

Application Note

AM13x の Simulink モデル ベンチマーク



Nilabh Anand, Arun Munuswamy, Val Singh

概要

モデル ベースの設計ワークフローにおいて、シミュレーションだけでなく、実際のプロセッサ実行条件下でターゲットとなる組み込みハードウェアで直接アルゴリズムを検証する必要性がますます高まっています。ターゲット ハードウェアで Simulink アルゴリズム モデルのベンチマーク測定を行うと、シミュレーションだけでは得られない実行時間やリソース使用率に関する定量的なデータが得られます。これらは、プラットフォームの選択、スケジューリングの決定、マルチレートシステムの設計に不可欠な情報です。このアプリケーション ノートでは、PIL (プロセッサ イン ザ ループ) シミュレーションを使用して、テキサス インストルメンツの AM13x マイコンで Simulink アルゴリズム モデルのベンチマーク測定を行うための構造化方法論をご紹介します。

目次

1 はじめに.....	2
2 PIL (プロセッサ イン ザ ループ).....	2
3 前提条件.....	2
4 AM13x での電流制御アルゴリズムのベンチマーク測定.....	3
4.1 C28x 参照シミュレーション モデル例.....	3
4.2 F28379D 用に構成されたシミュレーション モデル.....	3
4.3 シミュレーション結果.....	4
5 アルゴリズムの分離: サブシステムのモデル参照への変換.....	5
5.1 変換手順.....	5
5.2 PIL 用に構成された変換済みモデル参照ブロック.....	6
6 AM13x PIL のモデル参照の構成.....	7
6.1 ターゲット ハードウェアの選択.....	7
6.2 SysConfig 構成.....	7
7 SIL/PIL マネージャー.....	9
7.1 SIL/PIL マネージャーの起動と PIL の実行.....	9
8 最適化および再プロファイリング.....	11
8.1 PIL の結果: 単一データ型と正弦/余弦 TMU を使用.....	11
8.2 TMU 機能を示す Codegen レポート.....	12
8.3 実行の比較オプションからコード ビューを有効化.....	12
8.4 PIL の結果: 単一データ型と正弦/余弦 TMU での実行プロファイリング.....	13
9 まとめ.....	13
10 参考資料.....	13

商標

すべての商標は、それぞれの所有者に帰属します。

1 はじめに

Simulink で開発された組込み制御アルゴリズムは、量産用ハードウェアが最終決定される前に、シミュレーションで検証されること、また初期ターゲット プラットフォーム上でプロトタイプ作成されることがよくあります。モーター制御、電力変換、信号処理のアプリケーションでは、磁界方向制御 (FOC) などのアルゴリズムは、Simulink で数値的に検証された確立済みのシミュレーション実績を持つことが一般的です。また、TI C28x リアルタイム マイコンや TI AM261x といった既存の組込みターゲット、あるいは TI 以外のプラットフォームですでに導入およびプロファイリングが行われ、特定のターゲットに依存しない自己完結型の Simulink サブシステムとして構成されている場合もあります。

2 PIL (プロセッサ イン ザ ループ)

AM13x を候補となるプラットフォームとして導入する際、「このアルゴリズムは当該プロセッサ上でどのような性能を発揮するのか」という根本的な工学的疑問が生じます。シミュレーションでは、この問いの答えを見つけることはできません。実行時間、命令キャッシュの動作、パイプライン使用率、ハードウェア アクセラレーションによる数学演算の影響 (AM13x の場合はアクセラレータは三角関数演算演算ユニット – TMU) は物理プロセッサのプロパティであり、ターゲット ハードウェアで実行されることによってのみ観測できます。

PIL (プロセッサ イン ザ ループ) は、この機能を正確に実現します。PIL は、接続された AM13x ハードウェア上で Simulink アルゴリズム モデルを直接実行し、テスト ハーネスとしてシミュレーション環境を維持し、テスト ベクトルを提供し、出力を収集し、参照シミュレーションに対して数値的な正確性を検証します。実行プロファイリング データは、各 PIL 実行の終了時に自動的に返され、物理デバイス上のアルゴリズムで消費されたプロセッサ サイクルと実行時間を報告します。

3 前提条件

- [AM13x サポート パッケージ](#)
- [C2000 Microcontroller Blockset](#)
- [Motor Control Blockset](#)

4 AM13x での電流制御アルゴリズムのベンチマーク測定

4.1 C28x 参照シミュレーション モデル例

このワークフローは、MathWorks が提供している例『[Code Verification and Profiling Using PIL Testing](#)』が良好な出発点となります。この例には、TI C28x プラットフォームでデモ PIL を行う包括的な FOC モーター制御モデルが含まれています。同じモデルを使用して AM13x を再ターゲットし、同じモデルを使用して電流制御アルゴリズムのベンチマークを行います。

4.2 F28379D 用に構成されたシミュレーション モデル

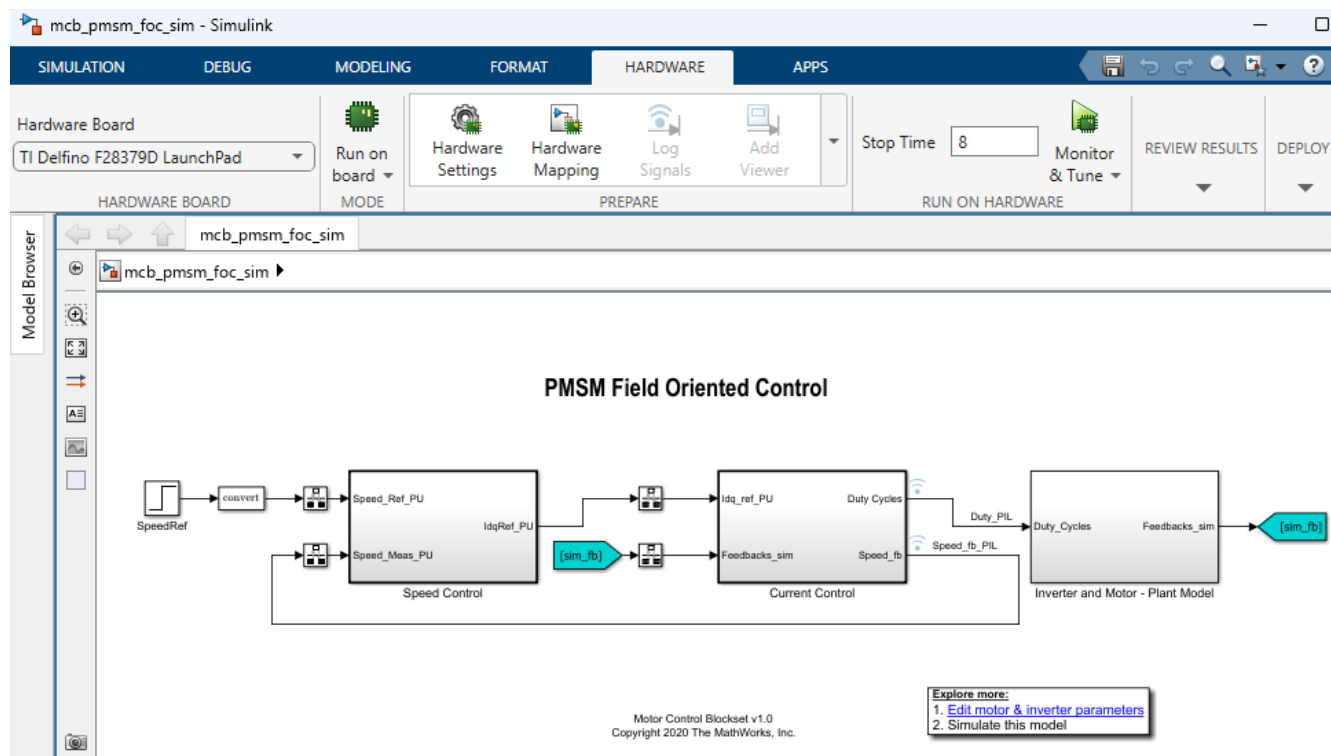


図 4-1. PMSM Blockset

PIL に進む前に、ベースライン シミュレーションの正確性を検証

- モデル例を開き、シミュレーションを実行して完了します
- シミュレーション データ インспекターで速度リファレンスと速度フィードバック信号を観察します。フィードバック軌道は、ターゲット展開の前にアルゴリズムが数値的に健全であることを確認し、リファレンスを正しく追跡する必要があります
- このシミュレーション結果は、PIL 出力と比較するための数値的基準となります

4.3 シミュレーション結果



図 4-2.

5 アルゴリズムの分離: サブシステムのモデル参照への変換

モデル ブロックレベルの PIL では、テスト中のアルゴリズムをモデル参照ブロックとしてカプセル化する必要があります。これにより、電流制御アルゴリズムの周囲に入出力の明確な境界が確立されます。その結果、参照モデルのみを AM13x ターゲットに展開しつつ、周囲のシステム モデルは Simulink ホスト上で実行を継続できます。

5.1 変換手順

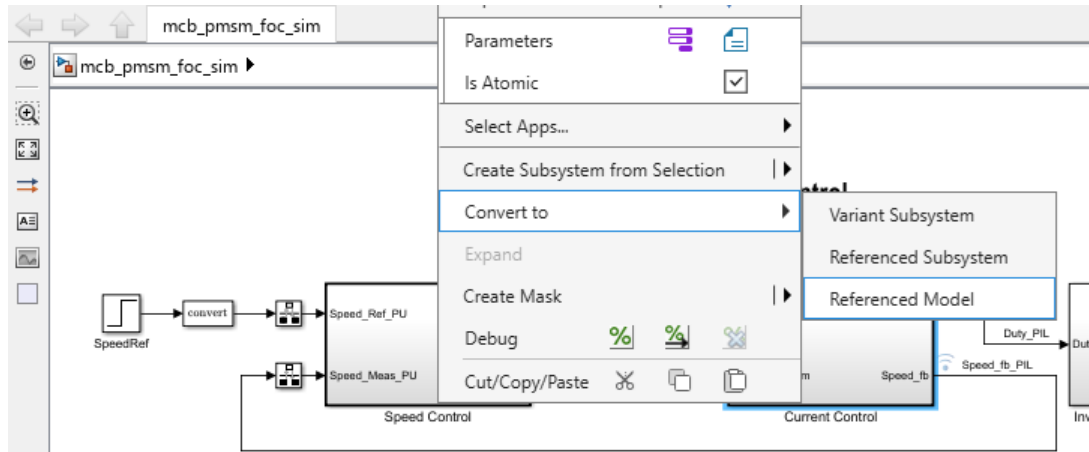


図 5-1. 参照モデル

1. トップレベル モデルで、電流制御サブシステム ブロックを右クリックします
2. 「Subsystem」 → 「Convert to」 → 「Referenced Model」の順に選択します
 - a. 「convert the model」をクリックすると、モデル アドバイザー (Model Advisor) に変換の成功を知らせるメッセージが表示されます

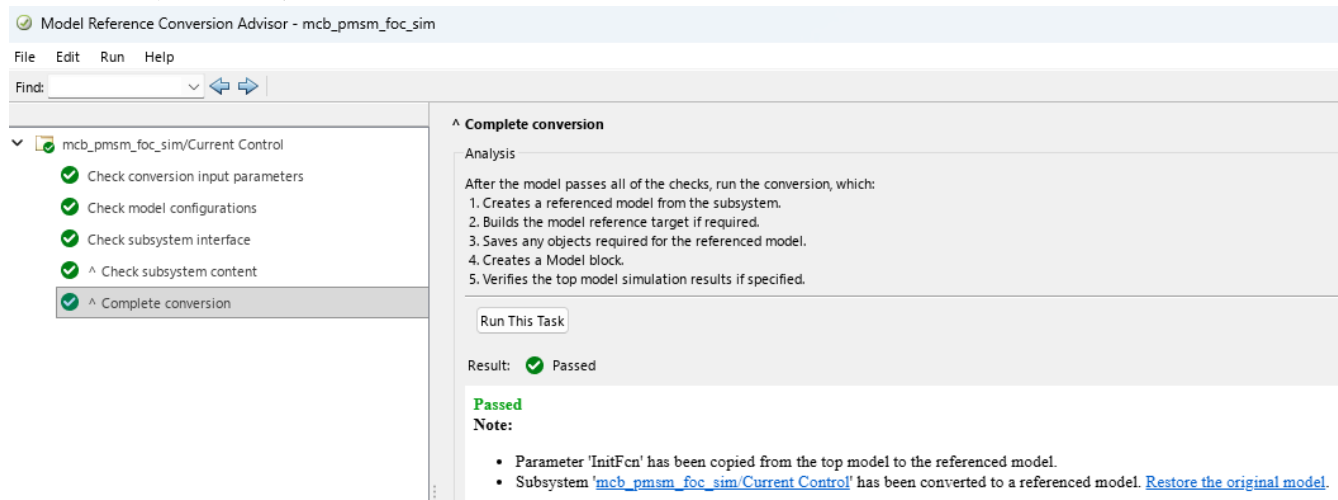


図 5-2. 変換手順

3. Simulink で、電流制御アルゴリズムの内容のみを含む新しい個別のモデル ファイルが作成されます
4. トップレベル モデルの元のサブシステムは、新しく作成されたモデルを参照するモデル参照ブロックに自動的に置き換えられます。参照モデル例との競合を避けるために、トップ モデルの名前を変更します。
5. 変換後にトップレベル モデルが正しくシミュレーションされることを確認します。数値結果は変換前のベースラインと同じである必要があります

トップレベル FOC システムは、Simulink ホストで完全に動作し続けます。PIL 実行中、AM13x には、電流制御モデル参照ブロックのみが展開されます。

5.2 PIL 用に構成された変換済みモデル参照ブロック

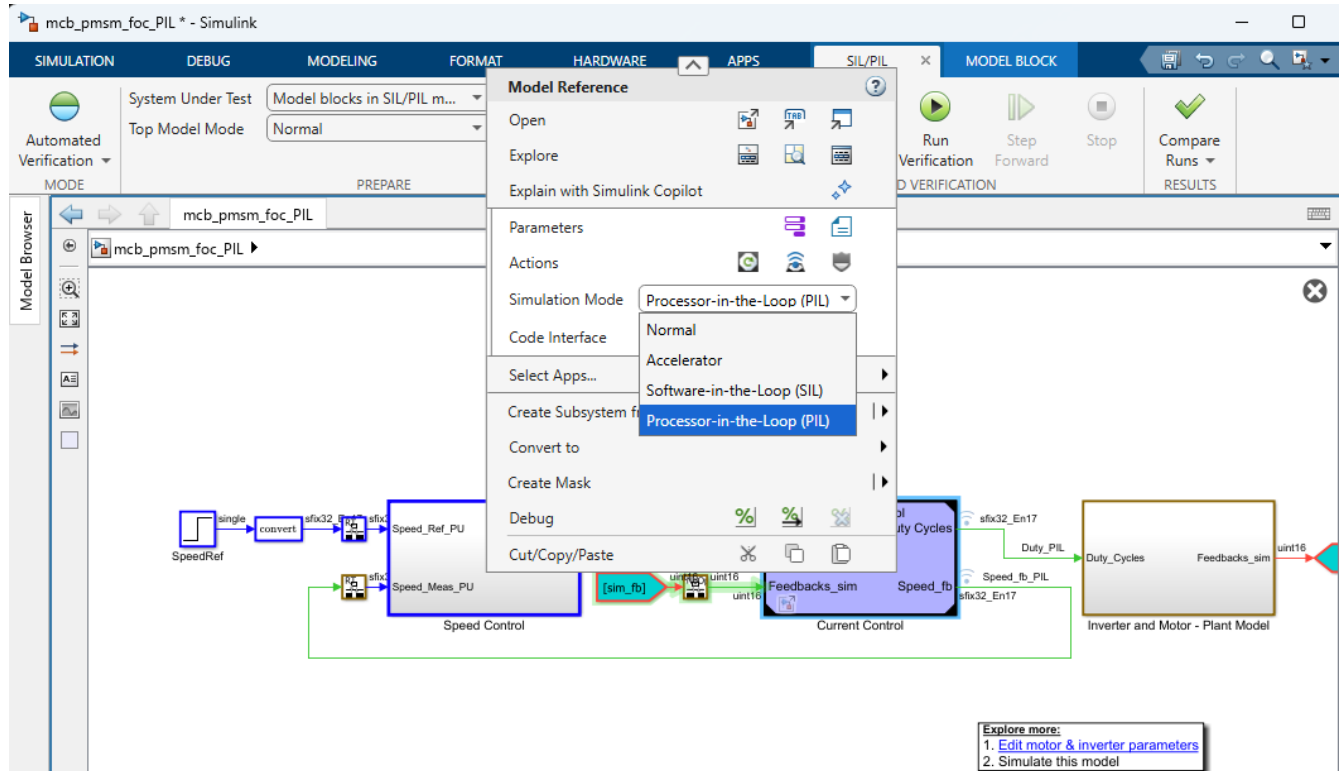


図 5-3. PIL

6 AM13x PIL のモデル参照の構成

モデル参照を確立したら、以下の構成手順で、AM13x Launchpad で PIL 実行できるように、電流制御モデルを準備します。

6.1 ターゲット ハードウェアの選択

1. 電流制御参照モデルを開きます (トップレベル モデルではありません)。
2. 「*Model Configuration Parameters*」 → 「*Hardware Implementation*」の順に移動します。
3. 「ハードウェア ボード」を AM13E230x に設定します。
4. 選択時に、「TMU (三角関数演算ユニット) コード置換ライブラリ」が自動的にアクティブになります。手動操作は不要です。

6.2 SysConfig 構成

1. 「*Target Hardware Resources*」 → 「*SysConfig*」の順に移動します
2. PIL については、AM13x PIL の例に付属している「**SysConfig file**」を選択します。このファイルには、ホストとターゲット間の PIL データ交換に必要なペリフェラルの設定があらかじめ構成されています
3. SysConfig ファイルの場所を指定する際は、「相対パス」を使用します

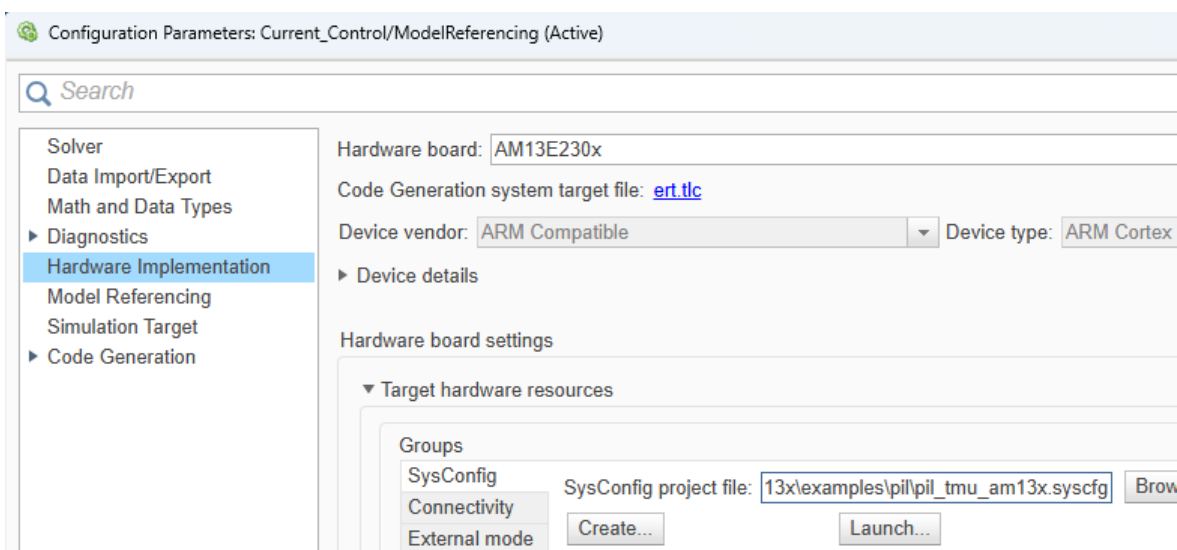


図 6-1. 展開ターゲットの選択

Hardware board: AM13E230x

Code Generation system target file: [ert.tlc](#)

Device vendor: ARM Compatible Device type: A

► Device details

Hardware board settings

▼ Target hardware resources

Groups

SysConfig Connection interface for PIL or external mode: Serial

Connectivity

External mode Serial port: COM7

Baud rate: 921600

図 6-2. UART シリアル接続

7 SIL/PIL マネージャー

7.1 SIL/PIL マネージャーの起動と PIL の実行

1. SIL/PIL マネージャー アプリケーションを開きます (Simulink ツール ストリップで「Apps」→「SIL/PIL Manager」の順に選択)
2. アプリが、PIL 実行、プロファイリング データの収集、数値比較を自動的に管理します
3. 下の画像に示されているように、「Settings」で「Task Profiling」オプションを有効化します

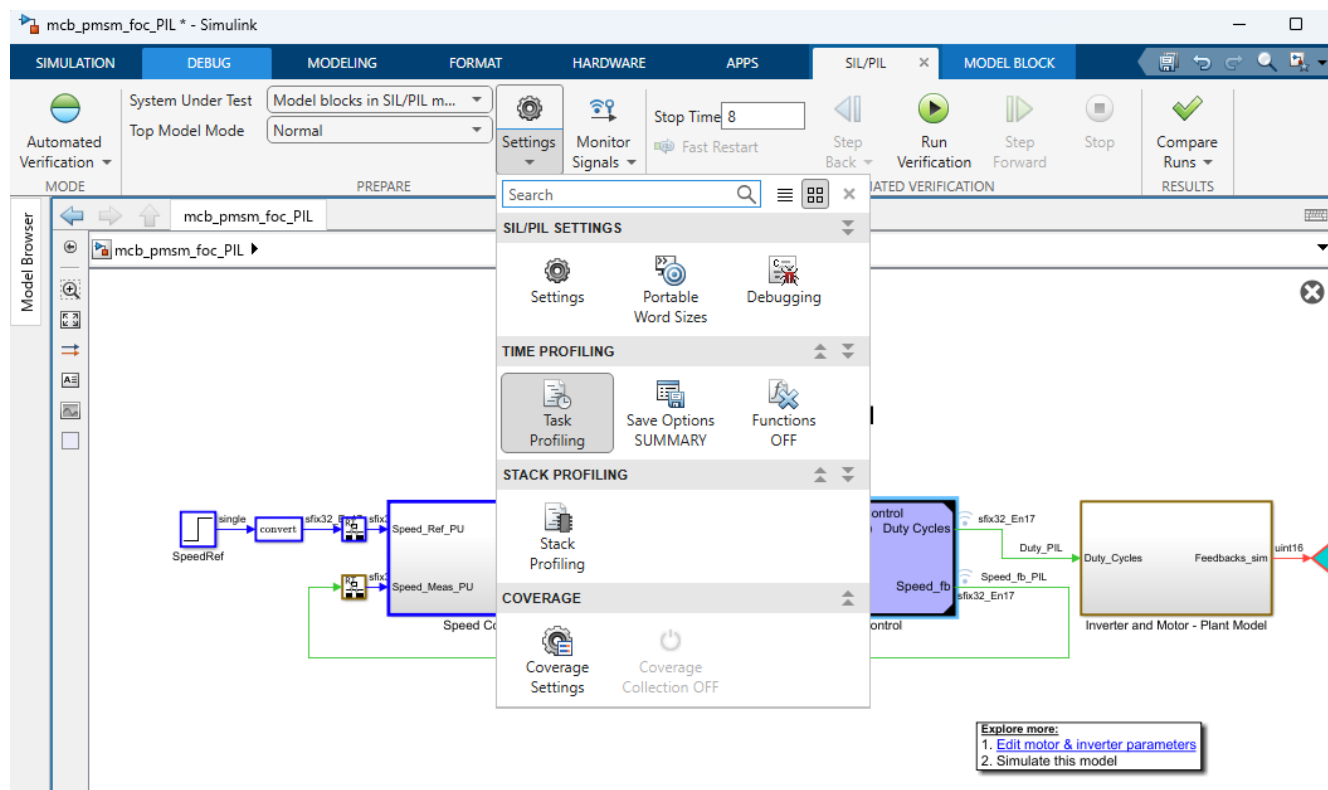


図 7-1. Simulink タスク プロファイリング

7.1.1 シミュレーション時間に関する検討事項

PIL シミュレーションは非リアルタイムです。各ステップでは、データが Simulink ホストから AM13x ターゲットに移動し、シリアル通信リンクを介して戻される必要があります。ベース サンプリング時間が短いモーター制御モデルの場合、シミュレーションを全期間にわたって実行すると、PIL セッションの所要時間が非現実的なほど長くなります。シミュレーションの停止時間を 100 ミリ秒 (0.1) に設定します。有意義なプロファイリング データをキャプチャし、過度の実行時間なしにアルゴリズムの動作を観察するにはこれで十分です。

1. SIL/PIL マネージャーで「実行」をクリックして、PIL 実行を開始します。
2. Simulink は電流制御モデルをコンパイルして AM13x に展開し、通信リンクを確立して、データ交換を段階的に開始します。

7.1.2 PIL の結果:数値検証の表示



図 7-2. 結果

7.1.3 PIL の結果:実行プロファイリング

Code Profile Analyzer

GENERAL | TIME PROFILING

Time unit: Microseconds (us) | Timer Ticks per Second (MHz): 200

Task Execution | Task Validation | Function Execution | Highlight Source | Highlight Code | Highlight Profiling | Generate Report

Task Execution | Task Validation | Function Execution | Comparison

Task Execution Times

Section Name	Maximum Execution Time	Average Execution Time	Maximum Self Time	Average Self Time	Calls
Current_Control_initialize	4.3300	4.3300	4.3300	4.3300	1
Current_Control_Init	0.4950	0.4950	0.4950	0.4950	1
Current_Control [5e-05 0]	16.4550	16.4298	16.4550	16.4298	2487

図 7-3. ベンチマーク測定

8 最適化および再プロファイリング

PIL ベンチマーク測定のワークフローは本質的に反復的です。ベースライン プロファイリング結果が確立されると、エンジニアはコード生成とアルゴリズム最適化を適用し、PIL を再実行して各変更の正確な影響を測定できます。

AM13x の既知の最適化レバーには以下が含まれます。

- **TMU コード置換ライブラリ:** AM13x のターゲット選択時に自動的に有効になり、標準的な三角関数呼び出しを TMU ハードウェア アクセラレーション対応の同等品で置き換え、FOC アルゴリズムの実行時間を大幅に短縮できます。
- **コンパイラ最適化レベル:** 「Model Configuration Parameters」→ 「Code Generation」→ 「Build configuration」の順に選択して調整できます
- **データ型の絞り込み:** アルゴリズムの精度が許す限り、不要な倍精度演算を単精度に削減します
- **ルックアップ テーブルの近似:** TMU が適用できない場合、超越関数呼び出しをテーブル ベースの近似に置き換えます
- その他の **Simulink コード生成の最適化:** 『生成コードの最適化戦略』に記載されています
- 適用される最適化ごとに、PIL 実行を繰り返し、更新されたプロファイル結果を記録します。ベースラインと最適化されたプロファイリング数値を比較することで、AM13x ターゲットの最適化効果を定量的かつハードウェア検証済みで測定することができます。

8.1 PIL の結果: 単一データ型と正弦/余弦 TMU を使用

- mcb_pmsm_foc_sim_data.m のデータ型を「Fixed-point」から「single」に変更します

```
mcb_pmsm_foc_sim_data.m × +
C:\Users\la1255762\OneDrive - TEXAS INSTRUMENTS\Documents\MATLAB\Examples\R2026a\mcb\CodeVerification
19      %% Set data type for controller & code-gen
20      % dataType = fixdt(1,32,17);    % Fixed point code-generation
21      dataType = 'single';           % Floating point code-generation
22
23      %% System Parameters // Hardware parameters
```

図 8-1. データ型の変更

- 近似方法を「Lookup」から「none」に変更して、正弦/余弦関数を生成します

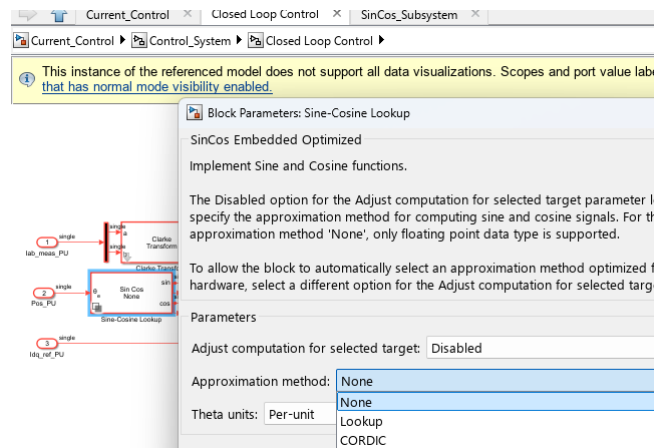


図 8-2. 近似方法

- コード生成により、正弦/余弦が TMU 関数 `sinf_tmu/cosf_tmu` に置き換えられます

8.2 TMU 機能を示す Codegen レポート

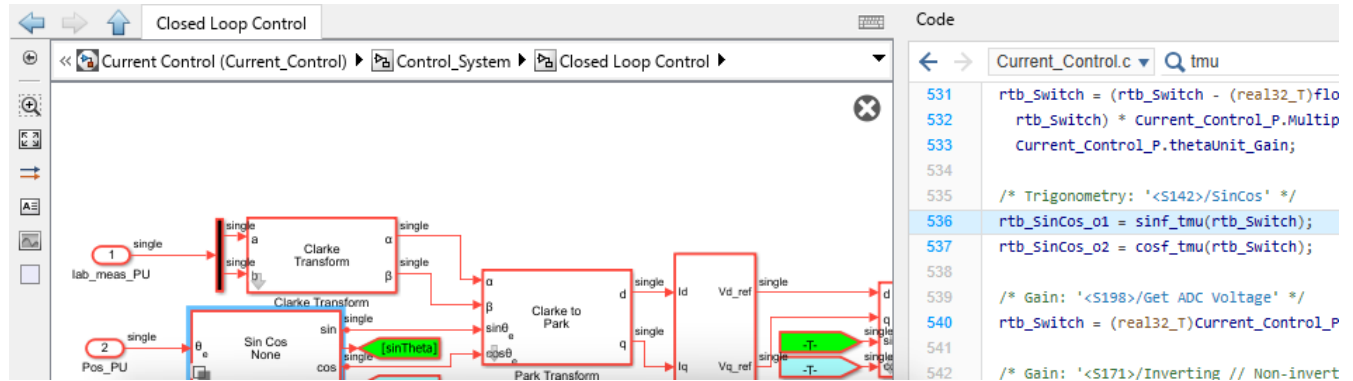


図 8-3. TMU の本質の代替品

8.3 実行の比較オプションからコード ビューを有効化

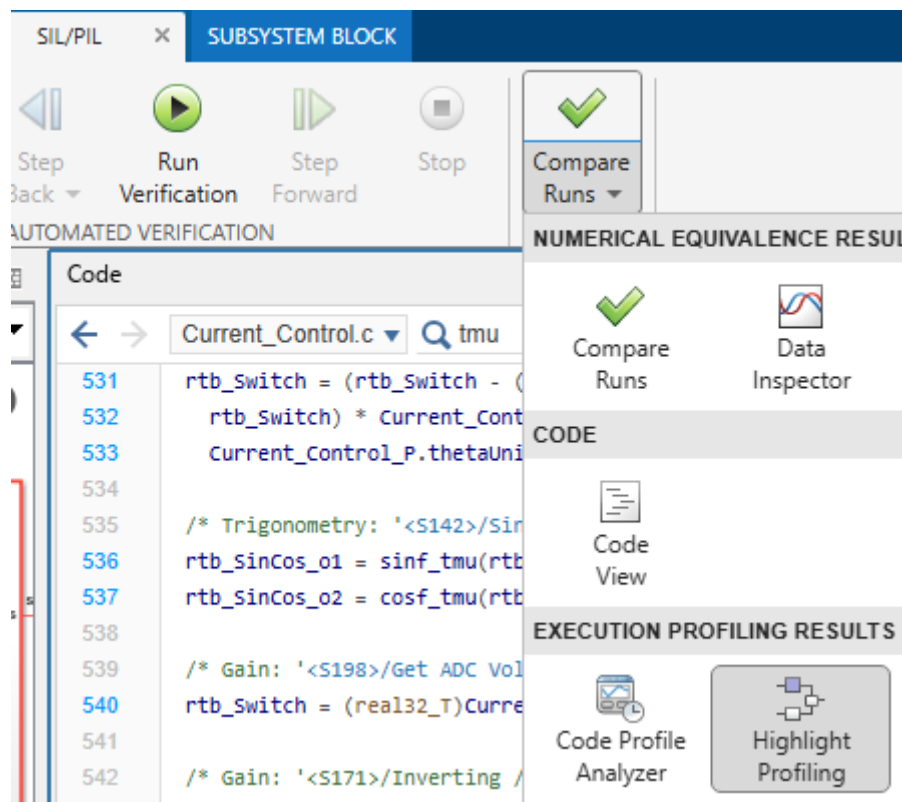
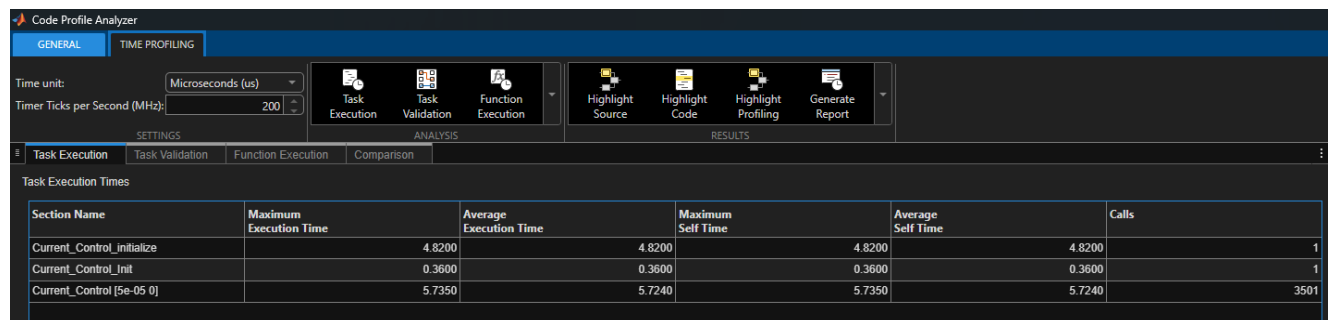


図 8-4. プロファイリング

8.4 PIL の結果: 単一データ型と正弦/余弦 TMU での実行プロファイリング



Section Name	Maximum Execution Time	Average Execution Time	Maximum Self Time	Average Self Time	Calls
Current_Control_initialize	4.8200	4.8200	4.8200	4.8200	1
Current_Control_Init	0.3600	0.3600	0.3600	0.3600	1
Current_Control [5e-05 0]	5.7350	5.7240	5.7350	5.7240	3501

図 8-5. TMU での結果

9 まとめ

このアプリケーション ノートでは、アルゴリズム モデル ソース (C28x、他のマイコン) から AM13x デバイスに移植可能な構造化ベンチマーク測定ワークフローをご紹介しました。これにより、AM13x へのスムーズかつ高速な移行が可能になります。

10 参考資料

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含みいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月