

Introduction to the RT Box using PLECS

RT Box 1での簡単なPLECSモデルの構築

Step-by-step introduction to the RT Box architecture, real-time workflow, inputs and outputs, and model structure RT Boxアーキテクチャ、リアルタイムワークフロー、入出力、モデル構造の段階的な紹介

Tutorial Version 1.0



1 はじめに

このチュートリアルでは、PLECS RT Boxで基本的なPLECSモデルを実行する方法を段階的に学習します。このチュートリアルは、PLECS Standaloneを使用してPLECSモデルからRT Box上のリアルタイムシミュレーションに移行するユーザ向けに設計されています。チュートリアルの具体的な学習目標は次のとおりです:

- ・ RT Boxハードウェア、ワークフロー、およびRT Boxターゲットサポートライブラリ コンポーネントについて深く理解します。
- ・ リアルタイムシミュレーションからのデータでPLECSスコープを更新する方法、リアルタイムシミュレーション中にパラメータ を変更する方法、RT Box Web Interfaceからシミュレーションに関する情報を収集する方法を学びます。
- ・ PLECSモデルからリアルタイムシミュレーションへの移行を容易にするためにモデルを構築する方法を学びます。

始める前に このチュートリアルでは、RT BoxとPLECS Coderのライセンスが必要です。RT Boxを初めて設定する合は、RT Box ユーザマニュアルの第1章**クイックスタート**を参照してください。

このチュートリアルは、RT Boxのみで完了できるように設計されています。ただし、追加のハードウェアを必要とする、斜体で示されている追加の演習がいくつかあります。これらの演習では、オシロスコープ、RT Boxインタフェースカード、またはループバックケーブルを使用して、重要な概念をさらに詳しく説明します。

オシロスコープのプローブを RT Boxに接続するには、Pleximの RT Boxインタフェースカード、標準のD-SUB 37ピンのオス およびメスコネクタ、または手元にあるその他のブレッドボードジャンパワイヤを使用すると便利です。追加の演習では、ループ バックケーブルを使用して、RT Boxの出力から RT Boxのアナログ入力とデジタル入力を駆動します。これを行うには、D-SUB 37ピンのオス - メスケーブルを使用します。

2 RT Boxアーキテクチャ

PLECS RT Boxは、デュアルコアのARM CPUとFPGAを搭載したXilinx Zynqシステムオンチップ(SoC)をベースにした最先端 のリアルタイムシミュレータです。リアルタイムシミュレーションはCPUコア上で実行され、FPGAはRTBoxの入力と出力(I/O) へのインタフェースとして機能します。CPUコアには、物理モデルまたは制御システムを表現するCコードを実行する柔軟性が あります。ホストPC上のPLECSシミュレーションに使用するのと同じコードを、RT Box上のモデルのシミュレーションで使用 することもできます。このアプローチにより、オフラインおよびリアルタイムのシミュレーションのための一貫したモデリング フレームワークが保証されます。

FPGAは、CPU上で実行されるシミュレーションとRT BoxのアナログおよびデジタルI/O間の高分解能、低遅延のインタフェース として機能します。RT Boxには、16個のアナログ入力チャネルと16個のアナログ出力チャネルが装備されています。すべて のアナログI/Oは16-bitの分解能と最大2MHzのサンプリング周波数を備えています。さらに、RT Boxには32点のデジタル 入力チャネルと32点のデジタル出力チャネルがあります。各デジタルI/Oは、PWM信号の生成や、7.5nsの時間分解能でPWM 信号をキャプチャするように構成できます。デジタルI/Oは、汎用信号としても使用できます。

RT Boxのアーキテクチャと動作原理の詳細な概要については、RT Boxユーザマニュアルを参照してください[1]。

3 リアルタイムワークフロー

リアルタイムワークフローは、個別のモデルを開発および維持することなく、PLECSモデルからRT Box上のリアルタイム シミュレーションに簡単に移行できるように設計されています。PLECSにおける典型的なリアルタイムワークフローは、図1に まとめられている以下の主要なステップで構成されています:

図1: 典型的なRT Boxワークフロー



- 1 PLECSモデルを作成: ワークフローはPLECSモデルから開始します。PLECSモデルは、リアルタイムシミュレーション用に モデルを構成する前に、期待どおりに動作する必要があります。一般的に、RT Boxに展開されたリアルタイムシステムと、 RT Boxに接続された外部ハードウェアを表すモデルを開発することが有益です。これにより、オフラインモデルとリアル タイムモデルをワークフローの後半でベンチマークできるようになります。RT Boxにデプロイされるモデルには、PLECS マニュアル[2]の"コード生成"に記載されている"サポートしていないコンポーネント"を除き、任意のPLECSドメインの コンポーネントを含めることができます。
- 2 RT Boxライブラリコンポーネントの追加と構成: 次のステップは、ライブラリブラウザから RT Box固有のコンポーネント を追加して、シミュレーション信号をRT BoxフロントパネルのI/Oにマッピングすることです。RT Boxから物理ハードウェア までの信号パスを定義し、アナログI/Oスケールとオフセット値を計算する必要があります。この段階では、RT Boxにデプロイ されるモデルの部分が1つのサブシステム内に含まれるようにモデルを構造化することも役立ちます。
- 3 オフラインシミュレーションを実行: すべてのRT BoxのI/Oコンポーネントには代表的なオフラインモデルがあります。 モデル全体をオフラインでシミュレートして、モデル内のすべてのスケーリングと信号接続が正しいことを確認します。 この段階では物理的な接続もチェックする必要があります。
- 4 離散化ステップサイズを選択: PLECSモデルをリアルタイムでシミュレートするには、固定ステップソルバを使用して、 モデルを離散化して固定実行レートで実行する必要があります。PLECS Coderは、モデルの連続状態空間方程式を、 固定ステップソルバと互換性のある離散状態空間表現に変換します。PLECS Coderは、モデルを汎用ターゲットまたは リアルタイムターゲットで実行するように設計されたCコードに変換します。コードを生成する前に、離散化ステップサイズを 指定する必要があります。電力変換システムに固有の高速な時定数により、離散化ステップサイズは通常、マイクロ秒の オーダーになります。理想的なステップサイズは、システムモデルの忠実度とシミュレーション結果の精度の間の妥協点です。
- 5 コード生成シミュレーションを実行: コード生成モードを使用すると、リアルタイム シミュレーションの前にモデルの離散化の 影響を評価できます。コード生成モードは、PLECS Coderを利用して、ホストPC上のPLECSで実行できる汎用Cコードを作成 します。汎用Cコードは、連続状態空間モデルに対してベンチマークできる離散化モデルを表します。離散化シミュレーション 結果と連続シミュレーション結果を比較すると、離散化がモデルの忠実度にどのように影響するかが示され、選択した離散化 ステップ サイズが適切かどうかが示されます。この段階では、必要なモデル忠実度を達成するために離散化ステップサイズ を調整する必要があります。

- 6 モデルをビルドしてRT Boxにアップロード: PLECS Coderは、モデルをRT Box上でリアルタイムに実行することに特化した Cコードに変換します。モデルコードは、ワンクリックで自動的に生成、コンパイルされ、RT Boxにアップロードされます。 エラーがなければシミュレーションは自動的に開始します。エラーが発生した場合、RT Box Web Interfaceはアプリケーション ログに重要なデバッグ情報を提供します。この段階では、リアルタイムパフォーマンスとRT Boxプロセッサの負荷に基づいて、 モデルをさらに最適化し、離散化ステップサイズを調整する必要がある場合があります。
- 7 外部モードでRT Boxに接続してテストを実行: RT Box上でリアルタイムコードが実行されると、ユーザは"外部モード"に 入り、PLECSスコープで波形をで更新したり、特定のシミュレーションパラメータを変更したり、後処理用にリアルタイム データをキャプチャしたりできるようになります。

4 RT Box Web Interface

RT Box Web Interfaceには、重要な診断情報が表示されます。Web Interfaceには、"Coderオプション"ウィンドウからアクセス できます。"ターゲット"でRT Boxデバイスを選択した後、青いモニタアイコンをクリックして"Webインターフェイス"を開きます。 ここに、RT Boxで実行されているモデルのモデル名とサンプルステップサイズが表示されます。現在のサイクル時間は、RT Box がモデルステップを実行するのにかかる時間を示します。サイクル時間がサンプル時間よりも短いことを常に確認してください。

Applicationタブからモデルをアップロードしたり、RT Boxを起動または停止したりできます。事前コンパイルされたRT Box モデルは、アクティブモデルディレクトリに移動し、生成されたコードを含むフォルダを開いて、・elf拡張子を持つファイルを アップロードすることで、RT Boxにアップロードできます。RT Box Web Interfaceには、有効なPLECSライセンスがなくても、 Webブラウザ経由でアクセスすることもできます。つまり、PLECSソフトウェアを開かなくても、事前にコンパイルされた・elf ファイルを RT Boxにロードして実行できます。

Front panel タブには、フロントパネルのアナログコネクタとデジタルコネクタのピン配置情報と、ステータスLEDの現在の 状態を表示します。フロントパネルコネクタの画像をクリックすると、各コネクタ ピンの機能を示す表を表示します。

Diagnosticsタブでは、エラーがあればそれも含めてアプリケーションログがキャプチャします。赤色のエラーLEDが点灯した 場合、アプリケーションログにはモデルの実行時エラーに関する診断情報が含まれます。

Infoタブには、ホスト名、IPアドレス、MACアドレス、シリアル番号、ファームウェアバージョンなどを表示します。

5 アナログ入力および出力

このセクションでは、RT Boxターゲットサポートライブラリのアナログ入力およびアナログ出力コンポーネントを使用して、 簡単なモデルを作成します。Analog InブロックとAnalog Outブロックは、RT Box のフロントパネルからのアナログ信号を リアルタイム シミュレーションの値にマッピングするために使用されます。Analog InブロックとAnalog Outブロックは、RT Box のフロントパネルからのアナログ信号をリアルタイムシミュレーションの値にマッピングするために使用されます。アナログ 出力チャネルは、ハードウェアインザループ(HIL)シミュレーションでよく使用されます。Analog Outブロックの用途の1つは、 RT Boxでシミュレートされた電力コンバータの電圧または電流の測定値を表すことです。各アナログ出力にはオフセット (Offset)とスケーリング(Scale)のパラメータがあります。オフセットとスケーリングの各パラメータは、センシング回路のゲイン とDCオフセットを動作的にモデル化できます。 RT Boxを使用してラピッドコントロールプロトタイピング(RCP)アプリケーションで電力コンバータを制御する場合、アナログ入力は電圧と電流を感知するために不可欠です。アナログ入力には、オフセット(Offset)とスケーリング(Scale)パラメータもあります。

スケーリングとオフセットの計算は、Analog OutコンポーネントとAnalog Inコンポーネントの両方に対して FPGA上で実行 され、CPUプロセッサの負荷には影響しません。

5.1 アナログ出力の生成

この演習では、PLECSモデル内の信号をRT BoxのAnalog Outピンに送信する方法を学習します。スケーリングとオフセットのパラメータを操作して、アナログ出力信号の波形を調整します。また、アナログ出力電圧の範囲設定とモデルの離散化がアナログ出力信号に与える影響を観察します。

この演習では、**外部モード**について説明します。外部モードは、RT BoxとPLECSユーザインタフェース間の通信リンクを確立 します。外部モードで接続している間、"PLECSスコープ"と"数値表示"はリアルタイムの測定値で更新されます。<u>演習6.1</u>で説明 しているように、外部モードでは特定のパラメータをリアルタイムで変更することもできます。



あなたのタスク:

1 新しいモデルを作成し、RT Boxターゲットライブラリから"Analog Out"コンポーネントを回路図に配置します。振幅1、周波数 50Hzの"正弦波信号"を"Analog Out"コンポーネントに接続します。Analog output to channelを0に設定します。他の" 正弦波信号"と"Analog Out"コンポーネントの値はすべてデフォルトのままにします。信号を表示するには、PLECSライブラリ から"PLECSスコープ"コンポーネントを組み込みます。回路図は図2のようになります。シミュレーションを実行し、モデルを 保存します。

図2: 正弦波信号と接続したアナログ出力チャネル



2 Coderドロップダウンメニューから、Coderオプション...ダイアログを選択します。"コードオプション"ウィンドウで基本設定 タブを選択し、"離散ステップサイズ"として2e-6秒を入力します。ターゲットタブで、PLECS RT Boxをターゲットとして 選択し、双眼鏡アイコンをクリックして利用可能なRT Boxを見つけます。

このタブから、すべてのチャネルのアナログ出力の電圧範囲を選択できることに注意してください。他のすべてのオプション はデフォルトのままにして、ウィンドウの下部にあるビルドボタンをクリックしてビルドプロセスを開始します。モデルが RT Boxターゲットに正常にアップロードされ、シミュレーションが実行されると、RT Boxの前面パネルにある青色の実行 LEDが点灯します。

3 コードオプションウィンドウの外部モードタブに移動します。接続をクリックし、自動トリガを有効化ボタンをクリックします。 PLECS回路図の"PLECSスコープ"を開くと、リアルタイムデータが定期的に更新されていることがわかります。コードオプション ウィンドウから、サンプル数や間引き設定を増やして、正弦波の複数のサイクルをキャプチャできます。 PLECSスコープを更新することにより、データキャプチャを特定のトリガイベントに同期できます。Coderオプションウィンドウ の外部モードタブから、トリガチャネルとして"正弦波信号"を選択します。スコープウィンドウの縦軸には、トリガレベルと 遅延を示す小さな四角形があります。レベルまたは遅延が現在の軸の制限外にある場合は、代わりに小さな三角形が表示 されます。マウスの左ボタンを押しながら四角いアイコンを垂直にドラッグして、スコープウィンドウからトリガレベルを 調整します。トリガ遅延は、Shiftキーを押しながらマウスでトリガアイコンを水平にドラッグすることで、スコープから調整 できます。トリガレベルとトリガ遅延はどちらも外部モードタブから設定できます。このタブからでトリガ更新レートを調整 することもできます。トリガチャネルを"正弦波信号"の出力に変更します。

4 オシロスコープが必要:オシロスコープを"Analog Out"のチャネル0に接続します。Plexim社が設計したインタフェース カードをお持ちの場合は、BNCコネクタを使用してAnalog Out信号にアクセスできる場合があります。Analog Outコネクタ のピン1はアナログ出力チャネルの0に対応し、ピン20は信号グランドに対応します。4章で説明したように、フロントパネル のピン配列は、RT Box Web InterfaceのFront panelタブで確認できます。より詳細なピン配置情報とRT Boxの接地の 考え方については、RT Boxユーザマニュアルを参照してください[1]。オシロスコープで正弦波形を観察し、最大電圧と 最小電圧を記録します。



PLECSスコープとオシロスコープの波形は一致していますか?

- アナログ出力電圧の範囲は、Coderオプションウィンドウのターゲットタブで指定された設定により0~5Vに制限 されることに注意してください。オシロスコープの波形では、正弦波の負の半サイクル中に大きなクリッピングが 発生します。PLECSスコープと外付けのオシロスコープは厳密には同じ波形を表示しません。
- 5 Analog Outコンポーネントのスケールとオフセットのパラメータを変更して、50Hz正弦波入力をピークツーピーク電圧 5.0 V、オフセット2.5Vのアナログ出力に変換します。Coderオプションウィンドウを開き、ビルドをクリックしてモデルをRT Boxにアップロードします。PLECSスコープでシミュレーション結果を調べると、元の波形のピーク振幅が1Vのままで、オフ セットがないことがわかります。
- 6 *オシロスコープが必要*:オシロスコープでアナログ出力の電圧を見ます。クリッピングの影響なく0~5.0Vの間で変化する 波形が表示されます。PLECSスコープは、スケーリングとオフセットが行われる前の信号を表します。
- 7 オシロスコープが必要:モデルの離散ステップサイズを1e-3秒に変更し、モデルを RT Boxにアップロードします。オシロスコープを使用して、離散電圧レベルが表示されるまで正弦波信号を拡大します。離散電圧レベルが維持される合計期間を 測定します。各離散電圧レベルの持続時間は、選択した離散化時間と等しくなります。アナログ出力測定は、シミュレーション タイムステップごとに1回更新されます。
 - 離散化ステップ サイズによってアナログ出力信号の周波数内容が制限されますか?
 - はい。離散化ステップサイズは、アナログ出力信号の周波数内容だけでなく、シミュレーション内のすべての信号を 制限します。連続した周期信号の場合、周期あたりの離散的な時間ステップが多いほど信号の忠実度が高くなります。 表現できる最大周波数はナイキスト周波数によって制限されます。ナイキスト周波数に違反するとエイリアシング が発生します。たとえば、1e-3秒の離散化時間ステップを持つ1100Hzの正弦波は、シミュレーションとアナログ 出力ピンの両方で100Hzの信号を持ちます。

PLECS Scopeではアナログ信号の離散化は表現されますか?

信号の離散化された性質は、必ずしもPLECSスコープで表現されるわけではありません。リアルタイム波形の離散的な性質を示す1つの方法は、信号タイプの設定を変更することです。これを行うには、PLECSスコープでカーソルをアクティブにします。カーソルを表示すると、まだ開いていない場合は"データ"ウィンドウも表示されます。データウィンドウの信号名の横に、信号の種類を表す小さなアイコンが表示されます。信号のアイコンをダブルクリックし、"離散表示"に変更します。

この段階では、モデルは参照モデルintroduction_rtbox_analog_1.plecsと同じになるはずです。

5.2 アナログ入力の測定

ここで、アナログ入力電圧を検出するようにモデルを構成します。演習中のRT BoxのAnalog Inコネクタのヘッダピンに指で 物理的に触れます。ピンに触れると、AC主電源周波数で周期的な歪みが観察されます。信号発生器など、研究室で利用できる 他の機器に基づいてこの演習を変更することもできます。RT Boxのアナログ入力には、シングルエンド入力と差動入力の両方 が装備されています。RT Boxのアナログ入力特性のより詳細な仕様については、RT Boxユーザマニュアル[1]を参照してください。



あなたのタスク:

1 前の演習のモデルに"Analog In"コンポーネントを追加します。アナログ入力チャネルを1に設定します。アナログ入力にはScale(スケール)とオフセットのOffset(パラメータ)もあることに注意してください。これらはデフォルト値のままにしておいても構いません。"Analog In"ブロックと"正弦波信号"ブロックの両方を"マルチプレクサ"に接続します。"マルチプレクサ"の出力を"Analog Out"コンポーネントに接続します。

注意: すべてのRT BoxのI/Oに対して、単一のI/Oチャネルまたはチャネルインデックスのベクトルを入力できます。 複合化されたチャネルが2つあるため、Analog output channelパラメータに"0:1"と入力します。スケールとオフ セット値は、チャネル数と同じ長さのベクトルとして入力することも、すべてのチャネルに適用されるスカラとして 入力することもできます。モデルは図3のようになるはずです。この時点で、生成された正弦波の周波数を50Hzから お住まいの地域のAC主電源周波数に調整する必要があります。

Coderオプションウィンドウを開き、モデルの離散化時間を2e-6秒に設定して、モデルをRT Boxにアップロードします。

図3: アナログ出力に接続されたアナログ入力チャンネルと正弦波



- 2 外部モード経由でRT Boxに接続し、PLECSスコープにリアルタイムデータを表示します。RT Boxの前面パネルにあるAnalog In コネクタのピン2と21を指で触れます。PLECSスコープが周期的な歪みを伴って更新されることがわかります。ピン2とピン21 は、それぞれアナログ入力1のpositive(正)とnegative(負)の入力接続です。
- 3 トリガチャネルをアナログ入力1に変更し、トリガレベルを調整して、アナログ入力ピンに触れているときにのみスコープ が更新されるようにします。トリガ条件が満たされない場合、スコープは更新されません。

- **4** *オシロスコープが必要*: アナログ入力ピンに触れながら、オシロスコープでAnalog Outの0チャネルを観察します。RT Box のアナログ出力に同じ波形が表示されますが、前の演習で決定したAnalog Out設定に従ってスケーリングおよびオフセット されています。指を離すと歪みは消えます。
- 5 *ループバックが必要*: ループバックケーブルを使用して、RT Boxのアナログ出力をアナログ入力に接続します。これで アナログ入力1は、アナログ出力1で生成された正弦波信号に接続されます。

0~5.0Vの入力信号を-1~1Vの信号範囲に調整する"Analog In"ブロックのスケーリングおよびオフセット パラメータを 決定します。アナログ入力信号は、"正弦波信号"の出力と同じ大きさとオフセットを持つことになります。

1e-3秒の離散ステップサイズでモデルを構築してRT Boxにアップロードし、外部モードのサンプルの数を300に減らします。"正弦波信号"の出力が、検出したアナログ入力電圧とどのように相関するかを観察します。

- (?) 生成した正弦波と測定した信号間の遅延はどれくらいですか?
- A 2e-3秒の遅延、つまり2つのタイムステップの遅延が表示されます。
- ステップ5を完了する前に、モデルは参照モデルintroduction_rtbox_analog_2.plecsと同じである必要が あります。ループバックケーブルを必要とするステップ5のオプションを完了すると、モデルはintroduction_ rtbox_analog_3.plecsと同じになるはずです。

6 デジタル入力、出力、PWM信号

RT Boxターゲットサポートライブラリには、2つの異なるクラスのデジタル入力信号と出力信号が含まれています。Digital In およびDigital Outコンポーネントは、モデルのタイムステップごとに1回更新される汎用I/Oを表します。これらの信号は、緊急 シャットダウン信号やセンサ障害アラームをモデル化するなど、リアルタイムシミュレーションと外部システム間でステータス 情報を伝達するためによく使用されます。これらのコンポーネントは、時間分解能がモデルの離散化時間に制限されるため、 PWM信号の検出や生成には適していません。100 µ sのスイッチング周期と1 µ sのモデルステップを持つPWM信号は、100 種類の異なるデューティサイクルのみを表すことができます。

PWM CaptureおよびPWM Outブロックは、FPGAの7.5nsの時間分解能を利用して、PWM分解能をシミュレーションタイム ステップから切り離します。PWM Outブロックは、RCPアプリケーションで使用され、RT Boxのデジタル出力で構成可能な PWM信号を生成します。PWM Outブロックへの入力は、モデルステップごとに更新される変調インデックスです。ただし、 モデルステップ間でPWM出力の立上がりエッジと立下がりエッジが発生する場合があります。同じ100 μ 秒のスイッチング 周期で、PWM Outコンポーネントは13,333種類の異なるデューティサイクルを生成できます。図4は、モデル ステップ間で PWM Out信号の立下がりエッジが発生している間、Digital Out信号がモデルステップ全体でラッチされる例を示しています。



正確なPWMサンプリングが必要なリアルタイムHILアプリケーションでは、PWM Captureブロックを使用します。PWM Capture ブロックは、FPGAサンプルレート7.5nsでデジタル入力をサンプリングします。次に、モデルのタイムステップにわたるすべての サンプルの平均が計算され、信号のオン時間の持続時間が0から1の間の値として決定されます。PWM Captureブロックは、 PLECSライブラリブラウザのパワー素子モジュールセクションにある"hybrid power module"と組み合わせて使用する必要 があります。前のモデルステップのPWM入力の時間平均値が、ハイブリッド電源モジュールの有効デューティサイクルとして 適用されます。デューティサイクル検出の精度は、モデルの時間ステップによって制限されなくなりました。図5は、Digital In コンポーネントと PWM Captureブロックが同じPWM入力を処理する方法の例を示しています。PWM Captureの結果は、 デジタル入力と比較した追加のデューティサイクル情報が含まれます。



6.1 デジタル出力を使用したPWM信号の生成

この演習では、デジタル出力信号を使用してPWM信号を生成します。オシロスコープの測定では、デジタル出力の状態が タイムステップごとに1回だけ更新されることが示されます。また、"パラメータのインライン化"機能を使用して、PLECSを 経由してRT Boxのパラメータを変更する方法も学習します。"パラメータのインライン化"は、生成されたコード内で調整可能 なパラメータを定義します。 デフォルトでは、すべてのパラメータは生成されたコード内で数値定数としてコード化されます。デフォルトの動作に対する 例外を指定できるため、再コンパイルせずに実行時にパラメータを変更できます。抵抗やインダクタンスなどの物理モデル 方程式に影響を与えるコンポーネントのパラメータは通常、変更した場合は方程式システム全体を再計算する必要がある ため、調整可能な状態に保つことはできません。

あなたのタスク:

 \mathbf{V}

- 1 新しいモデルを作成し、ライブラリブラウザからDigital Outコンポーネントを回路図に配置します。Digital Output channel を30に割り当てます。"三角波発生器"を追加して、5kHzの下降エッジののこぎり波を作成します。三角波発生器の最小値 と最大値を0と1に変更します。"関係演算子"を回路図に追加し、三角波キャリアの出力を"定数"値0.3と比較します。一定の デューティ比がのこぎり波キャリア以上の場合、関係演算子の出力はtrueになります。
- 2 Coderオプションウィンドウを開き、パラメーターのインライン化タブに移動します。0.3デューティ比を定義する"定数"を 回路図から例外のリストにドラッグアンドドロップします。
- **3** RT Box上で、スイッチング周期の1/20thまたは10e-6秒の離散化時間でモデルを実行します。外部モード経由でRT Box に接続します。
- 4 オシロスコープが必要:オシロスコープのプローブをデジタル出力30に接続し、周期とデューティサイクルが予想値と一致していることを確認します。デジタル出力を測定するには、デジタル出力ピンと信号グランドピンの間にスコープを接続する必要があります。RT Boxの前面パネルにあるデジタル出力コネクタでは、ピン36はデジタル出力30に対応し、ピン37は利用可能な信号グランド接続です。より詳細なピン配列仕様については、Web InterfaceのFront panelタブまたはRT Boxのマニュアルを参照してください[1]。より簡単にプローブできる場合は、他のデジタル出力(RT Box Interface Cardsのデジタル出力8~15など)を使用できることに注意してください。

メイン回路図で、"デューティ比"を0.3から0.32に変更します。



デジタル出力のデューティ比は変化しましたか?予想どおりの変化がありましたか?

出力デューティ比の変化が見られますが、結果のデューティサイクルは0.32より大きくなります。デジタルオンタイムは、予想値の64 µ sではなく、60 µ sから70 µ sに変化します。

デジタル出力デューティが変化するまで、デューティ比を0.01ずつ増加させ続けます。



デジタル出力のデューティ比はどの値で変化しますか? デジタル出力信号のオン時間が特定の値で変化する 理由を説明してください。



出力は、デューティ比が0.35より大きい場合にのみ変化します。離散化ステップサイズがスイッチング周期の1/20thの場合、出力信号の最大デューティサイクル分解能は0.05、つまり時間分解能は10 µ sになります。

5 *RT Box Interface Cardが必要*: Interface CardをRT Boxに接続します。デジタル出力30は、LaunchPad Interfaceおよび ControlCard Interfaceボード上のLEDに対応します。DO-30というラベルの付いたLEDが点灯していることに注目して ください。外部モード経由でRT Boxに接続し、デューティ比をより高い値または低い値に変更します。



デューティ比を変更すると LEDに何が起こりますか?

A LEDの明るさが変わります。

LEDが点灯しないようにデューティ比を設定します。



LEDをオフにするにはどのくらいの値が必要ですか?選択した離散化時間は、デジタル出力コンポーネントに 接続されたLEDの最小輝度にどのような影響を与えますか?

LEDをオフにするには負の値が必要です。三角のランプ波形は正確にゼロから始まるため、ゼロ値に対して"以上 "の比較が真になります。論理High出力は、1つのシミュレーション時間ステップの間ラッチされます。PWM信号 の分解能はLEDの最小輝度に直接関係しており、離散化ステップサイズに応じて変化します。

この段階では、モデルは参照モデルintroduction_rtbox_digital_1.plecsと同じになるはずです。

6.2 高忠実度の PWM信号生成

PWM Outコンポーネントを使用すると、RT Boxによって生成されるPWM信号の精度が向上します。この演習では、Digital OutコンポーネントとPWM Outコンポーネントによって生成されるデジタル出力を比較します。



あなたのタスク:

1 前の演習のモデルに"PWM Out"コンポーネントを追加します。PWMデューティを表す"定数"をPWM Outブロックに接続します。Carrier limitsを[0,1]に変更し、Carrier typeをSawtooth(のこぎり波)に変更し、PWM Outをデジタル出力 チャネル31に割り当てます。回路図については図6を参照してください。モデルを RT Boxにアップロードし、外部モード 経由で接続します。

図6: PWM信号を生成する2つの異なる方法の比較



- 2 *オシロスコープが必要*: オシロスコープでデジタル出力30と31を測定します。デジタル出力31は、RT Boxの前面パネル のピン18に対応します。"外部モード"に接続した後、"デューティ比"を0.3に変更し、2つの波形を比較します。デューティ比 を0.01ずつ増やしていくと、PWM Outブロックに関連付けられたデジタル出力の PWM分解能が大幅に高くなることが わかります。
- **3** *RT Box Interface Cardが必要*: Interface CardをRT Boxに接続します。デジタル出力31は、Interface Card上のLEDにも 接続されています。デューティ比を変更し、LED出力の変化を確認します。デューティ比を0.01に設定します。

⑦ デューティ比が0.01の場合、両方のLEDの明るさは同じになりますか?異なる場合は、その理由を説明してください。

Digital Outコンポーネントに関連付けられたLEDは、PWM Outコンポーネントに関連付けられたLEDよりも明るくなります。PWM分解能はLEDの最小輝度に直接関連しています。PWM Outコンポーネントは、Digital Outコンポーネントよりも高いPWM分解能を備えています。さらに、PWM Outコンポーネントの分解能は、シミュレーションのタイムステップと直接関連付けられていません。

この段階では、モデルは参照モデルintroduction_rtbox_digital_2.plecsと同じになるはずです。

6.3 Digital Inと高忠実度のPWM Capture

この演習では、同一のPWM入力刺激を使用して、Digital InとPWM Captureコンポーネントの動作を比較します。この演習 では、RT BoxのPWM出力をデジタル入力ピンに接続するためのループバック ケーブルが必要です。あるいは、この目的の ために外部PWMソースまたは信号発生器を使用することもできます。

Digital Inコンポーネントは、入力信号をモデルステップの直前のデジタル入力状態を表す0または1として記録し、PWM Captureは、図5に示すように、前のモデルステップでデジタル入力信号がアクティブだった時間の割合を表す0~1の値を 生成します。平均オン時間情報を"hybrid power modules"で使用すると、モデルの精度は完全にスイッチングされたモデル と比較して大幅に向上します。

あなたのタスク:

1 新しいPLECSモデルを作成し、図7に示されている回路を構築します。この回路は、チャネル0に割り当てられた"Digital In" コンポーネントと、チャネル1に割り当てられたPWM Captureコンポーネントで構成されます。

Digital Inブロックと PWM Captureブロックの出力は、"マルチプレクサ"コンポーネントに接続され、次にスコープに接続 されます。"PWM Out"ブロックはデジタル出力0と1 を使用し、他のすべての"PWM Out"のパラメータは10 kHzのCarrier frequency(キャリア周波数)と[-1,1]のCarrier limits(キャリア制限値)のデフォルト値のままになります。PWMデューティ を表す定数は、"信号セレクタ"経由で"PWM Out"ブロックに接続されるため、両方のPWM出力のデューティと設定は同じ になります。デューティサイクルの"定数"をパラメータのインライン化リストに追加します。

図7: Digital InとPWM Captureコンポーネントの比較



2 ループバックが必要: ループバックケーブルを使用して、RT Boxのデジタル出力をデジタル入力に接続します。モデルを RT Boxにアップロードし、外部モード経由で接続します。"Digital In"と"PWM Capture"コンポーネントの出力信号を比較 します。自動トリガを有効化では、ターゲットチャネルはターゲットレベル0.5でDigital In(channel 0)に設定します。 サンプル数は300に設定します。次に、デューティサイクルを0から0.1に調整し、スコープ信号がどのように変化するか を観察します。 (?)

"Digital In"と"PWM Capture"の結果を比較するとどうなるでしょうか? "Digital In"信号の幅は一定ですか、それとも 動作ポイントによって変化しますか? これは、降圧コンバータなどの単純なモデルのリアルタイムパフォーマンス にどのような影響を与えますか?

アDigital In"信号は常に0または1のいずれかです。"PWM Capture"ブロックは0~1の間の値です。デューティサイクルが0の場合、"PWM Capture"信号が変化しても、デジタル入力の動作は一定のままであるように見えます。これは、PWM生成とPWMサンプリングの非同期動作によるものです。

デューティサイクルを0.1に変更すると、この非同期動作により、"Digital In"のデューティサイクルが急速に変動 することが観察されます。ただし、RT Boxが生成する PWM信号に基づいて、デューティサイクルは一定であることが わかっています。"PWM Capture"信号の平均オン期間が増加することがわかります。理想的な降圧コンバータの 場合、出力電圧はデューティサイクルに比例します。"Digital In"コンポーネントを使用してPWM信号を検出すると デューティサイクルが変化し、結果、検出エラーにより出力電圧も変化します。

この段階では、モデルは参照モデルintroduction_rtbox_digital_3.plecsと同じになるはずです。

7 モデル構造とステップサイズ選択の概要

RT Boxの最終用途がHILであるかRCPであるかにかかわらず、PLECSモデルを個別のサブシステムに構造化することは、 リアルタイムワークフローの重要な部分です。RT Boxにデプロイされるモデルの部分は、1つのサブシステム内に含まれて いる必要があります。そのサブシステムの外部には、RT Boxに接続された物理ハードウェアが表現されます。これら2つの システム間のインタフェースは、アナログ、デジタル、PWM信号を含む必要なすべてのI/Oが定義されます。この構造により、 HILの場合はRT Boxに接続されたコントローラ、RCPの場合は外部に接続されたパワーステージを表すオフラインモデル を作成できます。

このモデル構造を使用すると、オフラインとリアルタイムのシミュレーション結果を直接ベンチマークして比較できます。 このアプローチのもう1つの大きな利点は、コード生成モードを使用してモデルの離散化の影響を評価できることです。 このモードでは、連続モデル表現と離散モデル表現の精度をベンチマークできます。

この演習では、単純なモデル構造の例と、コード生成モードを使用して離散化ステップサイズが単純な1次システムの動作 に与える影響を評価する方法を示します。

あなたのタスク:

1 新しいモデルを作成し、ライブラリブラウザのシステムブロック内にある"サブシステム"コンポーネントを回路図にドラッグ アンドドロップします。"サブシステム"をダブルクリックしてその内容を表示し、既存の"入力信号ポート"および"出力信号 ポート"コンポーネントを削除します。サブシステムに"Analog In"ブロックと"Analog Out"ブロックを追加します。RT Box のアナログ入力0とアナログ出力0を使用します。図8に示すように、G(s) = 10.023+1の形式で1次システムを実装する"伝達関数" コンポーネントを追加します。"伝達関数"の入力信号と出力信号を"PLECSスコープ"に接続します。





最上位レベルの回路図のサブシステムに、制御信号を接続できる新しいポートが追加されていることに注意してください。 新しいポートは、Target InportコンポーネントとTarget Outportコンポーネントによって作成され、これらのコンポーネント は、ライブラリオブジェクトのマスクの下を見ることで確認できます。"Analog In"および"Analog Out"コンポーネントの名前 は、親サブシステムのポート名に対応します。これらのポートは、リアルタイムシミュレーションとRT Boxに接続されたハード ウェア間のインタフェースを表します。"Analog In"コンポーネントを右クリックし、サブシステム。 サブシステムのモデル 表示を選択して、"Analog In"モデルのオフライン実装を確認します。RT Boxアナログ入力のスケーリングとオフセット パラメータがオフライン実装に反映されていることがわかります。

2 Analog Outポートに接続された最上位レベルの回路図に"PLECSスコープ"を追加します。周波数50Hzの"パルス発生器" を追加します。"パルス発生器"の出力をAnalog Inポートに接続します。その他のパラメータはすべてデフォルトのままに しておきます。最上位レベルの回路図は図9のようになります。

図9: Analog InとAnalog Outコンポーネントで作成したポートを表示するサブシステム



- 3 シミュレーションを実行し、PLECSスコープの現在のトレースを保持ボタンを使用してすべてのトレースを保持します。 トレースしたラベルを"Offline"とします。最上位レベルの回路図のスコープには、"Analog Out"ブロックのスケーリング値 とオフセット値が反映されていることに注意してください。
- 4 次に、コード生成用のサブシステムを有効にします。サブシステムを右クリックし、サブシステム → 実行の設定…に移動して、 コード生成機能の有効化チェックボックスをクリックします。サブシステムの境界線が太字になりました。Coderオプション ウィンドウから、左側のメニューのSubsystemに移動します。タスクタブで、離散化ステップサイズに1e-3秒と入力します。 ターゲットタブで、ターゲットドロップダウンメニューからGenericを選択します。ビルドボタンをクリックします。モデル を保存したディレクトリに、離散化されたサブシステムを表すCコードを含む新しいフォルダが作成されます。
- 5 サブシステムを右クリックし、サブシステム -> シミュレーションモード -> コード生成オプションを選択します。コード生成 モードを選択すると、サブシステムの境界線が図10に示すように破線の太枠に変わります。コード生成モードでは、PLECS はそのサブシステムを表す離散化されたCコードを使用します。



図10: コード生成モードのサブシステム(破線の太枠)

6 シミュレーションを実行し、新しいスコープトレースをすべて"CodeGen_1e-3"として保持します。サブシステム内のスコープ が更新されていないことに注目してください。最上位レベルの回路図でPLECSスコープを開き、システムの離散化の影響が 明らかになるまで波形を拡大します。図11にシミュレーションの出力を示します。



図11: "オフライン"と"コード生成"の結果の比較

7 コード生成の結果を、次の離散化ステップサイズと比較します: 1e-3秒、2e-3秒、10e-3秒、および25e-3秒。

- このシステムにはどのような離散化ステップサイズが適切でしょうか?離散化ステップサイズがどのくらいになると、 期待される結果からの大幅な逸脱が予想され始めるでしょうか?
- 離散化ステップサイズを1e-3秒と2e-3秒にすると、連続信号を適切に近似できます。10e-3秒のステップサイズは 入力信号の基本周期を示しますが、基準ソリューションからは大幅に逸脱します。25e-3秒の結果など、ナイキスト比 を超えるステップサイズの場合、許容できないパフォーマンスが予想されます。実際のHILおよびRCPシミュレーション では、再構成された信号に必要な精度、および離散化と信号サンプリングによって生じる許容可能な遅延は、アプリ ケーションごとに異なります。
- 8 シミュレーションモードを"通常"に戻し、ターゲットをRT Boxに更新することで、リアルタイムモデルに戻します。1e-3秒の離散化ステップサイズでモデルをRT Boxにデプロイし、外部モード経由で接続します。この段階では、サブシステム内のPLECSスコープは更新されますが、最上位レベルの回路図のスコープは更新されません。セクション5.2の演習と同様に、アナログ入力のピンに触れて、PLECSスコープの更新で波形を観察します。



アナログ入力ピンに触れることによって発生するノイズは、通常、AC主電源周波数に影響されます。これまでの 結果では、1e-3秒の離散化ステップサイズがこの1次システムに適していることが示されていました。結果として 得られるフィルタ処理された出力は、連続伝達関数の応答を適切に表しています。

この演習ではコード生成モードの簡単なアプリケーションを示しましたが、同じアプローチをより高度なモデルに拡張 することもできます。



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

8 結論

このチュートリアルでは、PLECS RT Boxで基本的なPLECSモデルを実行する方法を段階的に説明しました。RT Boxアーキ テクチャの理解から始めて、いくつかの重要なRT Boxライブラリ コンポーネントがどのように動作するか、その主要な用途 について学習しました。アナログI/O範囲のスケーリングと離散化に関する重要なトピックについて検討しました。さまざまな デジタル I/O実装が紹介され、FPGAの7.5nsの時間分解能を利用して、デジタル信号のパフォーマンスをモデルのステップ サイズから切り離す方法が強調されました。"外部モード"および"パラメータのインライン化"ツールを使用すると、RT Boxの 波形を観察してパラメータをリアルタイムで変更できるだけでなく、Web Interfaceから RT Boxの動作状態に関する詳細 情報を収集する方法も理解できるようになります。

コード生成モードを使用して、モデルの離散化の影響を観察し、RT Boxでモデルを実行する前にモデルのステップサイズが 適切かどうかを確認する方法を学習しました。モデルのリアルタイム部分を表すサブシステムを中心にモデルを構築する ことにより、単一のPLECSモデル内からオフライン、リアルタイム、およびコード生成でのシミュレーション結果を直接比較 できます。

これらのトピックを理解することで、HILとRCPアプリケーションの両方に対して独自のリアルタイムシミュレーションモデルを 開発できるようになります。

9 参照

 [1] RT Box User Manual, Plexim GmbH, Jan. 2024, Online: https://www.plexim.com/sites/default/files/rtboxmanual.pdf
日本語版は以下から:

https://adv-auto.co.jp/products/plexim/manual.html

 [2] PLECS User Manual, Plexim GmbH, Aug. 2024, Online: https://www.plexim.com/sites/default/files/plecsmanual.pdf
日本語版は以下から:

https://adv-auto.co.jp/products/plexim/manual.html

改訂履歷: Tutorial Version 1.0 初版

	+41 44 533 51 00 Plexim	ヽの連絡方法: Phone
	+41 44 533 51 01	Fax
	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	info@plexim.com	Email
	http://www.plexim.com	Web
Advai	ncing Automation AUTO	オートメーションへの連絡方法:
P	+81 3 5282 7047	Phone
	+81 3 5282 0808	Fax
	ADVAN AUTOMATION CO.,LTE 1-9-5 Uchikanda, Chiyoda-ku Tokyo, 101-0047 Japan	Mail
@	plecs_adva@adv-auto.co.jp	Email

RT Box Tutorial

 $^{\odot}$ 2002–2024 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。