

Embedded Code Generation *DEMO MODEL*

H-Bridge Converter

Hブリッジコンバータ

- TI C2000 MCU用の組み込みコード生成機能を備えたHブリッジコンバータの電流制御 -

Last updated in C2000 TSP 1.9.1

1 はじめに

このデモモデルは、誘導負荷に電力を供給する電流制御Hブリッジ回路を特徴としています。Texas Instruments (TI) C2000 Microcontrollers(MCU)を使用したPLECS Embedded Coderの一般的なワークフローについて説明します。PLECS RT Boxと組み合わせることで、MCUの性能を直接検証できます。モデルは"Plant"と"Controller"に分かれています。このプラントには誘導負荷を備えたHブリッジコンバータが含まれており、コントローラは比例積分(Proportional-Integral: PI)電流制御方式を採用しています。各サブシステムは個別のリアルタイムターゲットにデプロイされます。Controllerサブシステム内の制御ロジックを構築し、TI C2000 MCUに書き込まれます。Plantサブシステムは、生成された組み込みコードのハードウェアインザループ(Hardware-in-the-loop: HIL)テスト用にPLECS RT Boxにデプロイされます。[次章](#)では、モデルの簡単な説明と、それをシミュレートする方法について説明します。

デモモデルは、外部RL負荷に接続されたBOOSTXL-DRV8305EVM BoosterPack [\[1\]](#)の2相で構成される電源回路のハードウェアプロトタイプで検証されます。

注意 このモデルには、以下の方法で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:

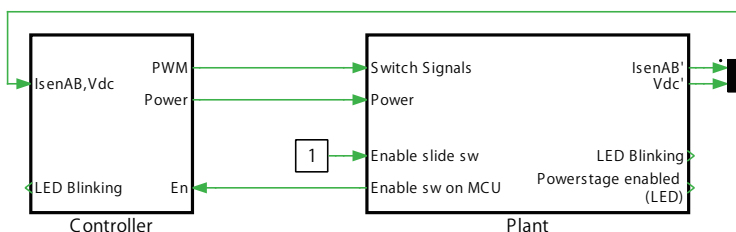
PLECS Standalone: シミュレーションメニュー -> シミュレーション・パラメータ... -> 初期化

PLECS Blockset: Simulinkモデルウィンドウで右クリック -> モデルプロパティ -> コールバック -> InitFcn*

2 モデル

最上位レベルの回路図には、[図1](#)に示すように、コントローラとプラントモデルを表す2つのサブシステムが含まれています。両方のサブシステムは、[編集](#)メニューからサブシステム -> 実行の設定...からコード生成機能の有効化にチェックすると、コード生成が有効になります。この手順は、RT Boxのモデルコードを生成するために必要です。

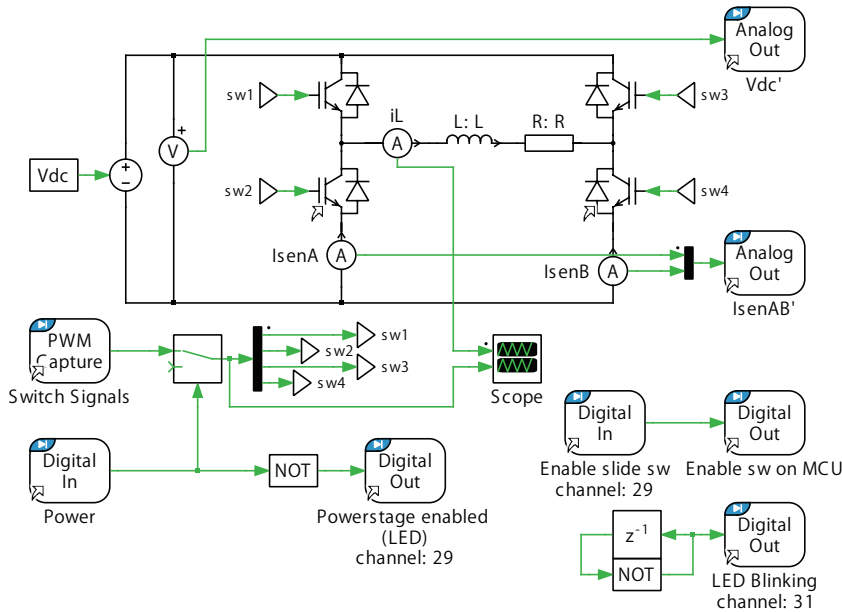
図1: プラントとコントローラサブシステムのあるトップレベルの回路図



2.1 電源回路

[図2](#)に示す電源回路には、 $V_{dc} = 24V$ のDC電源電圧が供給されます。Hブリッジは、誘導負荷に電力を供給する2つのIGBTハーフブリッジパワー素子モジュールコンポーネントで構成されています。パルス幅変調(pulse-width modulation: PWM)スイッチング信号は、PLECS RT BoxライブラリのPWM Captureブロックから取得します。パワー素子モジュールのコンポーネントとPWM信号のサブサイクル平均化の詳細については、[\[2\]](#)で説明されています。DC入力電圧と出力インダクタ電流の測定値は、PLECS RT BoxライブラリのAnalog Outブロックに接続します。プラントサブシステムの離散化ステップサイズは $2 \mu s$ に設定しています。

図2: 誘導負荷を備えたHブリッジの電源回路



アナログ出力のスケーリング

PLECS RT Boxターゲットのアナログ入力および出力電圧の範囲は、**Coderオプション...**ウィンドウの**ターゲットタブ**(Analog input voltage rangeおよびAnalog output voltage range)で -10Vから10Vで設定できます。

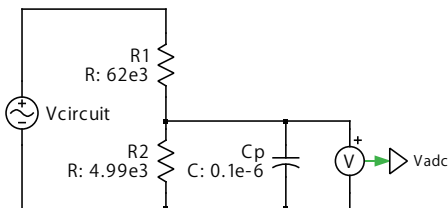
ただし、一般的な HILシミュレーションでは、RT Boxからエクスポートしたアナログ出力値を組み込みコントローラに供給するため、値は、MCUのアナログ/デジタルコンバータ(Analog-to-digital converter: ADC)の電圧要件を満たすように0V～3.3Vの範囲内にスケーリングおよびオフセットされます。

プラント モデルは第4章で説明する電源回路プロトタイプに基づいているため、入力電圧とシャント電流はBOOSTXL-DRV8305EVM BoosterPackの電圧および電流検出回路に応じて調整されます。

電圧センシング

[1]で説明したものと同様の電圧センシング回路を図3に示します。

図3: 電圧センシングの回路図



以下の式は電圧センシング回路の応答を示しています。キャパシタ C_p を選択してセンシング回路に極を作成し、検出した電圧のスイッチング周波数リップルを減衰させることができます。極周波数は以下 f_p と表記します:

$$\frac{V_{adc}}{V_{circuit}} = \frac{R_2}{(R_1 + R_2) + sR_1R_2C_p}, C_p = \frac{R_1 + R_2}{2\pi f_p R_1 R_2}$$

電圧は、以下のゲインでスケールリングします:

$$V_{m.K} = \frac{R_2}{R_1 + R_2} = \frac{4990}{62000 + 4990}$$

電流センシング

[1]で説明したように、ローサイドシャント電流は両方のハーフブリッジで検出されます。これらの電流は、 $I_{m.K}$ のスケールリング係数で検知および増幅し、電流シャントアンプを使用して、 $I_{m.O}$ のオフセット係数でオフセットします。

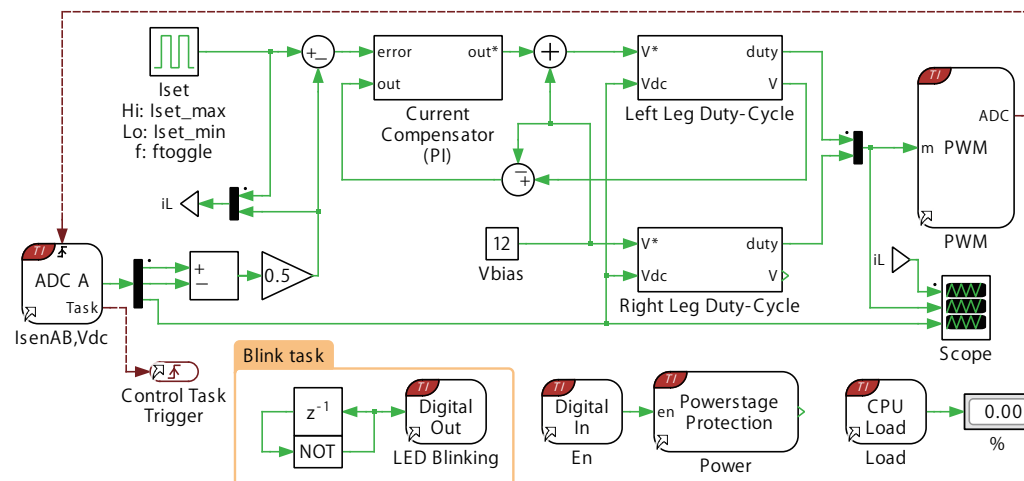
$$I_{m.K} = 0.007 * 10$$

$$I_{m.O} = \frac{3.3}{2}$$

2.2 コントローラ

Controllerサブシステムを図4に示します。"Right Leg Duty-Cycle"サブシステムは、検知した入力電圧 V_{dc} の変動を考慮しながら、右レッグHブリッジ出力で平均12Vを維持するために必要なデューティ比を決定します。

図4: Hブリッジ回路のコントローラ



"Left Leg Duty-Cycle"サブシステムは、図5に示すように、PIコントローラに基づいてHブリッジの左レッグの変調指数を決定します。検出したインダクタ電流は、-3Aから3A間に切り替える設定点と比較します。この誤差は、アンチワインドアップロジックを備えたデジタルPIコントローラによる電流補償に使用します。

本章で説明しているコントローラ開発に関連する概念のいくつかは、参考文献[3]に基づいています。

プラント伝達関数

PIコントローラのゲインパラメータを設定するには、プラント伝達関数 $P(s)$ が必要です。 $P(s)$ は、インダクタ(L)と抵抗器(R)の両端の電圧 V_{RL} (入力変数)の変化と、インダクタ電流 I_L (出力変数)の応答を関連付けます:

$$P(s) = \frac{I_L}{V_{RL}} = \frac{1/R}{1 + sL/R} = \frac{K_1}{1 + sT_1}, \text{ ただし } K_1 := 1/R \text{ および } T_1 := L/R$$

時間定数

複雑なシステムの次数を減らすには、主要時定数と小時定数を区別すると役立つことがよくあります。通常、主要時定数はプラント伝達関数の一部ですが、制御システムはいくつかの小時定数(センサ、アクチュエータ、サンプリング、計算遅延、高速内部制御ループなど)を導入します。

最大の小時定数が最小の主要時定数の少なくとも4分の1である場合、すなわち、

$$\min(T_{\text{dominant}}) \geq 4 \cdot \max(T_{\text{small}})$$

それにより、システムのモデリングにおいて重要な簡略化が可能になり、システム内の全ての小時定数の合計である一つの小さな等価時定数を定義することができます。

コントローラのサンプリング周波数($1/T_s$)がスイッチング周波数($1/T_{\text{sw}}$)と同じ場合、この特定の実装におけるHブリッジコンバータモデルに存在する小時定数は次のようになります:

- 制御計算の小時定数 T_{calc} は $\frac{1}{2} T_{\text{sw}}$
- PWM出力生成の小時定数 T_{pwm} は $\frac{1}{2} T_{\text{sw}}$
- 連続パラメータを離散パラメータに変換する小時定数 T_{sh} は $\frac{1}{2} T_{\text{sw}}$ (以降で説明)

等価の小時定数 T_{Σ} は:

$$T_{\Sigma} = T_{\text{calc}} + T_{\text{pwm}} + T_{\text{sh}}$$

$$D_{\Sigma}(s) = \frac{1}{1 + sT_{\Sigma}}$$

制御パラメータの計算

PIコントローラの制御パラメータ(K_p および K_i)は、MOC(Magnitude Optimum Criterion)を使用して計算します。システムの開ループ伝達関数 $H_{\text{OL}}(s)$ は、コントローラ、プラント、および時間遅延からの伝達関数の積によって与えられます:

$$H_{\text{OL}}(s) = \frac{1 + sT_n}{sT_i} \cdot \frac{K_1}{1 + sT_1} \cdot \frac{1}{1 + sT_{\Sigma}}, \quad \text{where } K_p = \frac{T_n}{T_i} \text{ and } K_i = \frac{1}{T_i}$$

制御パラメータ T_n は、プラント伝達関数の極がキャンセルされるように(すなわち、 $T_n = T_1$ になるよう)選択されます。極ゼロのキャンセルの後、閉ループ伝達関数は2次システムを表します。残りのパラメータ T_i は、2次システムの減衰係数(ζ)を $1/\sqrt{2} = \zeta$ に設定して計算し、 $T_i = 2K_1T_{\Sigma}$ となります。

連続制御パラメータから離散制御への変換

PI制御器のデジタル実装は次のように記述できます:

$$y[k] = K_{pz} e[k] + K_{iz} \sum_{i=0}^k e[i]$$

サンプル時間(T_s)がPI制御器の最小時定数よりも十分に小さい場合、つまり、

$$T_s \leq \frac{T_n}{2}$$

したがって、合計 $\sum_{i=0}^k e[i]$ は等価な伝達関数で近似できます:

$$\sum e[i] \approx \frac{1 + sT_s/2}{sT_s}$$

これにより、連続PI制御器と離散PI制御器の間に次の関係が得られます:

$$K_{pz} = \frac{T_n - T_s/2}{T_i}$$

$$K_{iz} = \frac{T_s}{T_i}$$

ただし、デジタル制御器の入力でのサンプル&ホールドにより追加の遅延が発生するため、連続時間設計時にこれを考慮する必要があります:

$$SH(s) = \frac{1}{1 + sT_s/2}$$

アンチwindアップ

制御システムのアクチュエータが限界に達するか飽和状態になると、プラントは制御出力の変化に応答なくなり、制御システムは制御誤差を低減できなくなります。このような状況では、コントローラ内の積分器が極端な値に達するwindアップの発生を防ぐことが重要です。windアップを防ぐ1つの方法は、[図5](#)の積分器*I*を飽和出力*y'*と同じ値に保つことです。これにより、誤差が(最終的に)ゼロを下回った場合、積分器出力(*i*)の変化がコントローラの飽和によって制限されることがなくなります。

飽和状態:

$$I = y'$$

$$i = 0$$

したがって、アンチwindアップ補正ゲイン(K_c)は次のように計算できます:

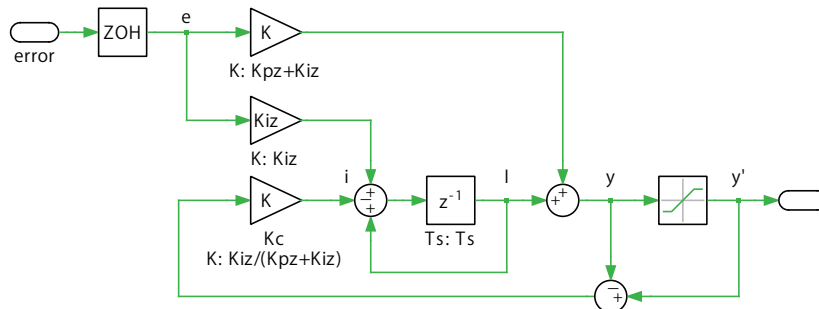
$$i = K_{iz} \cdot e - K_c(I + e(K_{pz} + K_{iz}) - y') = 0$$

$$i = K_{iz} \cdot e - K_c(y' + e(K_{pz} + K_{iz}) - y') = 0$$

$$K_c = \frac{K_{iz}}{K_{pz} + K_{iz}}$$

アンチwindアップを含むPIコントローラの全体構造を[図5](#)に示します。

図5: アンチwindアップ機能付きPIコントローラ



TI C2000 Targetライブラリコンポーネントの構成

[図4](#)のコントローラには、TI C2000Targetライブラリのコンポーネントがいくつか含まれています。ゲート信号はPWMコンポーネントによって生成され、左右レグのデューティ比がPWMブロックに入力されます。ゲート信号を生成するためのデューティ比は[0, 1]の範囲にあります。PWMブロックのパラメータウィンドウの**全般**タブから**Carrier type**、**Carrier frequency**、**Output**タブから**Dead time**パラメータを設定できます。ePWMモジュールへの入力相補的なPWMペアを生成することに注意してください。つまり、デッドタイムを除いて、ePWMxAとePWMxBは反対の極性になります。

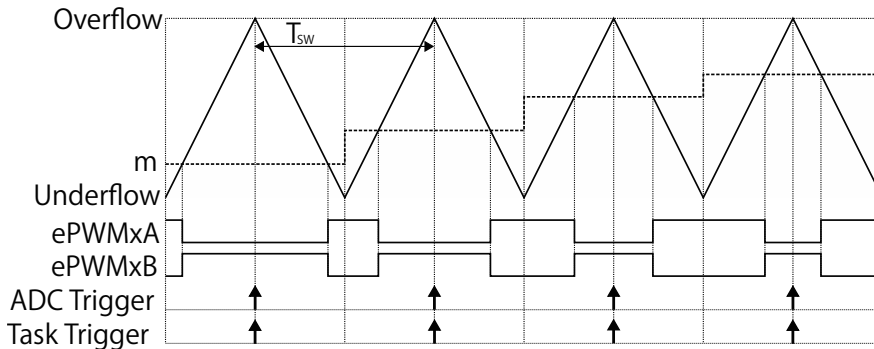
入力電圧とインダクタ電流の測定値は、TI C2000 TargetコンポーネントライブラリのADCブロックからモデル環境に導入されます。検出されたアナログ電圧を制御アルゴリズムに使用する物理単位の値に変換するために、ADCブロックのパラメータウィンドウを介して各チャンネルにスケーリング係数とオフセット係数が提供されます。**ADC unit**と**Analog input channel**パラメータは、異なるMCUの利用可能なリソースに応じて変更できます。

このモデルでは、PWM発生器、ADC、および制御タスクトリガ間のトリガ信号を接続することで、組み込みアプリケーションの割り込みシーケンスが明示的に定義されます。トリガ信号は赤い破線で表示しています。PWMブロックのパラメータウィンドウの**Events**タブで、**ADC Trigger**パラメータが**Overflow**として設定しています。これは、このブロックに関連付けられた最初のePWMモジュールは、キャリア値が最大値に達するたびにADCの変換開始信号を生成します。そして、ADCブロックの**Trigger source**パラメータを**Show trigger port**に設定します。次に、ADCブロックのトリガポートをPWMブロックからのADCトリガに接続します。

制御タスクは、ADCモジュールでの最後の変換後に実行されます。これは、ADCのタスク出力をTI C2000 Targetコンポーネントライブラリの**制御タスクトリガ**ブロックに接続することで構成されます。

ADCとタスクトリガの方式を図6に示します。このモデルでは、ePWMxBがHighのときに下側のスイッチが閉じます。ADCは、ローサイドスイッチに電流が流れている場合にのみシャント電流を測定します。

図6: PWMキャリア、ADCおよびタスクトリガ方式



実行中にPWM信号を有効または無効にするには、RT Box LaunchPad Interfaceボード上のDIPスイッチ"DI-29"を使用します。この入力信号"DI-29"は、"Plant"サブシステムの"Enable slide sw"というラベルのDigital inブロックに接続され、その後、RT Box LaunchPad Interfaceボードを介して"Controller"サブシステムのPowerstage Protectionブロックの入力としてルーティングされます。Powerstage Protectionブロックは、ターゲットMCU上のすべてのPWM出力を有効化または無効化するための有限状態マシンを実装します。ロジックがLowからHighに遷移するとPWM出力を有効化し、HighからLowに遷移すると無効化します。詳細については、このブロックの**ヘルプ**セクションを参照してください。

パワー ステージが有効になると、**Powerstage Protection**ブロックの**Powerstage enable GPIO number**で構成されたデジタル出力が切り替わります。この信号は、"Plant"サブシステムの"Power"というラベルのDigital Inブロックに接続されています。これにより、キャプチャされたPWM信号が、ゲートドライバ回路をモデル化するインバータブリッジのゲートに渡されるようになります。LaunchPad Interfaceボード上の赤色LED "DO-29"が点灯し、スイッチング信号がゲートドライバに接続されていることが視覚的に示されます。

3 シミュレーション

このデモ モデルのシミュレーションをコンピュータ上でオフライン モードで実行するだけでなく、"Controller"サブシステムをTI C2000 MCUのターゲット固有のコードに直接変換することもできます。すべてのペリフェラルブロック(ADC、PWMなど)

のデフォルトのI/O構成は、TI 280039C[4]、TI280049C[5]、TI28069[6]、TI28379D[8]、TI28P550SJ9[9]、TI28P650DK9[10] LaunchPads 28388D [13] controlCARDをサポートしています。

さらに、デモモデルでは、TI 28377S [7] LaunchPad、TI 280039C [11]、TI 28379D [12] controlCARDのコード生成が可能です。この設定を行うには、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ー... -> 初期化タブからモデル初期化コマンドウィンドウに移動し、board_typeの値を変更して、目的のボードを選択します。また、Coderオプションウィンドウで対応するターゲットとBoardタイプを適切に構成する必要があります。

以下の手順に従って、"Controller"サブシステムをTI MCUにアップロードします。

- USBケーブルでMCUをホストコンピューターに接続します。
- **Coder** -> **Coderオプション...**ウィンドウのシステムリストから、"Controller"を選択します。
- 次に、**ターゲット**タブで、ドロップダウンメニューから適切なターゲットを選択します。そして、**General**サブタブで、目的の**Build type**を選択します。
- そして、PLECSからMCUを直接**Build type**で選択し、**Build configuration**でFlash or Run from RAMを選択して、MCUフラッシュメモリまたはRAMにプログラムし、**Board type**でLaunchPadを選択して、**ビルド**をクリックします。

正しくプログラムされていれば、モデル初期化コマンドにリストされているGPIO "DO_DSP_LED"に対応するLEDが点滅します。

Code Composer Studioに精通している上級ユーザ向けには、Generate code into CCS projectオプションがあります。CCSプロジェクトから適切なcgフォルダを見つけ(手順については[14]を参照)、そのパスを**CCS project directory**フィールドに入力して**ビルド**をクリックします。"Controller"サブシステムのコードが自動的に生成されます。そして、通常のCCSプロジェクトとしてプロジェクトのビルドとデバッグに進みます。

注意 RT Box LaunchPad Interfaceボードを使用する場合は、シミュレーション中はRSTジャンパが開いていることを確認してください。

プログラムしたMCUを使用して実際のパワー ステージを制御する前に、まずPLECS RT Boxを使用してコントローラの動作を検証し、ハードウェアインザループ(hardware-in-the-loop: HIL)テストを実行することを強くお勧めします。一般的なハードウェア構成を図7に示します。ここでは、評価ボードであるTI 28069 LaunchPad(赤いボード)が、**RT Box LaunchPad Interface**(緑のボード)を介してRT Boxに接続されています。

図7: HIL検証のハードウェアセットアップ



RT Boxでリアルタイムモデルを実行するには、以下の手順に従ってください。RT Boxでビルドする前に、適切なTI MCUで"Controller"サブシステムがビルド済みであることを確認してください。

- **Coder** -> **Coderオプション**ウィンドウの**システム**リストから、"Plant"を選択します。**ターゲット**タブをクリックし、ターゲットデバイスを選択します。**ビルド**をクリックして、モデルをターゲットのRT Boxにデプロイします。
- モデルがアップロードされたら、**Coderオプション...**ウィンドウの**外部モード**タブから RT Boxに接続し、**自動トリガを有効化**にチェックして、テスト結果をリアルタイムで観察します。

正しくプログラムされていれば、RT Box LaunchPad Interfaceボードの"DO-31"に対応するLEDが点滅します。

2.2の最後で説明しているように、RT Box LaunchPad Interfaceボード上のスイッチ"DI-29"をLowからHighに切り替えて、MCUを有効にします。パワーステージが有効になると、LaunchPad Interfaceボードの"DO-29"に対応するLEDが点灯します。"Plant"サブシステム内のPLECSスコープのリアルタイム波形を観察します。

注意 この段階で、RT Box LaunchPad Interfaceボード上の"DO-29"に対応するLEDがオンになっていることを確認します。

RT Box LaunchPad Interfaceボードのスイッチ"DI-29"をHighからLowに切り替えると、すべてのゲート信号が無効になります。LaunchPad Interfaceボードの"DO-29"はオフになります。"DI-29"をHighに戻すと、PWM出力が再び有効になります。

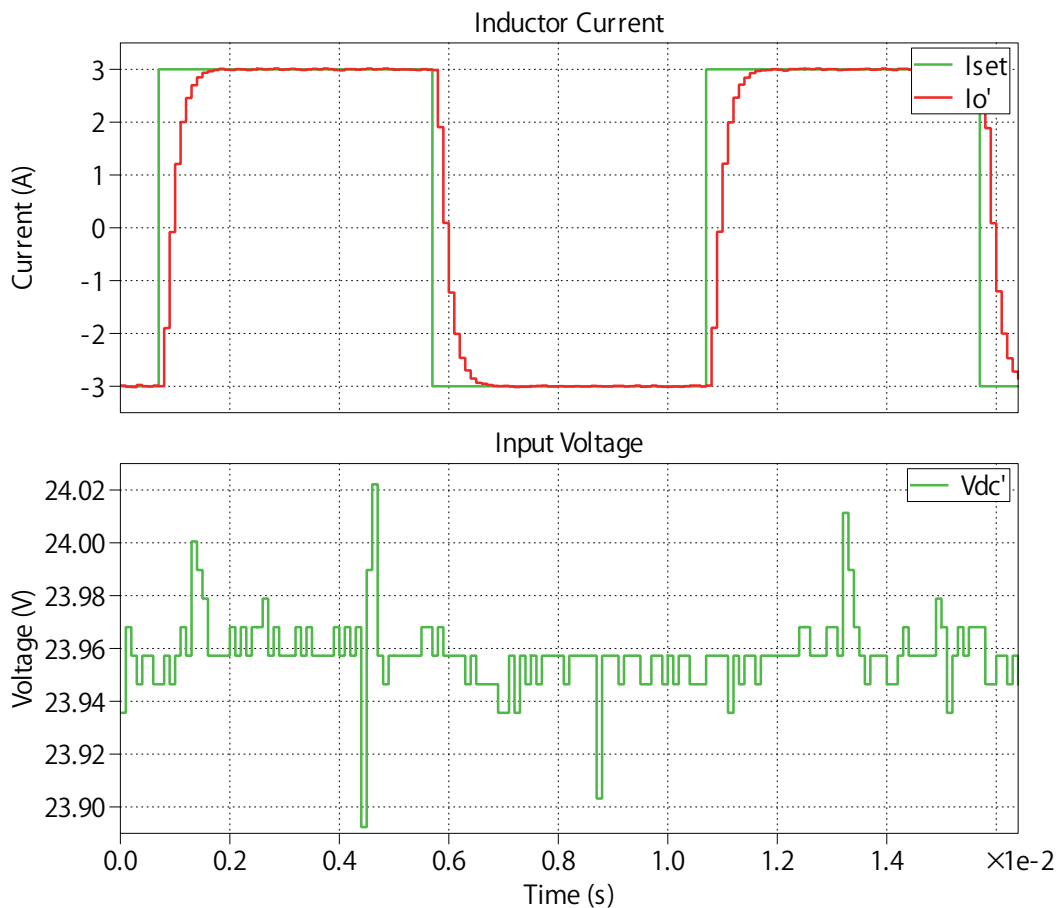
MCU内の制御プログラムのパラメータを調整し、中間値を観察するには、以下の手順に従ってTI MCUの外部モードに接続します。

- まず、PLECS RT Boxの"Plant"サブシステムの**外部モード**に接続されている場合、"切断"します。
- そして、"**Coder** -> **Coderオプション...**"ウィンドウの左側にある**システム**リストから、"Controller"を選択します。
- 次に、**外部モード**タブから適切な**ターゲットデバイス**を選択し、**接続**をクリックします。
- そして、**自動トリガを有効化**にチェックして、"Controller"サブシステム内のPLECSスコープでテスト結果を確認します。

インダクタ電流の測定値は、"Plant"と"Controller"サブシステム内のPLECSスコープを使用して表示できます。インダクタ電流リファレンス値は、"Controller"サブシステムの"Iset"(パルス発生器)コンポーネントを使用して、-3Aと3A間で切り替えます。これらのリファレンス値は、モデルを構築する前に、**Coderオプション**ウィンドウの**パラメータのインライン化**タブにある**例外**リストに"Iset"コンポーネントが追加されているため、リアルタイムで即座に変更できます。

インダクタ電流のリアルタイムステップ応答を図8に示します。

図8: PLECS RT Box 1を使用したインダクタ電流と入力電圧のリアルタイム測定



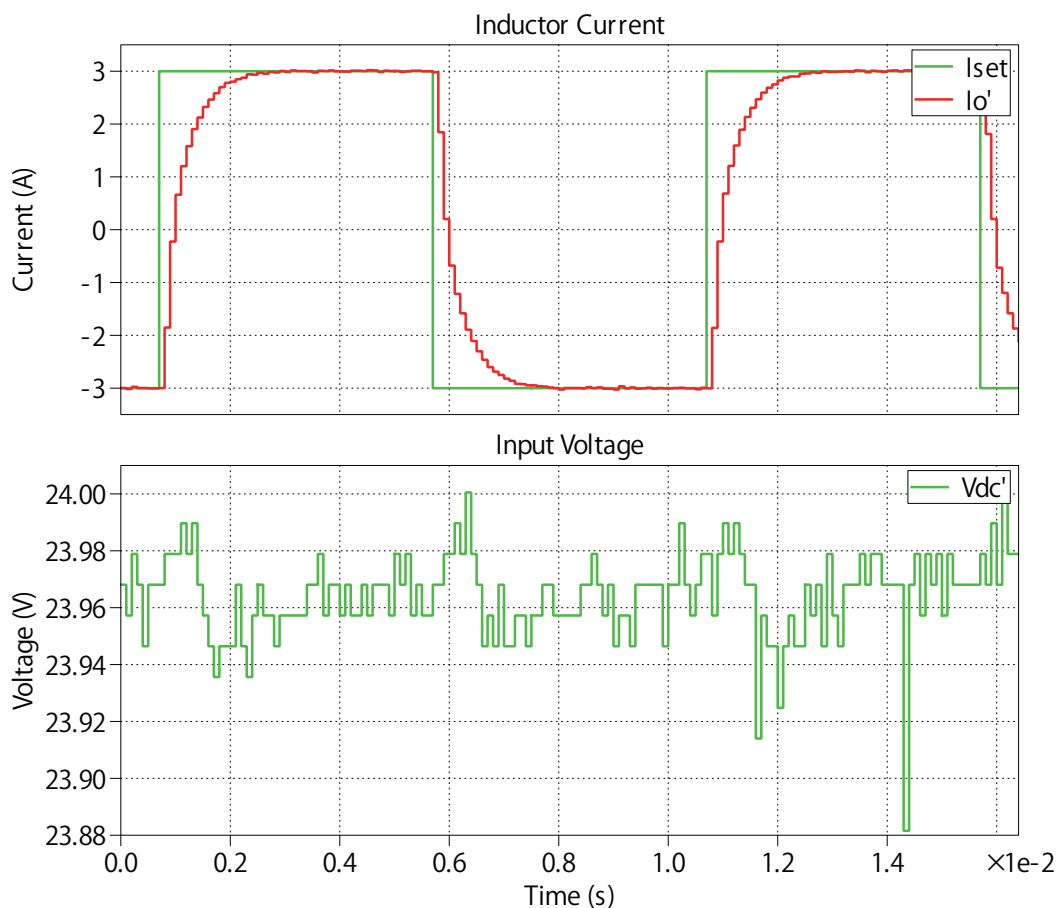
外部モードタブから、目的の**Trigger channel**のトリガコントロールを設定できます。

4 電源回路プロトタイプ

回路の検証には、BOOSTXL-DRV8305EVM BoosterPack [1]の2相を外部RL負荷に接続した電源回路のハードウェアプロトタイプを使用します。28069 LaunchPadのデモモデルにリストされているGPIO は、BoosterPackと完全に互換性があります。

第3章で説明したように、コードをデプロイして28069 LaunchPadターゲットに接続した後、図9に示すように、“Controller”サブシステムのPLECSスコープを使用して電流と電圧の測定値を表示します。

図9: 電源回路プロトタイプのインダクタ電流と入力電圧の測定



PLECS RT Boxを使用したリアルタイムシミュレーション結果が、ハードウェアプロトタイプの結果と一致していることに注目してください。

5 まとめ

このモデルは、TI C2000 MCU用の組み込みコード生成をサポートする、離散電流コントローラを備えたHブリッジコンバータを示しています。オフラインモードとリアルタイムモードの両方で実行できます。このモデルは電源回路のプロトタイプでも検証しています。このモデルは、リアルタイムで変更できる電流コントローラリファレンスを使用した**パラメータのインライン化**機能も示しています。

6 参考文献

- [1] TI DRV8305N 3-Phase Motor Drive BoosterPack Evaluation Module
URL: <http://www.ti.com/tool/BOOSTXL-DRV8305EVM>.
- [2] J. Allmeling, and N. Felderer, "Sub cycle average models with integrated diodes for real-time simulation of power converters," *IEEE Southern Power Electronics Conference (SPEC), 2017*.
- [3] Conception de Systèmes automatiques, Hansruedi Bühler, Presses Polytechniques Romandes, Lausanne 1988, ISBN 2-88074-149-1
- [4] TI C2000 F280039C LaunchPad development kit,
URL: <http://www.ti.com/tool/LAUNCHXL-F280039C>.

- [5] TI C2000 Piccolo MCU F280049C LaunchPad development kit,
URL: <http://www.ti.com/tool/LAUNCHXL-F280049C>.
- [6] TI C2000 Piccolo MCU F28069M LaunchPad development kit,
URL: <http://www.ti.com/tool/LAUNCHXL-F28069M>.
- [7] TI C2000 Delfino MCU F28377S LaunchPad development kit,
URL: <https://www.ti.com/lit/pdf/sprui25>.
- [8] TI C2000 Delfino MCU F28379D LaunchPad development kit,
URL: <http://www.ti.com/tool/LAUNCHXL-F28379D>.
- [9] TI C2000 F28P550SJ9 LaunchPad development kit,
URL: <https://www.ti.com/tool/LAUNCHXL-F28P55X>.
- [10] TI C2000 F28P650DK9 LaunchPad development kit,
URL: <https://www.ti.com/tool/LAUNCHXL-F28P65X>.
- [11] TI C2000 F280039C controlCARD evaluation module,
URL: <https://www.ti.com/tool/TMDSCNCD280039C>.
- [12] TI C2000 F28379D controlCARD development kit,
URL: <https://www.ti.com/tool/TMDSCNCD28379D>.
- [13] TI C2000 F28388D controlCARD evaluation module,
URL: <https://www.ti.com/tool/TMDSCNCD28388D>.
- [14] PLECS TI C2000 Target Support User Manual,
URL: <https://www.plexim.com/download/documentation>.
日本語版: <https://adv-auto.co.jp/products/plexim/manual.html>

改訂履歴:

C2000 TSP 1.1	初版
C2000 TSP 1.3.1	2つのIGBTハーブブリッジパワー素子モジュールのアサーションをオンにする
C2000 TSP 1.4.5	Webリンクを更新
C2000 TSP 1.5.1	コントローラでの倍精度演算の使用を最小限に抑え、28388Dおよび28379D controlCARDターゲットのサポートを追
C2000 TSP 1.6.1	280039C LaunchPadおよびcontrolCARDターゲットのサポートと自動ピン選択を追加
C2000 TSP 1.9.1	28P550SJ9 LaunchPadターゲットのサポートを追加

 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

Advancing Automation
 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 6285 0250 Fax

✉ ADVAN AUTOMATION CO.,LTD Mail

1-9-5 Uchikanda, Chiyoda-ku
Tokyo, 101-0047
Japan

@ info-advan@adv-auto.co.jp Email

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

Embedded Code Generation Demo Model

© 2002–2024 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。