Motor Control Blockset™

Field-Oriented Control (FOC) of PMSM Using NXP™ S32K144 Development Kit

MATLAB®

R2021a

MathWorks®

How to Contact MathWorks

Latest news:	www.mathworks.com
Sales and services:	www.mathworks.com/sales_and_services
User community:	www.mathworks.com/matlabcentral
Technical support:	www.mathworks.com/support/contact_us
Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Motor Control Blockset™ (Nonrelease)

Revision History

March 2021	First printing	"Release for R2021a"

Contents

1 About This Example

2 Run 3-Phase PMSM in Open-Loop Control and Calibrate ADC Offset

3 Calibrate Hall Sensor Offset for PMSM Motor

4 Field-Oriented Control of PMSM Using Hall Sensor

About This Example

Introduction

This example implements a motor control system using the NXP MCSPTE1AK144 hardware. The MCSPTE1AK144 development kit, which uses the NXP S32K144 microcontroller, can run motor control applications designed for either a three-phase brushless DC (BLDC) or three-phase permanent magnet synchronous motor (PMSM). For more details about the hardware, see MCSPTE1AK144 development kit.

You can use this example to implement a field-oriented control (FOC) algorithm for the three-phase PMSM available in the MCSPTE1AK144 development kit.

This example also helps you:

- Verify your hardware setup using an open-loop control algorithm and calculate offset of the analog to digital converter (ADC).
- Calculate offset of the Hall sensors available in the three-phase PMSM.

The control algorithm available the example needs these offset values to optimally run the PMSM.

Follow the example workflows in this order to successfully run the PMSM using FOC:

Sequence	Workflow Title
Workflow - 1	Run 3-Phase PMSM in Open-Loop Control and Calibrate ADC Offset
Workflow - 2	Calibrate Hall Sensor Offset for PMSM Motor
Workflow - 3	Field-Oriented Control of PMSM Using Hall Sensor

For information regarding the FOC algorithm that is used in this example, see Field-Oriented Control (FOC).

Hardware Specifications

MCSPTE1AK144 development kit:

- Uses NXP S32K144 microcontroller.
- Uses 12 V DC power supply.
- Provides USB connectivity.
- Includes a three-phase PMSM (BLDC) containing three Hall sensors.

For detailed specifications, see MCSPTE1AK144 development kit.

Software Requirements

This section lists the software products from MathWorks® and NXP that you need to simulate and run the example models on the MCSPTE1AK144 development kit.

Required MathWorks Products

To simulate an example model:

- Motor Control Blockset

To generate code and deploy an example model:

- Motor Control Blockset
- Embedded Coder®

Required NXP Products

To simulate or deploy an example model:

- NXP Model-Based Design Toolbox (MBDT)

For instructions to download and install NXP MBDT, see NXP Model-Based Design Toolbox (MBDT).

Contents of Downloaded ZIP Folder

The ZIP folder that you downloaded from MATLAB File Exchange, includes:

File Name	Description	
mcb_nxp_MCSPTE1AK144_example_doc.pdf	This document.	
mcb_open_loop_control_MCSPTE1AK144.slx	Target model	for Workflow - 1 (Run 3-Phase PMSM in Open-Loop Control and Calibrate ADC Offset)
mcb_open_loop_control_MCSPTE1AK144_data.m	Target model Initialization script	
mcb_openloop_host.slx	Host model	
mcb_pmsm_hall_offset_MCSPTE1AK144.slx	Target model	for Workflow - 2 (Calibrate Hall Sensor Offset for PMSM Motor)
mcb_pmsm_foc_hall_MCSPTE1AK144_data.m	Target model Initialization script	
mcb_hall_offset_host.slx	Host model	
mcb_pmsm_foc_hall_MCSPTE1AK144.slx	Target model	for Workflow - 3 (Field-Oriented Control of PMSM Using Hall Sensor)
mcb_pmsm_foc_hall_MCSPTE1AK144_data.m	Target model Initialization script	
mcb_hall_foc_host.slx	Host model	

Each workflow in this example uses a host and a target model. The host model is a user interface to the controller hardware board. You can run the host model on the host computer. The prerequisite to use the host model is to deploy the target model to the controller hardware board. The host model uses serial communication to command the target Simulink® model and run the motor either in an open-loop or closed-loop control. For more details about the host and target model setup, see Host-Target Communication.

To learn more about configuring the target model initialization script, see Estimate Control Gains from Motor Parameters.

Run 3-Phase PMSM in Open-Loop Control and Calibrate ADC Offset

Introduction

This workflow uses open-loop control (also known as scalar control or Volts/Hz control) to run the permanent magnet synchronous motor (PMSM) available in the MCSPTE1AK144 development kit. This technique varies the stator voltage and frequency to control the rotor speed without using any feedback from the motor. You can use this technique to check the integrity of the hardware connections of the MCSPTE1AK144 development kit. A constant speed application of open-loop control uses a fixed-frequency motor power supply. An adjustable speed application of open-loop control needs a variable-frequency power supply to control the rotor speed. To ensure a constant stator magnetic flux, keep the supply voltage amplitude proportional to its frequency.

Open-loop motor control does not have the ability to consider the external conditions that can affect the motor speed. Therefore, the control system cannot automatically correct the deviation between the desired and the actual motor speed.

This workflow helps you run the motor by using an open-loop motor control algorithm. It helps you get started with Motor Control Blockset™ and verify the hardware setup by running the motor. The target model algorithm also reads the ADC values from the current sensors and sends these values to the host model by using serial communication.

You can use this model to:

- Check connectivity with the target.
- Check serial communication with the target.
- Verify the hardware and software environment.
- Check ADC offsets for current sensors.
- Run a new motor with an inverter and target setup for the first time.

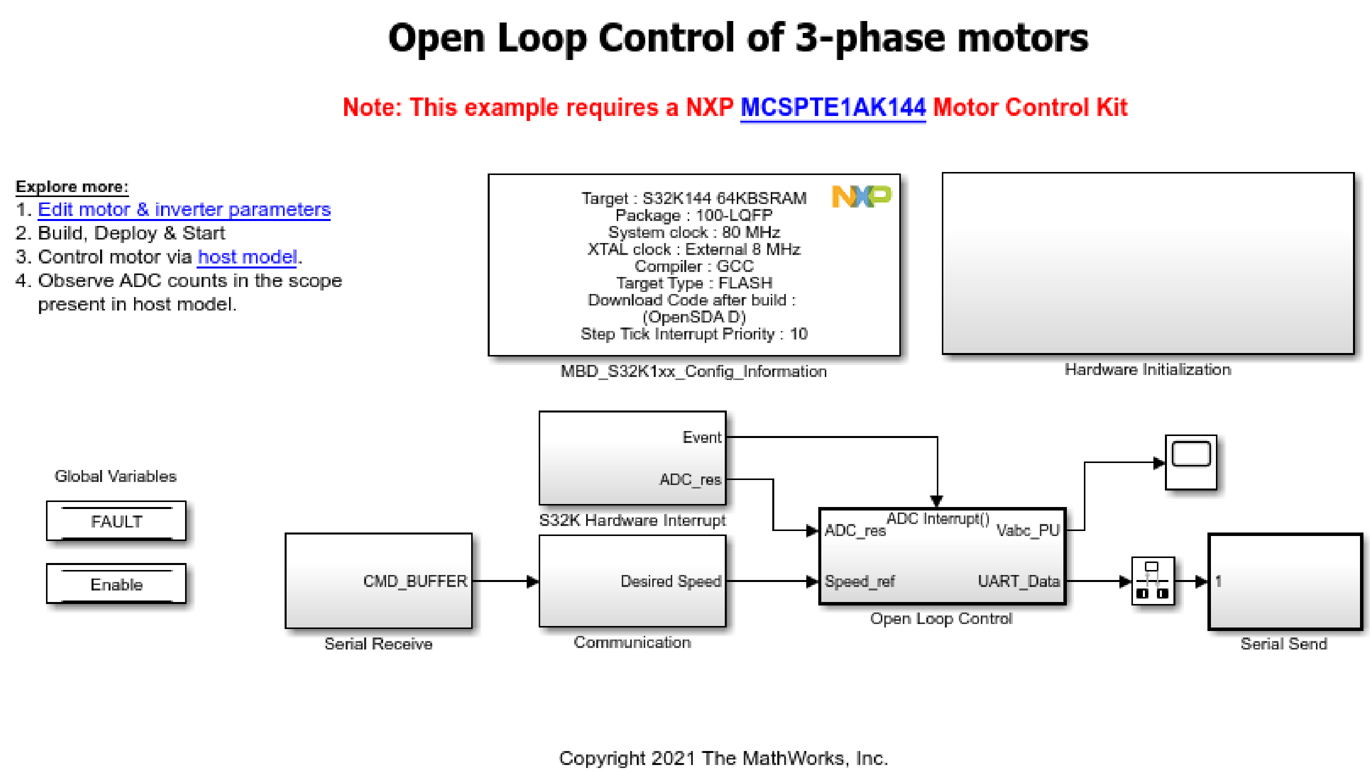
Models

The example includes these models:

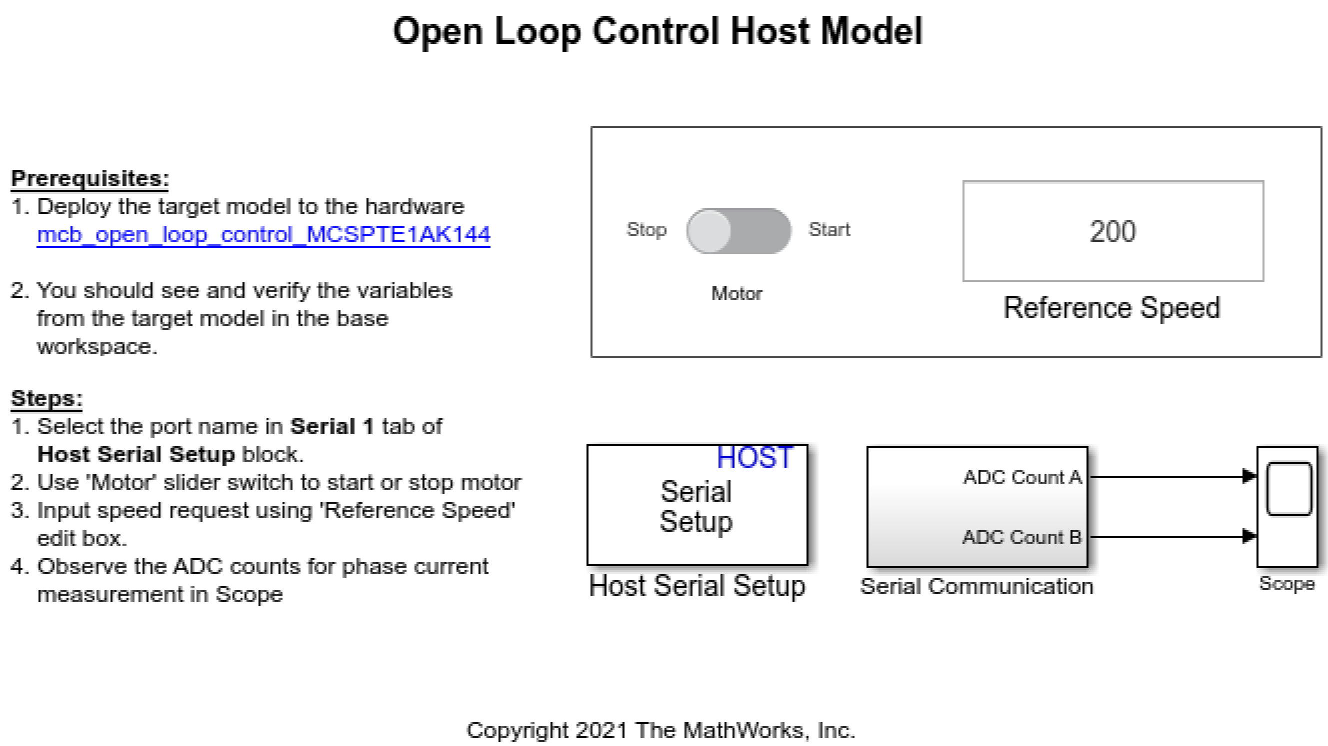
- mcb_open_loop_control_MCSPTE1AK144.slx (target model)
- mcb_openloop_host.slx (host model)

You can use these models for both simulation and code generation. You can use the `open_system` command (or use the target model name) at the MATLAB command prompt to open the target and host models.

```
open_system('mcb_open_loop_control_MCSPTE1AK144.slx');
```



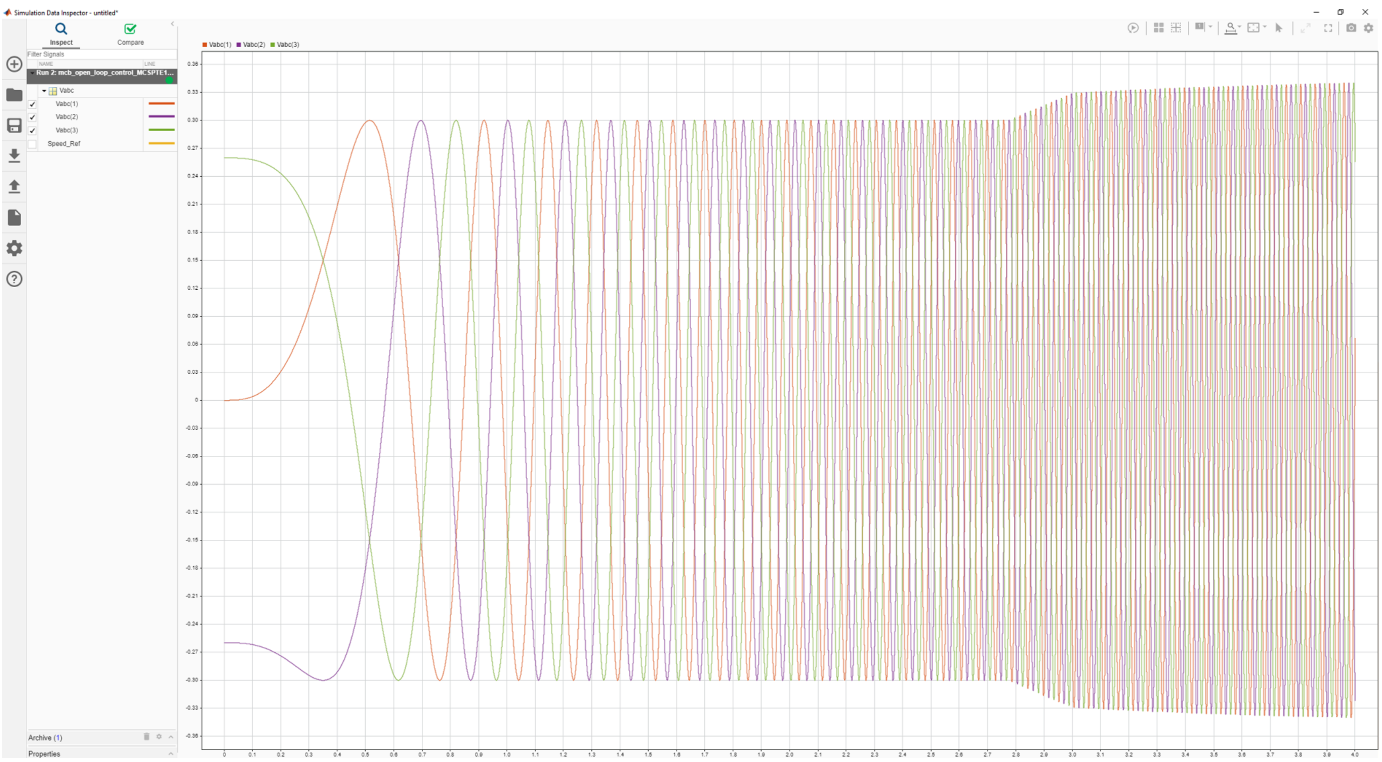

```
open_system('mcb_openloop_host.slx');
```


Open Loop Control Host Model
Prerequisites:
1. Deploy the target model to the hardware mcb_open_loop_control_MCSPTE1AK144
2. You should see and verify the variables from the target model in the base workspace.
Steps:
1. Select the port name in Serial 1 tab of Host Serial Setup block.
2. Use 'Motor' slider switch to start or stop motor
3. Input speed request using 'Reference Speed' edit box.
4. Observe the ADC counts for phase current measurement in Scope
Stop
Start
Motor
200
Reference Speed
HOST
Serial Setup
Host Serial Setup
ADC Count A
ADC Count B
Serial Communication
Scope
Copyright 2021 The MathWorks, Inc.

Simulate Target Model

This example supports simulation. Follow these steps to simulate the model.

1. Open the target model included in this workflow.
2. Click **Run** on the **Simulation** tab to simulate the model.
3. Click **Data Inspector** on the **Simulation** tab to view and analyze the simulation results.



Generate Code and Deploy Model to Target Hardware

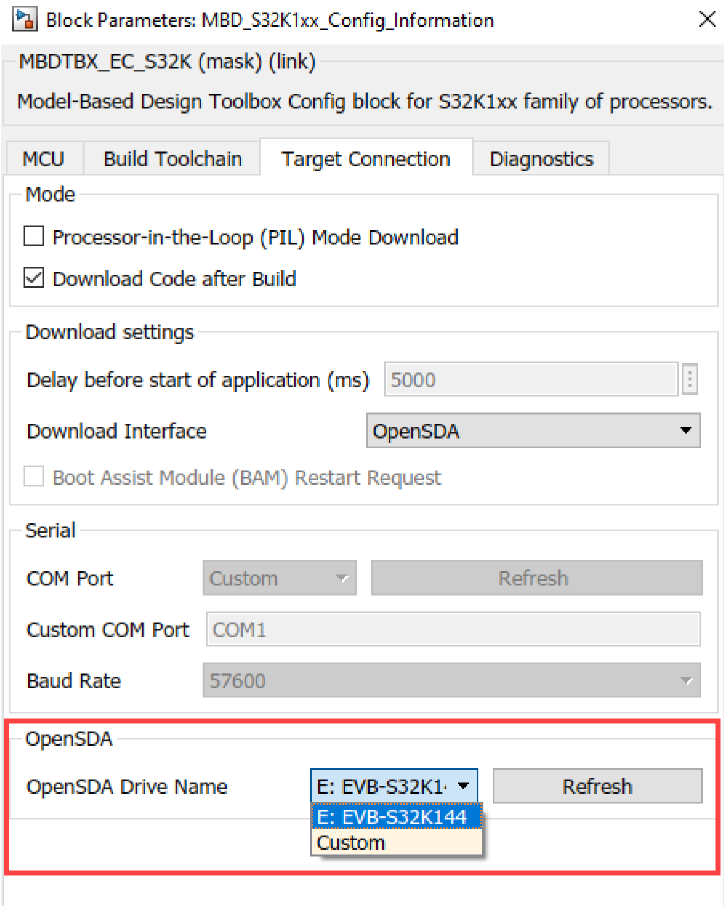
This section explains how to generate code and run the motor by using open-loop control. The example uses a host and a target model. The host model is a user interface to the controller hardware board. You can run the host model on the host computer. The prerequisite to use the host model is to deploy the target model to the controller hardware board. The host model uses serial communication to command the target Simulink model and run the motor in a open-loop control.

Note The PMSM may not run at higher speeds, especially when the shaft is loaded. To resolve this issue, you should apply more voltages corresponding to a given frequency. You can use these steps to increase the applied voltages in the model:

1. Navigate to this path in the model: `/Open Loop Control/Control_System/VabcCalc/`.
2. Update the gain `Correction_Factor_sinePWM` as 20%.
3. For safety reasons, regularly monitor the motor shaft, motor current, and motor temperature.

Generate Code and Run Model to Implement Open-Loop Control

1. Simulate the target model and observe the simulation results.
2. Connect the MCSPTE1AK144 development kit to the host computer.
3. Open the target model.
4. Open the MBD_S32K1xx_Config_Information block parameters dialog box and select the drive on which the hardware is mounted in the **OpenSDA Drive Name** field available in the **Target Connection** tab.



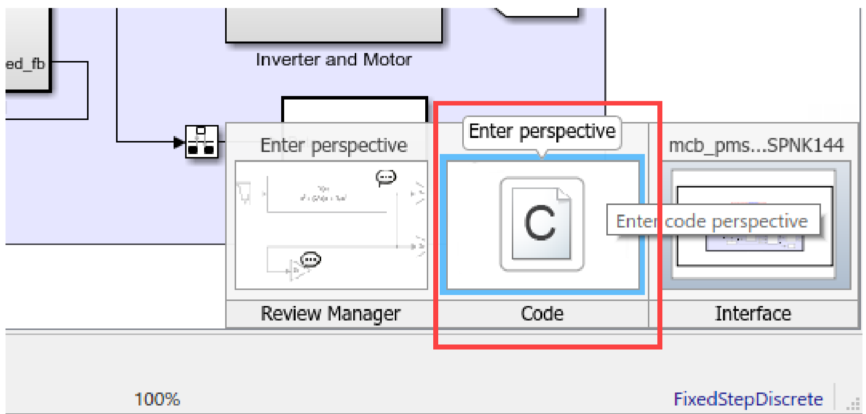
5. Click the **Edit motor & inverter parameters** hyperlink available in the target model to open the target model initialization script. You can also use this command to open the initialization script.

   ```
   edit mcb_open_loop_control_MCSPTE1AK144_data.m
   ```

6. Verify and edit the motor and inverter parameters that are pre-configured in the model initialization script `mcb_open_loop_control_MCSPTE1AK144_data.m`.

   For instructions to configure the script, see Estimate Control Gains from Motor Parameters.

7. Click the Show Perspectives views button  on the bottom-right corner of the target model and select **Code** perspective.



8 Click **Build** on the **C Code** tab (or press **Ctrl+B**) to build and deploy the target model to the hardware.

Note Ignore the warning message "Multitask data store option in the Diagnostics page of the Configuration Parameter Dialog is none" displayed by the model advisor, by clicking the Always Ignore button. This is part of the intended workflow.

---

9 Verify the variables updated by the target model in the MATLAB base workspace.

10 Click the **host model** hyperlink in the target model to open the associated host model. You can also use the `open_system` command to open the host model.

For details about the serial communication between the host and target models, see Host-Target Communication.

11 In the Host Serial Setup block parameters dialog box of the host model, select a **Port name**.

12 Click **Run** on the **Simulation** tab to run the host model.

13 Change the position of the Start / Stop Motor switch to On, to start running the motor.

14 Enter the **Reference Speed** value in the host model.

15 After the motor is running, observe the ADC counts for the *a* and *b* phase currents in the Time Scope.

Note This example may not allow the motor to run at full capacity. Begin running the motor at a small speed. In addition, it is recommended to change the Reference Speed in small steps (for example, for a motor having a base speed of 3000 rpm, start running the motor at 500 rpm and then increase or decrease the speed in steps of 200 rpm).

If the motor does not run, change the position of the Start / Stop Motor switch to Off, to stop the motor and change the Reference Speed in the host model. Then, change the position of the Start / Stop Motor switch to On, to run the motor again.

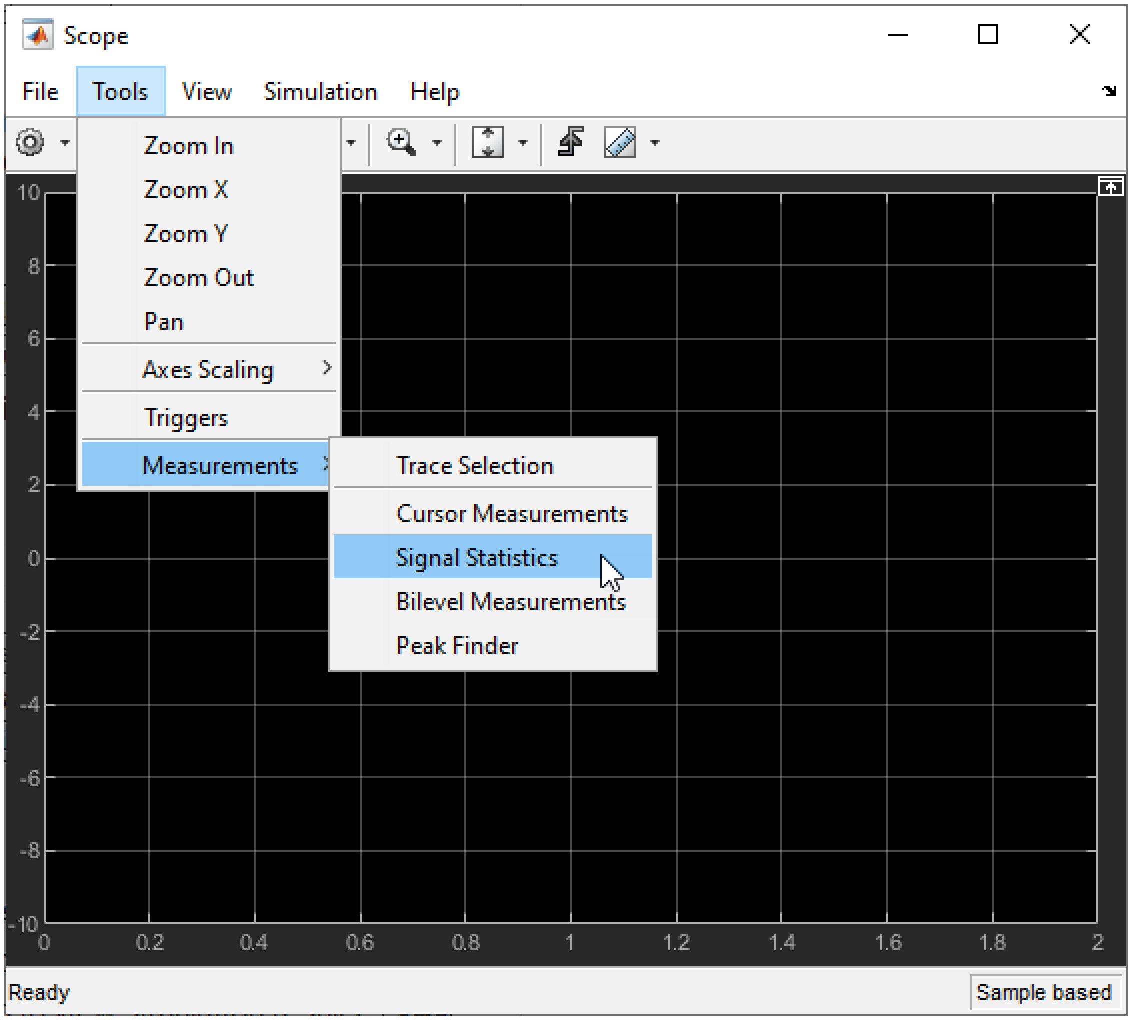
Generate Code and Run Model to Calibrate ADC Offset

1. Simulate the target model and observe the simulation results.
2. Connect the MCSPTE1AK144 development kit to the host computer.
3. Disconnect the motor wires for three phases from the hardware board terminals.
4. Open the target model.
5. Open the MBD_S32K1xx_Config_Information block parameters dialog box and select the drive on which the hardware is mounted in the **OpenSDA Drive Name** field available in the **Target Connection** tab.
6. Click the Show Perspectives views button  on the bottom-right corner of the target model and select **Code** perspective.
7. Click **Build** on the **C Code** tab (or press **Ctrl+B**) to build and deploy the target model to the hardware.

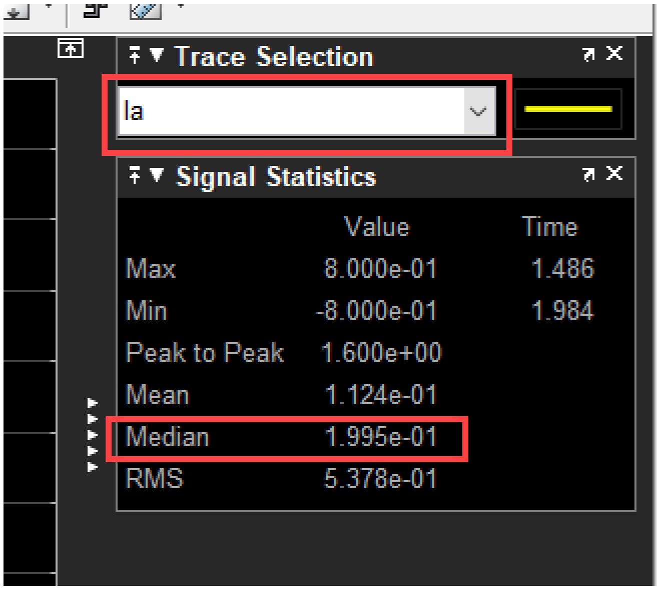
   Note Ignore the warning message "Multitask data store option in the Diagnostics page of the Configuration Parameter Dialog is none" displayed by the model advisor, by clicking the Always Ignore button. This is part of the intended workflow.
8. Verify the variables updated by the target model in the MATLAB base workspace.
9. Click the **host model** hyperlink in the target model to open the associated host model. You can also use the `open_system` command to open the host model.

   For details about the serial communication between the host and target models, see Host-Target Communication.
10. In the Host Serial Setup block parameters dialog box of the host model, select a **Port name**.
11. Click **Run** on the **Simulation** tab to run the host model.
12. Observe the ADC counts for the *a* and *b* phase currents (I_a and I_b) in the Time Scope. The average values of the ADC counts are the ADC offset corrections for the currents I_a and I_b. Follow these steps to obtain the average (median) values of ADC counts:

    - In the **Scope** window, navigate to **Tools** > **Measurements** and select **Signal Statistics** to display the **Trace Selection** and **Signal Statistics** areas.



- Under **Trace Selection**, select a signal (I_a or I_b). The characteristics of the selected signal are displayed in the **Signal Statistics** pane. You can see the median value of the selected signal in the Median field.



Update the ADC offset correction values (for the currents I_a and I_b) in the *inverter.CtSensAOffset* and *inverter.CtSensBOffset* variables in the target model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m` of workflow - 3 (Field-Oriented Control of PMSM Using Hall Sensor).

Calibrate Hall Sensor Offset for PMSM Motor

Introduction

This workflow calculates the offset between the rotor direct axis (d-axis) and position detected by the Hall sensors available in the permanent magnet synchronous motor (PMSM) of the MCSPTE1AK144 development kit. The field-oriented control (FOC) algorithm needs this position offset to run the PMSM correctly. To compute the offset, the target model runs the motor in the open-loop condition. The model uses a constant V_d (voltage along the stator's *d*-axis) and a zero V_q (voltage along the stator's *q*-axis) to run the motor (at a low constant speed) by using a position or ramp generator. When the position or ramp value reaches zero, the corresponding rotor position is the offset value for the Hall sensors.

The control algorithm in workflow - 3 uses this offset value to compute an accurate position of *d*-axis of the rotor. The controller needs this offset to optimally run the PMSM.

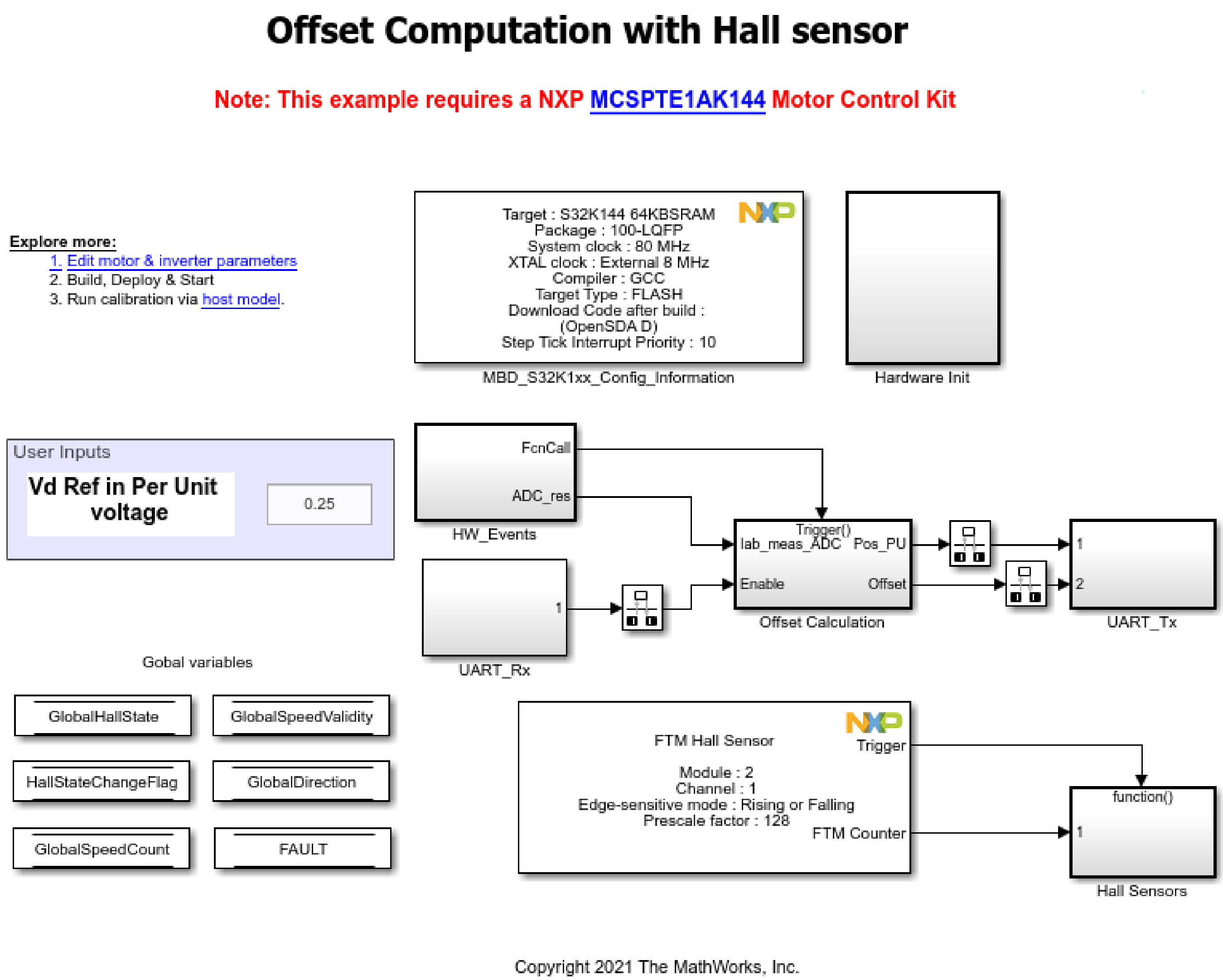
Models

The example includes these models:

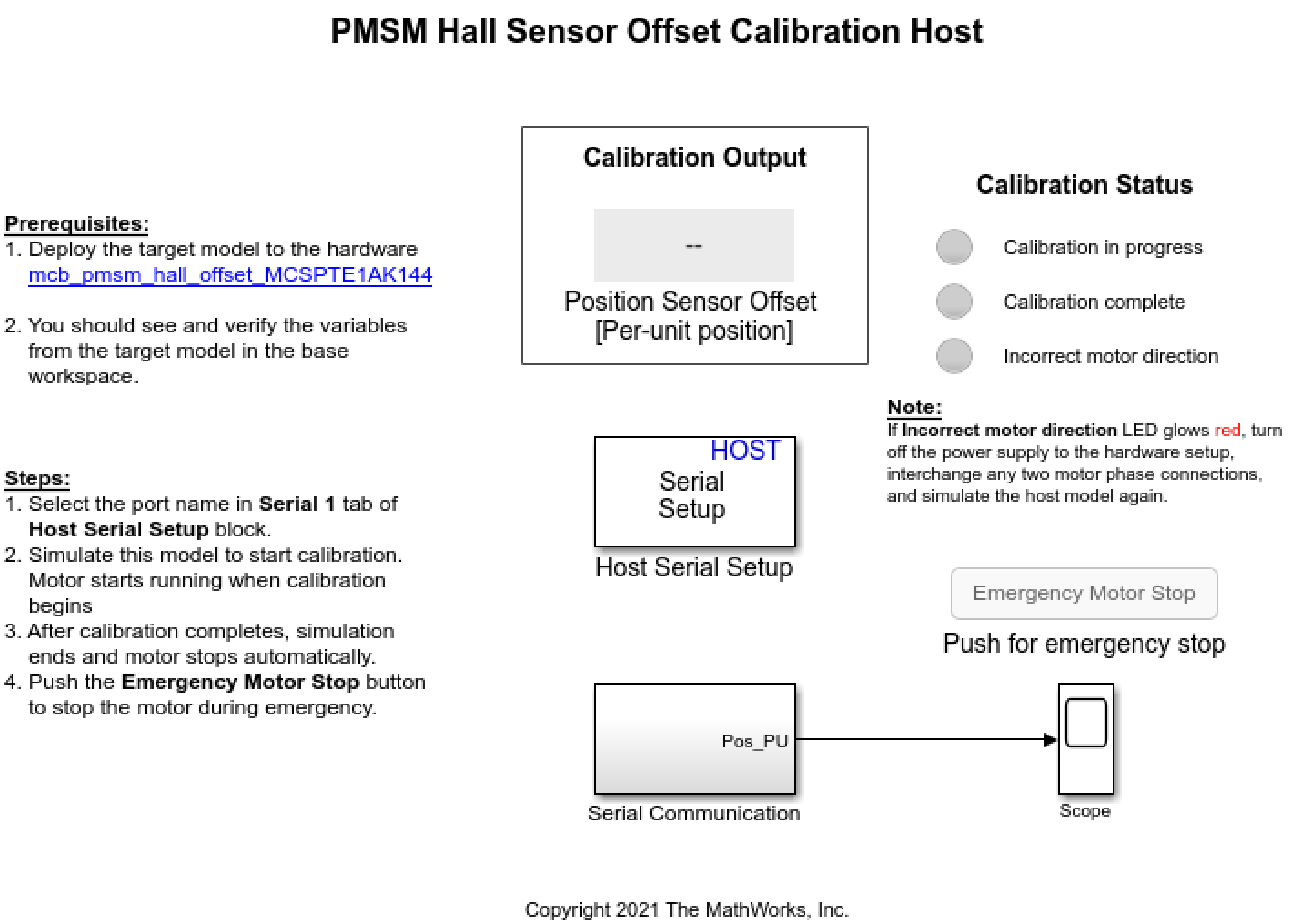
- mcb_pmsm_hall_offset_MCSPTE1AK144.slx (target model)
- mcb_hall_offset_host.slx (host model)

You can use these models only for code generation. You can use the `open_system` command (or use the target model name) at the MATLAB command prompt to open the target and host models.

```
open_system('mcb_pmsm_hall_offset_MCSPTE1AK144.slx');
```




```
open_system('mcb_hall_offset_host.slx');
```


PMSM Hall Sensor Offset Calibration Host
Prerequisites:
1. Deploy the target model to the hardware mcb_pmsm_hall_offset_MCSPTE1AK144
2. You should see and verify the variables from the target model in the base workspace.
Steps:
1. Select the port name in Serial 1 tab of Host Serial Setup block.
2. Simulate this model to start calibration. Motor starts running when calibration begins
3. After calibration completes, simulation ends and motor stops automatically.
4. Push the Emergency Motor Stop button to stop the motor during emergency.
Calibration Output
--
Position Sensor Offset [Per-unit position]
Calibration Status
Calibration in progress
Calibration complete
Incorrect motor direction
Note:
If Incorrect motor direction LED glows red, turn off the power supply to the hardware setup, interchange any two motor phase connections, and simulate the host model again.
HOST
Serial Setup
Host Serial Setup
Emergency Motor Stop
Push for emergency stop
Pos_PU
Serial Communication
Scope
Copyright 2021 The MathWorks, Inc.

Generate Code and Deploy Model to Target Hardware

This section explains how to generate code and run the motor by using open-loop control. This workflow uses a host and a target model. The host model is a user interface to the controller hardware board. You can run the host model on the host computer. The prerequisite to use the host model is to deploy the target model to the controller hardware board.

The host model uses serial communication to command the target model and run the motor in an open-loop configuration. You can use the host model to control the motor rotations and validate direction of rotation of the motor. The **Incorrect motor direction** LED in the host model turns red to indicate that the motor is running in the opposite direction. When the LED turns red, you must reverse the motor phase connections to change the direction of rotation. The host model displays the calculated offset value.

Follow these steps to deploy and run the target model on the MCSPTE1AK144 development kit.

1. Connect the MCSPTE1AK144 development kit to the host computer.
2. Open the target model.
3. Open the MBD_S32K1xx_Config_Information block parameters dialog box and select the drive on which the hardware is mounted in the **OpenSDA Drive Name** field available in the **Target Connection** tab.
4. Click the **Edit motor & inverter parameters** hyperlink available in the target model to open the target model initialization script. You can also use this command to open the initialization script.

   ```
   edit mcb_pmsm_foc_hall_MCSPTE1AK144_data.m
   ```
5. Verify and edit the motor and inverter parameters that are pre-configured in the model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m`.

   For instructions to configure the script, see Estimate Control Gains from Motor Parameters.
6. In the **Vd Ref in Per Unit voltage** field under the **User Inputs** area, enter the constant V_d value.
7. Click the Show Perspectives views button  on the bottom-right corner of the target model and select **Code** perspective.
8. Click **Build** on the **C Code** tab (or press **Ctrl+B**) to build and deploy the target model to the hardware.
9. Verify the variables updated by the target model in the MATLAB base workspace.
10. Click the **host model** hyperlink in the target model to open the associated host model. You can also use the `open_system` command to open the host model.

    For details about the serial communication between the host and target models, see Host-Target Communication.
11. In the Host Serial Setup block parameters dialog box of the host model, select a **Port name**.
12. Click **Run** on the **Simulation** tab to run the host model. The motor runs and calibration begins when you start simulation. After the calibration process is complete, simulation ends and the motor stops automatically.
13. See the **Calibration Status** section to know the status of the calibration process:

- The **Calibration in progress** LED turns orange when the motor starts running. Notice the rotor position and the variation in the offset value in the Scope (the position signal indicates a ramp signal with an amplitude between 0 and 1). After the calibration process is complete, the LED turns grey.
- The **Calibration complete** LED turns green when the calibration process is complete. Then the **Calibration Output** field displays the computed offset value.
- The **Incorrect motor direction** LED turns red if the motor runs in the opposite direction. Then the **Calibration Output** field displays the value "NaN." Turn off the DC power supply (24V) and reverse the motor phase connections from ABC to CBA. Repeat steps 6 to 10 and check if the **Calibration complete** LED is green. Verify that the **Calibration Output** field displays the offset value.

Note To immediately stop the motor, click the **Emergency Motor Stop** button.

This example does not support simulation. The example automatically saves the computed offset value in the *PositionOffset* variable available in the base workspace.

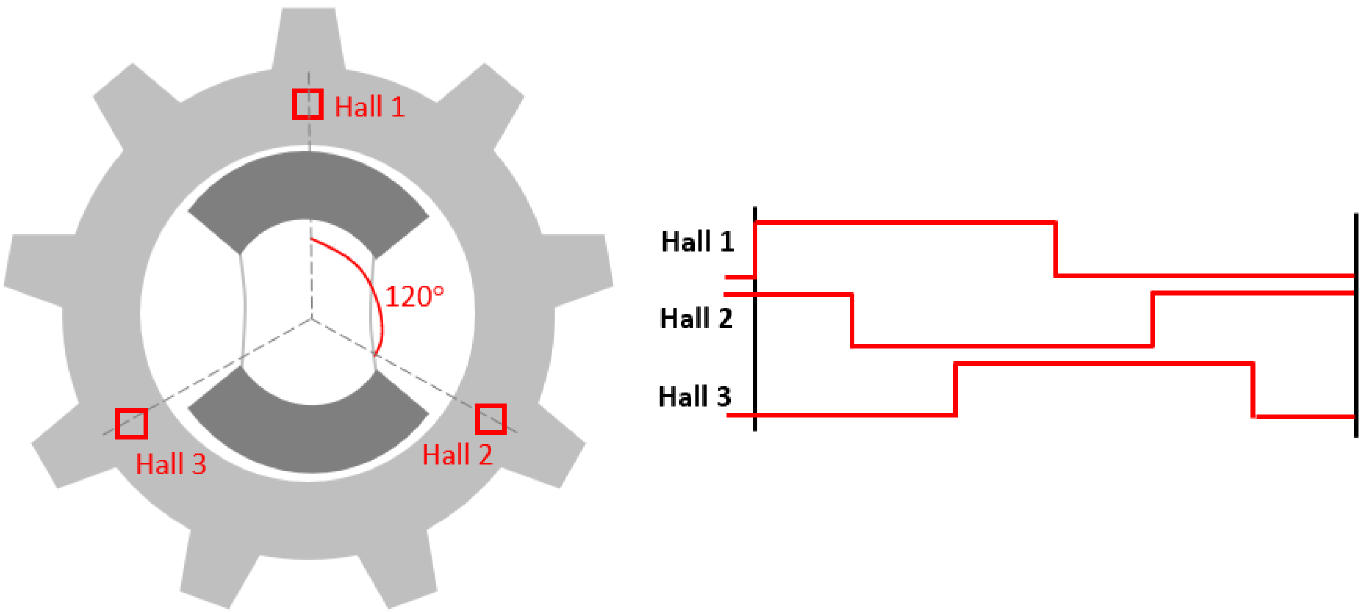
Update the computed Hall sensor offset value in the *pmsm.PositionOffset* variable in the target model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m` of workflow - 3 (Field-Oriented Control of PMSM Using Hall Sensor).

Field-Oriented Control of PMSM Using Hall Sensor

Introduction

This workflow implements the field-oriented control (FOC) technique to control the speed of a three-phase permanent magnet synchronous motor (PMSM) available in the MCSPTE1AK144 development kit. The FOC algorithm requires rotor position feedback, which is obtained by the Hall sensors available inside the PMSM. For details about FOC, see Field-Oriented Control (FOC).

This example uses the Hall sensor to measure the rotor position. A Hall effect sensor varies its output voltage based on the strength of the applied magnetic field. A PMSM consists of three Hall sensors located electrically 120 degrees apart. A PMSM with this setup can provide six valid combinations of binary states (for example, 001,010,011,100,101, and 110). The sensor provides the angular position of the rotor in the multiples of 60 degrees, which the controller uses to compute the angular velocity. The controller can then use the angular velocity to compute an accurate angular position of the rotor.



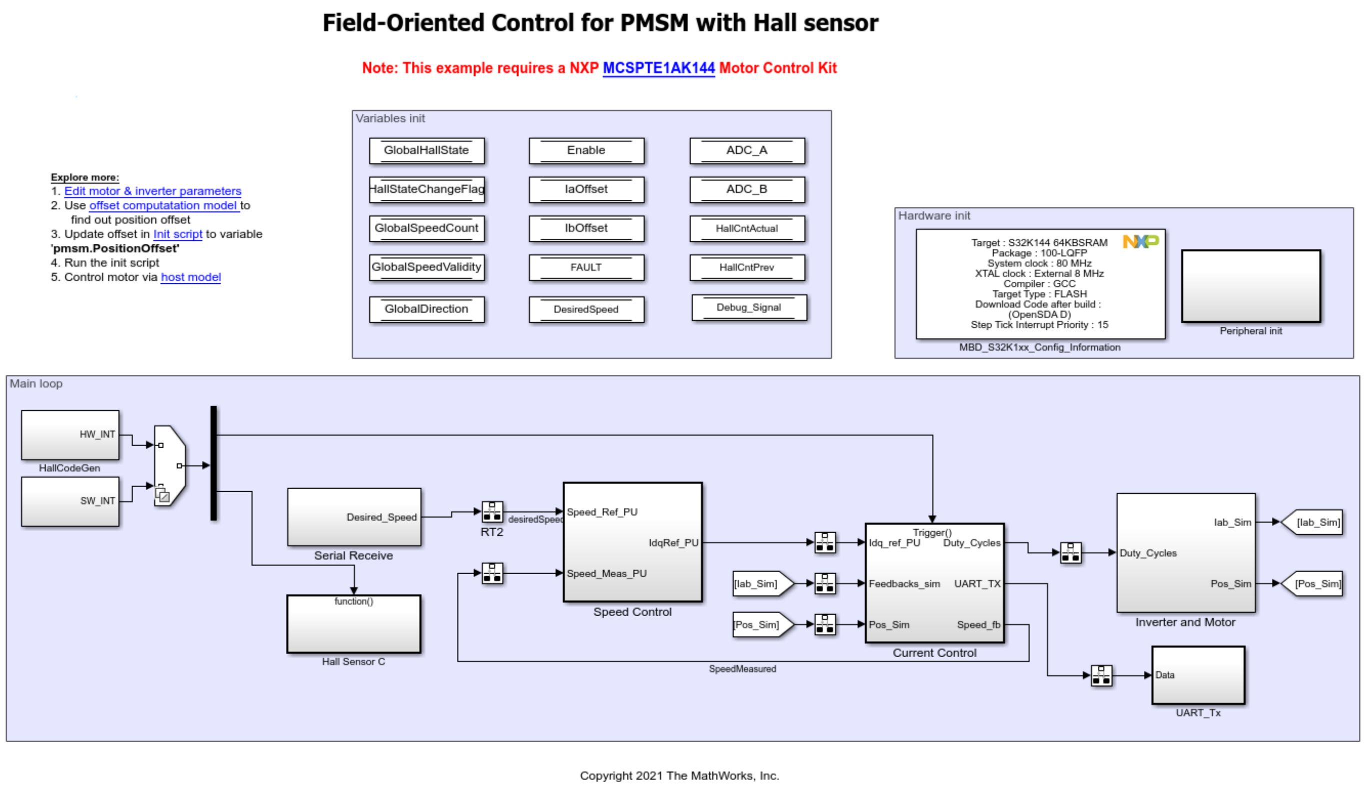
Models

The example includes these models:

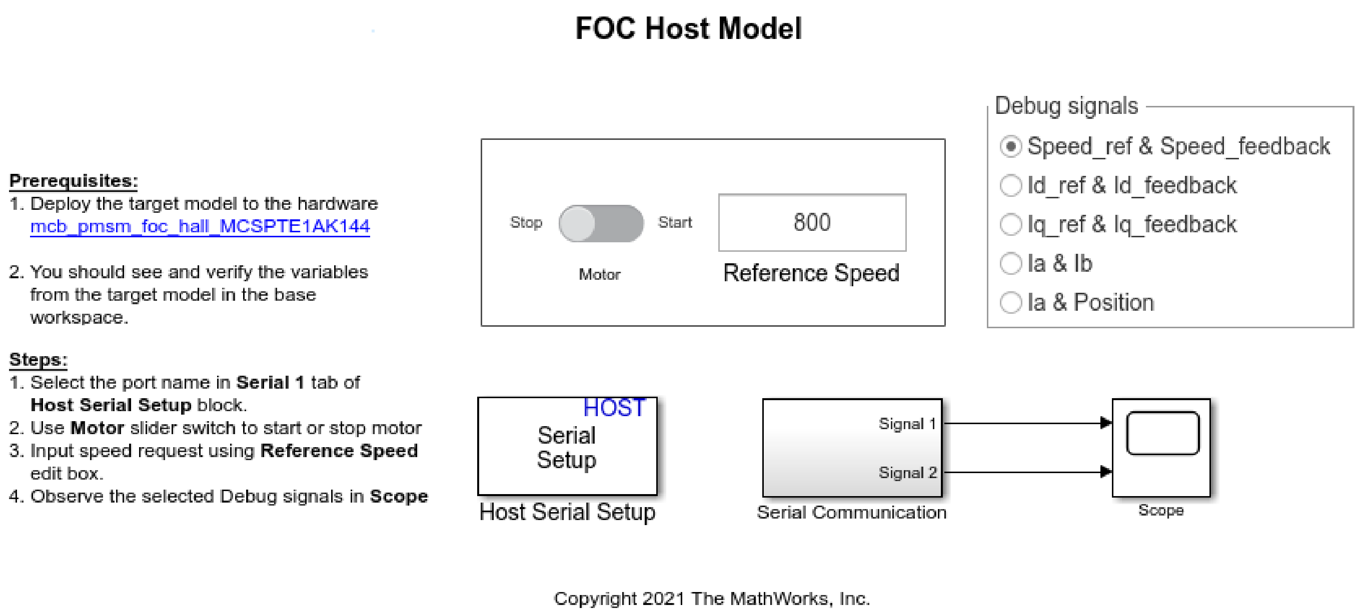
- mcb_pmsm_foc_hall_MCSPTE1AK144.slx (target model)
- mcb_hall_foc_host.slx (host model)

You can use these models for both simulation and code generation. You can use the `open_system` command (or use the target model name) at the MATLAB command prompt to open the target and host models.

```
open_system('mcb_pmsm_foc_hall_MCSPTE1AK144.slx');
```



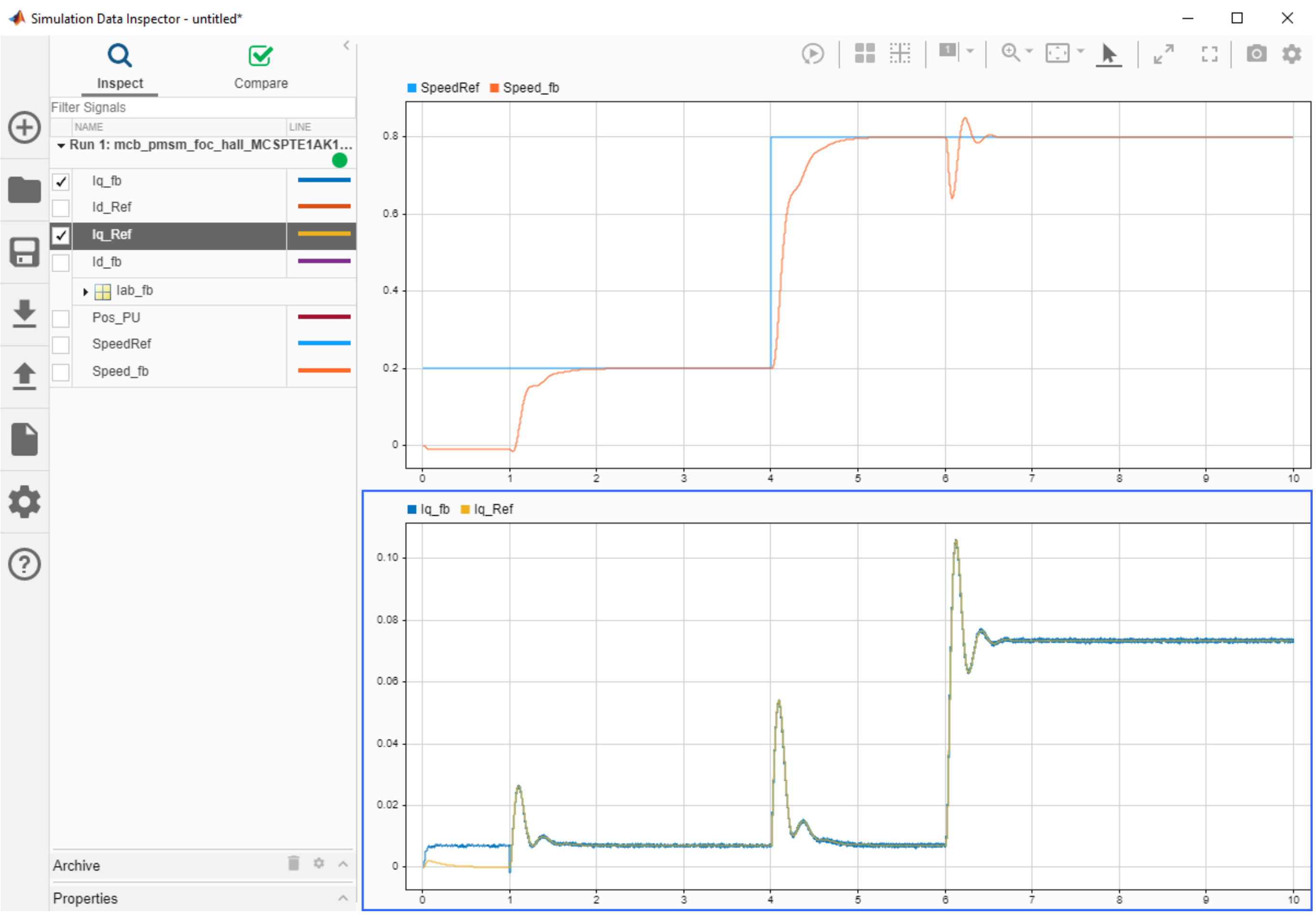

```
open_system('mcb_hall_foc_host.slx');
```


FOC Host Model
Prerequisites:
1. Deploy the target model to the hardware mcb_pmsm_foc_hall_MCSPTE1AK144
2. You should see and verify the variables from the target model in the base workspace.
Steps:
1. Select the port name in Serial 1 tab of Host Serial Setup block.
2. Use Motor slider switch to start or stop motor
3. Input speed request using Reference Speed edit box.
4. Observe the selected Debug signals in Scope
Stop
Start
Motor
800
Reference Speed
Debug signals
Speed_ref & Speed_feedback
Id_ref & Id_feedback
Iq_ref & Iq_feedback
Ia & Ib
Ia & Position
HOST
Serial Setup
Host Serial Setup
Signal 1
Signal 2
Serial Communication
Scope
Copyright 2021 The MathWorks, Inc.

Simulate Target Model

This example supports simulation. Follow these steps to simulate the model.

1. Open the target model included in this workflow.
2. Click **Run** on the **Simulation** tab to simulate the model.
3. Click **Data Inspector** on the **Simulation** tab to view and analyze the simulation results.



Generate Code and Deploy Model to Target Hardware

This section explains how to generate code and run the FOC algorithm on the target hardware.

This example uses a host and a target model. The host model is a user interface to the controller hardware board. You can run the host model on the host computer. The prerequisite to use the host model is to deploy the target model to the controller hardware board. The host model uses serial communication to command the target Simulink model and run the motor in a closed-loop control.

Follow these steps to deploy and run the target model on the MCSPTE1AK144 development kit.

1. Simulate the target model and observe the simulation results.
2. Connect the MCSPTE1AK144 development kit to the host computer.
3. Follow workflow - 1 to calculate the ADC offset values and update them in the *inverter.CtSensAOffset* and *inverter.CtSensBOffset* variables in the target model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m`.
4. Follow workflow - 2 to calculate the Hall sensor offset value and update it in the *pmsm.PositionOffset* variable in the target model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m`.
5. Open the target model.
6. Open the MBD_S32K1xx_Config_Information block parameters dialog box and select the drive on which the hardware is mounted in the **OpenSDA Drive Name** field available in the **Target Connection** tab.
7. Click the **Edit motor & inverter parameters** hyperlink available in the target model to open the target model initialization script. You can also use this command to open the initialization script.

   ```
   edit mcb_pmsm_foc_hall_MCSPTE1AK144_data.m
   ```
8. Verify and edit the motor and inverter parameters that are pre-configured in the model initialization script `mcb_pmsm_foc_hall_MCSPTE1AK144_data.m`.

   For instructions to configure the script, see Estimate Control Gains from Motor Parameters.
9. Click the Show Perspectives views button  on the bottom-right corner of the target model and select **Code** perspective.
10. Click **Build** on the **C Code** tab (or press **Ctrl+B**) to build and deploy the target model to the hardware.
11. Verify the variables updated by the target model in the MATLAB base workspace.
12. Click the **host model** hyperlink in the target model to open the associated host model. You can also use the `open_system` command to open the host model.

    For details about the serial communication between the host and target models, see Host-Target Communication.
13. In the Host Serial Setup block parameters dialog box of the host model, select a **Port name**.
14. Update the **Reference Speed** value in the host model.
15. Change the position of the Start / Stop Motor switch to On, to start running the motor.
16. Observe the debug signals that you select in the **Debug signals** area, in the Time Scope of the host model. You can select one of the following options:

- **Speed_ref & Speed_feedback** — Display speed reference and speed feedback signals.
- **Id_ref & Id_feedback** — Display I_d reference and I_d feedback current signals.
- **Iq_ref & Iq_feedback** — Display I_q reference and I_q feedback current signals.
- **Ia & Ib** — Display I_a and I_b current signals.
- **Ia & Position** — Display I_a current and rotor-position signals.