

# Motor Control Blockset™

Sensorless Field-Oriented Control (FOC) of PMSM  
Using NXP™ S32K344 Development Kit



# MATLAB®

R2024b



# How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://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)*

© COPYRIGHT 2025 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

## Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

## Patents

MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

## Revision History

September 2025	First printing	"Release for R2024b"
----------------	----------------	----------------------

<b>1</b>	<b>About This Example</b>	
	<b>Introduction</b> .....	<b>1-2</b>
	<b>Hardware Specifications</b> .....	<b>1-3</b>
	<b>Software Requirements</b> .....	<b>1-4</b>
	Required MathWorks Products .....	<b>1-4</b>
	Required NXP Products .....	<b>1-4</b>
	<b>Contents of Downloaded ZIP Folder</b> .....	<b>1-5</b>
<b>2</b>	<b>Run 3-Phase PMSM in Open-Loop Control and Validate Hardware Condition</b>	
	<b>Introduction</b> .....	<b>2-2</b>
	<b>Models</b> .....	<b>2-3</b>
	<b>Simulate Target Model</b> .....	<b>2-5</b>
	<b>Generate Code and Deploy Model to Target Hardware for Open-Loop Control</b> .....	<b>2-6</b>
	Generate Code and Run Model to Implement Open-Loop Control .....	<b>2-6</b>
<b>3</b>	<b>Sensorless Field-Oriented Control of PMSM</b>	
	<b>Introduction</b> .....	<b>3-2</b>
	<b>Models</b> .....	<b>3-3</b>
	<b>Simulate Target Model</b> .....	<b>3-5</b>
	<b>Generate Code and Deploy Model to Target Hardware for Sensorless FOC</b> .....	<b>3-6</b>



# About This Example

---

- “Introduction” on page 1-2
- “Hardware Specifications” on page 1-3
- “Software Requirements” on page 1-4
- “Contents of Downloaded ZIP Folder” on page 1-5

## Introduction

This example implements a motor control system using the NXP MCSPTE1AK344 hardware. The MCSPTE1AK344 development kit, based on the 32-bit Arm® Cortex®-M7 S32K3 microcontroller and the GD3000 pre-driver, 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 [\*MCSPTE1AK344 development kit\*](#).

You can use this example to implement a sensorless field-oriented control (FOC) algorithm for the three-phase PMSM available in the MCSPTE1AK344 development kit.

This example also helps you:

- Verify your hardware setup using an open-loop control algorithm and check the integrity of the hardware, motor, and current measurement.
- Perform sensorless FOC of the three-phase PMSM.

The control algorithm available in 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 Validate Hardware Condition
Workflow - 2	Sensorless Field-Oriented Control of PMSM

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

## Hardware Specifications

MCSPTTE1AK344 development kit:

- Uses 32-bit Arm® Cortex®-M7 S32K3 microcontroller and the GD3000 pre-driver.
- Uses 12 V DC power supply.
- Provides USB connectivity.
- Includes a three-phase PMSM (BLDC).

For detailed specifications, see *MCSPTTE1AK344 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 MCSPT1AK344 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 S32K3, version 1.7.0

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 the GitHub® repository, includes:

File Name	Description	
mcb_nxp_MCSPTE1AK344_example_doc.pdf	This document.	
Open loop speed control \s32k344_mcb_open_loop_s32ct.mdl	Target model	for Workflow - 1  (Run 3-Phase PMSM in Open-Loop Control and Validate Hardware Condition)
Open loop speed control \s32k344Data.m	Target model Initialization scripts	
Open loop speed control \mcbOpenLoopData.m		
Open loop speed control \OpenLoopControlHost.slx	Host model	
Sensorless closed loop speed control \s32k344_mcb_sensorless_s32ct.mdl	Target model	for Workflow - 2  (Sensorless Field-Oriented Control of PMSM)
Sensorless closed loop speed control \s32k344Data.m	Target model Initialization scripts	
Sensorless closed loop speed control \mcbSensorlessData.m		
Sensorless closed loop speed control \SensorlessFocHost.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 Validate Hardware Condition

---

- “Introduction” on page 2-2
- “Models” on page 2-3
- “Simulate Target Model” on page 2-5
- “Generate Code and Deploy Model to Target Hardware for Open-Loop Control” on page 2-6

# 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 MCSPT1AK344 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 MCSPT1AK344 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.

You can use this model to:

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

## Models

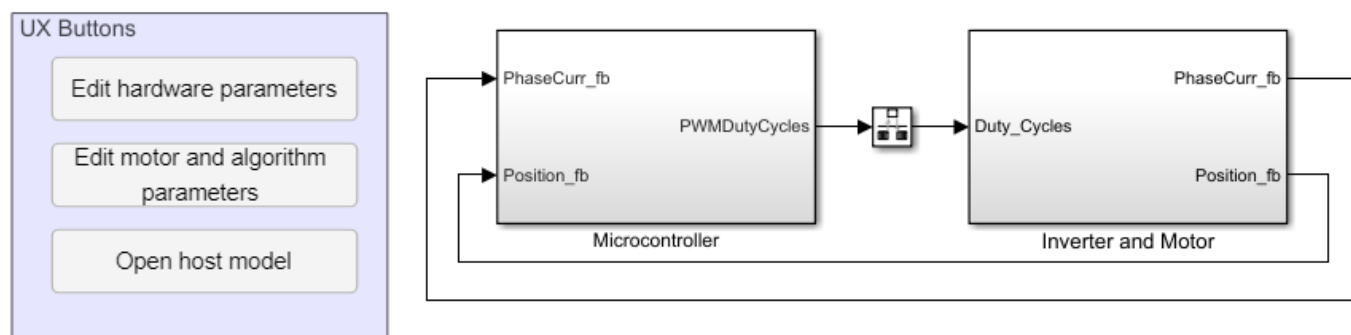
The example includes these models:

- s32k344\_mcb\_open\_loop\_s32ct.mdl (target model)
- OpenLoopControlHost.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('s32k344_mcb_open_loop_s32ct.mdl');
```

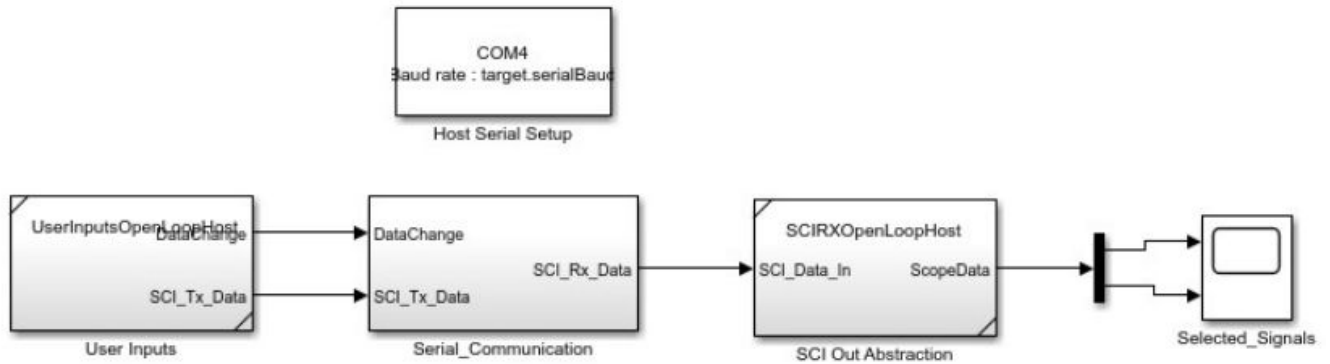
### Open Loop Control using NXP K3 motor control kit



Copyright 2025 The MathWorks, Inc.

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

## NXP K3 Host

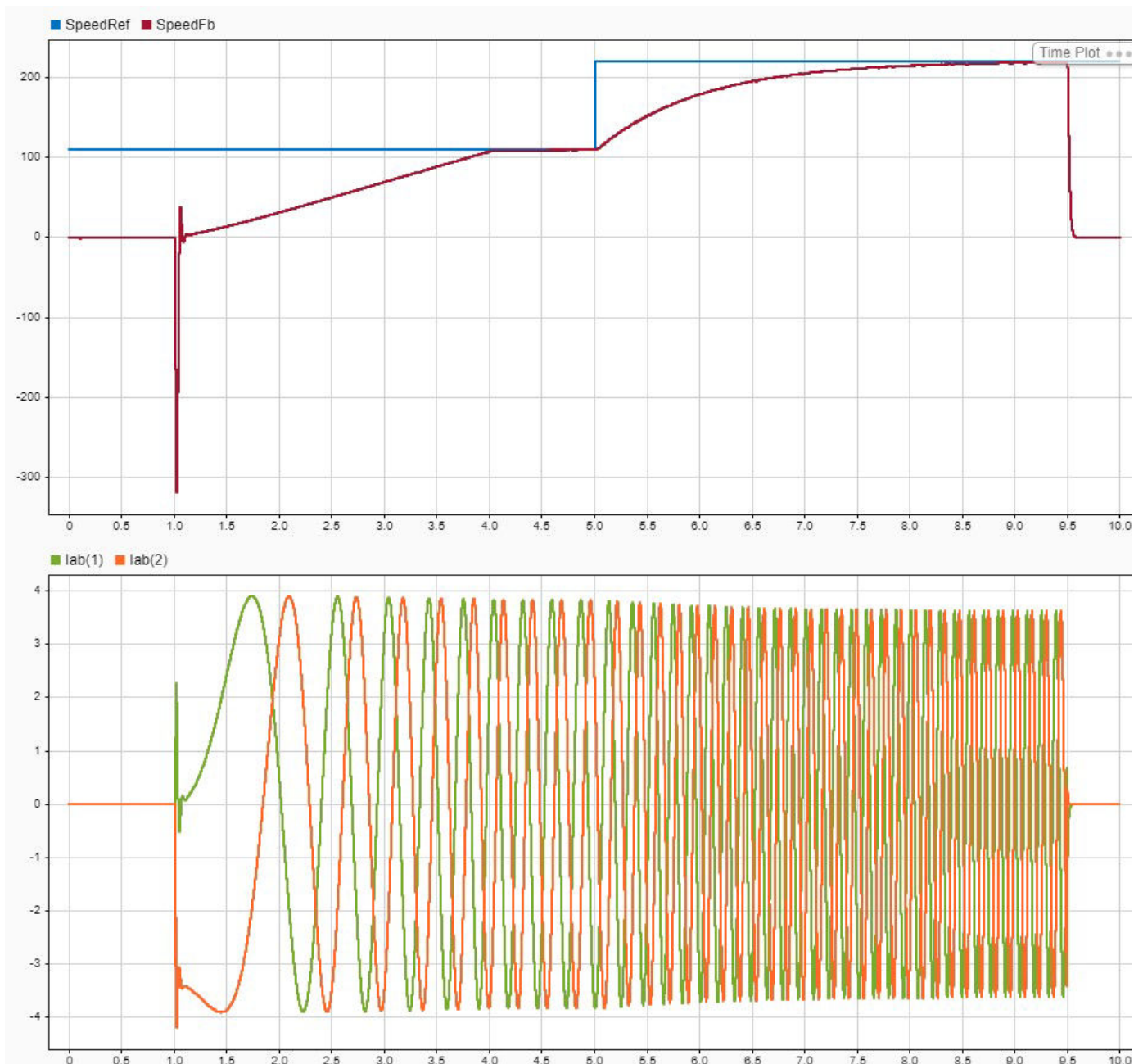


Copyright 2025 The MathWorks, Inc.

## 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 for Open-Loop Control

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.

### Generate Code and Run Model to Implement Open-Loop Control

- 1** Simulate the target model and observe the simulation results.
- 2** Connect the MCSPT1AK344 development kit to the host computer.
- 3** Open the target model.
- 4** Click the **Edit hardware parameters** button available in the target model to open the target model's hardware initialization script, `s32k344Data.m`. Use this script to:
  - Update the device driver specifications (for example, PWM values, communication baud rate, and so on) as well as the inverter configuration parameters (for example, inverter voltage, current sense gain, and so on).
  - Update the target hardware device COM port to match the hardware connection by using the variable `target.comport`. The example uses this variable to update the Port parameter in the Host Serial Setup, Host Serial Receive, and Host Serial Transmit blocks in the host model.



```

> MCSPT1AK344 > MCSPT1AK344 > Open loop speed control
s32k344Data.m x +
\MCSPT1AK344\MCSPT1AK344\Open loop speed control\s32k344Data.m

1  % Model      :   Open loop control of PMSM
2
3  clear ;
4  clc
5
6  %% Sample Times
7  PWM_Frequency      = 20e3;           % Hz - PWM frequency
8  PWM_Period         = 8000;           % uint16 - PWM Timer modulo
9  PWM_DutyScale       = 32768;         % uint16 - PWM duty cycle scale
10 T_pwm              = 1/PWM_Frequency; % sec - PWM switching time period
11 Ts                 = 2 * T_pwm;       % sec - Sample time for current controller
12 Ts_simulink         = T_pwm;          % sec - Sample time for model simulation
13 Ts_motor            = T_pwm;          % sec - Sample time for motor simulation
14 Ts_inverter          = T_pwm;         % sec - Sample time for average value inverter
15 Ts_speed            = 10 * Ts;        % sec - Sample time for speed controller
16
17 target.serialBaud    = 230400;
18 target.comport = 'COM4';
19 dataType = 'single';
20
21 %% Inverter Parameters
22 inverter.Rshunt       = 0.010;         % Ohm - Shunt resistor
23 inverter.MaxADCCnt    = 16383;         % 14-bit ADC resolution
24 inverter.HalfADCCnt   = 8192;          % Half of the 14-bit ADC resolution
25 inverter.AOffset      = 8184;          % Counts - Ia offset
26 inverter.BOffset      = 8184;          % Counts - Ib offset
27 inverter.Rds_on       = 35e-3;         % Ohm
28 inverter.R_board      = inverter.Rds_on + inverter.Rshunt/3; % Ohm

```

- 5 Click the **Edit motor and algorithm parameters** button available in the target model to open the target model initialization script, `mcbOpenLoopData.m`. You can also use this command to open the initialization script.

```
edit mcbOpenLoopData.m
```

- 6 Verify and edit the motor and inverter parameters that are pre-configured in the model initialization script `mcbOpenLoopData.m`. You can replace the parameter values with the values obtained from either the motor datasheet or other sources.

```

> MCSPT1AK344 > MCSPT1AK344 > Open loop speed control
s32k344Data.m x mcbOpenLoopData.m x +
\MCSPT1AK344\MCSPT1AK344\Open loop speed control\mcbOpenLoopData.m

1  %% Initialization script for open loop control of PMSM
2
3  %% Copyright 2025 The MathWorks, Inc.
4
5  %% Sample times used in control algorithm
6  Ts_speed      = 20*Ts;          %Sec    // Sample time for speed controller
7  Ts_SysControl = 0.01;          %Sec
8
9  %% Motor parameters
10 pmsm.p        = 2;
11 pmsm.I_rated   = 6;
12 pmsm.N_rated   = 2200;
13 pmsm.calibSpeed = 60;
14 pmsm.QEPSlits  = 1000;          %        // QEP Encoder Slits
15 pmsm.QEPIndexPresent = true;    %        // To avoid initial rotor alignment
16
17 pmsm.Rs        = 0.192;          %Ohm    // Stator Resistor
18 pmsm.Ld        = 0.096e-3;       %H      // D-axis inductance value
19 pmsm.Lq        = 0.107e-3;       %H      // Q-axis inductance value
20 pmsm.J         = 0.12e-4;        %Kg-m2  // Inertia in SI units
21 pmsm.B         = 0.1e-6;        %Kg-m2/s // Friction Co-efficient
22 pmsm.Ke        = 2.52;          %Bemf Const // Vpk_LL/krpm
23 pmsm.Kt        = 0.097;         %Nm/A    // Torque constant
24 pmsm.PositionOffset = 0;        % Position offset in radian;
25 pmsm.FluxPM    = (pmsm.Ke)/(sqrt(3)*2*pi*1000*pmsm.p/60); %PM flux computed from Ke
26 pmsm.T_rated   = mcbPMSMRatedTorque(pmsm); %Get T_rated from I_rated
27

```

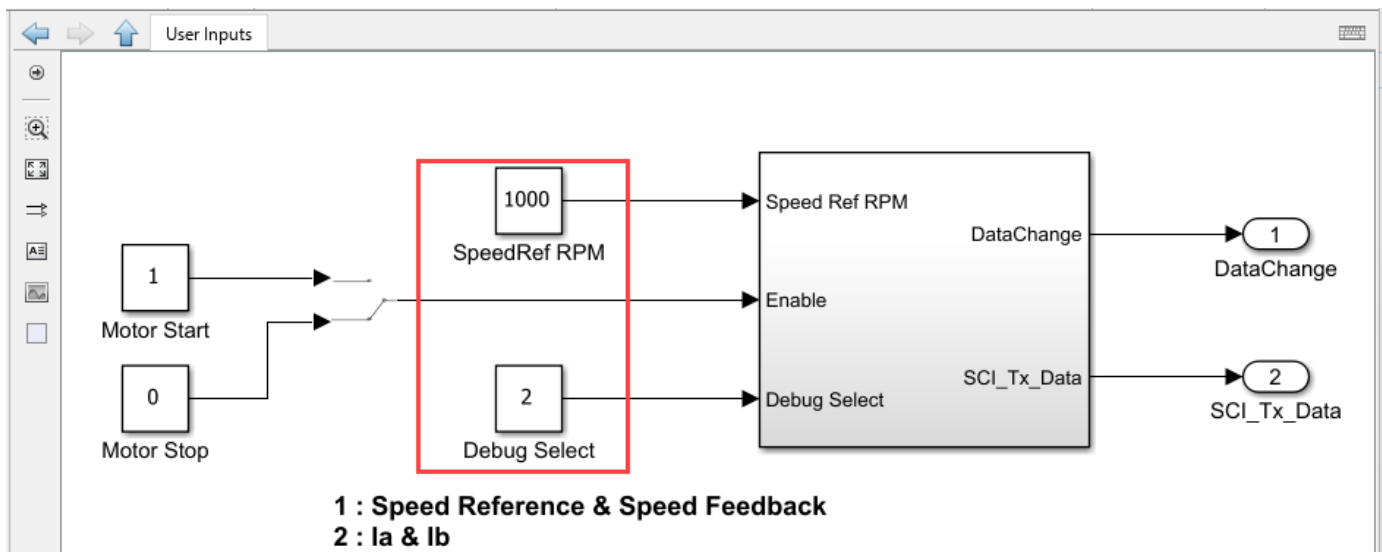
For instructions to configure the script, see [Estimate Control Gains from Motor Parameters](#).

- 7 Click **Build, Deploy & Start** on the **Hardware** tab to deploy the target model to the hardware.
- 8 Verify the variables updated by the target model in the MATLAB base workspace.
- 9 Click the **Open host model** button in the target model to open the associated host model, `OpenLoopControlHost.slx`. You can also use the `open_system` command to open the host model.

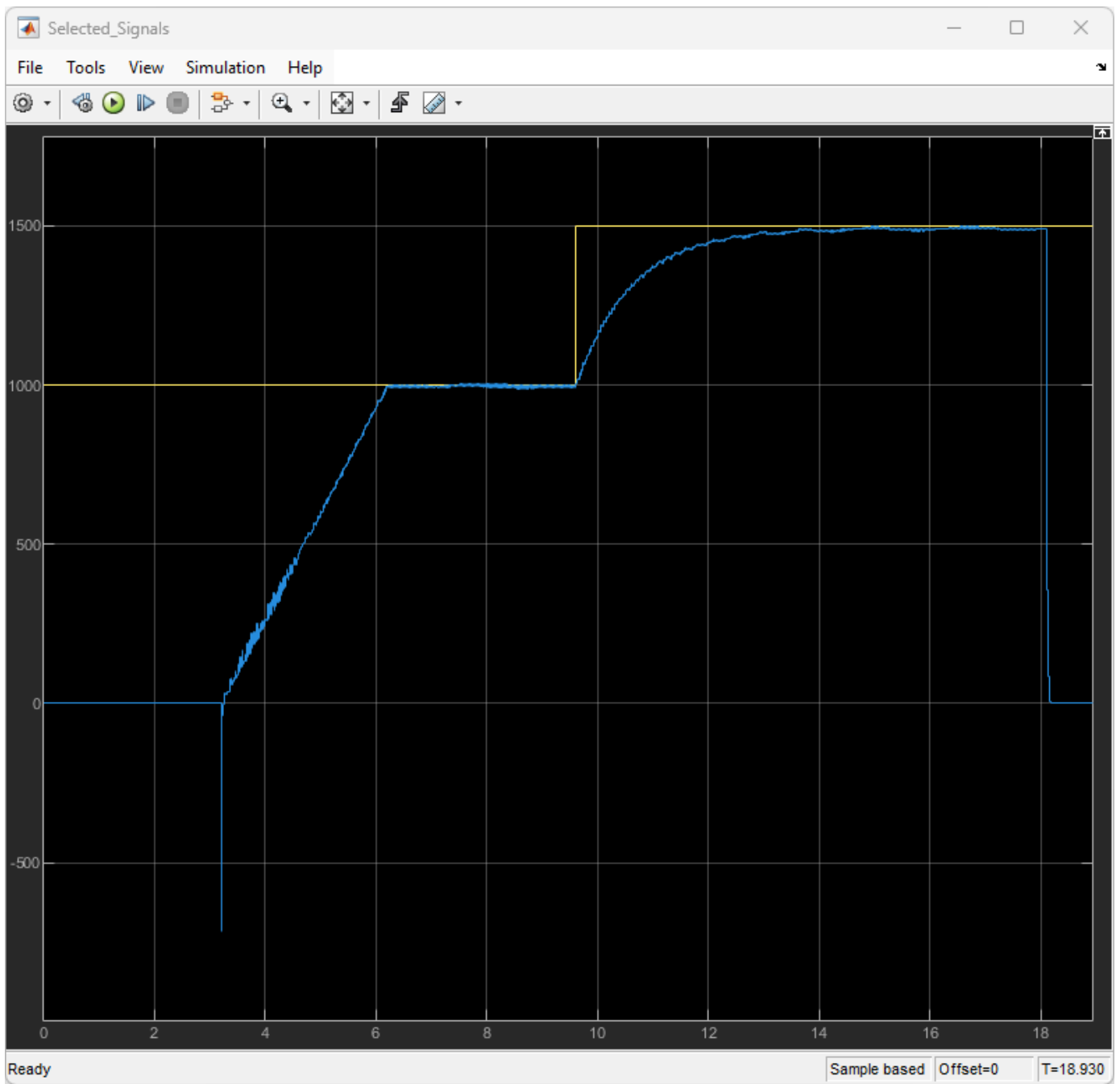
For details about the serial communication between the host and target models, see [Host-Target Communication](#).

- 10 Go to `OpenLoopControlHost/User Inputs` subsystem and verify that the value for `SpeedRef` RPM Constant block is set to 1000, and the value of `Debug Select` Constant block is set to 2.

**Note** It is recommended that you always limit the reference speed to 1000 RPM during open-loop run.



- 11** Click **Run** on the **Simulation** tab to run the host model.
- 12** After the motor starts running, observe the  $I_a$  and  $I_b$  phase currents in the Time Scope to verify a successful open-loop control using MCSPT1AK344 development kit. This image shows the speed response using open loop model.



You can now proceed to sensorless FOC of the PMSM using MCSPT1AK344 development kit, which uses an I-F control-based startup followed by transition from I-F to closed-loop control and from closed-loop to I-F control.

# Sensorless Field-Oriented Control of PMSM

---

- “Introduction” on page 3-2
- “Models” on page 3-3
- “Simulate Target Model” on page 3-5
- “Generate Code and Deploy Model to Target Hardware for Sensorless FOC” on page 3-6

### Introduction

This workflow implements the sensorless field-oriented control (FOC) technique to control the speed of a three-phase permanent magnet synchronous motor (PMSM) available in the MCSPT1AK344 development kit. The FOC algorithm requires rotor position feedback, which is obtained by the sensorless position estimators. The PMSM starts using the I-F control algorithm (using the I-F Controller block) followed by transition from I-F to closed-loop control and from closed-loop to I-F control. For details about FOC, see *Field-Oriented Control (FOC)*.

## Models

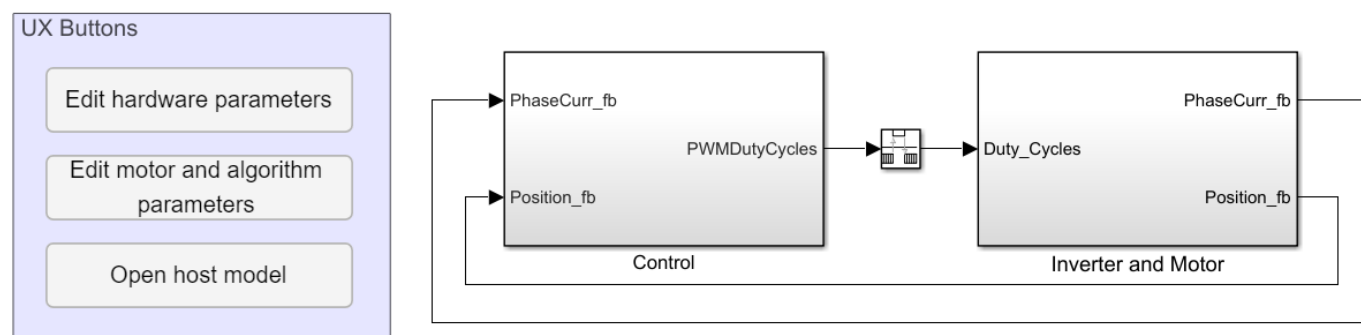
The example includes these models for sensorless closed loop control of PMSM using MCSPT1AK344 development kit:

- s32k344\_mcb\_sensorless\_s32ct.mdl (target model)
- SensorlessFocHost.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('s32k344_mcb_sensorless_s32ct.mdl');
```

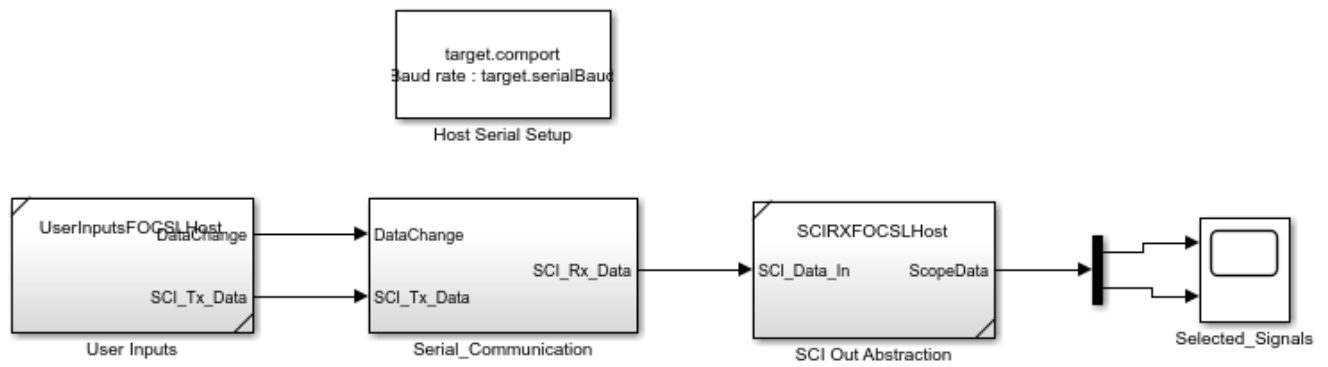
### Sensorless Control using NXP K3 motor control kit



Copyright 2025 The MathWorks, Inc.

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

## NXP K3 Sensorless FOC Host



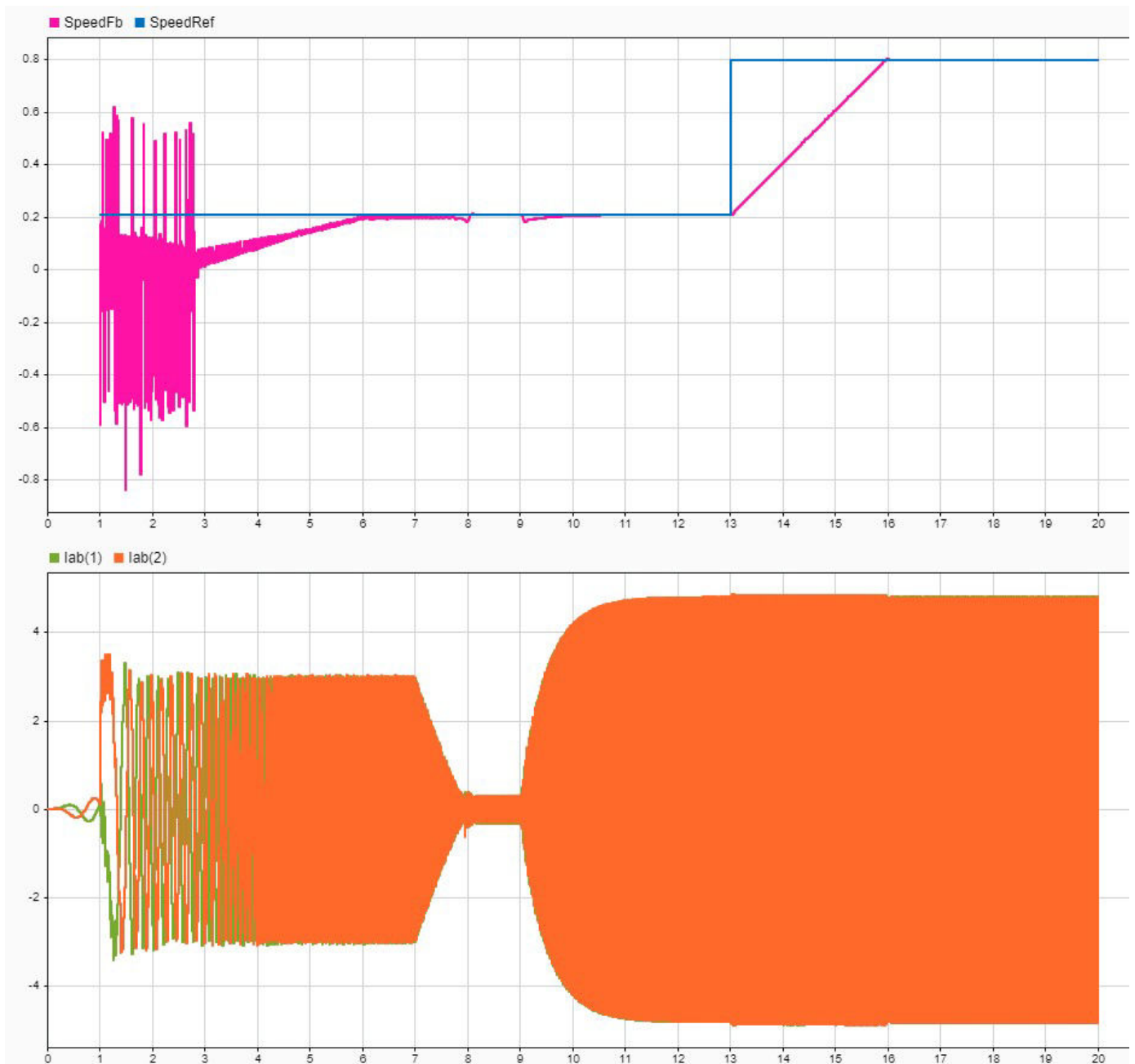
Copyright 2025 The MathWorks, Inc.



## Simulate Target Model

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

- 1 Open the target model `s32k344_mcb_sensorless_s32ct.mdl`.
- 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 for Sensorless FOC

This section explains how to generate code and run the sensorless 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 MCSPT1AK344 development kit.

- 1 Simulate the target model `s32k344_mcb_sensorless_s32ct.mdl`, and observe the simulation results.
- 2 Connect the MCSPT1AK344 development kit to the host computer.
- 3 Open the target model.
- 4 Click the **Edit hardware parameters** button available in the target model to open the target model's hardware initialization script, `s32k344Data.m`. Ensure that this script is the same one that you modified in the open-loop control phase (as described in Step 4 of *Generate Code and Run Model to Implement Open-Loop Control* section).
- 5 Click the **Edit motor and algorithm parameters** button available in the target model to open the target model initialization script, `mcbSensorlessData.m`. You can also use this command to open the initialization script.

```
edit mcbSensorlessData.m
```

The example implements field-oriented control (FOC) using sensorless position estimation and I-F control-based startup to control the speed of PMSM. The PMSM starts using the I-F control algorithm (using the I-F Controller block in ) followed by transition from I-F to closed-loop control and from closed-loop to I-F control.

Therefore, the `mcbSensorlessData.m` script allows you to specify the speed at which those transitions occur. Change the values, if required.

To estimate the position feedback for the FOC algorithm, the script also allows you to select one of the three options: flux observer (FO), extended EMF observer (EEMF) or sliding mode observer (SMO). By default, this example uses EEMF.

```

> MCSPT1AK344 > MCSPT1AK344 > Sensorless closed loop speed control
s32k344Data.m x mcbSensorlessData.m x +
C:\Users\lapillai\Downloads\MCSPT1AK344\MCSPT1AK344\Sensorless closed loop speed control\mcbSensorlessData.m
18
19 pmsm.Rs      = 0.244;           %Ohm      // Stator Resistor
20 pmsm.Ld      = 0.1e-3;         %H        // D-axis inductance value
21 pmsm.Lq      = 0.114e-3;       %H        // Q-axis inductance value
22 pmsm.J       = 0.12e-4;        %Kg-m2     // Inertia in SI units
23 pmsm.B       = 0.1e-6;        %Kg-m2/s   // Friction Co-efficient
24 pmsm.Ke      = 2.52;          %Bemf Const // Vpk_LL/krpm
25 pmsm.Kt      = 0.097;         %Nm/A      // Torque constant
26 pmsm.PositionOffset = 0;      % Position offset in radian;
27 pmsm.FluxPM   = (pmsm.Ke)/(sqrt(3)*2*pi*1000*pmsm.p/60); %PM flux computed from Ke
28 pmsm.T_rated  = mcbPMSMRatedTorque(pmsm); %Get T_rated from I_rated
29 pmsm.V_max    = 45;           % V - Max measured voltage
30
31 % pmsm.N_base = mcb_getBaseSpeed(pmsm,inverter); %rpm // Base speed of motor at given Vdc
32 pmsm.N_base = pmsm.N_rated;
33
34 pmsm.OL2CL    = 0.2 * pmsm.N_base;
35 pmsm.CL20L    = 0.1 * pmsm.N_base;
36 pmsm.Sensor   = 'EEMF'; % FO,SMO,EEMF
37
38 %% PU System
39 PU_System.V_base = inverter.V_dc/sqrt(3);
40 PU_System.I_base = inverter.ISenseMax;
41 PU_System.N_base = pmsm.N_base;

```

Additionally, verify and edit the motor and inverter parameters, if required. For instructions to configure the script, see [Estimate Control Gains from Motor Parameters](#)

- 6 After updating the initialization script, open the target model again, and click **Build, Deploy & Start** on the **Hardware** tab to deploy the target model to the hardware.
- 7 Verify the variables updated by the target model in the MATLAB base workspace.
- 8 Click the **Open host model** button in the target model to open the associated host model. You can also use the `open_system` command to open the host model, `SensorlessFocHost.slx`.

For details about the serial communication between the host and target models, see [Host-Target Communication](#).

- 9 Go to `SensorlessFocHost/User Inputs` subsystem and update the value for `SpeedRef RPM Constant` block to use a very low value (maximum 10% of base speed). The value of `Debug Select Constant` block is set to 1 to obtain the speed reference and speed feedback.
- 10 Click **Run** on the **Simulation** tab to run the host model. The motor starts rotating.
- 11 Increase the motor `SpeedRef RPM` in steps to exceed **Speed to exit I-F controller** (parameter of the I-F Controller block, which is same as `pmsm.OL2CL` variable) to switch from I-F control to closed-loop control.
- 12 During closed-loop operation, decrease the motor `SpeedRef RPM` such that it falls below **Speed to re-enter I-F controller** (parameter of the I-F Controller block, which is same as `pmsm.CL20L` variable) to switch from closed-loop control to I-F control.
- 13 Observe the debug signals in the scope and display blocks available in the host model. This image shows the speed reference and speed feedback signals.

