*Application Note*

# *Benchmark Simulink Model on AM13x*

*Nilabh Anand, Arun Munuswamy, Val Singh*

**ABSTRACT**

Model-Based Design workflows increasingly require algorithm validation not only in simulation but directly on target embedded hardware, under real processor execution conditions. Benchmarking a Simulink algorithm model on target hardware provides quantitative execution time and resource utilization data that simulation alone cannot supply information that is essential for platform selection, scheduling decisions, multi-rate system design. This application note presents a structured methodology for benchmarking Simulink algorithm models on the Texas Instruments AM13x microcontroller using Processor-in-the-Loop (PIL) simulation.

## Table of Contents



## Trademarks

All trademarks are the property of their respective owners.



Copyright © 2026 Texas Instruments Incorporated

# 1 Introduction

Embedded control algorithms developed in Simulink are frequently first validated in simulation or prototyped on an initial target platform before a production hardware decision is finalized. In motor control, power conversion, and signal processing applications, it is common for an algorithm such as Field-Oriented Control (FOC) to have an established simulation history verified numerically in Simulink, and potentially already deployed and profiled on a prior embedded target such as the TI C28x real-time microcontroller or the TI AM261x or from a non-TI platform entirely, represented as a self-contained Simulink subsystem with no target-specific dependencies.

# 2 Processor-in-the-Loop (PIL)

A fundamental engineering question arises when the AM13x is introduced as a candidate platform: *how does this algorithm perform on this processor?* Simulation cannot answer this question. Execution time, instruction cache behavior, pipeline utilization, and the impact of hardware-accelerated math operations (Accelerator in case of AM13x is Trigonometric Math Unit – TMU) are properties of the physical processor and are only observable through execution on target hardware.

Processor-in-the-Loop (PIL) provides precisely this capability. PIL executes the Simulink algorithm model directly on the connected AM13x hardware while maintaining the simulation environment as the test harness, supplying test vectors, collecting outputs, and verifying numerical correctness against the reference simulation. Execution profiling data is returned automatically at the conclusion of each PIL run, reporting the measured processor cycles and execution time consumed by the algorithm on the physical device.

# 3 Prerequisites

- AM13x Support Package
- C2000 Microcontroller Blockset
- Motor Control Blockset



Copyright © 2026 Texas Instruments Incorporated

# 4 Benchmarking the Current Control Algorithm on AM13x

## 4.1 Reference C28x Example Simulation Model

The starting point for this workflow is the MathWorks shipping example *Code Verification and Profiling Using PIL Testing* . This example ships with a complete FOC motor control model demonstrating PIL on the TI C28x platform. The same model is used for retargeting to the AM13x, to benchmark the current control algorithm using identical model.

## 4.2 Simulation Model Configured for F28379D

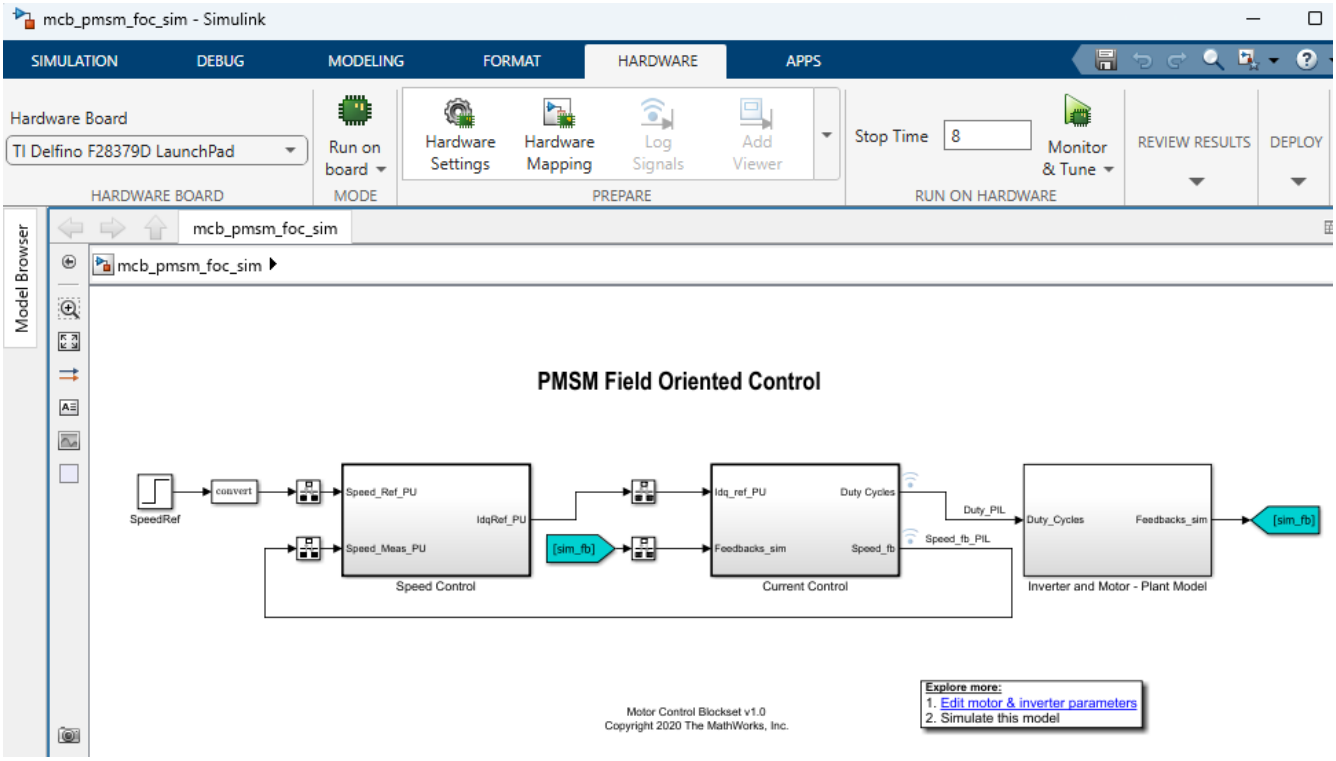


**Figure 4-1. PMSM Blockset**

**Before Proceeding to PIL, Verify Baseline Simulation Correctness**

- Open the example model and run the simulation to completion
- Observe the speed reference and speed feedback signals in the Simulation Data Inspector, the feedback trajectory must track the reference correctly, confirming that the algorithm is numerically sound prior to any target deployment
- This simulation result serves as the numerical reference against which PIL outputs are compared


Copyright © 2026 Texas Instruments Incorporated

## 4.3 Simulation Results


**Figure 4-2.**



Copyright © 2026 Texas Instruments Incorporated

# 5 Isolating the Algorithm: Converting the Subsystem to a Model Reference

PIL at the model block level requires the algorithm under test to be encapsulated as a Model Reference block. This establishes a clean input/output boundary around the current control algorithm and allows the surrounding system model to continue executing on the Simulink host while only the referenced model is deployed to the AM13x target.

## 5.1 Conversion Procedure

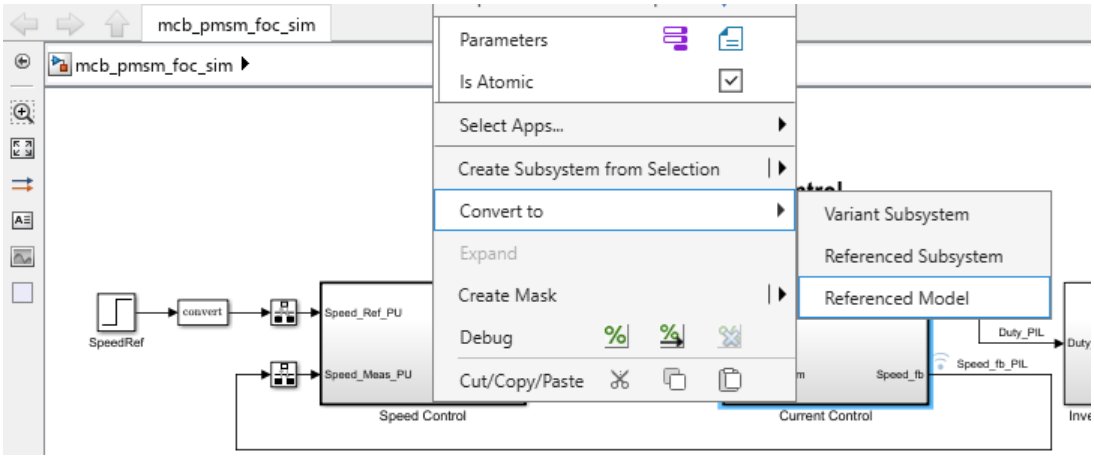


**Figure 5-1. Reference Model**

1. In the top-level model, right-click the current control subsystem block
2. Select *Subsystem → Convert to → Referenced Model*
   a. Click convert the model, Model Advisor shows a successful conversion message

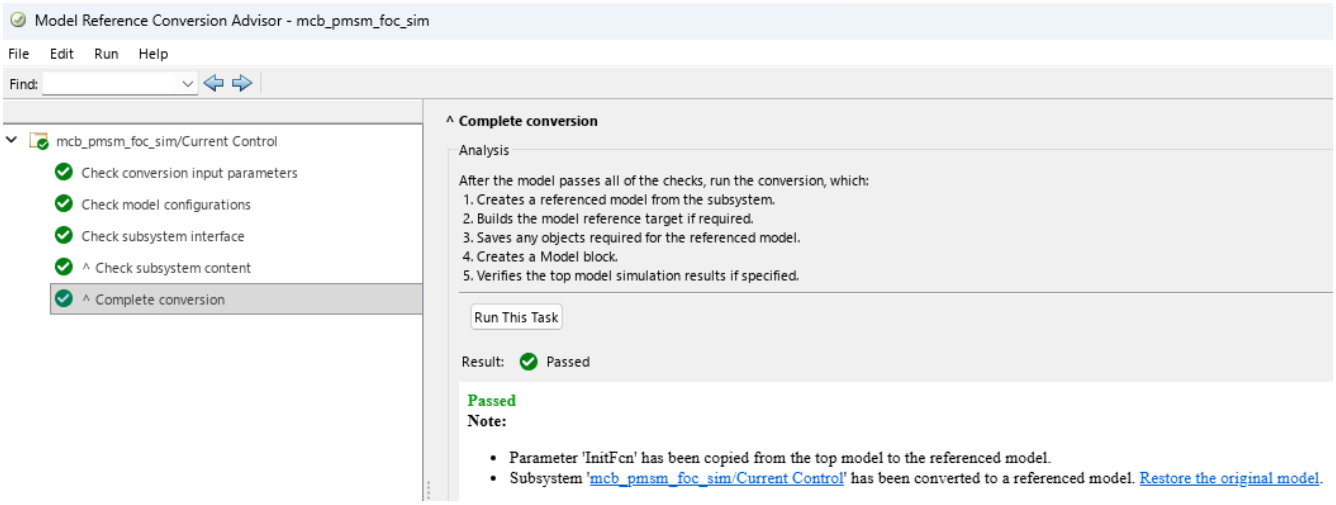


**Figure 5-2. Conversion Step**

3. Simulink creates a new, separate model file containing exclusively the current control algorithm contents
4. The original subsystem in the top-level model is automatically replaced by a model reference block pointing to the newly created model. Rename the top model to avoid conflict with reference example model.
5. Verify that the top-level model simulates correctly after conversion, the numerical results must remain identical to the pre-conversion baseline

The top-level FOC system continues to run entirely on the Simulink host. Only the current control Model Reference block is deployed to the AM13x during PIL execution.


Copyright © 2026 Texas Instruments Incorporated

## 5.2 Converted Model Reference Block Configured for Processor in the Loop

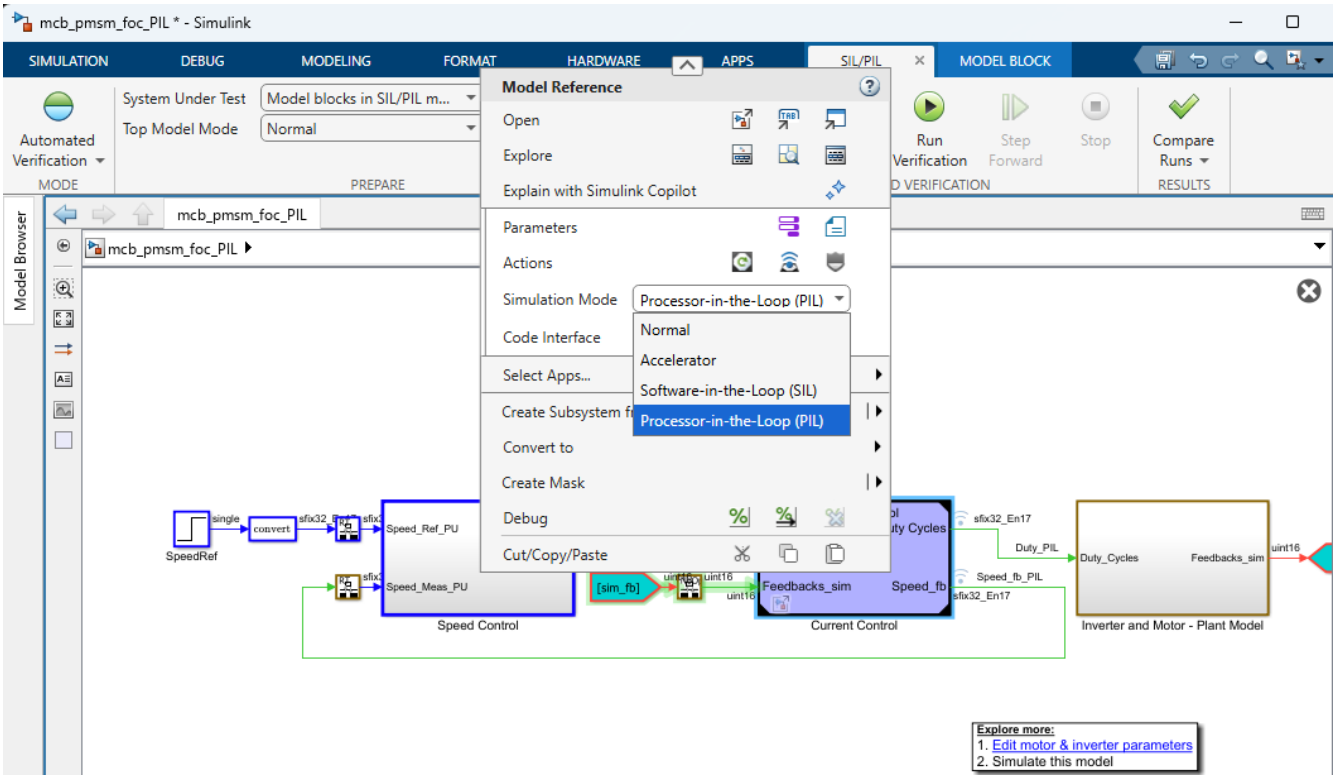


**Figure 5-3. PIL**



Copyright © 2026 Texas Instruments Incorporated

# 6 Configuring the Model Reference for AM13x PIL

With the model reference established, the following configuration steps prepare the current control model for PIL execution on the AM13x Launchpad.

## 6.1 Target Hardware Selection

1. Open the current control referenced model (not the top-level model).
2. Navigate to *Model Configuration Parameters → Hardware Implementation*.
3. Set *hardware board* to AM13E230x.
4. Upon selection, the *TMU (Trigonometric Math Unit) Code Replacement Library* is activated automatically, no manual intervention is required.

## 6.2 SysConfig Configuration

1. Navigate to *Target Hardware Resources → SysConfig*
2. For PIL, *select the SysConfig file shipped with the AM13x PIL example*, this file is pre-configured with the communication peripheral settings required for PIL data exchange between the host and the target
3. Use a *relative path* when specifying the SysConfig file location

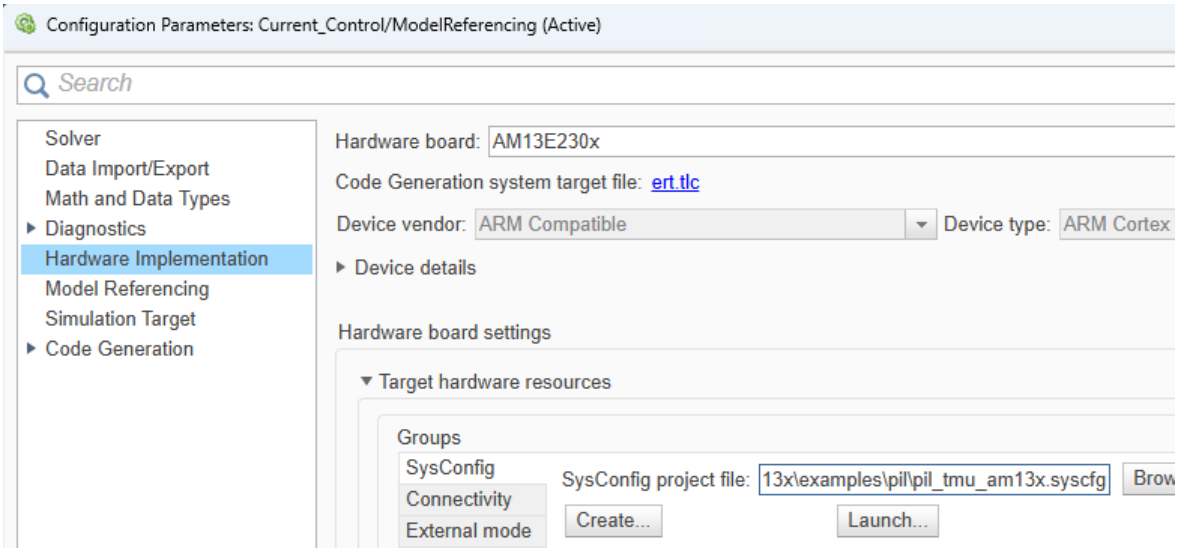


**Figure 6-1. Deployment Traget Selection**

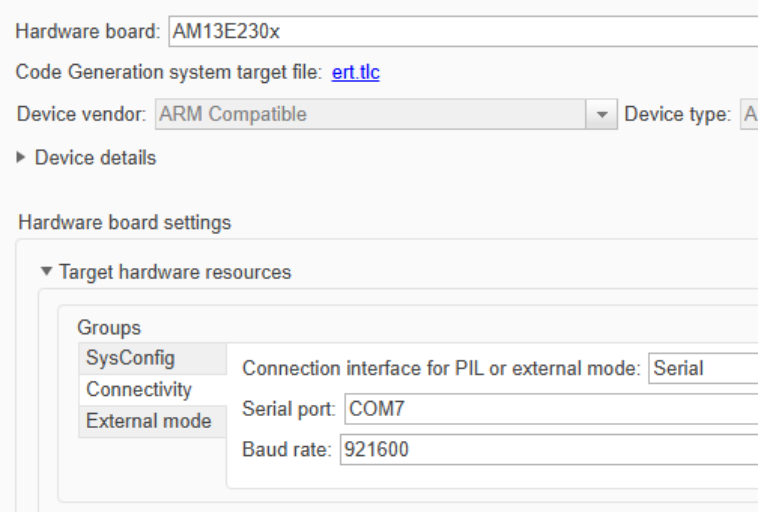


**Figure 6-2. UART Serial Connection**


Copyright © 2026 Texas Instruments Incorporated

# 7 SIL/PIL Manager

## 7.1 Launching the SIL/PIL Manager and Running PIL

1. Open the *SIL/PIL Manager* application (*Apps* → *SIL/PIL Manager* from the Simulink toolstrip)
2. The app manages PIL execution, profiling data collection, and numerical comparison automatically
3. Enable *Task Profiling* option from the Settings, as shown in the image below

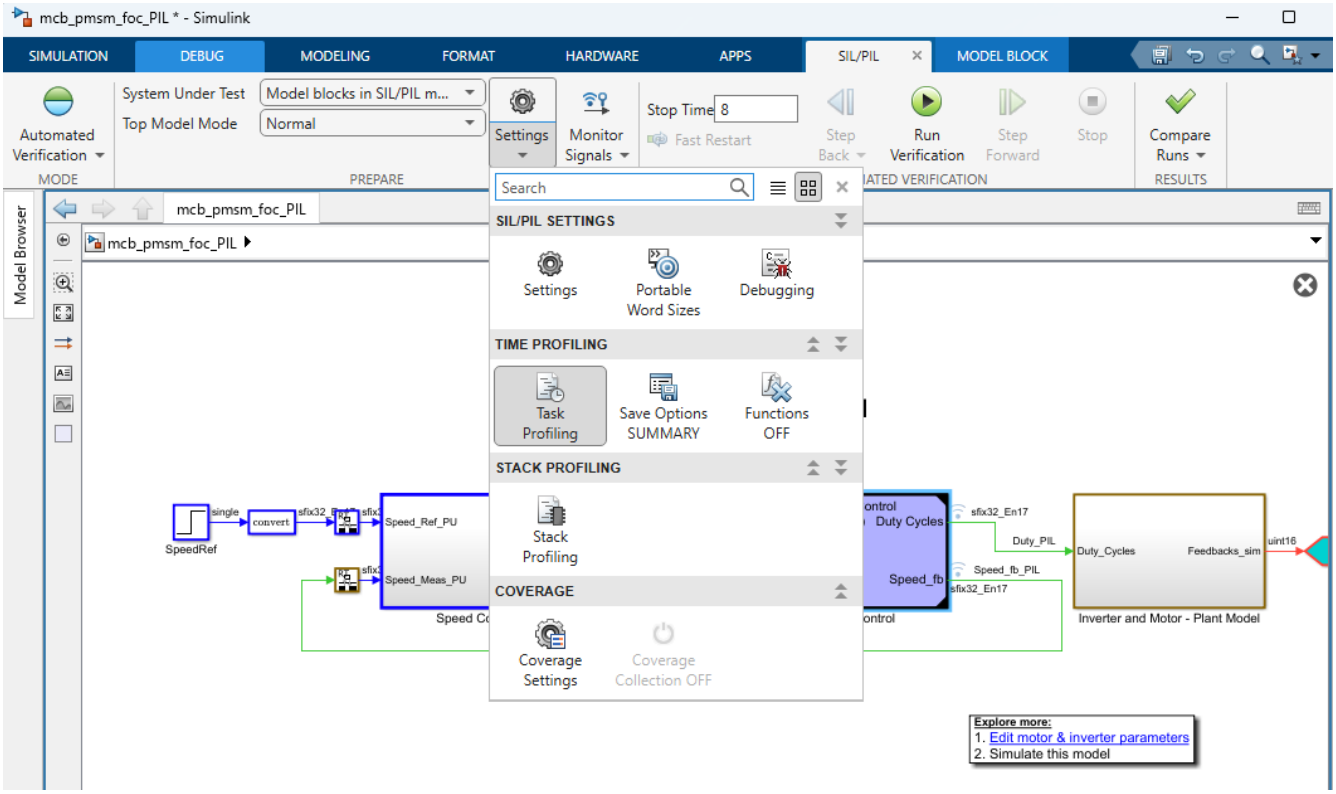


**Figure 7-1. Simulink Task Profiling**

### *7.1.1 Simulation Time Consideration*

PIL simulation is non-real time. Every time step requires data to travel from the Simulink host to the AM13x target and back over the serial communication link. For a motor control model with a fast base sample time, running the full simulation duration results in an impractically long PIL session. *Set the simulation stop time to 100 milliseconds (0.1)*, sufficient to capture meaningful profiling data and observe algorithm behavior without excessive run time.

1. Click *Run* in the SIL/PIL Manager to initiate PIL execution.
2. Simulink compiles and deploys the current control model to the AM13x, establishes the communication link, and begins exchanging data step by step.



Copyright © 2026 Texas Instruments Incorporated

### 7.1.2 PIL Results: Showing Numerical Verification


**Figure 7-2. Results**

### 7.1.3 PIL Results: Execution Profiling

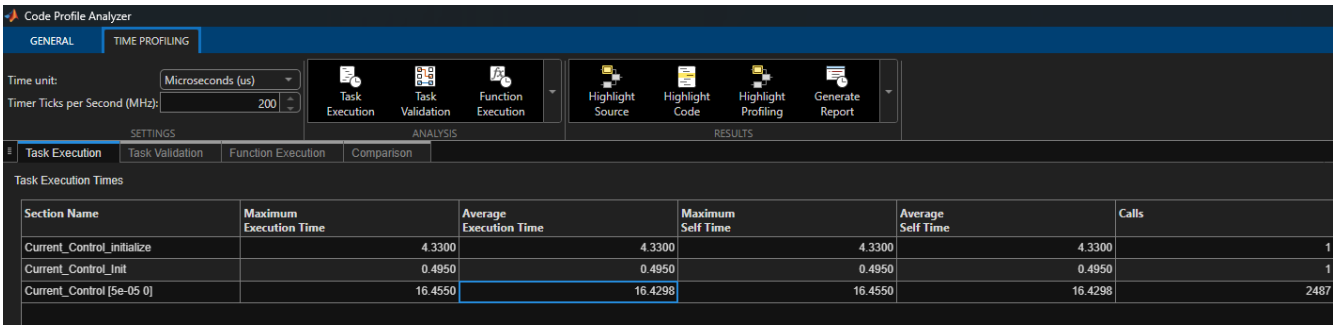


**Figure 7-3. Benchmark**



Copyright © 2026 Texas Instruments Incorporated

# 8 Optimization and Re-Profiling

The PIL benchmarking workflow is inherently iterative. Once a baseline profiling result is established, the engineer can apply code generation and algorithm optimizations and re-run PIL to measure the precise impact of each change.

Known optimization levers for the AM13x include:

- **TMU Code Replacement Library**: automatically enabled upon AM13x target selection; replaces standard trigonometric function calls with TMU hardware-accelerated equivalents, reducing execution time for FOC algorithms significantly.
- **Compiler Optimization Level**: adjustable in *Model Configuration Parameters → Code Generation → Build configuration*
- **Data Type Refinement**: reducing unnecessary double-precision operations to single-precision where algorithm accuracy permits
- **Lookup Table Approximations**: replacing transcendental function calls with table-based approximations where TMU is not applicable
- **Additional Simulink Code Generation Optimizations** as documented in *Optimization Strategies for Generated Code*
- For each optimization applied, repeat the PIL run and record the updated profiling results. Comparing baseline and optimized profiling numbers provides a quantitative, hardware-validated measure of optimization effectiveness on the AM13x target.

## 8.1 PIL Results: With Single Datatype and Sine/Cosine TMU

- Change the datatype to **single** from Fixed-point from mcb_pmsm_foc_sim_data.m

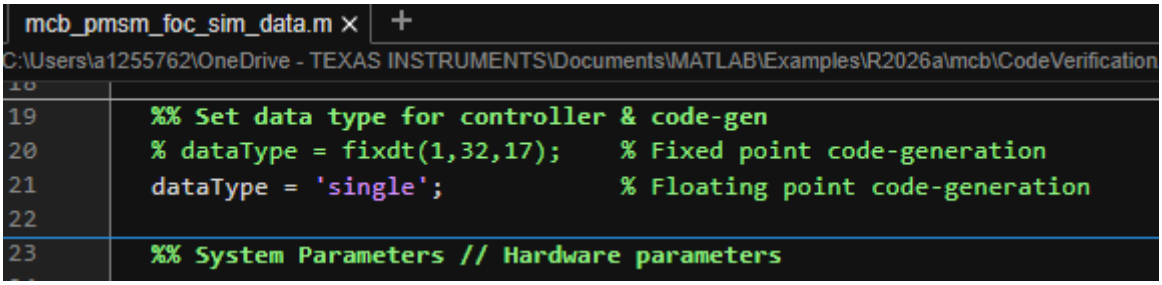


**Figure 8-1. Datatype change**

- Change the approximation method from Lookup to none to generate sine/cosine functions

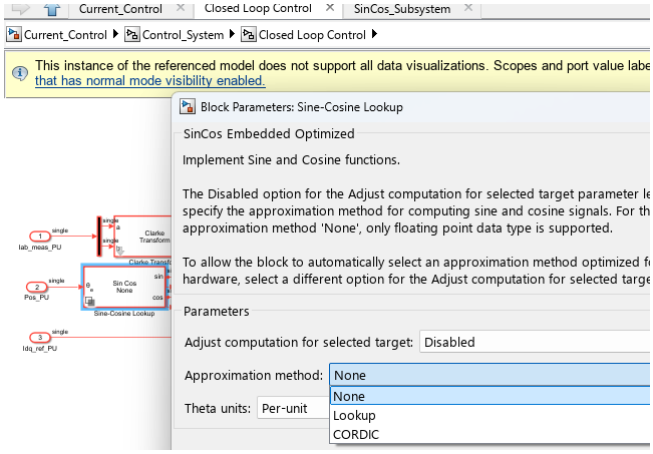


**Figure 8-2. Approximation method**

- Now the code generation replaces the sine/cosine with TMU functions sinf_tmu/cosf_tmu



Copyright © 2026 Texas Instruments Incorporated

## 8.2 Codegen Report Showing TMU Functions

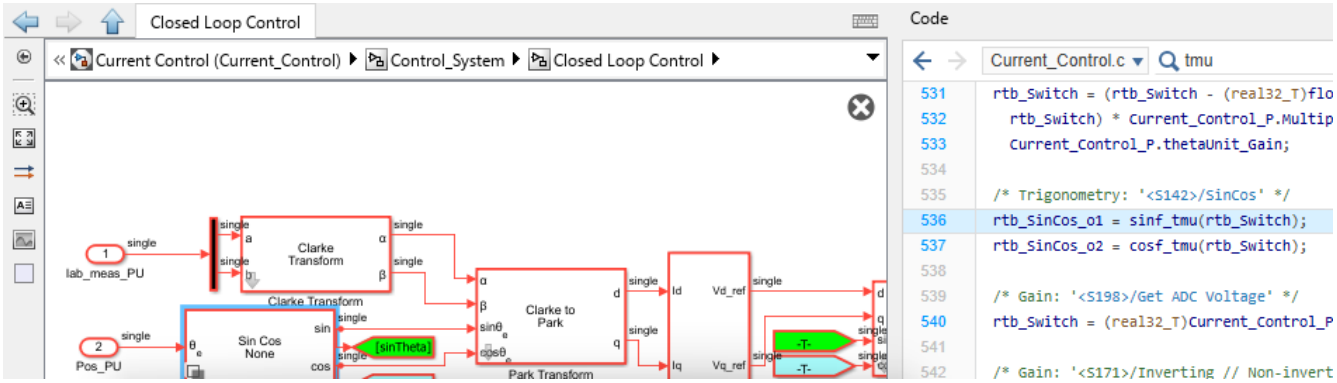


**Figure 8-3. TMU intrinsic replacement**

## 8.3 Enable Code View from Compare Runs Option

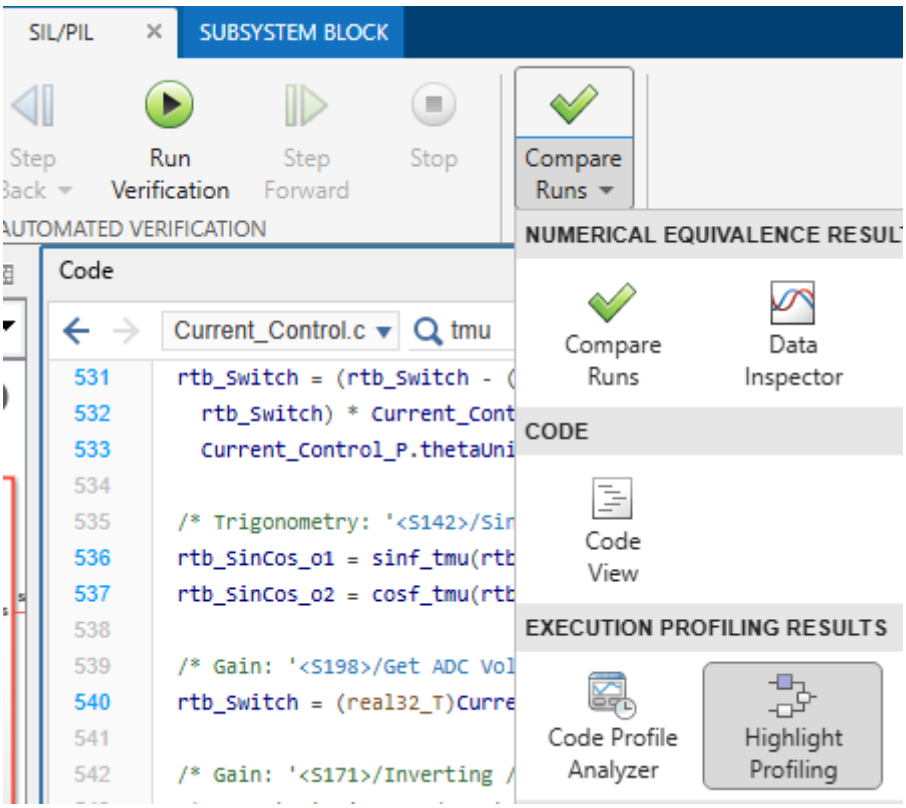


**Figure 8-4. Profiling**


Copyright © 2026 Texas Instruments Incorporated

## 8.4 PIL Results: Execution Profiling with Single Datatype and Sine/Cosine TMU

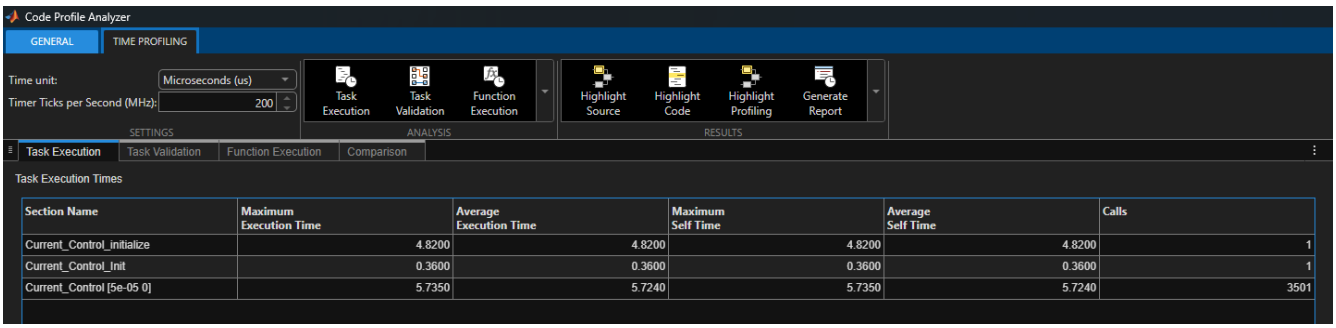


**Figure 8-5. Results with TMU**

## 9 Summary

This application note has presented a structured benchmarking workflow portable across algorithm model sources (C28x, other MCUs) to AM13x device. This will enable transition to AM13x smoother and faster.

## 10 References



Copyright © 2026 Texas Instruments Incorporated

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025