

Field-Oriented Control of a Permanent Magnet Synchronous Machine

In this example you will review a Field-Oriented Control (FOC) algorithm for a Permanent Magnet Synchronous Machine (PMSM). You will test the control algorithm with closed loop system simulation then generate HDL code for the control algorithm. You will also see how tunable parameter data is specified and how corresponding HDL port entities are generated.

This example uses:
Simscape Electrical

 Copy Command

Introduction

The example is partitioned such that you can generate code for the control algorithm as well as verify the behavior of the control algorithm using a simulation test bench. Simscape (TM) Electrical (TM) is required to run the system simulation test bench model `hdlcoderFocCurrentTestBench.slx` but is not required to generate code from the control algorithm model `hdlcoderFocCurrentFixptHdl.slx`.

Verify Behavior through Simulation

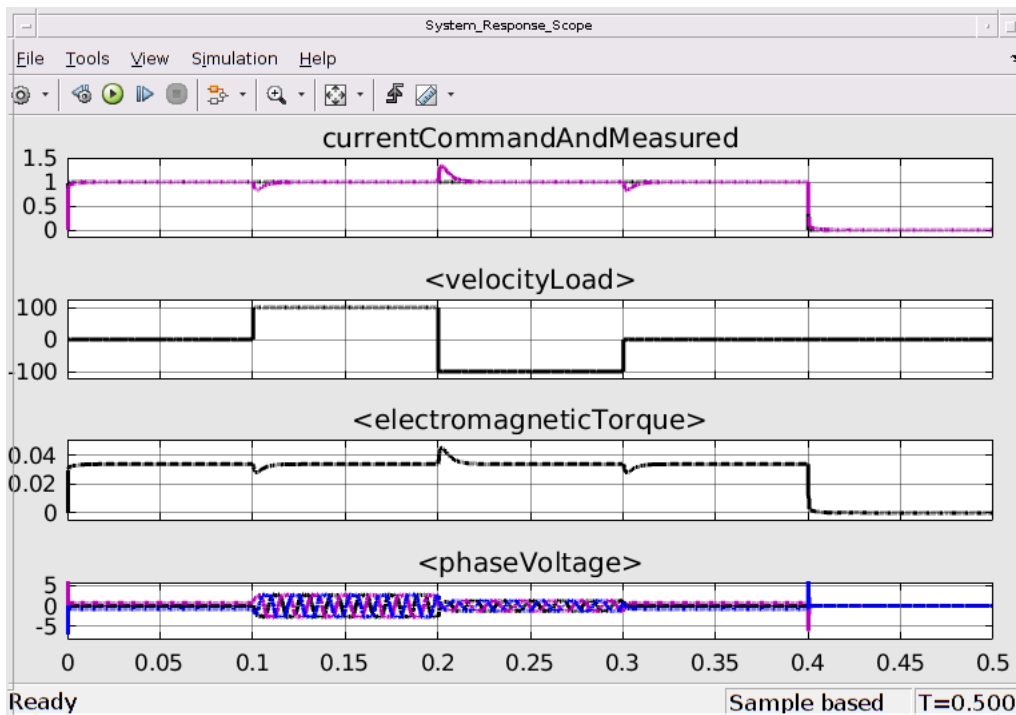
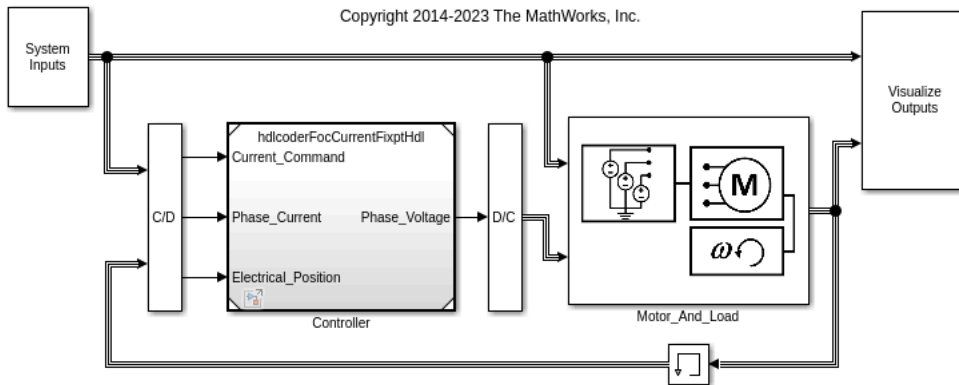
In this example FOC is used to regulate phase current to control torque of an electric machine. You can simulate a test bench to explore the behavior of the system. During the simulation, the solver may generate warnings related to zero crossing when the velocity load changes abruptly. You can disable these warnings during the simulation.

```
hasSimPowerSystems = license('test', 'Power_System_Blocks');
if hasSimPowerSystems
    open_system('hdlcoderFocCurrentTestBench')
    set_param('hdlcoderFocCurrentTestBench','IgnoredZcDiagnostic','none');
    sim('hdlcoderFocCurrentTestBench')
    set_param('hdlcoderFocCurrentTestBench','IgnoredZcDiagnostic','warn');
end
```

 Get ▾

Field-Oriented Control Current Control Test Bench

Copyright 2014-2023 The MathWorks, Inc.



The scope shows that a 1 Amp step current command is requested and the load velocity changes between locked rotor (zero), +100 rad/sec, and -100 rad/sec. The current command represents a quadrature current command to a non-salient PMSM. (The controller regulates the direct current to zero.) Note that for this motor and controller, the electromagnetic torque closely follows the measured quadrature current of the motor.

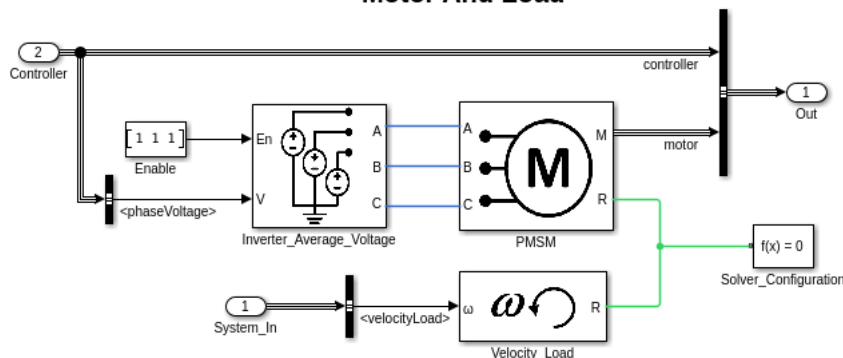
Explore Plant Specification

In the `Motor_And_Load` subsystem you will see a mathematical model of the components being controlled. An average model of the inverter is used to drive a constant parameter dq voltage equation model of a PMSM which is connected to a velocity load.

```
if hasSimPowerSystems
    open_system('hdlcoderFocCurrentTestBench/Motor_And_Load')
end
```

Get ▾

Motor And Load

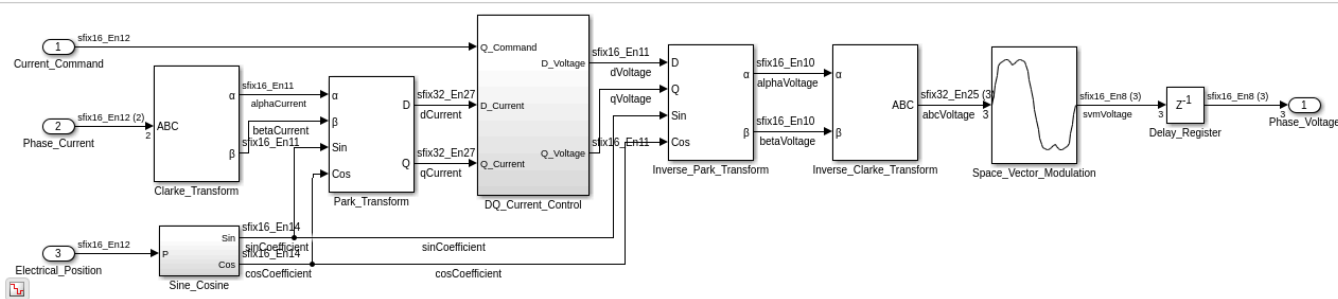


Explore Control Algorithm Specification

The FOC current control algorithm is specified in a separate model. In the control algorithm, the electrical equations of the machine are projected from the three-phase stationary reference frame onto a two phase rotating reference frame using Clarke and Park transforms. This simplifies the control by removing time and position dependencies. Space Vector Modulation enables the controller to achieve greater voltage across the phases than if just the sinusoidal outputs of the inverse Clarke transform were used.

```
load_system('hdlcoderFocCurrentFixptHd1');
open_system('hdlcoderFocCurrentFixptHd1/FOC_Current_Control')
```

Get ▾



Explore Data Specification

Both the controller and the plant (i.e. motor and load) reference data from the MATLAB workspace. A data definition file creates this data and is automatically run within the PreLoadFcn callback of the system test bench model.

```
%edit('hdlcoderFocCurrentFixptHd1Data.m')
%
```

Get ▾

When you review this file, notice that the parameters paramCurrentControlP and paramCurrentControlI are specified as Simulink.Parameters whose storage class is set to ExportedGlobal. This tells HDL Code to generate entity ports for these parameters instead of constant values.

Generate HDL Code for Control Algorithm

Before generating HDL code, it is important to ensure that the model adheres to certain important settings for HDL code generation. Below are some of the main steps:

- Create a DUT subsystem: For HDL code generation it is always better to create a DUT (Design Under Test) subsystem from which HDL code is generated. This subsystem serves several purposes including being a place-holder for HDL optimization settings.
- Setup for HDL: In order to get ready for HDL code generation, certain solver settings and model settings must be in place. The hdlsetup command takes care of all these settings and should be run before HDL code-generation.
- Checking sample times: Applying HDL optimizations requires all block sample times to be inferred as discrete. The main block-type to be cautious of are constants, which derive an 'inf' sample time, by default. We can find these blocks and explicitly set their sample-times to -1 so they get the correct back-propagated sample times.

```
% You can generate and review the HDL code for the controller.
makehdl('hdlcoderFocCurrentFixptHd1/FOC_Current_Control');
```

Get ▾

```

#### Generating HDL for 'hdlcoderFocCurrentFixptHdl/FOC_Current_Control'.
#### Using the config set for model <a href="matlab:configset.showParameterGroup('hdlcoderFocCurrentFixptHdl', { 'HDL Code Generation' } )">hdlcode
#### Running HDL checks on the model 'hdlcoderFocCurrentFixptHdl'.
#### Begin compilation of the model 'hdlcoderFocCurrentFixptHdl'...
#### Begin compilation of the model 'hdlcoderFocCurrentFixptHdl'...
#### Working on the model 'hdlcoderFocCurrentFixptHdl'...
#### Working on... <a href="matlab:configset.internal.open('hdlcoderFocCurrentFixptHdl', 'GenerateModel')">GenerateModel</a>
#### Begin model generation 'gm_hdlcoderFocCurrentFixptHdl' ....
#### Rendering DUT with optimization related changes (IO, Area, Pipelining)...
#### Model generation complete.
#### Generating new validation model: <a href="matlab:open_system('hdlsrc/hdlcoderFocCurrentFixptHdl/gm_hdlcoderFocCurrentFixptHdl_vnl')">gm_hdlco
#### Validation model generation complete.
#### Begin VHDL Code Generation for 'hdlcoderFocCurrentFixptHdl'.
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Clarke_Transform as hdlsrc/hdlcoderFocCurrentFixptHdl/Clarke_Transform.vhd.
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/DQ_Current_Control/D_Current_Control/Saturate_Output as hdlsrc/hdlcoderFocCurrentFi
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/DQ_Current_Control/D_Current_Control as hdlsrc/hdlcoderFocCurrentFixptHdl/D_Current_
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/DQ_Current_Control as hdlsrc/hdlcoderFocCurrentFixptHdl/DQ_Current_Control.vhd.
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Inverse_Clarke_Transform as hdlsrc/hdlcoderFocCurrentFixptHdl/Inverse_Clarke_Transf
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Inverse_Park_Transform as hdlsrc/hdlcoderFocCurrentFixptHdl/Inverse_Park_Transform.v
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Park_Transform as hdlsrc/hdlcoderFocCurrentFixptHdl/Park_Transform.vhd.
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Sine_Cosine/Sine_Cosine_LUT as hdlsrc/hdlcoderFocCurrentFixptHdl/Sine_Cosine_LUT.vh
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Sine_Cosine as hdlsrc/hdlcoderFocCurrentFixptHdl/Sine_Cosine.vhd.
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control/Space_Vector_Modulation as hdlsrc/hdlcoderFocCurrentFixptHdl/Space_Vector_Modulatio
#### Working on hdlcoderFocCurrentFixptHdl/FOC_Current_Control as hdlsrc/hdlcoderFocCurrentFixptHdl/FOC_Current_Control.vhd.
#### Generating package file hdlsrc/hdlcoderFocCurrentFixptHdl/FOC_Current_Control_pkg.vhd.
#### Code Generation for 'hdlcoderFocCurrentFixptHdl' completed.
#### Generating HTML files for code generation report at <a href="matlab:web('/home/amoses/Documents/MATLAB/ExampleManager/amoses.Bdoc23b.j2313133
#### Creating HDL Code Generation Check Report file:///home/amoses/Documents/MATLAB/ExampleManager/amoses.Bdoc23b.j2313133.field_example/hdlcoder-
#### HDL check for 'hdlcoderFocCurrentFixptHdl' complete with 0 errors, 1 warnings, and 0 messages.
#### HDL code generation complete.

```

Notice in the generated hdlcoderFocCurrentFixptHdl.vhd file that the entity has ports for paramCurrentControlP and paramCurrentControlI.