

# Embedded Code Generation *DEMO MODEL*

## Sensorless Field-Oriented Control of PMSM

PMSMのセンサレスフィールド指向制御

Last updated in STM32 TSP 1.4.1

# 1 はじめに

このSTM32デモモデルは、正弦波逆起電力を持つ三相のブラシレスDCモータ(brushless DC motor: BLDC)を備えたドライブシステムを搭載しています。以降では、パワーステージと制御の実装について詳しく説明します。以下に、このデモモデルの主な特長を示します。

- ここでは、STMicroelectronics社のSTM32G4xまたはSTM32F3xマイクロコントローラ(Micro Controller Unit: MCU)を使用した組み込みターゲット向けPLECS Coderの一般的なワークフローについて説明します。
- センサレスベクトル制御(Field-oriented control: FOC)戦略のためのロータ位置や速度観察の実装例を示します。
- 内部にd軸およびq軸の電流コントローラ、外部に速度コントローラを備えた、閉ループカスケードコントローラを実装します。
- このモデルは、"Plant"と"Controller"と呼ばれる2つのサブシステムに分かれています。これにより、同じモデルを以下の用途に使用できます:
  - PLECSによるオフラインシミュレーション
  - PLECS RT Boxを使用したハードウェアインザループ(Hardware-in-the-loop: HIL)テスト
  - STMicroelectronics社のP-NUCLEO-IHM03 motor control packを使用したラピッド制御プロトタイピング(Rapid control prototyping: RCP)[\[3\]](#)。

次章では、モデルの説明と、モデルをシミュレートする方法、および制御コードをSTM32ターゲットにデプロイする方法について説明します。

## 1.1 要求仕様

このデモモデルを実行するには、次の製品が必要です(www.plexim.comから入手可能):

- バージョン4.7.4以降のPLECS [Blockset](#)または[Standalone](#)
- [PLECS Coder](#)ライセンス
- バージョン1.4.1以降の[STM32 Target Support Package](#)
- [RT Box Target Support Package](#)

RCPの場合:

- [P-NUCLEO-IHM03 STM32 Motor Control Nucleo Pack](#)が1セット

HILの場合:

- [PLECS RT Box](#)が1台
- [NUCLEO-G431RB STM32 Nucleo](#)ボードが1枚
- [RT Box NUCLEO Interface](#)ボードが1枚

Plantモデルは、RT Boxのすべてのバージョンで実行できます。また、ControllerはNUCLEO-G474REまたはNUCLEO-F303REボード上で実行することも可能です。

---

**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

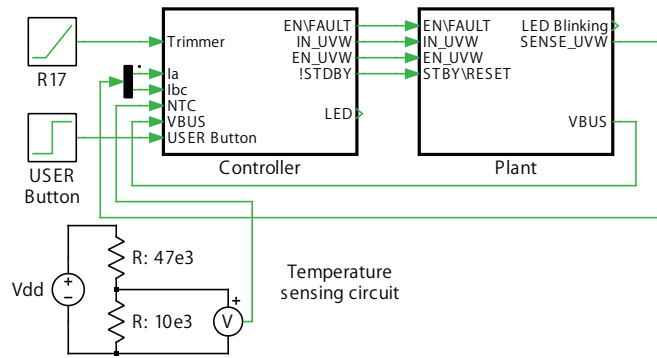
*PLECS Blockset:* Simulinkモデルウィンドウで右クリック -> モデル プロパティ -> コールバック -> InitFcn\*

---

## 2 モデル

[図1](#)に示すように、トップレベルの回路図には、コントローラとプラントモデルを表す2つの独立したサブシステムが含まれています。どちらのサブシステムも、**サブシステム** -> **実行の設定...**コンテキストメニューからコード生成が有効になっています。この手順は、PLECS Coderからサブシステムのモデルコードを生成するために必要です。

図1: デモモデルのトップレベルの回路図



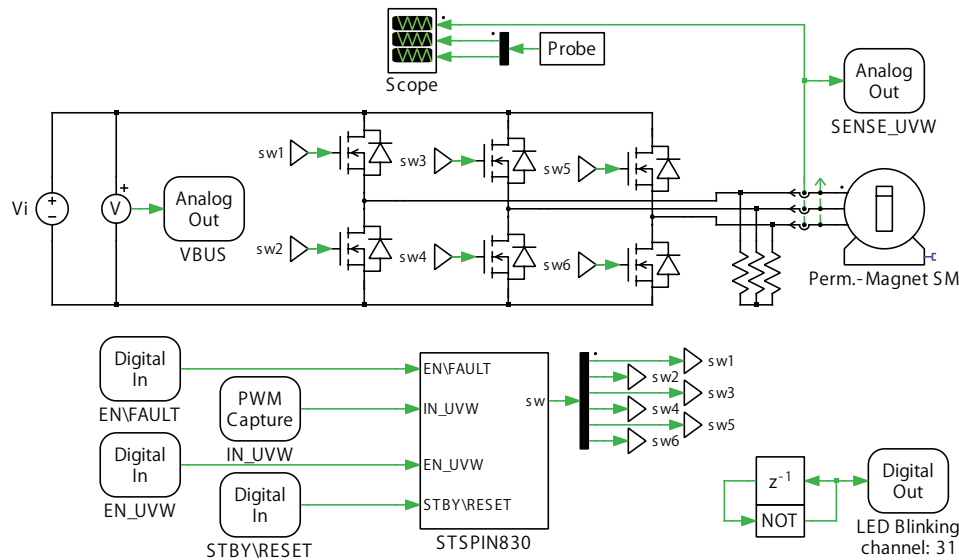
## 2.1 Plant

図2の電源回路は、永久磁石同期機(permanent magnet synchronous machine: PMSM)と三相フルブリッジ電圧源インバータ(voltage source inverter: VSI)を含み、 $V_{dc} = 12V$ の直流電源電圧( $V_i$ )で供給します。VSIは、3つのMOSFET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しています。

"STSPIN830"というラベルのサブシステムは、STSPIN830三相モータドライバの基本機能をモデル化しています[4]。

6パルスのPWMスイッチング信号は、PLECS RT BoxライブラリのPWM Captureブロックで取得します。パワーモジュールの構成要素とPWM信号のサブサイクル平均化に関する詳細は[6]に記載されています。DC入力電圧とAC出力ステータ電流の測定値は、PLECS RT BoxライブラリのAnalog Outブロックに接続されています。プラントサブシステムの離散化ステップサイズは $3 \mu s$ に設定されています。

図2: PMSMドライブシステムの電源回路



### モータパラメータ

モータパラメータは、STMicroelectronics社のSTM32向けモータ制御ソフトウェア開発キット(MCSDK)[5]を使用して実際のハードウェアで推定されたデータから一部取得したものを、以下の表に示しています。

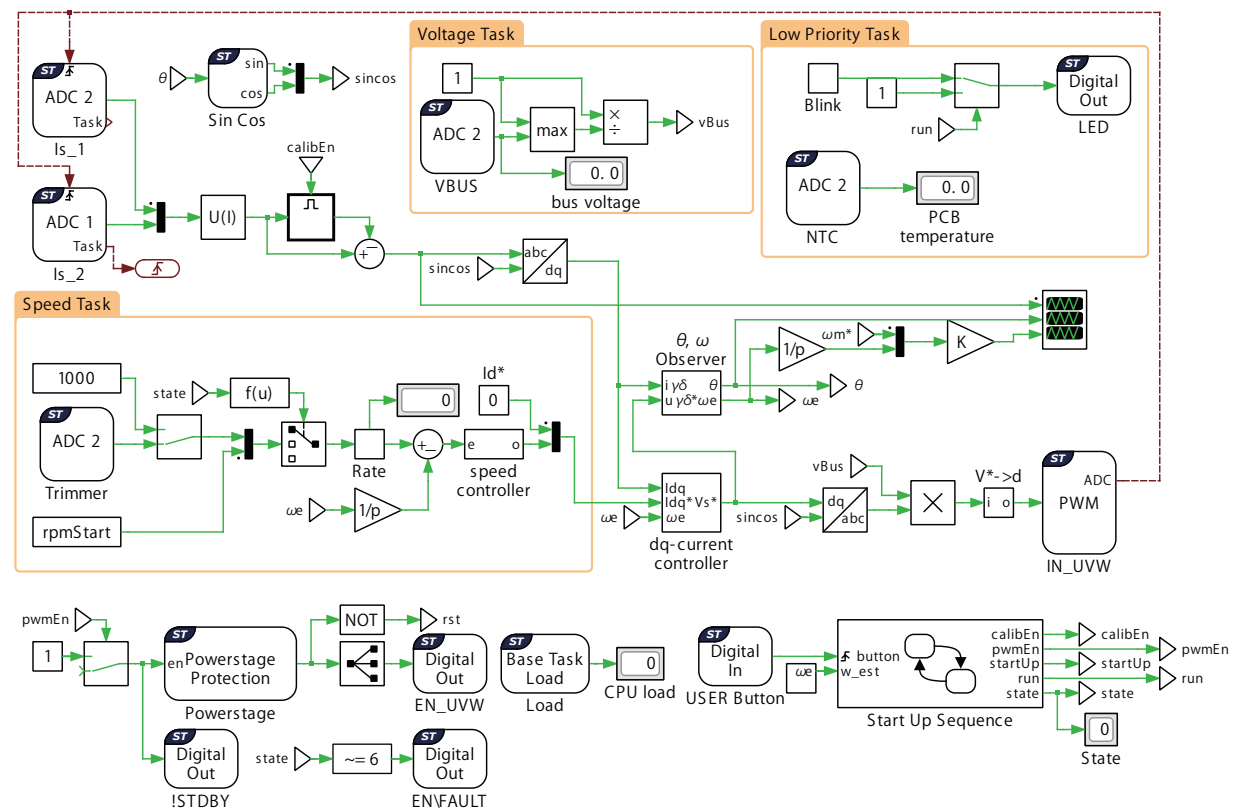
表1: GBM2804H-100Tモーターの仕様

Number of pole pairs	7
Maximum Torque (Nm)	0.981
Maximum Speed (rpm)	2180
Stator Resistance ( $\Omega$ )	4.7
Stator Inductance (mH)	0.96
Inertia (nNms <sup>2</sup> )	497
Friction (nNms)	755
Back EMF Voltage (V/krpm)	4.85

## 2.2 Controller

図3のControllerサブシステムでは、3つのステータ電流はSTM32 TargetコンポーネントライブラリのAnalog In (Triggered) ブロックが測定します。これらはインジェクト変換を表し、変換の開始は外部イベント、この場合はPWMアンダーフローイベントによってトリガされます。DCバス電圧は、通常のADCチャンネルを使用して監視します。通常のADCチャンネルは、CPUの介入なしにバックグラウンドで継続的に変換されます。通常のADCチャンネルは、CPUの介入なしにバックグラウンドで継続的に変換されます。同じタイプのADCを使用して、PCB上の温度と、X-NUCLEOIHM16M1ボードに搭載されたトリマ R17によって提供される速度リファレンスを検出します。

図3: センサレスロータフィールド指向制御のControllerモデル



ドライブシステムにはロータフィールド指向制御が適用され、その基本構造を図3に示します。ステータ電流はdq回転座標系で制御されます。速度とロータ位置は、"Observer"というラベルのサブシステムによって計算されます。速度コントローラである外側の制御ループが実装されており、ここでは、定数ブロックを介して、またはX-NUCLEO-IHM16M1ボード上のトリマR17を介して、RPM単位の速度リファレンスを手動で設定します。トリマによる回転速度の調整は、200から1500RPM間で可能です。トリマを時計回りに回転させると、速度リファレンス値が低下します。

**注意** コントローラ全体は、NUCLEOボード上の青色の"User"ボタンで有効または無効にできます。

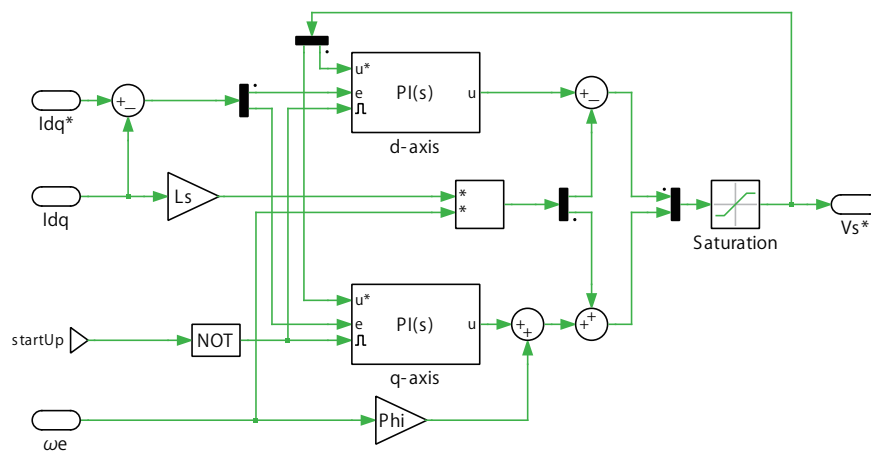
dq回転座標系における電圧制御変数は、 $L_{sd} = L_{sq}$ の場合、式1から導出されます。

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega L_{sq} \\ \omega L_{sd} & R_s \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} L_{sd} & 0 \\ 0 & L_{sq} \end{bmatrix} \cdot \begin{bmatrix} \frac{d}{dt} i_d \\ \frac{d}{dt} i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi_f \end{bmatrix} \quad (1)$$

### PI電流コントローラ

d軸とq軸の電流は、2つの独立したPIコントローラによって制御され、図4に示す"dq-current controller"というラベルのサブシステムに含まれています。比例および積分ゲインは、与えられた帯域幅と位相マージンを達成するように設計されています。

図4: dq座標系におけるPIコントローラ



### ロータ位置および速度観測

図5に"Observer"というラベルのサブシステムを示します。これは[1]に基づいて開発され、概念的には以下で説明します。このObserverは、表面PMSM(非突極 $L_d = L_q$ )、埋込PMSM(突極 $L_d < L_q$ )、同期リアクタンスモータなど、あらゆる種類の同期機に利用できます。

図5に示すObserverモデルでは、推定された回転 $\gamma$ フレームが使用され、これは位置誤差 $\theta_e$ に関してdq参照フレームとは異なります。位置誤差 $\theta_e$ は、Observerによって推定された拡張起電力(EMF)から得られます。この位置誤差 $\theta_e$ は、ロータの位置と速度を推定するために使用します。

図6は、 $e_\gamma$ を推定するための最小次数Observerのブロック図を示しています。gはObserver Gainであり、600に設定されています。 $e_\gamma$ のd軸成分は、 $e_\gamma$ と同様の方法で推定します。図6の青色の点線枠は、Observerが全体モデルにどのように組み込まれているかを示しています。

Observer内のすべての積分器は、リセット信号によってリセットできます。

図5:  $\theta, \omega$  Observerの回路図

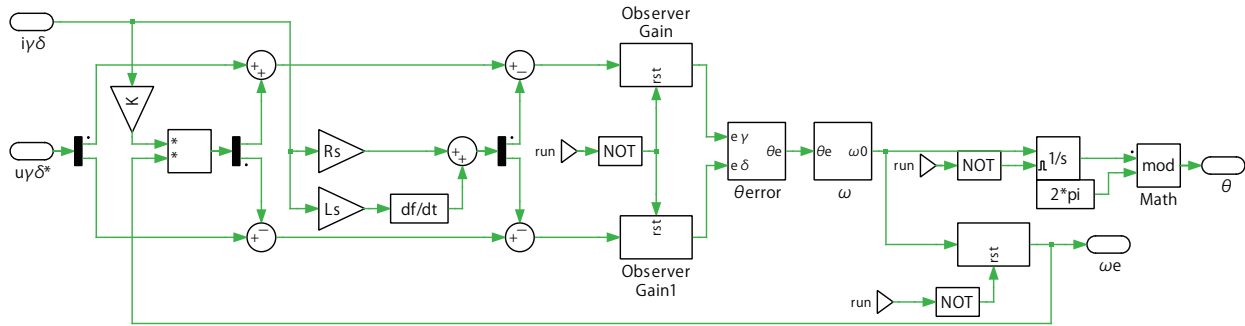
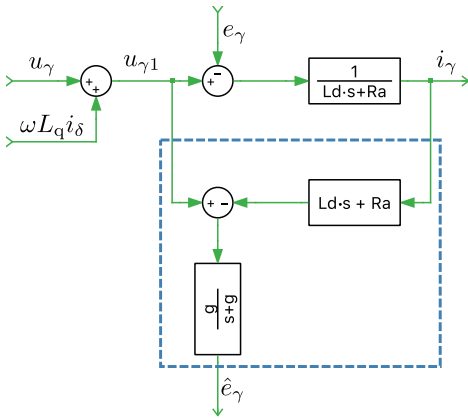


図6: 最小次数observerの等価ブロック図



### 速度コントローラ

外側の速度制御ループが駆動速度を制御します。外側の制御ループの時定数は、内側の電流制御ループの時定数に比べて大きい機械的時定数によって決まります。

比例積分速度コントローラは式2で表すことができます。

$$\frac{T_e^*}{\Delta\omega_m} = \left( K_p + \frac{K_i}{s} \right) \tag{2}$$

ここで、 $T_e^*$ は基準トルク、 $\Delta\omega_m = \omega_m^* - \omega_m$ 、 $K_p$ は比例ゲイン、 $K_i$ は積分ゲインです。

速度コントローラの操作変数である基準トルク $T_e^*$ は、内側の電流コントローラループの目標値です。したがって、式3を使用して、機械的トルク $T_e^*$ から電流設定値 $i_q^*$ への変換が行われます。

$$T_e^* = \frac{3}{2} \Psi_m i_q^* p \tag{3}$$

ここで、 $\Psi_m$ は永久磁石の磁束鎖交であり、 $p$ は回転子の永久磁石の極対の数です。

### マルチタスクコード

PLECS CoderとSTM32 Target Support Packageを使用すると、STM32ファミリのMCU向けにマルチタスクコードを生成できます。マルチタスクコードは、さまざまな時間スケールで動的に変化する複数のシステム出力を制御するための処理能力

を引き出します。このモデルでは、高速な内側の電流制御ループと、より低速な外側の速度制御ループを備えたカスケード制御方式を採用しています。マルチタスクコードは、こういった制御方式に非常に適しています。

マルチタスクのコード生成は、**Coder** -> **Coderオプション...**ダイアログの **タスク**タブで設定します。**タスクモード**をマルチタスクに変更し、**タスク設定**で、各タスクのサンプリング時間を設定します。基本サンプリング時間は常に**離散化ステップサイズ**と等しくなります。優先度の低いタスクの**サンプリング時間**設定は、基本サンプル時間の整数倍でなければなりません。実行速度が異なる低優先度の低速タスクを最大15個まで指定できるため、アプリケーション内で最も高速で優先度の高いタスクのためにプロセッサ時間を確保できます。詳細については、PLECSユーザマニュアル[7]の"コード生成"セクションを参照してください。

PLECSのブロック回路図は、タスクライブラリコンポーネントを使用して、優先度の低いタスクに割り当てられます。このモデルでは、[図3](#)に示すように、Base Taskに加えて、優先度の低い3つのタスクが定義されています。以下の表は、Base Taskに対する優先度の低いタスクの実行時間のサンプルを示しています。

タスク名称	サンプリング時間
Base Task	1/fsw
Voltage Task	10/fsw
Speed Task	20/fsw
Low Priority Task	10000/fsw

## スタートアップシーケンス

ドライブの起動プロセスを処理するために、ステートマシンが実装されています。ステートマシンは、NUCLEOボード上の青色の"User"押しボタンが押されるまで、Idle(アイドル)状態で待機します。ボタンを押すと、OffsetCalibまたはオフセットキャリブレーション状態に移行します。この状態では、シャント電流測定によるオフセットを一定期間校正します。この校正時間後に、起動を開始します。青色の"User"押しボタンを2度押すと、ステートマシンはIdle状態に戻り、PWM信号は無効になります。

## 3 シミュレーション

### 3.1 PLECSオフラインシミュレーション結果

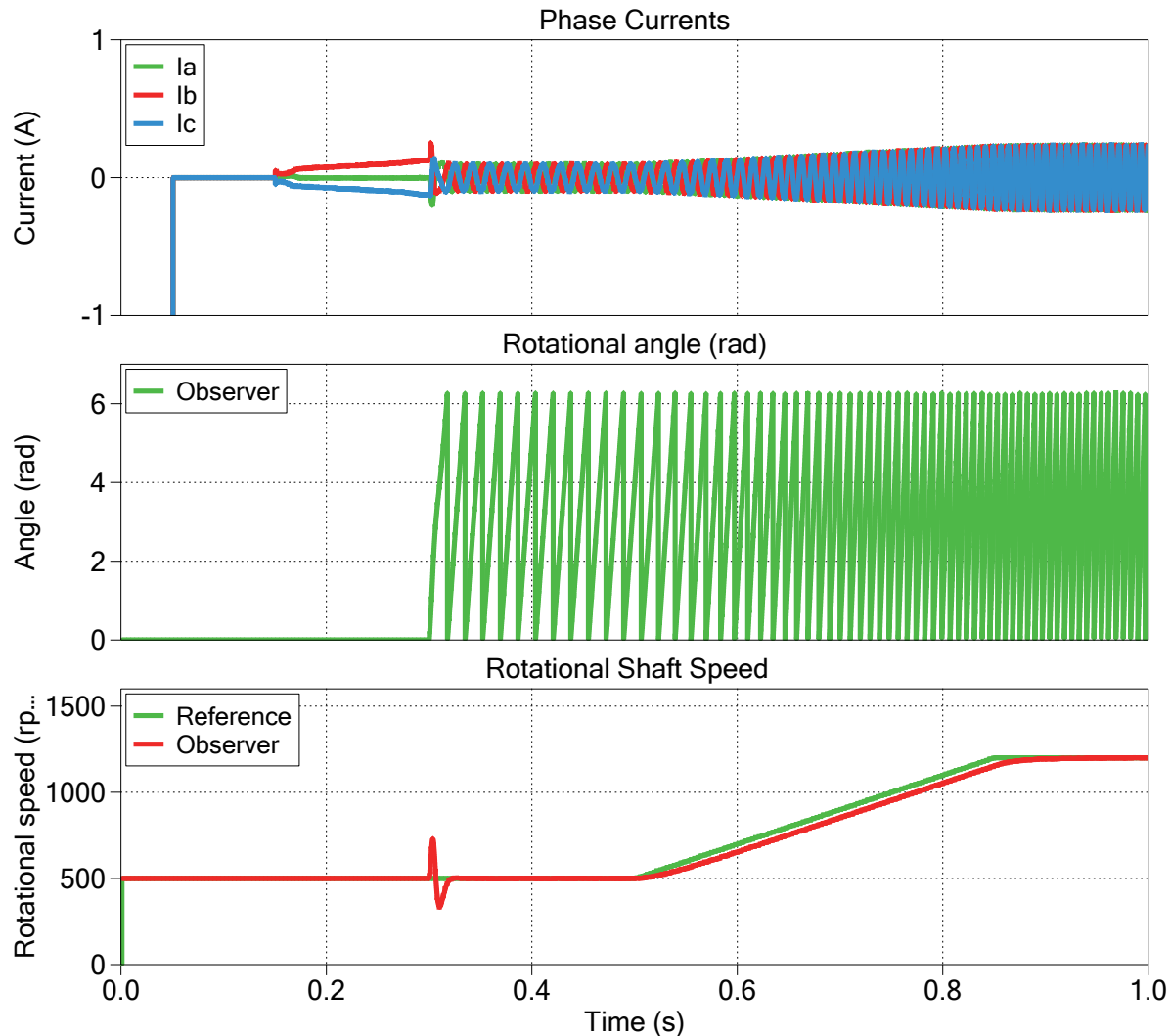
**シミュレーション** -> **開始**でモデルを実行し、オフラインシミュレーションの結果を確認します。[図7](#)は、"Controller"サブシステムPLECSスコープの結果を示しています。

0.05秒で起動プロセスを開始します。まず、電流ADCチャンネルの静的測定オフセットを除去します。0.05秒から、リファレンス電流が徐々に上昇します。モータの位置を調整し、500RPMの速度リファレンスで回転を開始します。推定速度がこの速度リファレンス値に達すると、モータは定数ブロックからの最終速度リファレンス値である1200RPMへのトラッキングを開始します。速度リファレンス値の変化率は2000RPM/sに制限されます。

### 3.2 STM32ターゲットの構成

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ことに加え、"Controller"サブシステムをSTM32G431RB、STM32G474RE、またはSTM32F303RE NUCLEOボード向け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。モデルはデフォルトではSTM32G431RB NUCLEOボード用に構成されていますが、他のターゲットもサポートされています。**Coderオプション**ウィンドウで、目的の**ターゲット**を選択してください。入出力構成は、選択されたターゲットに

図7: PMSMのオフラインシミュレーション結果(速度リファレンス500RPMで起動し、0.5秒で1200RPMまで上昇)



応じて自動的に調整されます。これは、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...-> 初期化タブにあるモデル初期化コマンドウィンドウで行います。オプションとして、"Plant"サブシステムをPLECS RT Boxにデプロイして、生成されたコードのハードウェアインザループ(hardware-in-the-loop: HIL)テストを実行できます。このコントローラは、ST社が提供するNUCLEO-IHM03拡張ボードと連携するように構成されています。

以下の手順に従って、"Controller"サブシステムをSTM32 MCUにアップロードします。

- MCUをUSBケーブルでホストコンピュータに接続します。
- **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**タブからドロップダウンメニューでSTM32G4xまたはSTMF3xを選択します。そして、**General**サブタブで、G431RB、G474RE、またはF303REを選択します。
- PLECSからMCUターゲットを直接デプロイするには、**Generate code only**パラメータのチェックを外し、ドロップダウンメニューから目的の**Programming interface**を選択します。デフォルトのプログラミングインタフェースはOpenOCDです。
- 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、NUCLEOボード上の緑色のLED "LD2"が点滅するはずです。

**注意** NUCLEOボード上のジャンパ構成を確認してください: JP5は5V\_STLK、JP8は位置[1-2]、JP6は閉じているはずで

### 3.3 PLECS RT Boxの構成

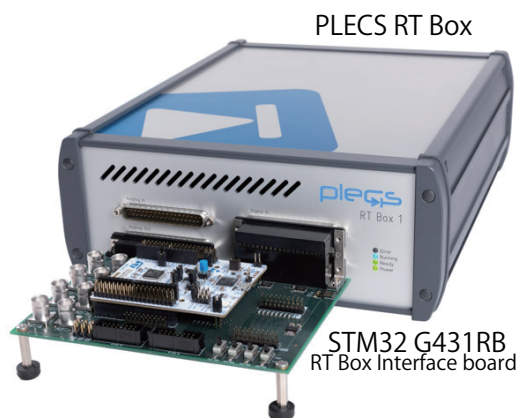
プログラムされたMCUで実際のパワーステージを制御する前に、PLECS RT Boxを使用してコントローラの動作を確認し、HILテストを実行することを強くお勧めします。図8に一般的なハードウェア構成を示します。NUCLEOボード(水色のボード)は、RT Box NUCLEO Interfaceボード(緑色のボード)を介してRT Boxに接続されています。

RT Box上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。"Plant"サブシステムは、すべてのRT Boxプラットフォームで実行可能です。

- **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**タブで、ターゲットデバイスを選択します。そして**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。RT Boxのピン配置はマイクロコントローラに依存するため、RT Box Plantシステムのコードを生成する前に、Controllerシステムでマイクロコントローラのターゲットを選択し、保存されていることを確認してください。
- モデルをアップロードしたら、**Coderオプション**ウィンドウの**外部モード**タブから**接続**を選択し、**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確認します。

正しくプログラムされていれば、RT Box NUCLEO Interfaceボードの"DO-31"に対応するLEDが点滅するはずで

図8: HIL検証のためのハードウェアセットアップ



### 3.4 閉ループHILテストの実行

NUCLEOボード上の青色の"User"ボタンを押して、MCUを有効にしてください。これにより、起動シーケンスとPWM信号の生成を開始します。起動が成功すると、速度コントローラは自動的に速度リファレンス値をトラッキングし、NUCLEOボード上の緑色のLED "LD2"が点灯します。"Plant"サブシステムのPLECSスコープで、リアルタイムの波形を観察します。シミュレーションメニューからシミュレーション・パラメータ... -> 初期化タブにある**モデル初期化コマンド**ウィンドウから、applicationの値を'rt\_box'に変更します。"Controller"サブシステム内の速度リファレンスソースが定数ブロックに接続されていることを確認してください。定数ブロックの速度リファレンス値は、モデルのビルド前に、**Coderオプション**ウィンドウの**パラメータインライン化**タブにある"例外"リストにコンポーネントが追加されているため、リアルタイムで変更できます。MCU上で計算された中間値を確認したり、定数ブロック内の速度リファレンス値を変更したりするには、以下の手順に従ってSTM32 MCUの外部モードに接続してください。

- 最初に、PLECS RT Boxの**外部モード**から"Plant"サブシステムを切断します(接続している場合)。
- **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"サブシステムのPLECSスコープでテスト結果を確認します。

次に、[3.3](#)に記載されている手順に従って、RT Boxの外部モードに再度接続します。

モータの回転を止めるには、NUCLEO基板上の青色の"User"ボタンをもう一度押します。これによりPWM信号の生成が停止し、コントローラはアイドル状態に入り、ユーザによって次の起動シーケンスが開始されるのを待ちます。

### 3.5 閉ループRCPテストの実行

最後に、NUCLEOボード上で動作する生成された制御コードは、ST社が提供するNUCLEO-IHM03拡張ボード付属のモータを回転させることもできます。**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...** -> **初期化**タブにある**モデル初期化コマンド**ウィンドウから、applicationの値を'rcp'に変更します。生成された制御コードをNUCLEO motor-control packでテストするためのハードウェア設定については、以下の手順に従ってください。

- X-NUCLEO-IHM16M1は、CN7およびCN10 ST Morphoコネクタを介してNUCLEOボードにスタックする必要があります。IHM16M1ボードの配置方向に注意してください。NUCLEOボード上の2つのボタンは、引き続き操作可能な状態しておく必要があります。
- CN1コネクタに3本のモータ線を接続します。
- X-NUCLEO-IHM16M1ボードのジャンパ設定を確認してください。J5とJ6が閉じられ、J2が位置[2-3]で閉じられ、J3が位置[1-2]で閉じられています。
- 12V DC電源(パックに付属の電源、または同等の電源を使用)をCN1またはJ4に接続します。

最終的には、[図9](#)のような構成になるはずですが。

次に、[3.2](#)に記載されている手順に従って、PLECS Coderを使用してMCUをプログラムします。正しくプログラムされていれば、NUCLEOボード上の緑色のLED "LD2"が点滅するはずですが。

NUCLEOボード上の青色の"User"ボタンを押して、MCUを有効にします。これにより、起動シーケンスとPWM信号の生成が開始されます。起動が成功すると、速度コントローラは自動的に速度リファレンス値をトラッキングし、NUCLEOボード上の緑色のLED "LD2"が点灯します。

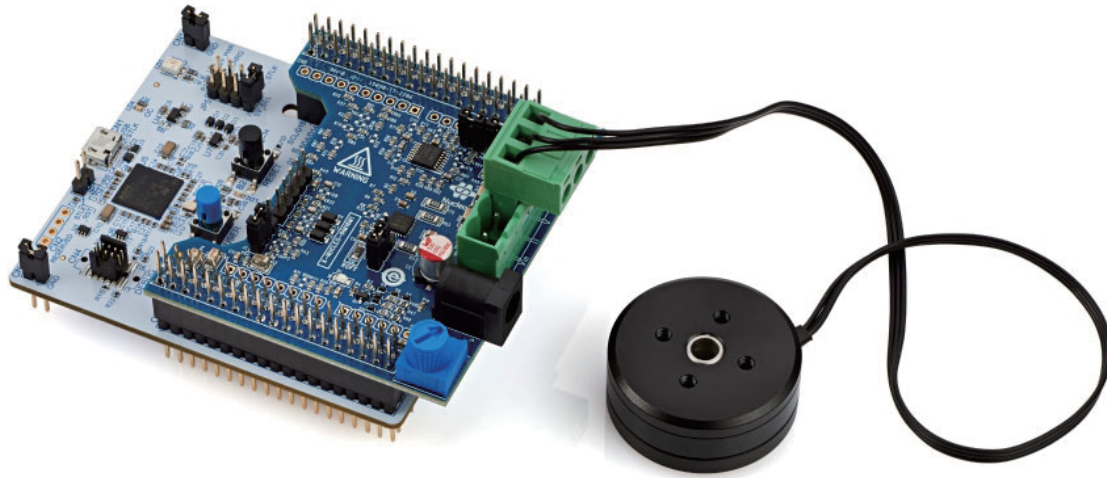
MCU上で計算された中間値を確認したり、速度リファレンスソースを変更したりするには、以下の手順に従ってSTM32 MCUの外部モードに接続してください。

- **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。
- **自動トリガを有効化**にチェックして、"Controller"サブシステムのPLECSスコープでテスト結果を確認します。

X-NUCLEO-IHM16M1基板上のポテンショメータを回転させることで、モータの速度リファレンス値を変更できます。この機能を実現するには、"Controller"サブシステム内の速度リファレンス源をADCの"Trimmer"信号に接続する必要があります。

モータの回転を止めるには、NUCLEO基板上の青色の"User"ボタンをもう一度押してします。これによりPWM信号の生成が停止し、コントローラはアイドル状態に入り、ユーザによって次の起動シーケンスが開始されるのを待ちます。

図9: RCP検証のためのハードウェア設定



## 4 まとめ

このモデルは、STM32 MCUにおける過電流および過電圧保護の一般的な構成を示しています。アナログ保護信号によって、数クロック周期以内にPWM出力をオフにすることができます。これにより、故障発生時におけるパワーステージの効率的な保護が可能になります。

このモデルは、NUCLEO-IHM03拡張ボードを用いたPMSM駆動システムのセンサレスFOC制御の応用を実証しました。3つのシミュレーションモードを用意して、それぞれに段階的な手順を示しました。

- PLECSによるオフラインシミュレーション
- PLECS RT BoxによるHILテスト
- STMicroelectronics社のP-NUCLEO-IHM03 motor-control pack[3]を用いたRCP

## 5 参考文献

- [1] S. Morimoto, K. Kawamoto, M. Sanada and Y. Takeda, "Sensorless control strategy for salient-pole PMSM based on extended EMF in rotating reference frame", IEEE Trans. Ind. Appl., vol. 38, no. 4, pp. 1054-1061, Jul./Aug. 2002.
- [2] R. De Doncker, D. Pule and A. Veltman, "Advanced electrical drives", Springer, 2011.
- [3] Motor Control Nucleo Pack with NUCLEO-G431RB and X-NUCLEO-IHM16M1, URL: <https://www.st.com/en/evaluation-tools/p-nucleo-ihm03.html>
- [4] STSPIN830 Compact and versatile three-phase and three-sense BLDC motor driver, URL: <https://www.st.com/en/motor-drivers/stspin830.html>
- [5] STM32 Motor Control Software Development Kit (MCSDK), URL: <https://www.st.com/en/embedded-software/x-cube-mcsdk.html>

- [6] 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.
  
- [7] PLECS User Manual,  
URL: <https://www.plexim.com/download/documentation>  
日本語マニュアル: <https://adv-auto.co.jp/products/plexim/manual.html>

改訂履歴:

STM32 TSP 1.0.1 初版  
STM32 TSP 1.4.1 初期化コマンド

## 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  
<https://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 System CO.,LTD. Mail  
1-9-5 Uchikanda, Chiyoda-ku  
Tokyo, 101-0047  
Japan  
<https://kesco.co.jp> Web

### *Embedded Code Generation Demo Model*

© 2002–2026 by Plexim GmbH

このマニュアルで説明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の書面による事前の同意なしに、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。

本マニュアルは、Plexim社の英文マニュアルを日本語に翻訳したものです。本マニュアルと英文マニュアルとで差異がある場合、英文マニュアルを正とします。

本マニュアルの内容に基づいて発生した負傷や損害などに対して、Plexim GmbHおよび計測エンジニアリングシステム株式会社は一切責任を負いません。製品とアプリケーションに関連したリスクを最小限に抑えるため、ユーザが適切な設計および保護対策を用意する必要があります。