_sweep_script.m(MATLAB/Octave)を使用して開始することもできます。これには、**PLECS設定**のGeneralタブで、RPCインターフェイスポート番号を、1080に設定する必要があることに注意してください。パラメータスイープのシミュレーション結果を図3に示します。

3.2 シミュレーションデータの量を削減

シミュレーションの時間範囲、平均シミュレーション時間ステップ、および記録された信号の数に応じて、結果として得られるシミュレーションデータは大量になる可能性があります。シミュレーションを並行して実行すると、メモリ消費量がさらに増加します。 以下では、シミュレーション データの量を最小限に抑える方法を示しています。

- **シミュレーションパラメータ**ダイアログの**最大時間刻み幅**パラメータを小さくしないでください。

図3: パラメータ スイープの出力結果



- シミュレーション結果の評価は、スクリプト`Parameter Sweep (Parallel)`に示されているように、対応するコールバック関数内で行うする必要があります。これにより、関連するシミュレーションデータ(`result`)のみがスクリプトに返されます。後処理用のコールバック関数がを用していない場合、すべてのシミュレーション結果がシミュレーションスクリプトに返されます (戻り値`out`)。
- シミュレーション時間内の任意のステップサイズを持つ時間ベクトル`OutputTimes`を指定して、シミュレーションデータを削減できます。`SolverOpts`と組み合わせて、シミュレーションパラメータダイアログで指定されたソルバ設定をオーバーライドできる構造体変数を使用する必要があります。これをParameter Sweep (Sequential)の14行目で示しています。

結果はOctaveコンソールに出力されます。Octaveコンソールには、**ウィンドウ**メニューから**Octaveコンソールの表示**を選択してアクセスできます。

3.3 PLECS Standalone用のPython 3スクリプト

前のセクションで説明したものと同じシミュレーションスクリプトがPython 3に実装されています。スクリプトは次のコマンドで実行できます:

```
python3 parameter_sweep_script.py
```

PLECS設定 → **基本設定**タブでRPCインターフェイスを有効にし、ポートを`1080`に設定してください。スクリプトを実行するには、最初にPLECSを手動で起動する必要があります。

3.4 PLECS Standalone用のMATLAB/Octaveスクリプト

同様のシミュレーションスクリプトは、MATLAB/Octaveから実行できるmスクリプトとして実装されています。これらのスクリプトは、MATLAB/Octaveのコマンドウィンドウから実行できます。

MATLAB/Octaveは、RPCインタフェースを介してPLECS Standaloneと通信するためにJSON-RPCクライアントを必要とします。Plexim社は、https://github.com/plexim/matlab-jsonrpcに適切なクライアントを提供しています。jsonrpc.mファイルは、スクリプトおよびPLECSモデルと同じディレクトリ、またはPATH環境変数で指定されたディレクトリに配置する必要があります。

シミュレーションの種類(例: 逐次シミュレーションまたは並列シミュレーション)を選択するには、Pythonスクリプト内の変数`METHOD`を適切に調整してください。**PLECS設定** → **基本設定**タブでRPCインターフェイスを有効にし、ポートを`1080`に設定してください。スクリプトを実行するには、まずPLECSを手動で起動する必要があります。

```
METHOD = 'Sequential'; % The options are 'Sequential' or 'Parallel'
%start RPC client (you need to enable it in PLECS preferences)
proxy = jsonrpc('http://localhost:1080');
%load the model (Start PLECS manualy)
MODEL_NAME = 'buck_converter_with_parameter_sweep';
dir = pwd();
proxy.plecs.load([dir '/' MODEL_NAME '.plecs']);
% Create simStruct with field 'ModelVars'
mdlVars = struct('varL', 50e-6);
simStruct = struct('ModelVars', mdlVars);
% Clear all previous traces in scope 'Scope' in the current model
proxy.plecs.scope([MODEL_NAME '/Scope'], 'ClearTraces');
% Parametric values to be swept
inductorValues = [40:20:220]; % in uH
switch METHOD
case 'Sequential'
for ix = 1:length(inductorValues)
 % Set value for L1
 simStruct.ModelVars.varL = inductorValues(ix) * 1e-6;
 simStruct.SolverOpts.OutputTimes = 0.001:0.2e-5:0.002;
 if ix == 5
  simStruct.ModelVars.varR = 0;
 else
  simStruct.ModelVars.varR = 1;
 end
 % Start simulation, return probed signal values in 'out'
 try
 out = proxy.plecs.simulate(MODEL_NAME,simStruct); % start AC Sweep analysis
 catch
 disp([' Error in Simulation ', num2str(ix),' for ', num2str(inductorValues(ix)),' µH']);
 end
 % Hold and label trace
 proxy.plecs.scope([MODEL_NAME '/Scope'], 'HoldTrace', ['L=' mat2str(inductorValues(ix))'µH']);
 % Find maximum current value and index
 [maxv, maxidx] = max(out.Values(1,:));
 % Output maximum current values to console
 disp(['Max current for L= ', num2str(inductorValues(ix)),'µH: ', num2str(maxv), ' A at ',
↪num2str(out.Time(maxidx)), ' s']);
end
case 'Parallel'
% Evaluate simulation results in callback function, the disp in the callback are shown in
↪the PLECSConsole
% In MATLAB use single quotes (') to get a single concatenated character array as in Octave
callback = ['if ischar(result)' ...
'disp([''There is a simulation error for the fifth case ('' name '') where varR = 0 is
↪artificially set to zero. Nevertheless the results for the other cases are still calculated'']);'
↪...
            'plecs(''scope'', ''./Scope'', ''HoldTrace'', name);' ...
            'else ' ...
            'plecs(''scope'', ''./Scope'', ''HoldTrace'', name);' ...
            'disp([''Simulation Nr. '' num2str(index) '' executed'']); '...
            '[maxi, maxidx] = max(result.Values(1,:));' ...
            'maxt = result.Time(maxidx);' ...
            'result = [maxi, maxt];' ...
            'end'];
% Initialize simStruct as cell array with all values for L1
for ix = 1:length(inductorValues)
 simStructs{ix}.ModelVars.varL = inductorValues(ix) * 1e-6;
 % Name of 'ModelVars' can be assigned for diagnostic purposes
 simStructs{ix}.Name = ['L=' mat2str(inductorValues(ix)) 'µH'];
 %
 simStructs{ix}.SolverOpts.OutputTimes = 0.001:0.2e-5:0.002;
```

(次ページに続く)

(前ページからの続き)

```
 end
 % Create a shortcut in simulation 5
 simStructs{5}.ModelVars.varR = 0;
 out = proxy.plecs.simulate(MODEL_NAME, simStructs, callback); % the callback processes the ↵
↪responses from every simulation and returns [maxv, maxt]
 for ix = 1:length(inductorValues)
  % Detect if errors occurred in parallel simulation
  if ischar(out{ix})
  disp([' Error in Simulation ', num2str(ix),' for ', num2str(inductorValues(ix)),' µH']);
  % Output maximum current values to console
  else
  % Find maximum current value and index
  [maxv, maxt] = deal(out{ix}(1), out{ix}(2));
   disp(['Max current for L= ', num2str(inductorValues(ix)),'µH: ', num2str(maxv), ' A at ', ↵
↪num2str(maxt), ' s']);
  end
 end
 end
```

結果はMATLABコンソールに表示されます。

3.5 PLECS Blockset用のスクリプト

Simulinkモデルのサブシステムブロックをダブルクリックして、MATLABエディタでmファイル`parameter_sweep_script.m`を表示して実行します。このファイルの内容は次のとおりです:

```
 % create path to scope
 scope = ('buck_converter_with_parameter_sweep/Circuit/Scope');
 % clear all previous traces in scope 'Scope' in the current model
 plecs('scope', scope, 'ClearTraces');
 % parametric values to be swept
 inductorValues = [40:20:220]; % in uH
 for ix = 1:length(inductorValues)
  % set value for L1
  varL = inductorValues(ix) * 1e-6;
  % start simulation, return probed signal values
  % to workspace using Output port '1'
  [t, x, y] = sim('buck_converter_with_parameter_sweep');
  % hold and label traces in scope
  plecs('scope', scope, 'HoldTrace', ['L=' mat2str(inductorValues(ix)) 'uH']);
  % find maximum current value and index
  [maxv, maxidx] = max(y(:,1));
  % Output maximum current values to MATLAB console
  fprintf('Max current for L=%duH: %fA at %fs\n',
  inductorValues(ix), maxv, t(maxidx));
 end
```

結果はMATLABコンソールに表示されます。

4 まとめ

このデモは、PLECS BlocksetまたはPLECS Standaloneにおいて、シミュレーションスクリプトを使用して物理回路値のパラメータスイープを実行する方法を示しています。このサンプルコードは、他のアプリケーションにも容易に適用できます。

改訂履歴:

PLECS 4.3.1	初版
PLECS 4.6.1	PLECS Standaloneにパラメータスイープの並列実装を追加。XML-RPCでパラメータスイープを実行するPython 3スクリプトを追加
PLECS 4.8.1	JSON-RPCを使用するオプションをインクルード Python 3スクリプトにパラメータスイープの逐次実装を追加しました。 JSON-RPCに関する軽微な問題を修正しました。
PLECS 5.0.2	PLECS StandaloneでRPCインタフェースを使用するためのMATLAB/Octaveスクリプトを追加

Plexim への連絡方法:

☎ +41 44 533 51 00 Phone

✉ Plexim GmbH Mail
Technoparkstrasse 1
8005 Zurich
Switzerland

@ info@plexim.com Email

http://www.plexim.com Web

計測エンジニアリングシステムへの連絡方法:

☎ +81 3 6273 7505 Phone

✉ Keisoku Engineering Systm CO.,LTD. Mail
1-9-5 Uchikanda, Chiyoda-ku
Tokyo, 101-0047
Japan

https://kesco.co.jp Web

PLECS Demo Model