Electric Motor Controller Virtual Test Bench

Real-Time Simulation Software, Hardware

Published by

Opal-RT Technologies, Inc. 1751 Richardson, suite 2525 Montréal (Québec) Canada H3K 1G6

www.opal-rt.com

Printed in Canada

CONTENTS

ELECTRIC MOTOR CONTROLLER VIRTUAL TEST BENCH

PREFACE

This document introduces OPAL-RT's unique solution, the Electric Motor Controller Virtual Test Bench, for testing electric motor controller. It uses the scalable hardware-in-the-loop simulator eDRIVEsim™ and integrates a powerful real-time computer, a motor and power electronic block set, accurate high speed solvers and versatile FPGA I/O modules that meet the demanding requirements of precise controller testing and motor drive simulation.

Figure 1: Sample test bench

Information on the following topics will be presented in this document:

- Hardware-in-the-loop concepts for motor testing and simulator modules. (see"Advanced Simulation Using the FPGA-PMSM Model")
- Quick overview of simulator's functionalities and their usage.
- More advanced concepts such as configuring the simulator and connecting the real controller in a closed-loop with the simulator.

INTRODUCTION

ELECTRIC MOTOR DRIVE HARDWARE-IN-THE-LOOP TESTING AND SIMULATION

Hardware-in-the-Loop (HIL) testing and simulation consists of connecting the device under test to the rest of the system that is simulated in real-time on a computer, equipped with all the inputs and outputs necessary to communicate with the real device.

For electric motor drives, this is done by simulating a part of or the complete drive on a very fast real-time system, including the motor and power electronics (including different types of converters and inverters). It is also able to emulate the mechanical load connected on the shaft of the motor. In addition, several motor drives, or a motor drive with other mechanical equipment can be simulated to emulate a complete system (like a hybrid powertrain or vehicle), using flexible I/Os, interfaces and software environment.

ELECTRIC MOTOR CONTROLLER VIRTUAL TEST BENCH OVERVIEW

The test bench system, illustrated in the diagram below, consists of two main parts: the real-time simulator (where the motor drive power circuit is simulated) and the external device under test (usually the electronic controller unit, or ECU). Both parts are usually connected in a closed loop by several analog and digital inputs and outputs.

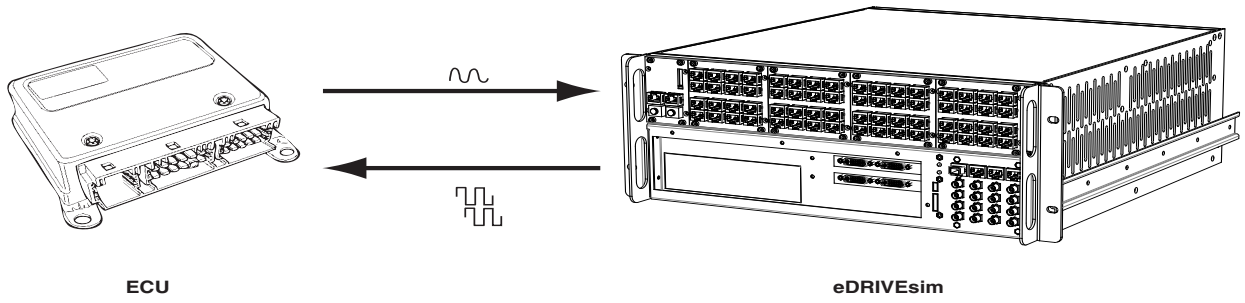


Figure 1: General concept of motor-HIL simulation

The real-time simulator is Opal-RT's eDRIVEsim®, a very powerful HIL platform for simulating motor drives and varous power converters. The simulator includes several modules, described in the following chapters.

SIMULATOR AND TEST BENCH FEATURES

- ability to apply various IGBT and Diode faults (only for nodal solver inverter).
- Ability to block the rotor.
- Ability to freeze the stator voltage.
- Ability to force rotor speed.
- Ability to apply an internal sinusoidal source directly to motor drive.
- Ability to generate an internal PWM source with dead-time to feed the inverter.

MOTOR AND POWER ELECTRONIC BLOCK SET AND SOLVERS

The simulator includes a power electronic and motor library that lets you simulate all components forming the electrical motor drive system. It includes detailed mathematical model of permanent magnet motors (PMM), power electronic converters and inverters and also devices such as resolver, encoder and Hall Effect sensor. The electric circuit represented below provides an overview of the type of motor drive model that can be simulated.

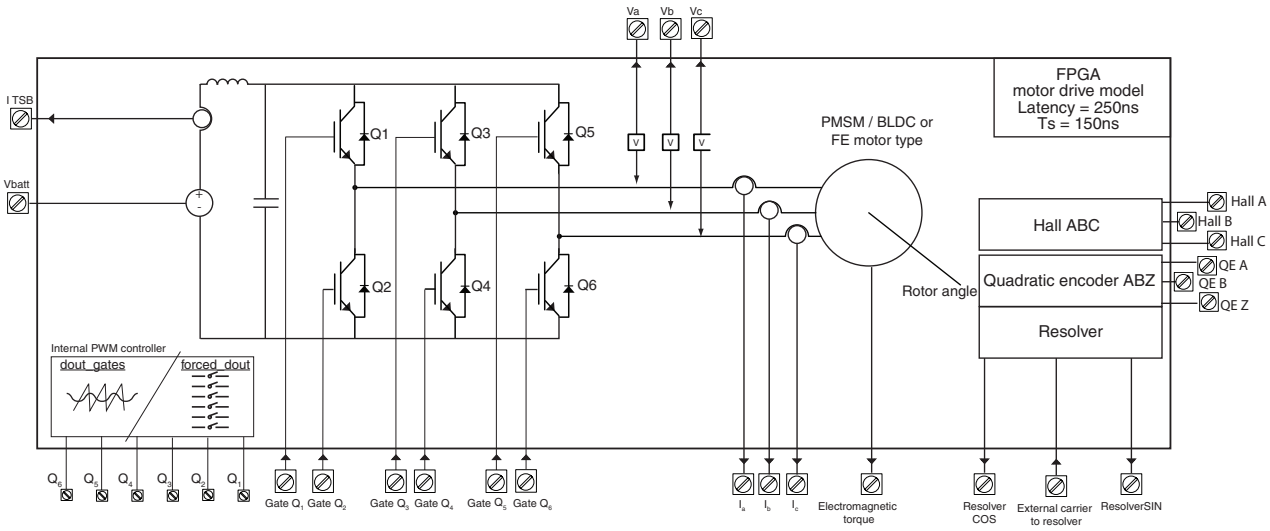


Figure 2: Motor drive circuit diagram

The motor can be either a PMSM or a BLDC, both based on a permanent magnet motor (PMM) differential and algebraic equations. Moreover, two different representations are supported that allows choosing the level of detail simulated:

1. Classic model that provides sinusoidal Back-EMF and inductances.
2. Finite-element based model based on JMAG software (see "jmag").

The simulator also provides a complete set of real-time solvers optimized for real-time simulation, allowing you to accelerate simulation but also to simulate complex transient and fault conditions that are difficult or impractical to achieve with a real system.

Using Xilinx System Generator, the user can also develop their own blocks and enhance the current block set library.

JMAG

RT-LAB.JMAG is a software module developed in collaboration with JMAG Group, developer of the popular JMAG simulation software for electromagnetic design of motor drives. RT-LAB.JMAG (which is only compatible with JMAG version 10.5 or earlier) is designed for use in the real-time simulation of motor models using inductance and back EMF data generated by Finite Element Analysis (FEA) methods. The FEA technique is required when classical D-Q models cannot be used due to large asymmetry in the motor construction generating non-sinusoidal fluxes and currents. This is important for hybrid vehicle manufacturers using high-speed motors optimized for space and cost and saturation effects.

Advantages of using motors with JMAG:

- Simulate motor models with a level higher of precision than standard D-Q models.
- Simulate the effect of rotor asymmetry.
- Simulate BACK EMF harmonic.
- Simulate saturation effects.
- Simulate cogging torque.

SOFTWARE ENVIRONMENT

The simulator also integrates a test and development software suite that includes the RT-LAB™ software for controlling the simulation execution, tuning the drive system parameters, and also applying fault conditions. Signal acquisition, data logging and scopes are other examples of commonly used features making test easier.

Note that this document will not explain the RT-LAB functionalities, so refer to the RT-LAB User Guide for more help.

HARDWARE PLATFORM

The simulator includes fast analog and digital I/O modules specifically designed for motor controller having extremely fast dynamics due to high-frequency switching action of power electronics switches. Their low latency and high sampling rates combined with short time steps ensures the best overall cycle time possible to interface controller or external devices.

The hardware includes the OP5600 Chassis including a Virtex-6 FPGA board, such as the ML605 board. It also includes analog and digital I/O signals and all convenient signal conditioning modules to connect the controller.

AREAS OF APPLICATION

Direct current and induction motor drives have gradually been replaced by PMSM and BLDC drives. PMSM and the BLDC have many advantages that makes them a preferred choice in automotive applications: higher efficiency, lower audible noise, absence of brushes (compared to DC motors), higher speed and higher power density.

BLDC and PMSM are very similar in construction and are also described with almost the same equations. Typically, the BLDC has a trapezoidal flux shape with concentrated windings and round rotor with equal inductances, while PMSM have distributed windings for a sinusoidal back-EMF shape and inductance that are different in the direct and quadrature axis.

Electric motors are found in a wide range of fields and this model is perfectly suited for each:

- Hybrid automotive systems
- Wind energy converters
- Electric trains
- Airplanes
- Submarines
- Boats
- Robotics

REQUIREMENTS

It is understood that the implementation and use of the FPGA motor drive model requires a good working knowledge of both MATLAB and RT-Lab. In addition, using the model requires certain software and basic hardware, listed below.

SOFTWARE

The following software is required:

- MATLAB, R2009b or 2010b.
- RT-LAB, V10.4.3
- OPJMAG, V2.0.
- RT-XSG V 2.2
- WinRAR, V3.8.0 or later versions.
- Xilinx ISE 12.3 to 13.3 (only required to develop custom bitstreams, standard JMAG/PMSM models do not require this software).

HARDWARE

The following eDRIVEsim hardware is required:

- OP5600 chassis equipped with Opal-RT's Virtex-6 FPGA board, or the Xilinx ML605.
- Convenient signal conditioning modules and enough analog and digital I/O signals to connect the controller to the simulator

GENERAL ARCHITECTURE

Figure 3 provides an overview of the simulator architecture. As shown, the motor drive model consists of two major parts or sub-models:

The FPGA model, which runs on the Virtex-6 board and includes:

- The motor drive equations with all its parts (motor, inverter, encoder, resolver, etc).
- Analog and digital input and output signals.

The CPU model, which runs on standard INTEL processor and includes:

- All other parts of the system (like mechanical load, other slower converters, etc...).
- A link to the user interface
- All other simulation features (data logging, triggering, CAN interface, etc...)

The two sub-models are two distinctive Simulink mdl files, with special interfaces to link them.

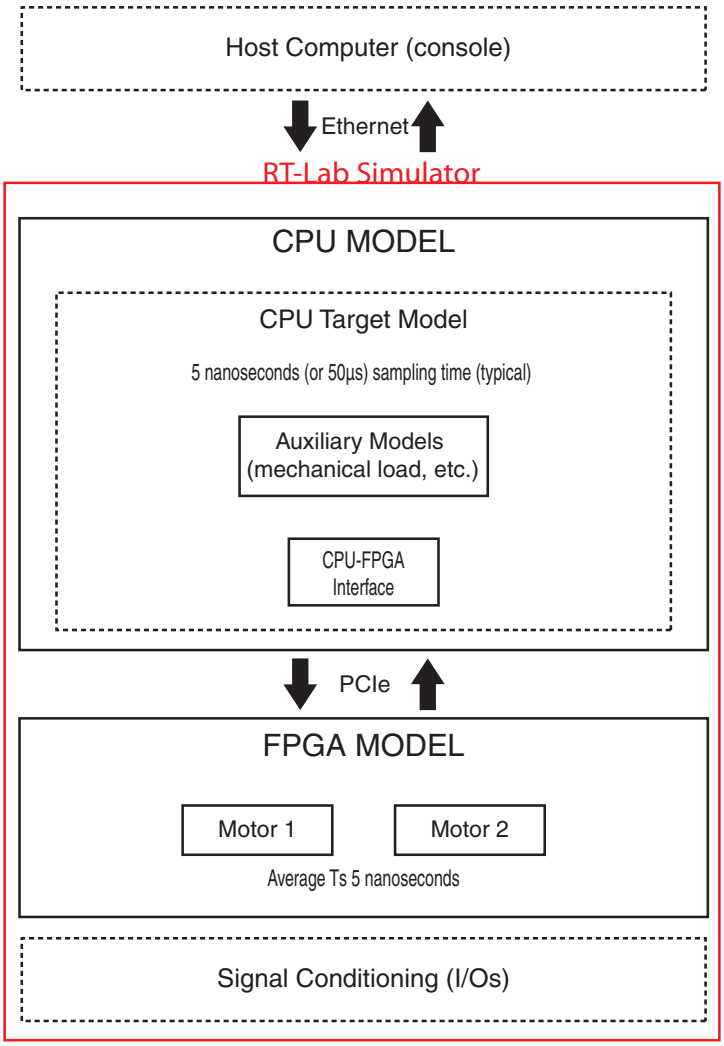


Figure 3: General simulator architecture

The FPGA motor drive model works in concert with the CPU target and the signal conditioning modules to simulate an actual electric drive model with speed and accuracy. The host's only role is to display and control the results using the RT-Lab software.

Although the FPGA model has an average time step speed of 5 nanoseconds, it is important to note that communication between the FPGA model and the CPU target model is limited to the average time step speed of the CPU model (50 microseconds, typical).

The details of the FPGA model and its functions, as well as the interface between the FPGA and CPU models, is described in subsequent chapters.

UNDERSTANDING THE MOTOR DRIVE MODEL

The following table shows all block elements that form the motor drive model. Users can select which elements to include during simulation. Experienced users can also develop additional elements to eventually replace existing elements or to enhance the simulator. Note that the simulator allows the simulation of two drive systems on a single Virtex-6 FPGA board.

Test Source	Inverter	Motor	Measurement
Sinusoidal source Internally generated PWM Internally generated PWM with I/O loopback or direct gate feed from I/O	TSB* Nodal	JMAG/PMSM	Torque Resolver Quadrature encoder Motor currents and voltage DC-link current Motor neutral point voltage IGBT currents (nodal inverter only)

* The TSB inverter does not allow fault generation. The Nodal inverter allows full fault generation capabilities.

The motor drive model can allows simulation of two types of inverters:

TSB: This inverter provides faster sampling but does not allow fault simulation.
Nodal: This inverter, although fractionally slower (150 ns latency), offers full fault simulation capability. The nodal inverter has a built-in L-C filter and can be provided with a second LC filter for DC-link simulation.

The motor drive model allows simulating PMSM and BLDC motor using two set of equations:

JMAG: Finite Element Analysis motor model that includes saturation, motor slot effect and cogging torque.
PMSM: Linear motor model based on classic Park d-q theory.

Note that test source are blocks used to simulate internal source that could be connected to the inverter or to the motor when the external controller is not available or when specific tests need to be performed on the motor drive circuit. The sources could be a sinusoidal voltage source that is connected to the motor, or it could be a DC voltage source connected to the inverter that is fed with PWM signals.

GETTING STARTED - ML605 BASED MODEL

The topics contained in this section are intended to provide a quick view of the virtual motor controller test bench and its features and capabilities. The steps are designed to be completed in a sequential order. Working through this tutorial gives you a preliminary understanding of the steps required to configure, then execute simulator parameters.

After you work through the sequence of topics in this tutorial, you will have a running test bench that lets you verify the model's fidelity and precision.

The first topic explains how to prepare and build the motor drive model and how to start the test bench using the RT-LAB software. A compressed archive file is given that contains a pre-configured model simulating two motor drives with a default set of parameters.

The last topic describes how to control the simulation and visualize data received from the simulation using the RT-LAB console subsystem. This console subsystem contains scopes to visualize the main motor drive quantities and provide access to various configuration parameters. For example, the console allows adjusting sensor positions and applies various faults on the motor and IGBT terminal and DC bus.

STARTING THE SIMULATION WITH RT-LAB

When you first receive the model, it will be in a compressed file format (Winrar), which must be extracted to a new directory:

1. Select the compressed file  , right-click and select "Extract files..."
2. Select the desired folder (or create a new folder) into which to extract the model files and click OK.
3. Open RT-Lab and create a new project, then add the motor drive model to the project.

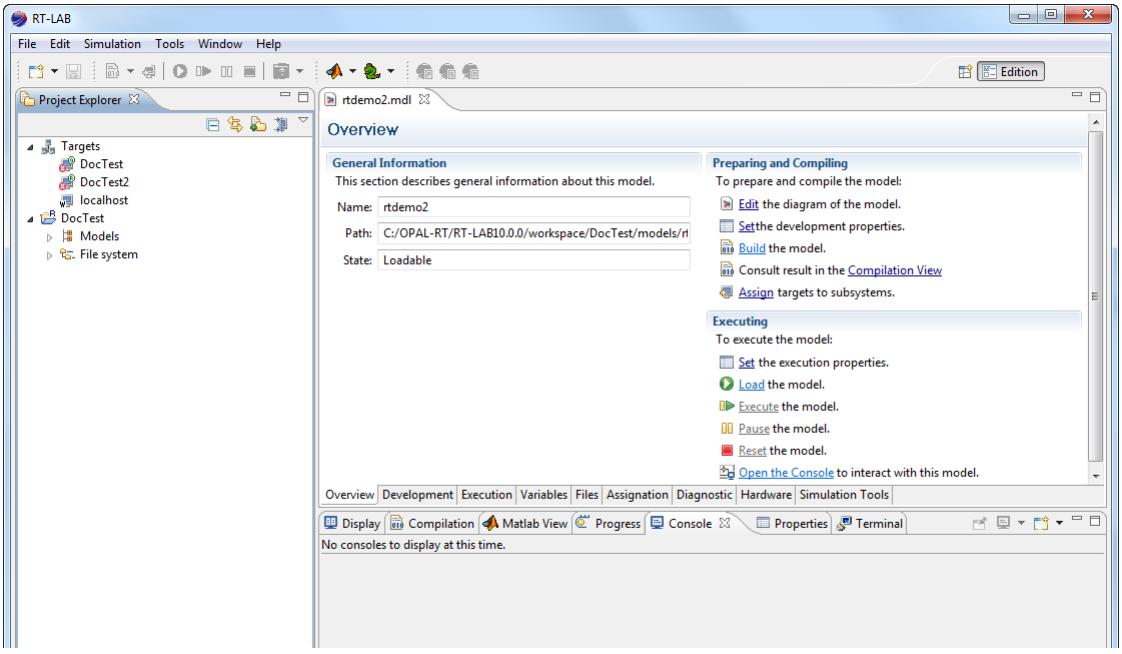


Figure 4: RT-Lab main screen

4. Compile the model in RT-LAB. 
5. Load  and Execute  the model in RT-LAB.

MODIFYING CONTROL AND TEST PARAMETERS

The RT-Lab motor drive model console contains various parameters used to control, test and monitor the motor model during real-time simulation. Most parameters are found in the motor's Function Block Parameters window, as we will describe in this section. Also note that the motor drive model is designed with the appropriate default values, but these values can be modified to suit your needs.

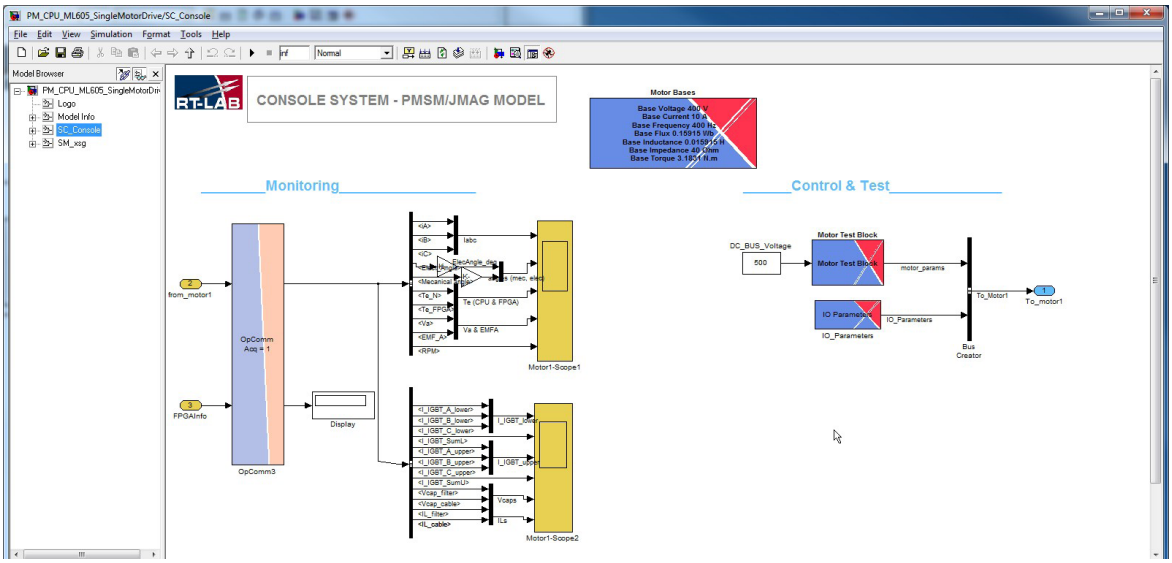


Figure 5: View of the model console system (motor #1 shown)

MOTOR TEST BLOCK

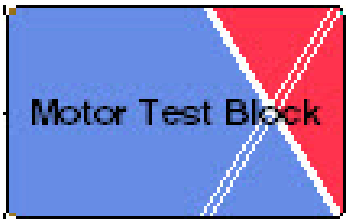


Figure 6: Motor Test Block parameters GUI

The motor test block contains various parameters to test the motor model. Double clicking the block opens the Function Block Parameters window, which provides 4 tabs for the various parameters: General, Test Source, Nodal Faults and Misc.

Motor Test Block Parameters, General Tab

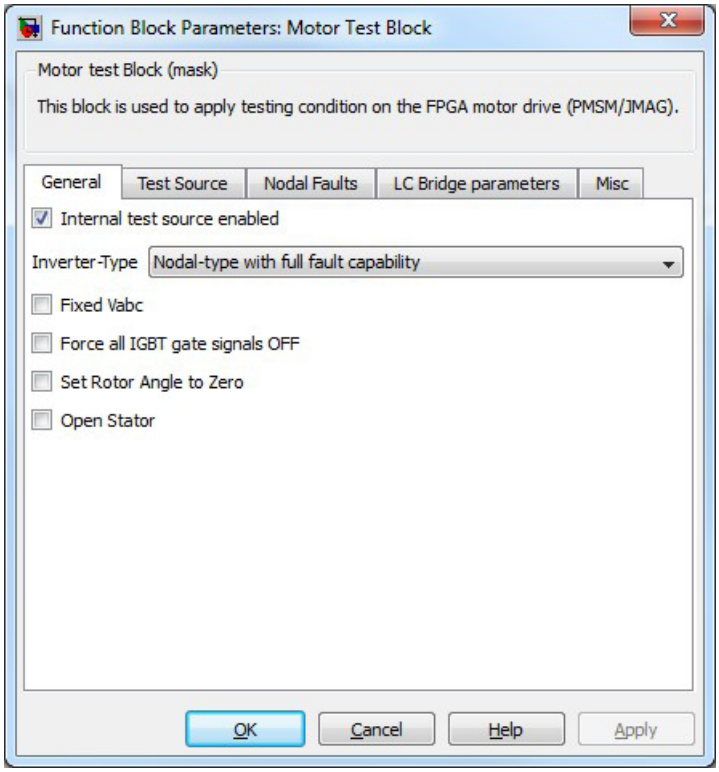


Figure 7: Motor Test Block1 parameters - General tab

Internal Test Source Enabled: This parameter connects the internal test sources to the motor drive. When selected, any external controller is disconnected from the FPGA motor model.

Inverter-Type: Choice of inverter model:
"TSB":switching function with rectification capability or

"Nodal-type with full fault capability": an inverter with L-C filter and full component fault capability.

Fixed Vabc: Blocks the terminal voltage to the specified angle, thus applying DC voltage or fixed PWM duty cycle to the PMSM.

Force all IGBT signals OFF: Option to internally force all IGBT signals to off.

Set Rotor Angle to Zero: Blocks the rotor to null angle.

Open Stator: Open the stator phase at the machine. This is made by increasing the stator resistance to a very high value (default=24 pu) so a small residual current can still flow when this option is checked.

Motor Test Block Parameters, Test Source Tab

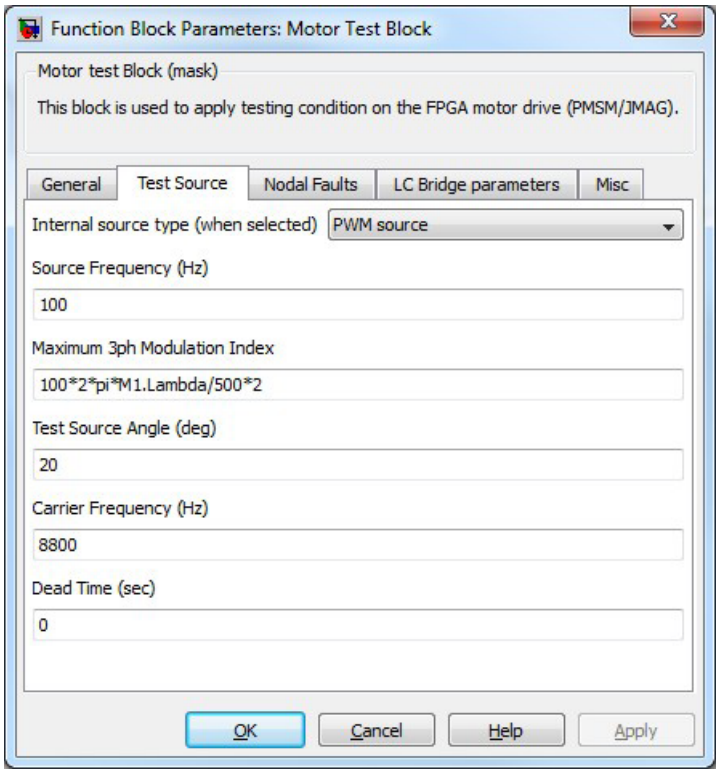


Figure 8: Motor Test Block1 parameters - Test Source tab

Internal source type (when selected): when Internal Test Source Enabled is selected in the General tab, it selects the machine's source. The options provided in this field are:

1. Sinusoidal source (applied directly to the machine terminal thus bypassing the inverter)
2. PWM source In this option, the inverter is driven by the internal PWM generator.
3. PWM source from IO loopback In this option, the inverter is also driven by the internal PWM generator but sent through the Digital Outout of the simulator and read back at the Digital input. The option will only work if a loopback card is connected at the I/O of the simulator.

Source Frequency (Hz): the frequency of the sinusoidal test voltage source applied to the machine terminal or the frequency of the PWM modulation.

Sinusoidal 3ph test source amplitude (V) : the amplitude in Volts of the internal sinusoidal test source (when selected).

Maximum 3ph Modulation Index: the maximum modulation index of the internal PWM test source when selected. If this parameter is called mi and the DC bus voltage, Vdc, then the equivalent tri-phase sinusoidal voltage source applied to the motor is Vdc/2*mi*sin (ωt)

Test Source Angle (deg): the phase in degree of the internal test source or equivalent PWM modulation signals.

Carrier frequency: PWM carrier frequency when PWM source is selected.

Dead time (us): dead-time of PWM waveform when PWM source is selected.

Motor Test Block Parameters, Nodal Fault Tab

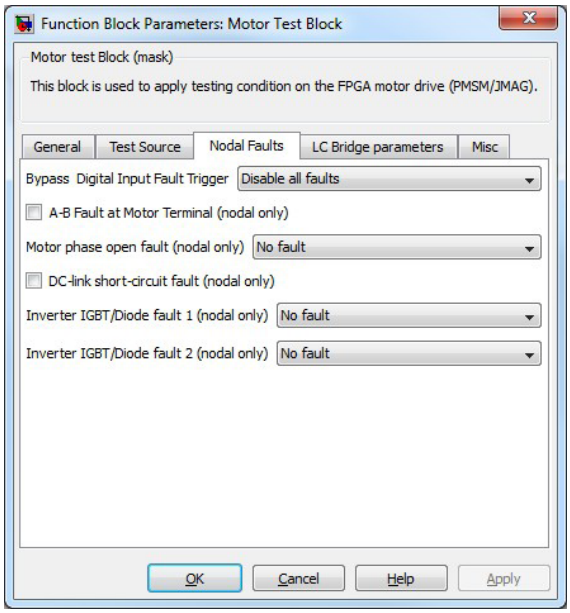


Figure 9: Motor Test Block1 parameters - Nodal Faults tab

This pane is used to select the faults to apply to the nodal inverter. Two IGBT and/or diode faults can be applied simultaneously in addition to motor-phase fault and DC-link fault.

Bypass Digital Input Fault trigger: this option is used to enable the triggering of the faults by the I/O. Choice are:

1. Disable all faults
2. Enable faults
3. Fault 1 triggered by Digital input: in this case, Fault 1 will be applied only if the corresponding Digital input of the simulator reads a logical 1. Fault 2 is ENABLED.
4. Fault 2 triggered by Digital input: in this case, Fault 2 will be applied only if the corresponding Digital input of the simulator reads a logical 1. Fault 1 is ENABLED.
5. Faults 1 & 2 triggered by Digital input: in this case, both Fault 1 and Fault 2 will be applied only if the corresponding Digital input of the simulator reads a logical 1.

A-B fault on motor terminal: makes an AB fault at motor terminal.

DC-link short circuit: makes a short circuit between the terminals of the capacitor of the DC bus.

Inverter IGBT/Diode Fault 1: make a fault on the upper IGBT/Diode, phase A. Choices are: no fault, short-circuit, open IGBT, open Diode.

Inverter IGBT/Diode Fault 2: make a fault on the lower IGBT/Diode, phase A. Choices are: no fault, short-circuit, open IGBT, open Diode.

Motor Test Block Parameters, LC Bridge parameters Tab

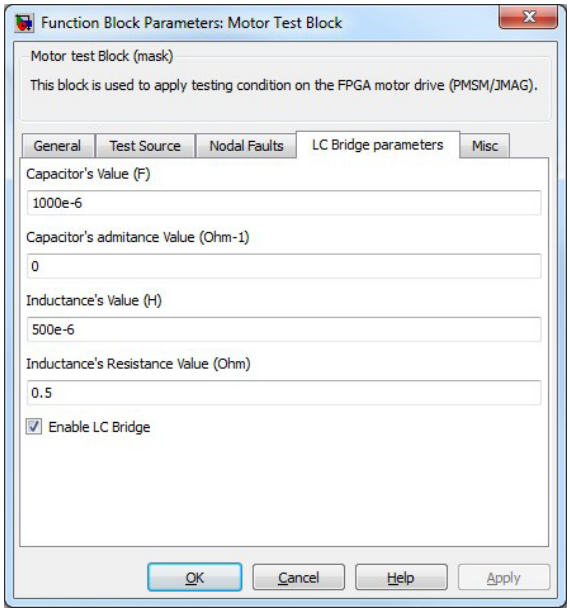


Figure 10: Motor Test block1 parameters – LC bridge parameters tab

This pane is used to set the parameters of the optional LC Bridge for the DC link simulation.

The solving method is a state space based algorithm with a sample time of 150ns (Trapezoidal discrete integration type is used). You need to know that the stability of the system could be compromised for very low value of capacitor or inductance.

You can disable the LC Bridge and change the values during the simulation (it can't be done dynamically, but occasionally).

Motor Test Block Parameters, Misc Tab

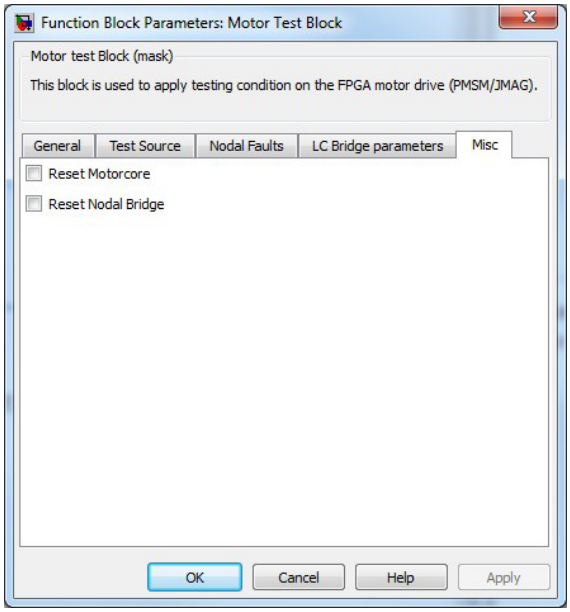


Figure 11: Motor Test Block1 parameters - Misc tab

This pane is used to reset the motorcore solver as well as the nodal bridge.

I/O GAINS SETUP

Double-click the **IO Parameters block** to open the Source Block Parameters window. This window provides access to three (3) tabs: DAC Gains, ADC Parameters and Sensors Parameters.

DAC Gains Tab

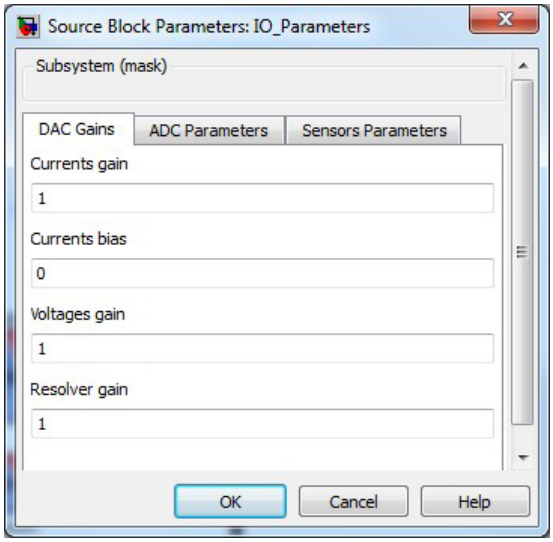


Figure 12: IO_Parameters, DAC Gains tab

The format used for the gain parameters is Ufix17_10; you must enter value between 0 and 128 (otherwise the gain will be saturated to 128).

The format used for the bias parameter is Fix18_10; you must enter value between -128 and +128 (otherwise the gain will be saturated to -/+ 128).

The default scaling of the analog outputs is 1v for 1pu. For example, if you set a gain to 4, the scaling changes to 4V for 1pu.

Currents Gain: gain applied to the motor`s current values on analog outputs.

Currents Bias: offset applied to the motor`s current values on analog outputs.

Voltage Gain: gain applied to the voltage values on analog outputs.

Resolver Gain: gain applied to the resolver`s values on analog outputs.

ADC Parameters Tab

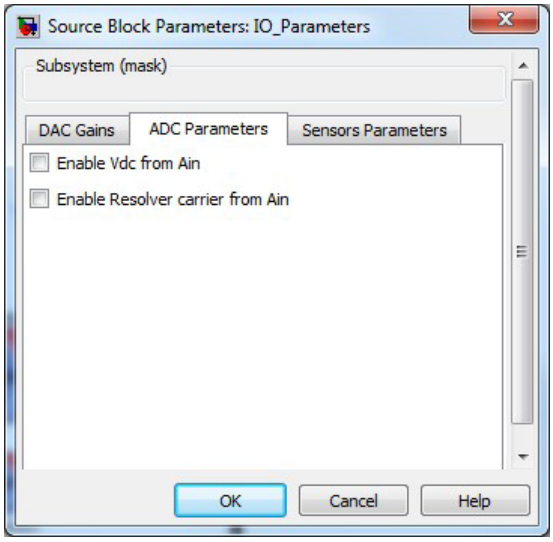


Figure 13: IO_Parameters, ADC Parameters tab

Enable Vdc from Ain: enable to use the Analog input channel for Vdc as internal Inverter's Vdc. If selected, a text field appears:

- *Vdc from Ain gain*: Gain to adjust the Vdc from ain amplitude. By default: 1V = 1pu = 1 M1.V_ base. For example, if you set the value to 0.5 so 1V = 0.5pu = 0.5 M1.Vbase.

Enable resolver carrier from Ain: Enable to use the Analog input channel for resolver carrier as internal resolver carrier. If selected, a text field appears:

- *Resolver excitation gain*: Gain to adjust the resolver carrier from Ain amplitude.

Sensors Parameters Tab

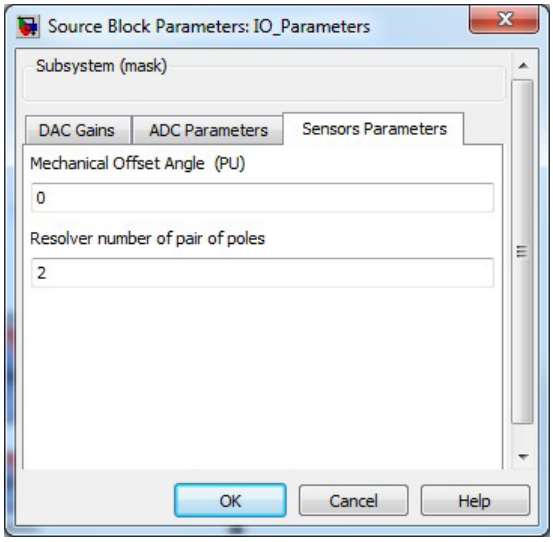


Figure 14: IO_Parameters, Sensors Parameters tab

You can specify a different pair of pole numbers for your resolver (between 0 and 15). Take special care when using this feature because, by default, the resolver has the same number of pair of poles as the motor, so it represents the electrical angle (and not the mechanical).

MONITORING

This part of the model Console is used to monitor real-time simulation values. It is composed of 2 scopes, one for simulation time step monitoring and another one for the PMSM values.

Note that other values can be selected at the Simulink main bus signals coming out of RT-Lab OpComm.

Entry #	Values	Units
1	Motor currents (3)	A
2	Angles (mechanical and electrical)	Degree
3	Torque	N.m.
4	Terminal voltage and Back-EMF (phase A)	V
5	Rotor speed	rpm
6	Lower IGBT currents (phase A and sum A,B,C)	A

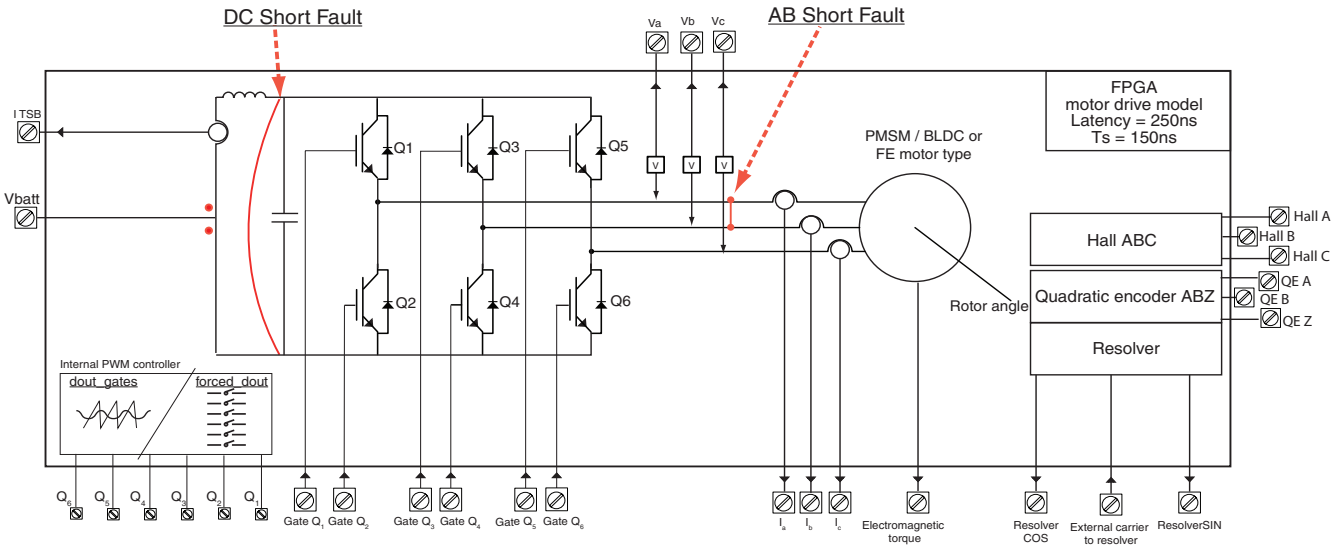
Table 1: 4PMSM motor (real-time only) Simulink scope

FPGA MODEL FAULTS

The FPGA motor drive model makes it possible to simulate a number of faults. Each of the faults described above is illustrated on the FPGA model drawings provided.

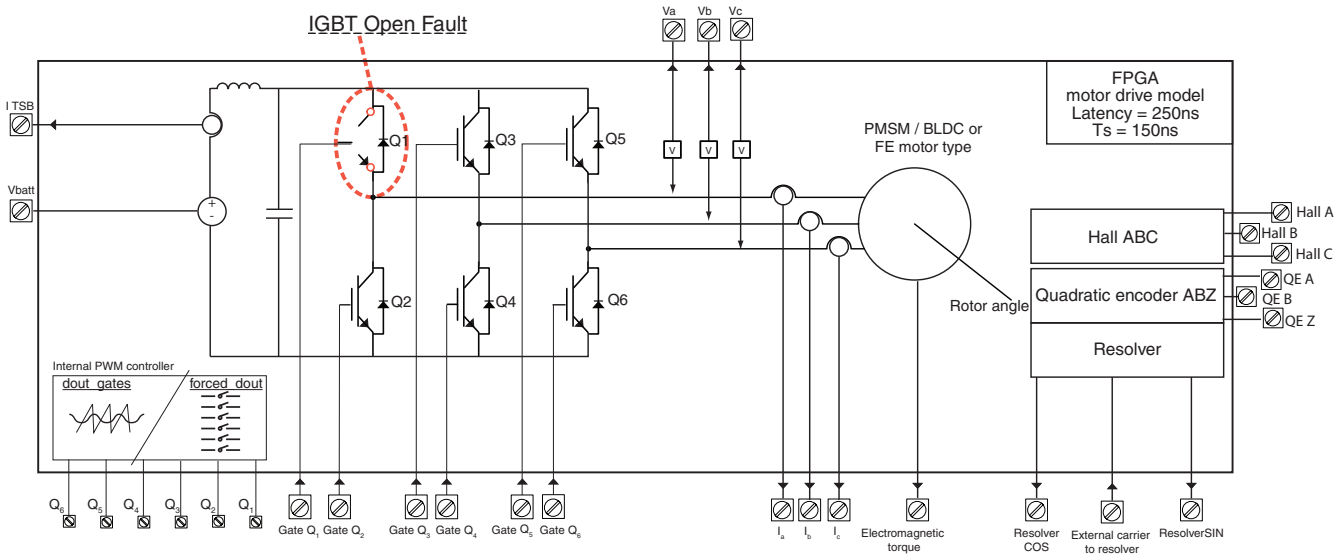
- AC and DC shorts
- IGBT open
- Diode open
- IGBT diode short
- Open machine terminal

FPGA model with AB and DC shorts

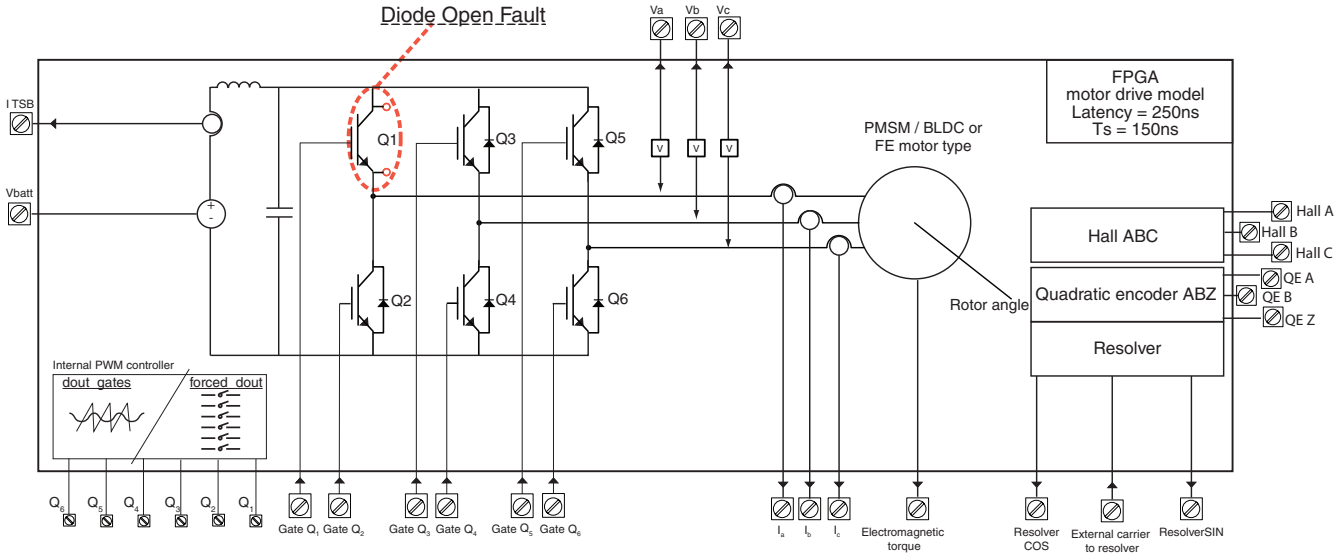


Note: It is only possible to simulate either an AB or a DC fault at one time. Both faults cannot be simulated simultaneously.

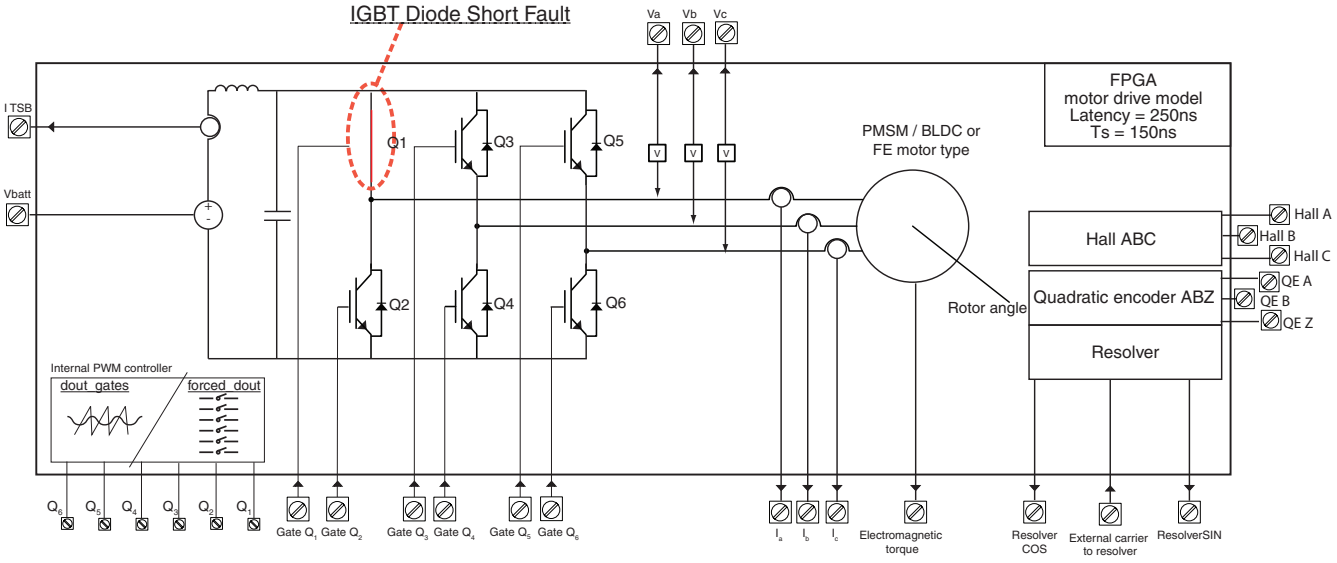
FPGA model with IGBT Open Fault



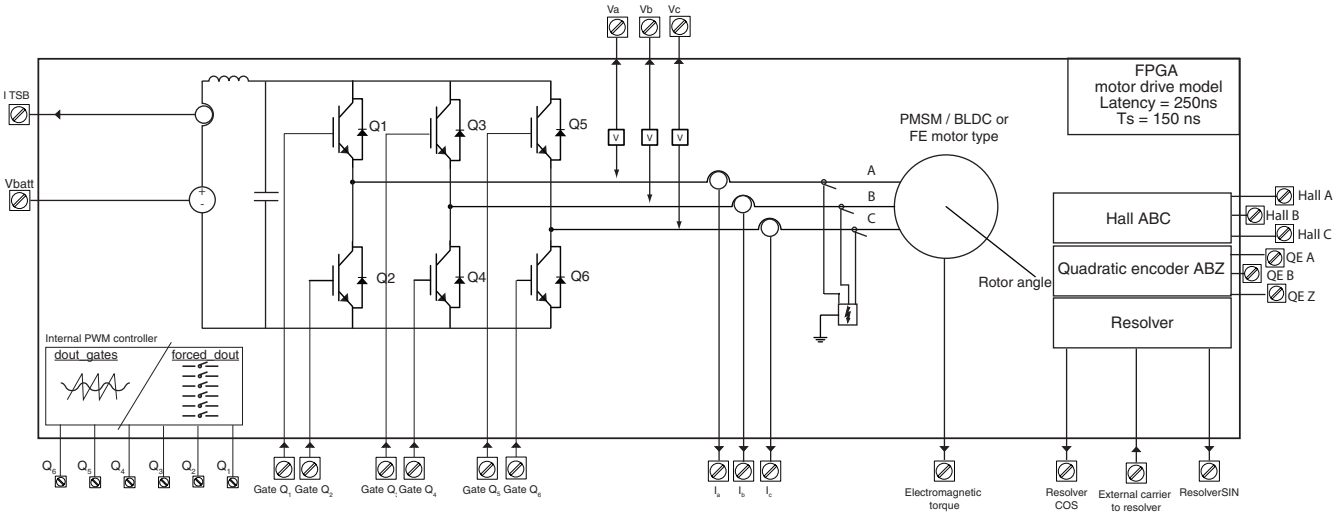
FPGA Model with Diode Open Fault



FPGA Model with IGBT Dlode Short Fault



FPGA Model with Open Machine Terminal Fault



GETTING STARTED - OP5142 BASED MODEL

The OP5142 FPGA motor drive model, “PM_CPU_OP5142_MotorDrive.mdl” is slightly different from the ML605 motor drive model. Both achieve the same simulation, however they use different parameters, therefore, there are some differences in configuration and use.

OP5142 MODEL CONTENTS

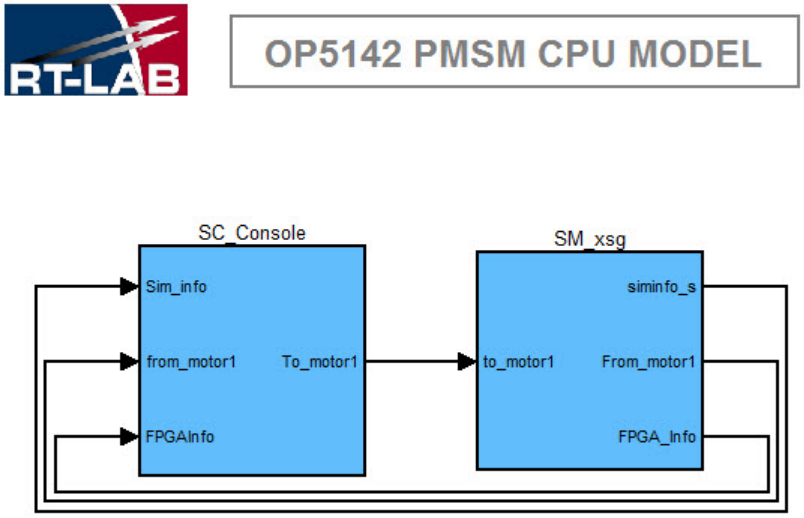


Figure 15: OP5142 PMSM CPU model

SC Console (User Interface)

- Online control of the model
  - Internal test source parameters
  - Gain + Bias Analog Output signals
  - Ld Lq Ra Rb Rc Flux Linkage motor parameters
- Monitoring
  - Motor and inverter states (Currents + Voltages + Torque + Angle)
  - Simulation Information
  - FPGA communication status

SM_xsg = CPU / FPGA interface + CPU processing

- Prepacking Data to send to FPGA
- Unpacking Data received from FPGA
- Mechanical part of the motor calculation

STARTING THE SIMULATION WITH RT-LAB

When you first receive the model, it will be in a compressed file format (Winrar), which must be extracted to a new directory:

1. Select the compressed file , right-click and select “Extract files...”
2. Select the desired folder (or create a new folder) into which to extract the model files and click OK.
3. Open RT-Lab and create a new project, then add the motor drive model to the project.

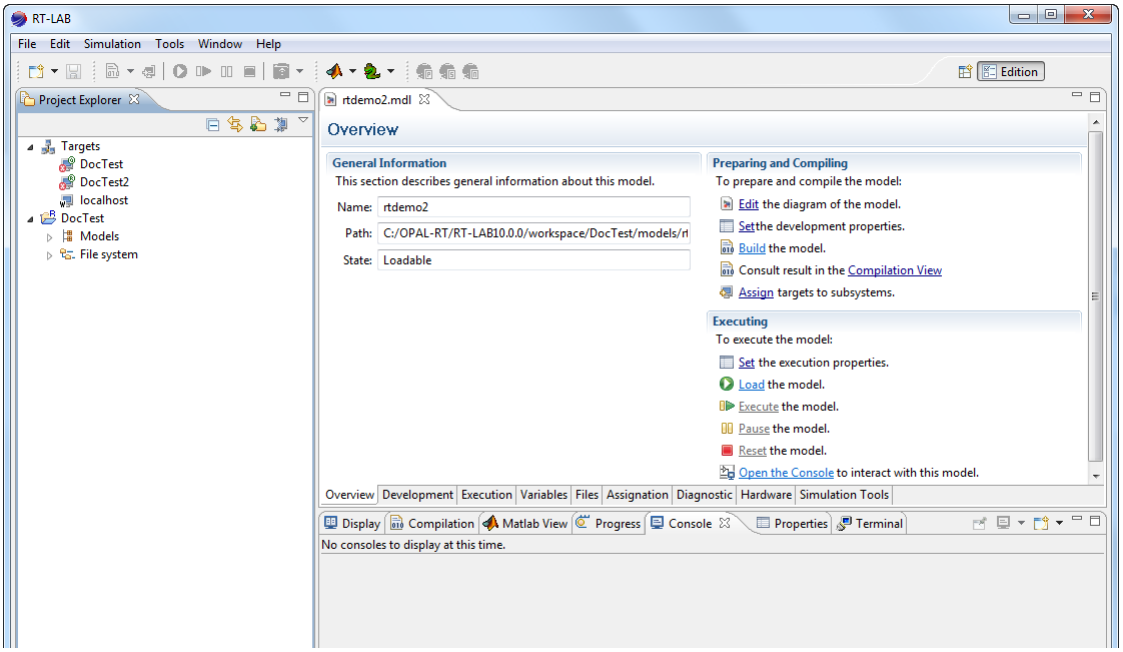


Figure 16: RT-Lab main screen

4. Click on Edit to open the model in Simulink
   - Make the required changes to the model’s Motor and Simulation Parameters (see “Configuring The Model” for instructions).

CONFIGURING THE MODEL

To change model parameters, you must change configuration parameters, in both the SM and the SC subsystems, according to your simulation requirements. Double-click the SM_xsg subsystem to open it.

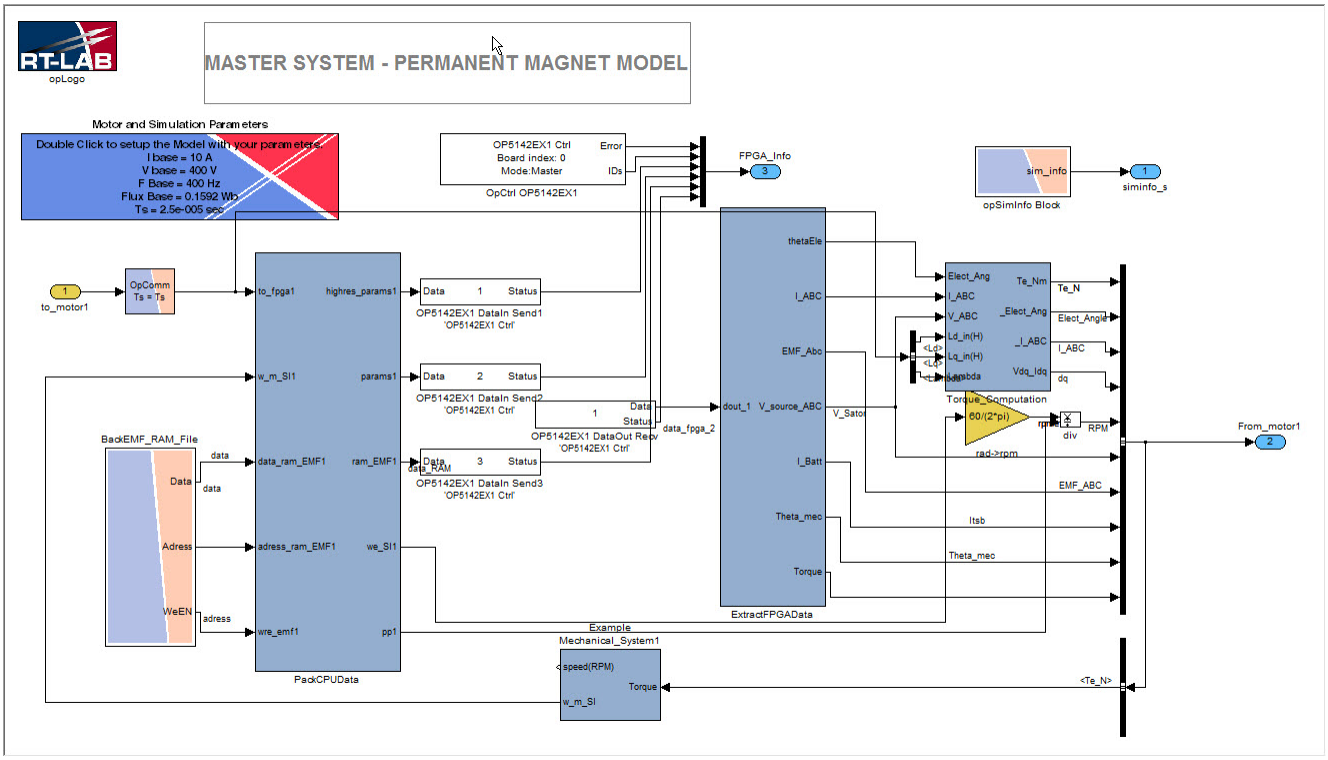


Figure 17: Model master system diagram

5. Double-click to open the “Motor and Simulation Parameters” block and set your custom parameters. Note that this block will only work if the “PM_CPU_OP5142_MotorDrive_CB.m” file is in your current MATLAB path.

Motor and Simulation Parameters, Base Tab

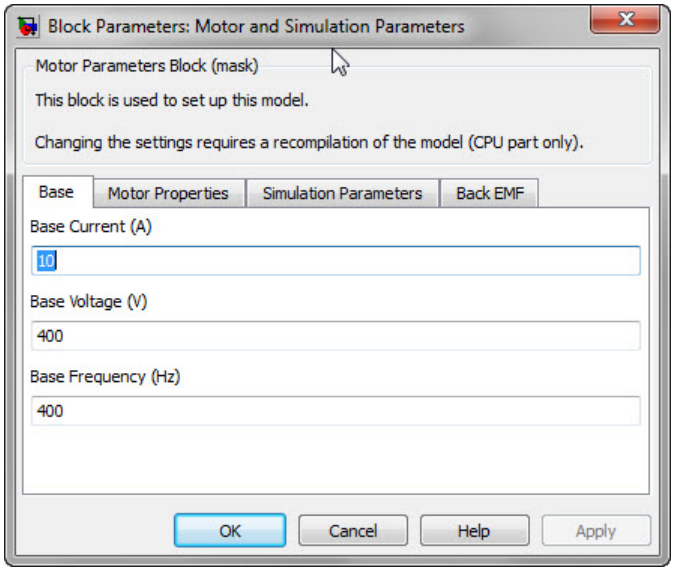


Figure 18: Motor and Simulation Parameters, Base tab

The motor drive model signals are in pu. This section lets you set the the different unit bases for rescaling.

Base Current (A): Scale factor of the Currents (I_Base amperes = 1pu)

Base Voltage (V): Scale factor of the Voltages (V_Base volts = 1pu)

Base Frequency (Hz): Scale factor of the Frequencies (F_Base Hz = 1pu)

All the other bases depend on other settings (torque base, pulsation base, flux base, inductance base and impedance Base) and are automatically processed from them.

A matfile is saved with all these parameters (FPGA_Machine_Params.mat). This file is generated during each model initialization.

Motor and Simulation Parameters, Motor Properties Tab

This section is used to set all the motor's physical parameters.

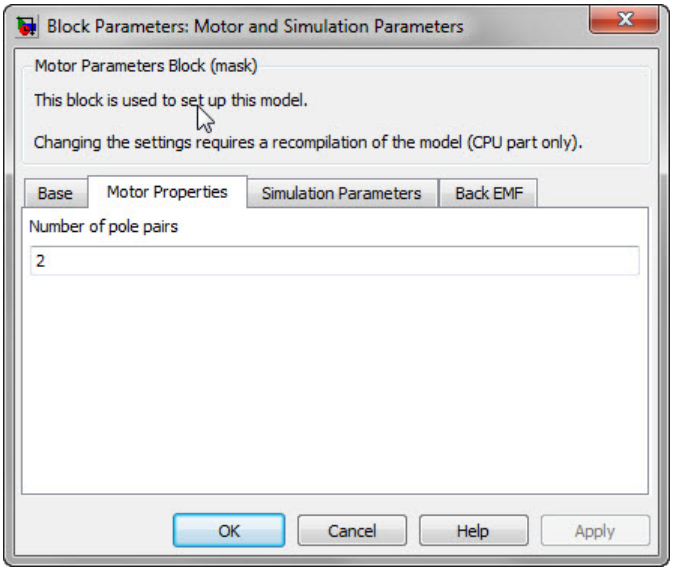


Figure 19: Motor and Simulation Parameters, Motor Properties tab

Number of pole pairs: Number of the machine pair of poles. Must be between 1 and 15. If your value is out of the limits, it will be set to 1 or 15 by default.

Motor and Simulation Parameters, Simulation Parameters Tab

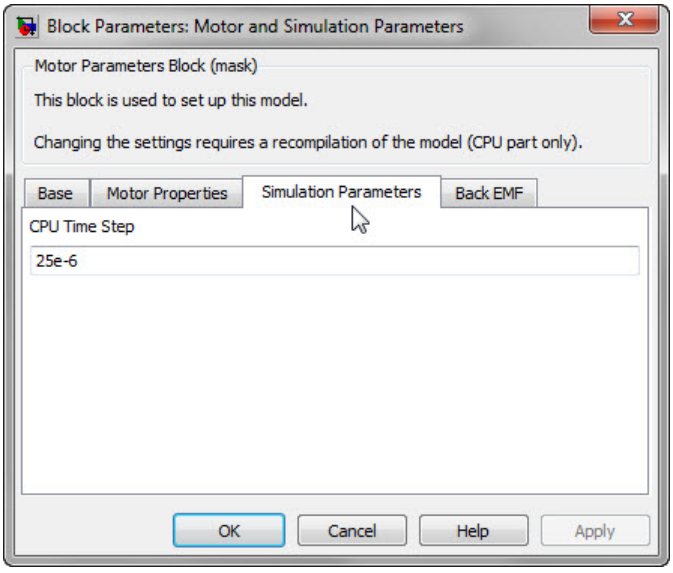


Figure 20: Motor and Simulation Parameters, Simulation Parameters tab

CPU Time Step: This value must be in seconds and represents the sample time in your CPU model. It will be used to set all the blocks that need a sample time to work properly. This value can be recovered with the variable Ts.

Motor and Simulation Parameters, Back EMF Tab

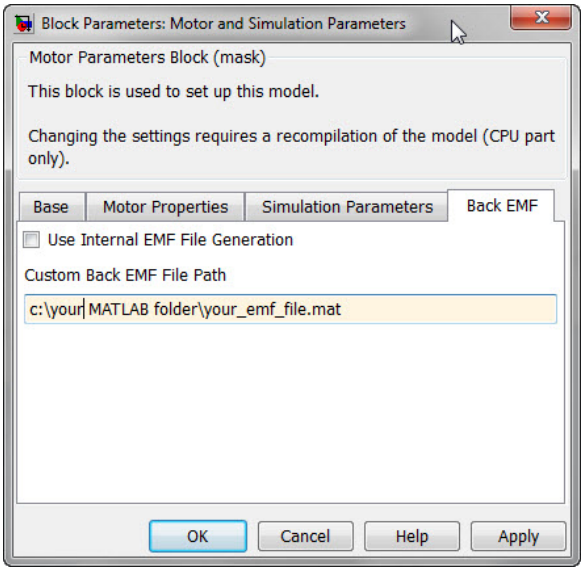


Figure 21: Motor and Simulation Parameters, Back EMF tab

In the model, the back-EMF shape is provided by a matfile. This file contains a variable of 4096 samples, which represents the back-EMF shape normalised to 1 between 0 and 360 degrees (electrical angle).

Use Internal EMF File Generation: If selected, the BackEMF File will be automaticaly generated (and it will be a cosinusoidal based back emf shape). If left unchecked, a field is available to let you specify a back-EMF shape (see below).

Custom Back-EMF File Path: Path of your own Back-EMF File. The Back-EMF file must be a v4 matfile type and contain a single variable. Its dimension must be 2x4096.

The first line is the time (by default : Back-EMF(1,:) = [Ts : Ts : Ts*4096]). The second line is your back Back-EMF (Example: by default Back-EMF(2,:) = cos([0:4095]*2*pi/4096)).

To save the file in the right format, use the following command: save your_emf_file.mat your_emf_variable -v4 (you must enter the complete document path or it will not be saved or accessible).

You can update the model by pressing Ctrl+D to verify the model, then save the model.

SETTING UP THE CONSOLE

Although you can set console parameters at any time, even during simulation, it is a good idea to modify parameters before proceeding to setting execution properties, assigning targets and loading the model.

CHECK THE MOTOR BASES

Open the console (SC) subsystem and view the parameters displayed in the Motor Bases block. If there are any discrepancies, return to “Motor and Simulation Parameters, Base Tab” and make the necessary corrections. You must click on Apply to save any changes.

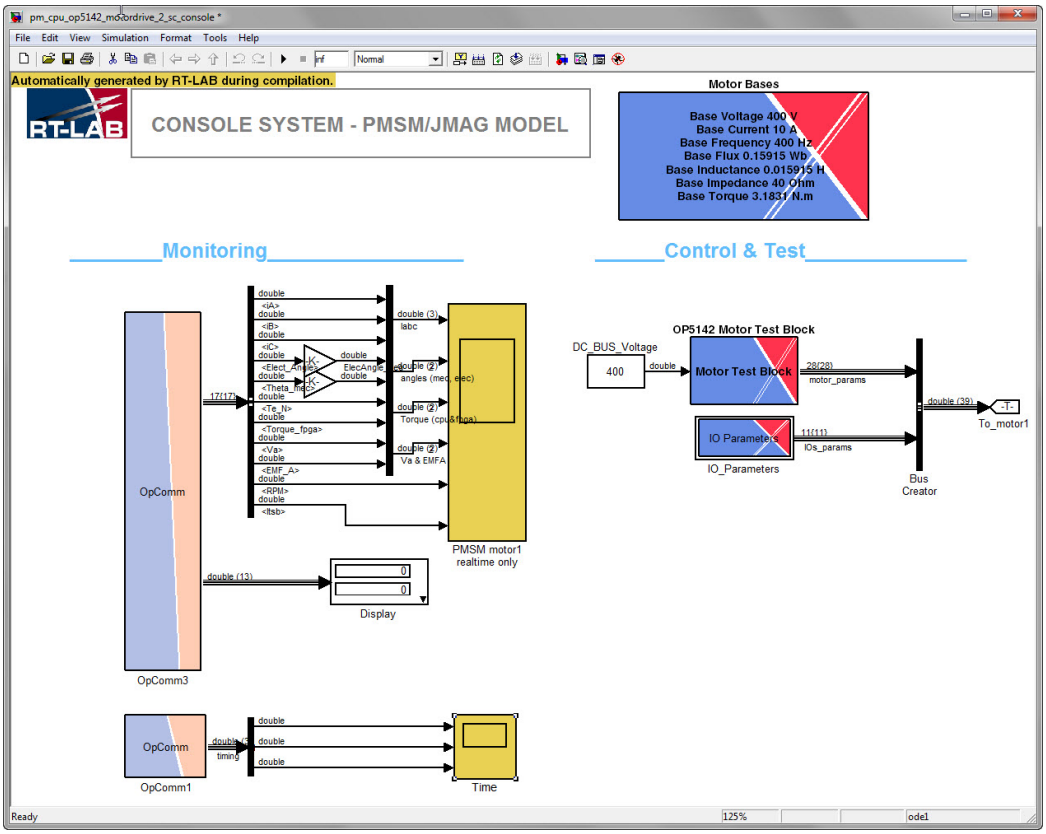


Figure 22: Console setup

SETUP THE MOTOR TEST BLOCK

Double-click on the OP5142 Motor Test Block to open the Function Block Parameters window. This window provides four (4) tabs where you can modify the block parameters: General, Test Source, Motor Parameters and Misc.

General Tab

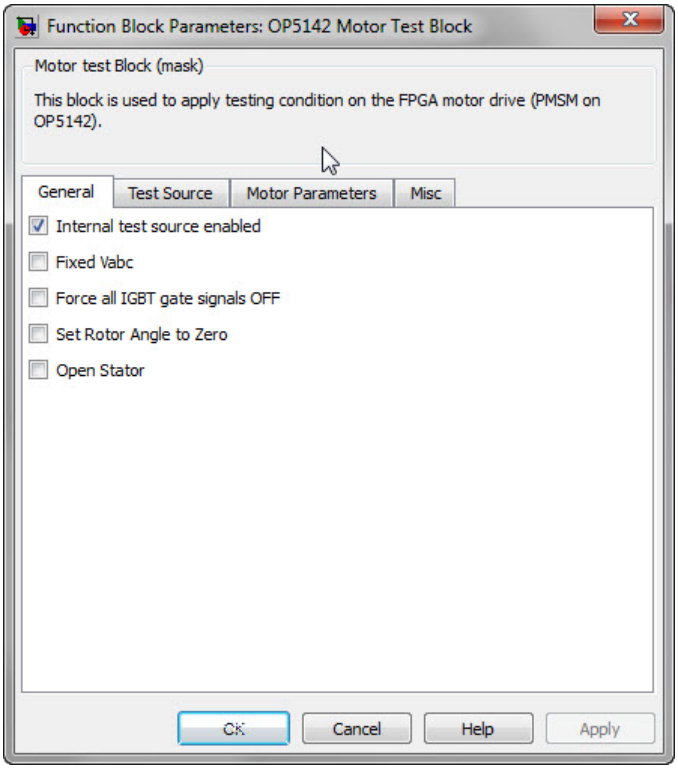


Figure 23: Motor Test Block Parameters (mask), General Tab

Internal Test Source Enabled: This parameter connects the internal test sources to the motor drive. When selected, any external controller is disconnected from the FPGA motor model.

Fixed Vabc: Blocks the terminal voltage to the specified angle, thus applying DC voltage or fixed PWM duty cycle to the PMSM.

Force all IGBT signals OFF: Option to internally force all IGBT signal to off.

Set Rotor Angle to Zero: Blocks the rotor to null angle.

Open Stator: Open the stator phase at the machine. This is made by increasing the stator resistance to a very high value (default=24 pu) so a small residual current can still flow when this option is checked.

Test Source Tab

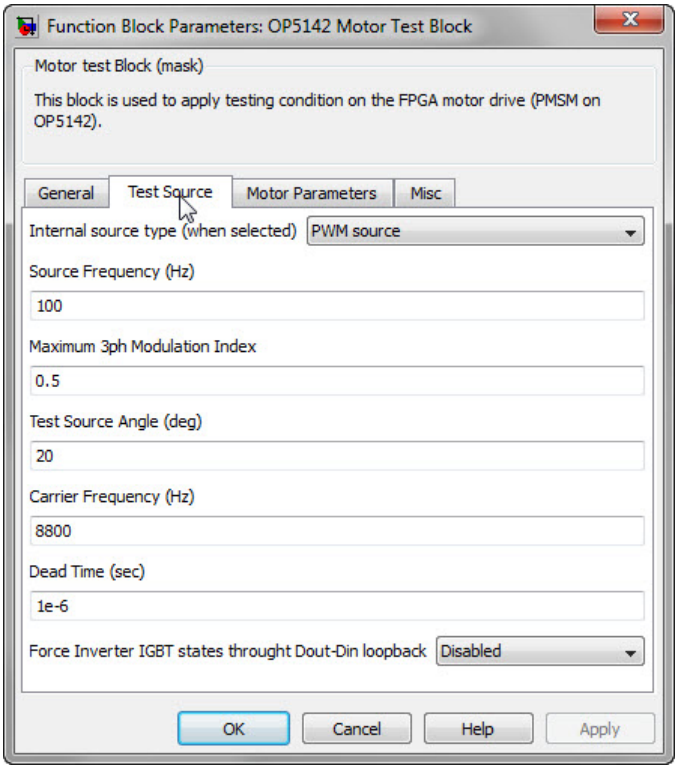


Figure 24: Motor test Block (mask), Test Source tab

Internal source type (when selected): When Internal Test Source Enabled is selected in the General tab, it selects the machine's source. The options provided in this field are:

1. Sinusoidal source (applied directly to the machine terminal thus bypassing the inverter)
2. PWM source; in this option, the inverter is driven by the internal PWM generator.
3. PWM source from IO loopback.
   In this option, the inverter is also driven by the internal PWM generator but sent through the Digital Outout of the simulator and read back at the Digital input. The option will only work if a loopback card is connected at the I/O of the simulator.

Source Frequency (Hz): the frequency of the sinusoidal test voltage source applied to the machine terminal or the frequency of the PWM modulation.

Sinusoidal 3ph test source amplitude (V): the amplitude in volts of the internal sinusoidal test source (when selected).

Maximum 3ph Modulation Index: the maximum modulation index of the internal PWM test source when selected. If this parameter is called mi and the DC bus voltage, Vdc, then the equivalent tri-phase sinusoidal voltage source applied to the motor is Vdc/2*mi*sin (t)

Test Source Angle (deg): the phase in degree of the internal test source or equivalent PWM modulation signal

Carrier frequency: PWM carrier frequency when PWM source is selected.

Dead time (us): dead-time of PWM waveform when PWM source is selected.

Motor Parameters Tab

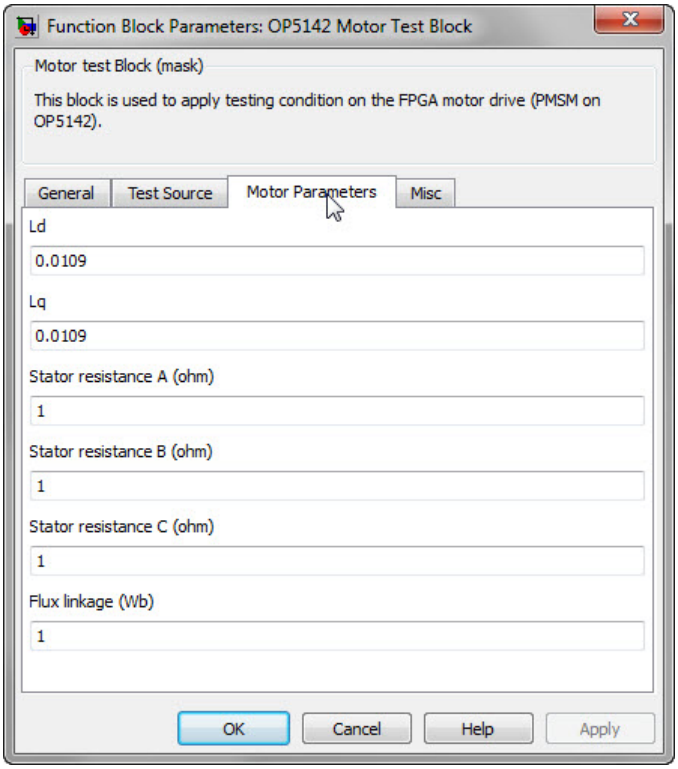


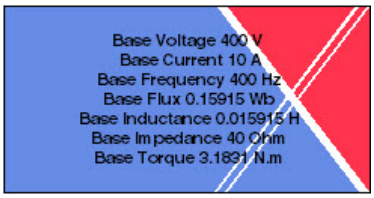
Figure 25: Motor test Block (mask), Motor Parameters tab

Ld, Lq (Henry): these parameters are the motor`s inductance in dq domain. The values must be between 16*M1.Lbase and 1/16*M1.L_base

Stator Resistances (ohm): these parameters are the resistance of the stator in Ohms. The values must be between 0 and 4*M1.Z_base.

Flux Linkage (Wb): this parameter is the motor flux linkage in Webers. The values must be between 0 and 4*M1.Flux_base. .

Refer to the “Motor Bases” block which displays all these bases. Only the first 3 bases are user-defined. Remaining base values are dependent on other configuration parameters and cannot be changed directly.



Misc Tab

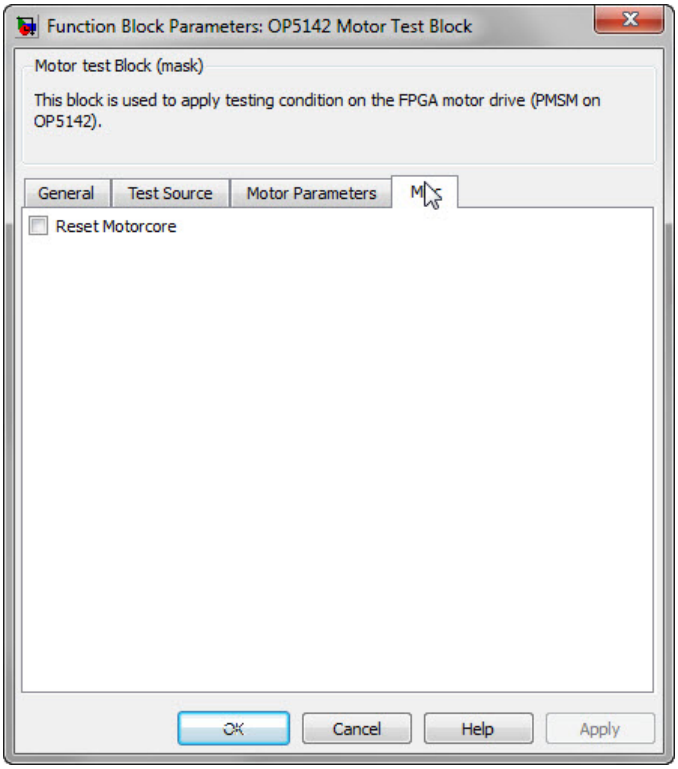


Figure 26: Motor test Block (mask), Motor Parameters tab

This tab lets you reset the motorcore. Check reset box then click Apply to reset the motor core. Uncheck then click Apply to return to the usual simulation mode.

SETUP YOUR I/O GAINS

Double-click the **IO Parameters block** to open the Source Block Parameters window. This window provides access to three (3) tabs: DAC Gains, ADC Parameters and Sensors Parameters.

The detailed I/O pin-out is provided in your I/O specification file.

DAC Gains Tab

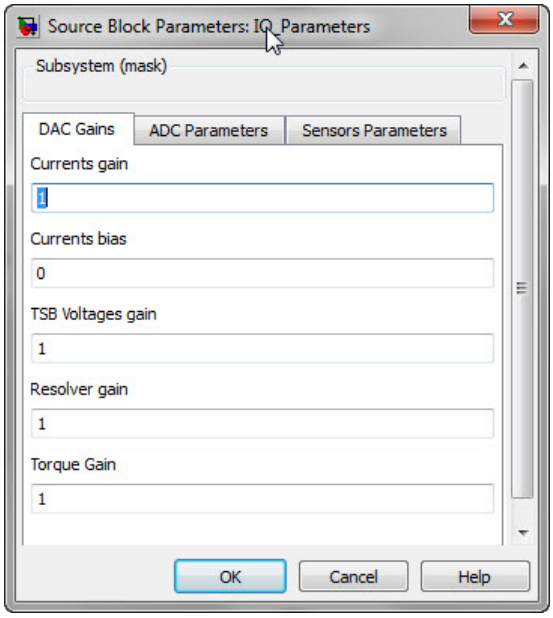


Figure 27: IO_Parameters, DAC Gains tab

The format uses for the gain parameters is Ufix17_10; you must enter value between 0 and 128 (otherwise the gain will be saturated to 128).

The format uses for the bias parameter is Fix18_10; you must enter value between -128 and +128 (otherwise the gain will be saturated to -/+ 128).

The default scaling of the analog outputs is 1V for 1pu. For example, if you set a gain to 4, the scaling changes to 4V for 1pu.

Currents Gain: gain applied to the motor`s current values on analog outputs.

Currents Bias: offset applied to the motor`s current values on analog outputs.

TSB Voltages Gain: gain applied to the TSB voltages values on analog outputs.

Resolver Gain: gain applied to the resolver`s values on analog outputs.

Torque Gain: gain applied to the motor`s torque value on analog outputs.

ADC Parameters Tab

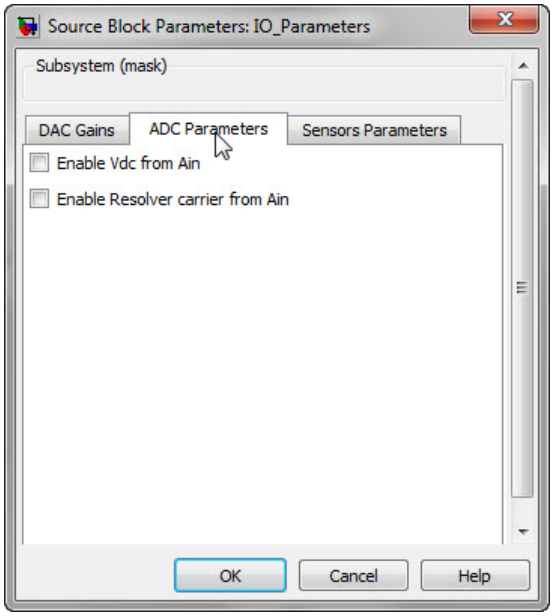


Figure 28: IO_Parameters, ADC Parameters tab

Enable Vdc from Ain: enable to use the Analog input channel for Vdc as internal Inverter's Vdc. If selected, a text field appears:

- *Vdc from Ain gain*: Gain to adjust the Vdc from ain amplitude. By default: 1V = 1pu = 1 M1.V_ base. For example, if you set the value to 0.5 so 1V = 0.5pu = 0.5 M1.Vbase.

Enable resolver carrier from Ain: Enable to use the Analog input channel for resolver carrier as internal resolver carrier. If selected, a text field appears:

- *Resolver excitation gain*: Gain to adjust the resolver carrier from Ain amplitude.

Sensors Parameters Tab

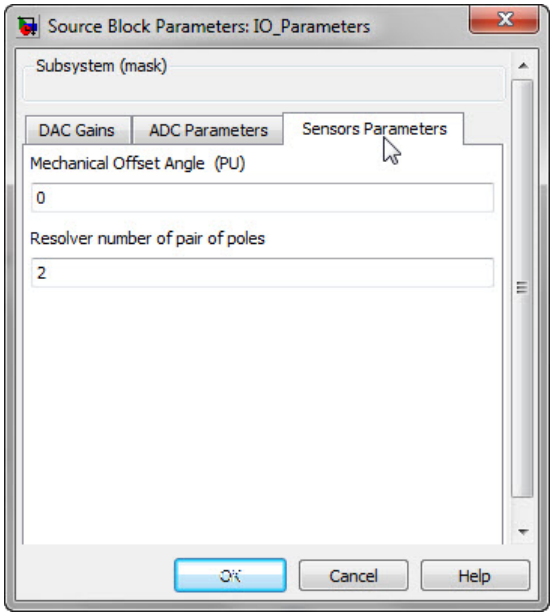


Figure 29: IO_Parameters, Sensors Parameters tab

You can specify a different pair of pole numbers for your resolver (between 1 and 15). Take special care when using this feature because, by default, the resolver has the same number of pair of poles as the motor, so it represents the electrical angle (and not the mechanical).

COMPILING, LOADING AND EXECUTING THE MODEL IN RT-LAB

Once you have made all the necessary parameter changes, you must assign a target and set the execution properties before you can can compile, load and execute the model.

1. In the RT-LAB main screen, select the model, then select the *Assignation* tab and select the physical node (target) for the model.

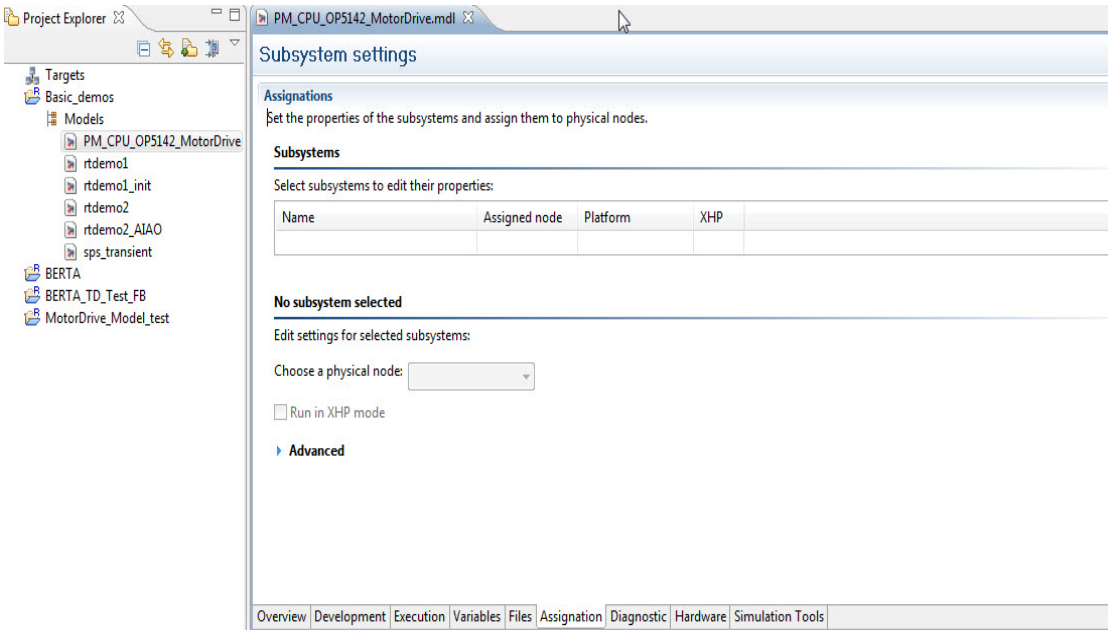


Figure 30: RT-LAB main window, Assignation tab

2. Click on the Execution tab and select the desired properties for executing the model.
3. Compile the model .
4. Load and Execute the model.

For more instructions on how to use RT-LAB, please consult the RT-LAB Quickstart Guide

VIEWING THE SIMULATION

When the model is running, you can click on the scope in the SC console subsystem to view the graphic representation of the simulation.

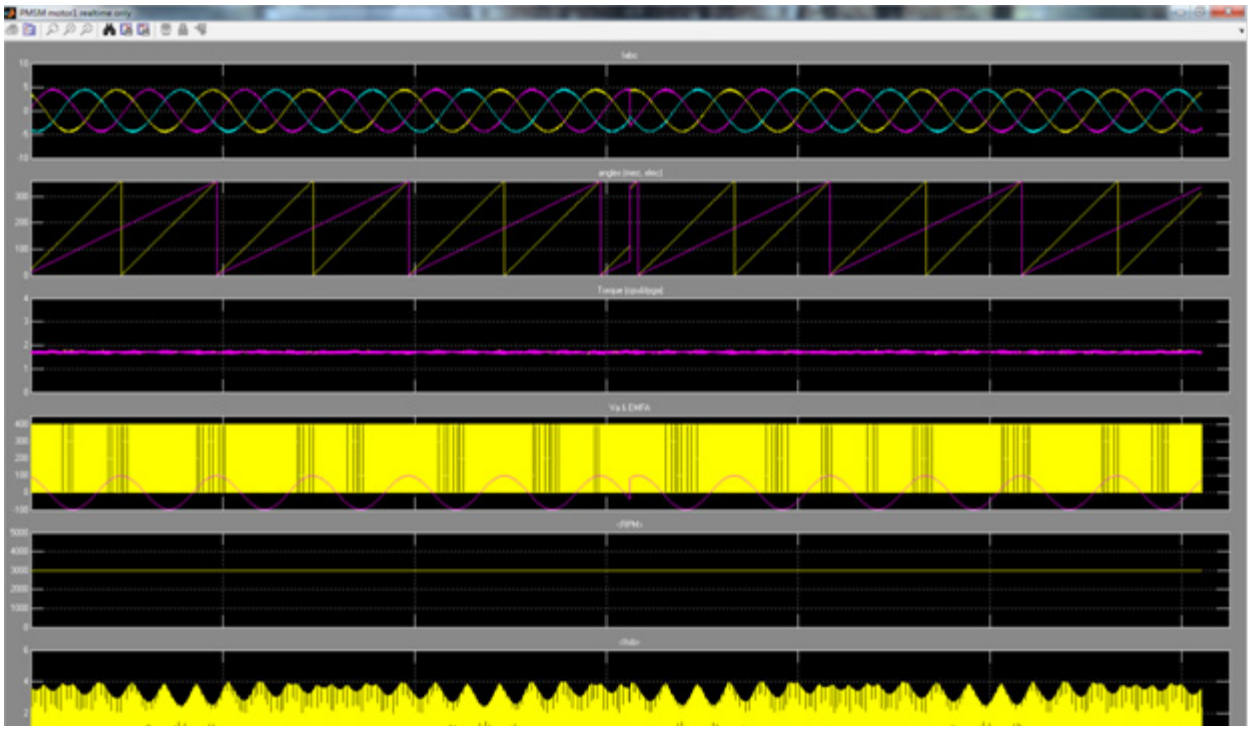


Figure 31: Model simulation scope

When you view the simulation in this manner, it is an asynchronous and under-sampled representation and does not reflect true FPGA signals. To view accurate representations, you must connect to hardware I/Os.

OP5142 MOTOR SIMULATION DIAGRAM

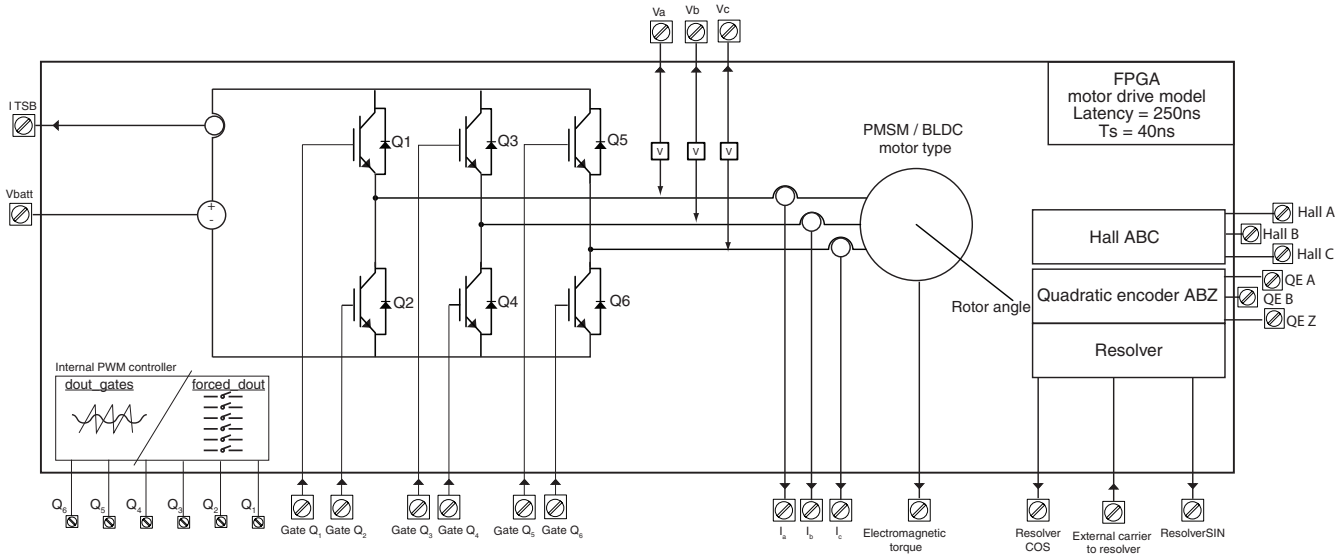


Figure 32: OP5142 motor simulation functional schematic

GENERATING PARAMETERS

This document demonstrates how to modify motor parameters for PMSM and JMAG-based FGPA motor emulator and use the provided script to produce the required data files. This operation is only applicable to the XSG drive model provided by Opal-RT Technologies.

HOW TO GENERATE PARAMETERS

The Opal-RT FPGA motor drive models require the user to go through a data generation process to obtain simulation data for the run-time model on the FPGA. In the example of the JMAG model, the procedure will launch Simulink simulation models to 'Extract' inductance and back-EMF data to be loaded on the FPGA chip.

This procedure is done using the Opal-RT 'FPGA data Generator' block, depicted in Figure 33.

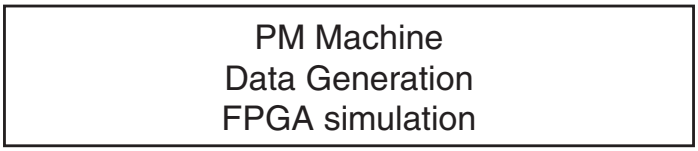


Figure 33: FPGA Data Generator block

This block enables the user to do the following:

1. Select the type of Motor to be run on the FPGA: PMSM or JMAG, and other general parameters
2. Set the motor base value used to normalize the calculations in the FPGA.
3. Set the PMSM parameters: Ld,Lq, flux, Stator Resistance, number of poles
4. Set the JMAG parameters: Stator Resistance, number of poles, .rtt filename
5. Set the nodal inverter parameters (in case this model is used on the FPGA)

FPGA DATA GENERATOR BLOCK

General Tab

Opening the FPGA Data Generator Block gives access to the General tab of the block parameters window.

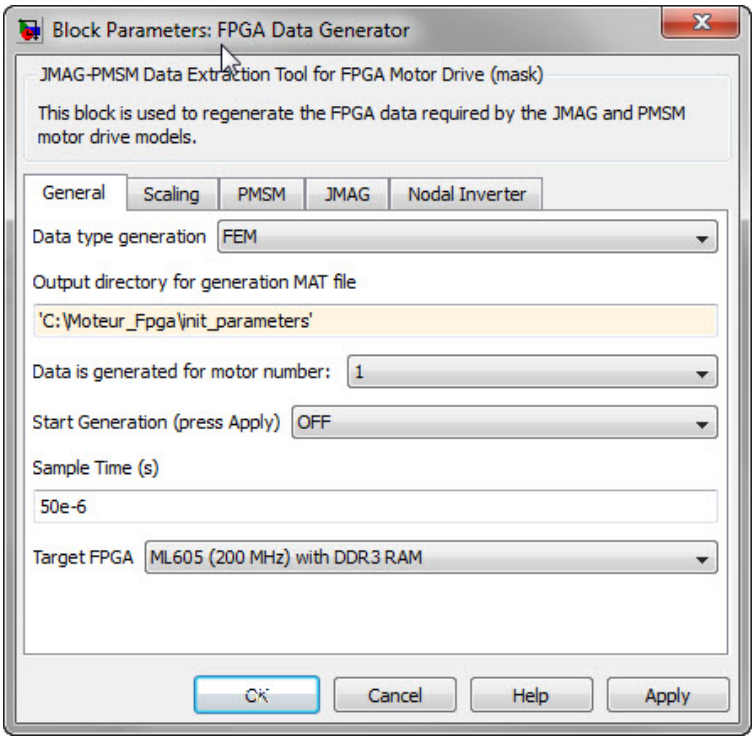


Figure 34: FPGA Data Generator block, General tab

This tab lets the user select the simulation data details from the following fields.

Data type generation: Select the type of motor from the drop-down list.

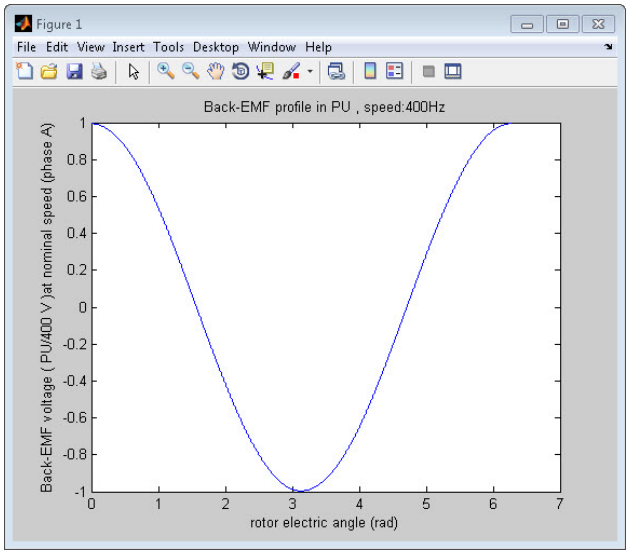
Output directory: Specify the exact directory into which the required data mat-files will be generated and recorded. The files will be generated according to the following format:
Motor inductance data files: RAM_MotorType_MotorNumber.mat
Motor cogging torque data file: TOG_MotorType_MotorNumber.mat
Motor Back-EMF data: EMF_MotorType_MotorNumber.mat
Nodal inverter data: NodalInverterData_MotorNumber.mat
Common general parameters: FPGA_Machine_Params.mat:

- With MotorType={PMSM or JMAG} and MotorNumber={1 or 2}
- FPGA_Machine_Params.mat contains 2 MATLAB structure variable M1 and M2 that contains the parameter of each motor

Data is generated for motor number: Select the motor number for which data will be generated.

Start Generation: Select either ON or OFF and click on Apply to launch or stop data generation.

The *Start Generation (press Apply)* field returns to 'OFF' and the MATLAB the inductance and back-EMF values are being generated. The process is finished when MATLAB displays a figure with the Back-EMF values at nominal speed.



Sample Time (s): Sample time for the RT-Lab CPU subsystem interface with FPGA.

Target FPGA: Select the type of FPGA board where the motor drive model will be generated.

Scaling Tab

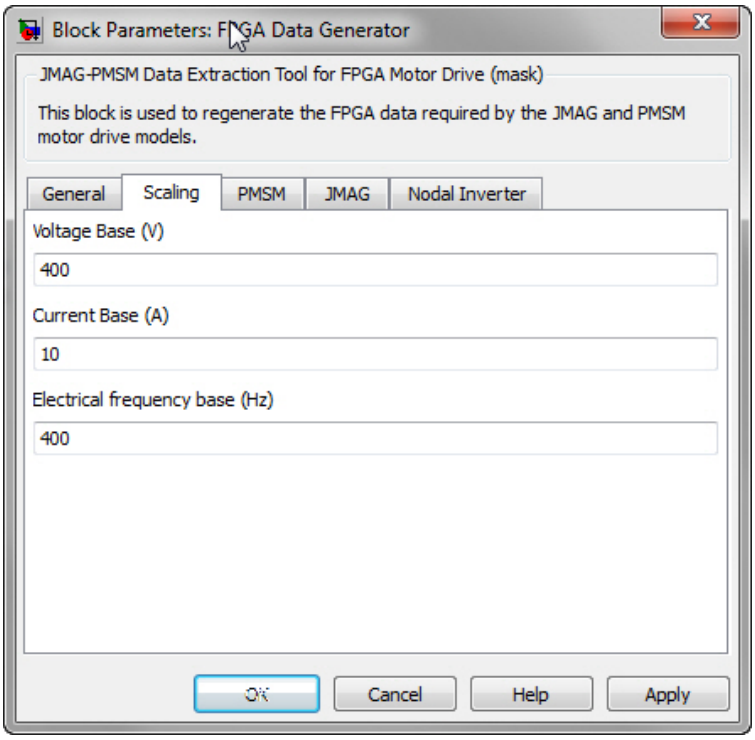


Figure 35: FPGA Data Generator block, Scaling tab

This tab enables the user to select scaling factors to be used for calculation inside the FPGA. From these bases, all other bases can be deducted: Power, Torque, Flux, Inductance, etc.

Typically, you will set these parameters with the rated values of the drive and motor. The FPGA model is designed to be able to correctly compute numbers reaching 4 times these values. It is important to set these values correctly: if the chosen bases are too low, some overflow can occur in the model calculation. Inversely, if the selected bases are too high, some resolution and precision will be lost.

Voltage base (V): The motor rated voltage. This is the peak voltage to be applied to the motor.

Current base (A): The motor rated current. This is the peak current to reach in the motor.

Electrical Frequency base (Hz): The motor rated electrical frequency. This parameter can be determined from RPM rating in the following manner: Electric_Frequency_base= RPM/60*Number_of_Pair_Pole, where Number_of_Pair_Pole is the number of pole pairs of the machine.

PMSM Tab

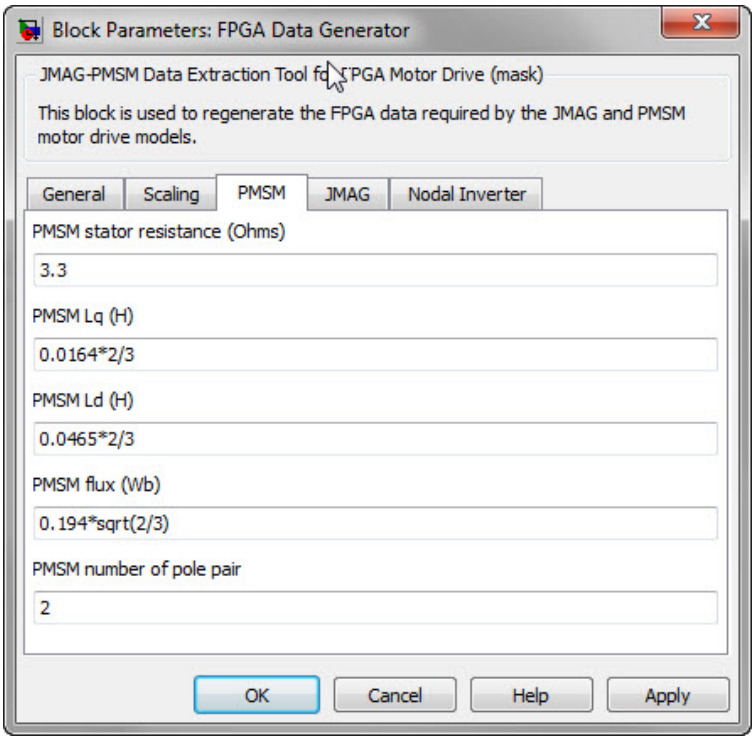


Figure 36: FPGA Data Generator block, PMSM tab

When PMSM is the selected Data generation type, this tab lets the user select the PMSM parameters.

PMSM Stator Resistance (Ohms): The stator resistance of the PMSM.
PMSM Lq Inductance (H): The Park equivalent quadrature axis inductance in Henry.
PMSM Ld Inductance (H): The Park equivalent direct axis inductance in Henry.
PMSM flux (Wb): The PMSM flux per pole in Weber.
PMSM number of pair of poles: The number of pair of poles of the PMSM

JMAG Tab

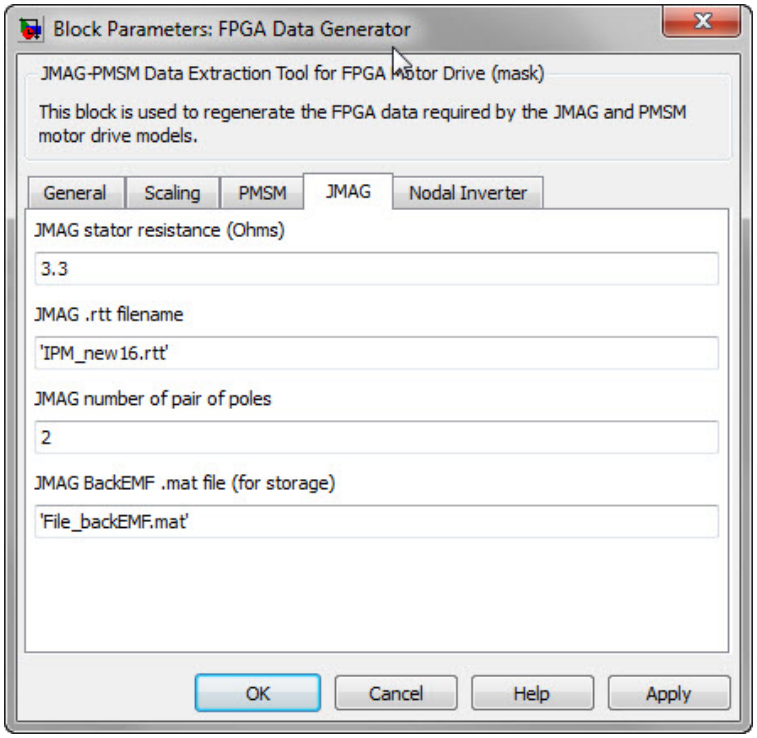


Figure 37: FPGA Data Generator block, JMAG tab

When JMAG is the selected Data generation type, this tab lets the user select the JMAG Finite-Element Analysis model parameters.

JMAG Stator Resistance (Ohms): The stator resistance of the PMSM.

JMAG .rtt filename The name of the JMAG .rtt file.

JMAG number of pair of poles: The number of pair of poles of the PMSM

JMAG BackEMF .mat file (for storage): Temporary filename for BackEMF data storage.

Nodal Inverter tab

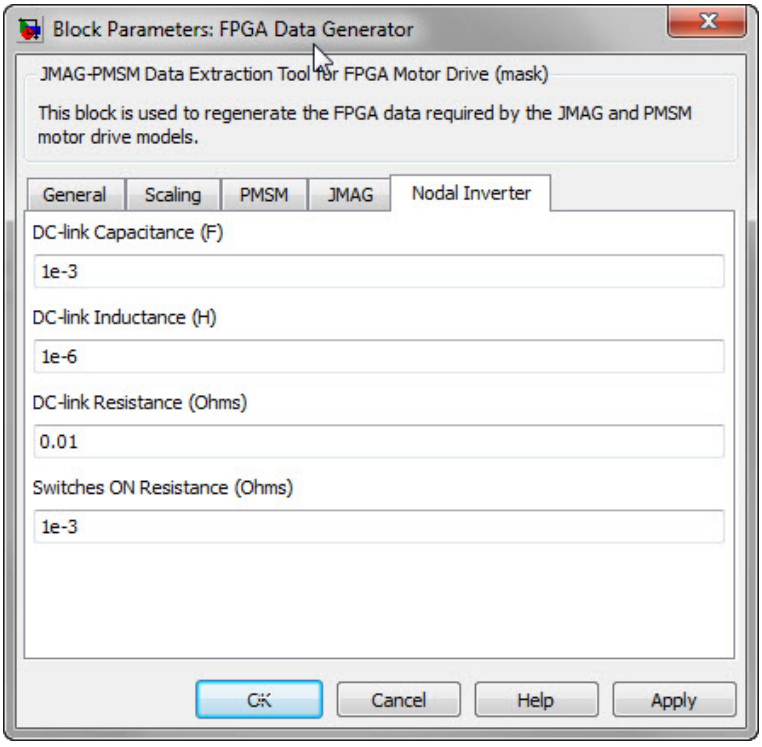


Figure 38: FPGA Data Generator block, Nodal Inverter tab

This tab lets users set the nodal inverter model parameters. The nodal inverter model uses a classic nodal solver that allows the user to apply various faults. The bridge model also includes an L-C filter in front of the inverter.

DC-link Capacitance (F): The capacitance in Farad of the LC filter of the DC-link.

DC-link inductance (H): The inductance in Henry of the LC filter of the DC-link.

DC-link Resistance (Ohms): The resistance in Ohms of the LC filter of the DC-link. This represents the losses of the inductor.

Switches ON Resistance (Ohms): The conduction state resistance in Ohms of the inverter IGBT and diodes.

REBUILDING AND EXECUTING THE MODEL

Whenever a set of model parameters changes, the RT-LAB model, namely the CPU model, must be recompiled to implement the modification.

The XSG Model need not be rebuilt, which means that no new bitstream generation is involved when changing motor parameters. Also, before loading, the data files for the inductance matrix and the back EMF matrix must be configured to transfer during the loading process in Binary mode, as shown in Figure 6.

These files have different suffixes (_JMAG or _PMSM) so you must transfer the appropriate files. As the model will be recompiled with the correct names, you can also transfer both suffixed file types.

The figure below sets the file transfer for motor #1 as PMSM and motor #2 as JMAG. Note that even if your model does not use the nodal inverter, the NodalinverterData matfiles must be provided.

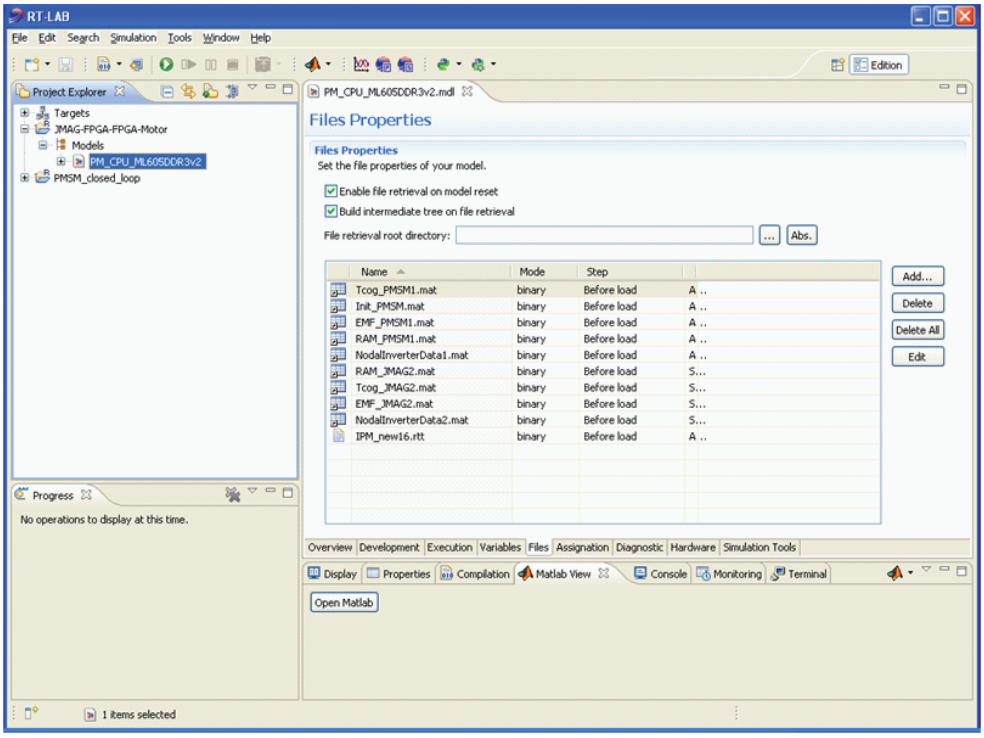


Figure 39: RT-Lab Files menu for transferring FPGA motor parameter files.

ADVANCED SIMULATION USING THE FPGA-PMSM MODEL

FPGA MOTOR DRIVE DATA GENERATOR

The Opal-RT FPGA Electric Motor Controller model is designed for a JMAG model and is configured with the appropriate default values. You can consult (or modify) these values by opening the FPGA Electric Motor Controller data generator.

MODIFYING FPGA MOTOR DRIVE DATA GENERATOR VALUES

See the previous chapter, “Generating Parameters”, for more detailed information and step by step instructions.

FPGA DATA GENERATOR BLOCK

General Tab

Opening the FPGA Data Generator Block gives access to the General tab of the block parameters window.

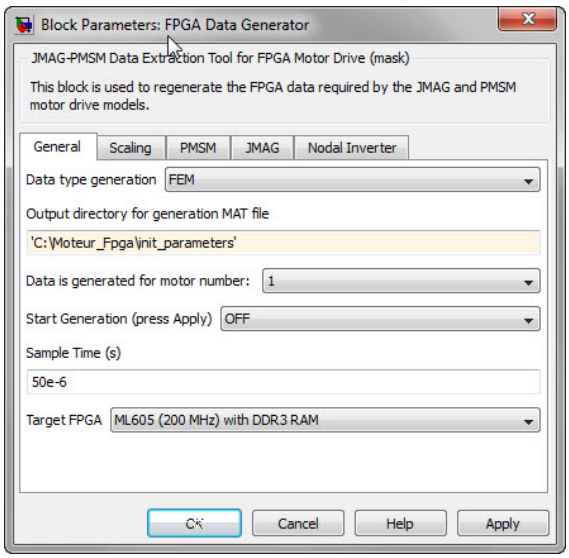


Figure 40: FPGA Data Generator block, General tab

This tab lets the user select the simulation data details from the following fields.

Data type generation: Select the type of motor from the drop-down list.

Output directory: Specify the exact directory into which the required data mat-files will be generated and recorded. The files will be generated according to the following format:
Motor inductance data files: RAM_MotorType_MotorNumber.mat
Motor cogging torque data file: TOG_MotorType_MotorNumber.mat
Motor Back-EMF data: EMF_MotorType_MotorNumber.mat
Nodal inverter data: NodalInverterData_MotorNumber.mat
Common general parameters: FPGA_Machine_Params.mat:

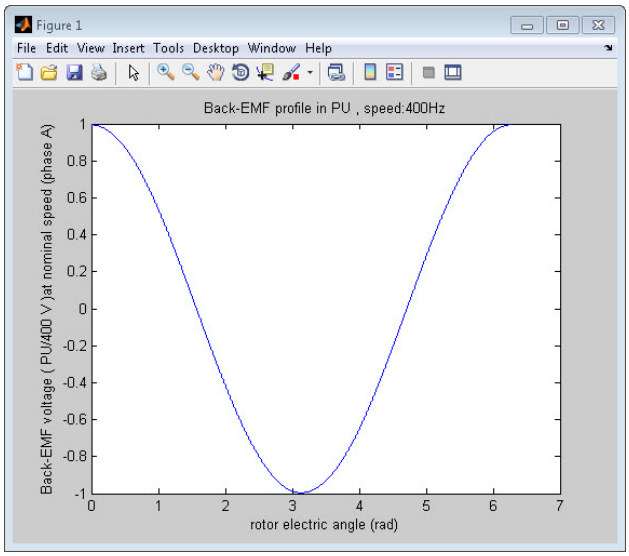
- With MotorType={PMSM or JMAG} and MotorNumber={1 or 2}
- FPGA_Machine_Params.mat contains 2 MATLAB structure variable M1 and M2 that contains the parameter of each motor

Data is generated for

motor number: Select the motor number for which data will be generated.

Start Generation: Select either ON or OFF and click on Apply to launch or stop data generation.

The *Start Generation (press Apply)* field returns to 'OFF' and the MATLAB the inductance and back-EMF values are being generated. The process is finished when MATLAB displays a figure with the Back-EMF values at nominal speed.



Sample Time (s): Sample time for the RT-Lab CPU subsystem interface with FPGA.

Target FPGA: Select the type of FPGA board where the motor drive model will be generated.

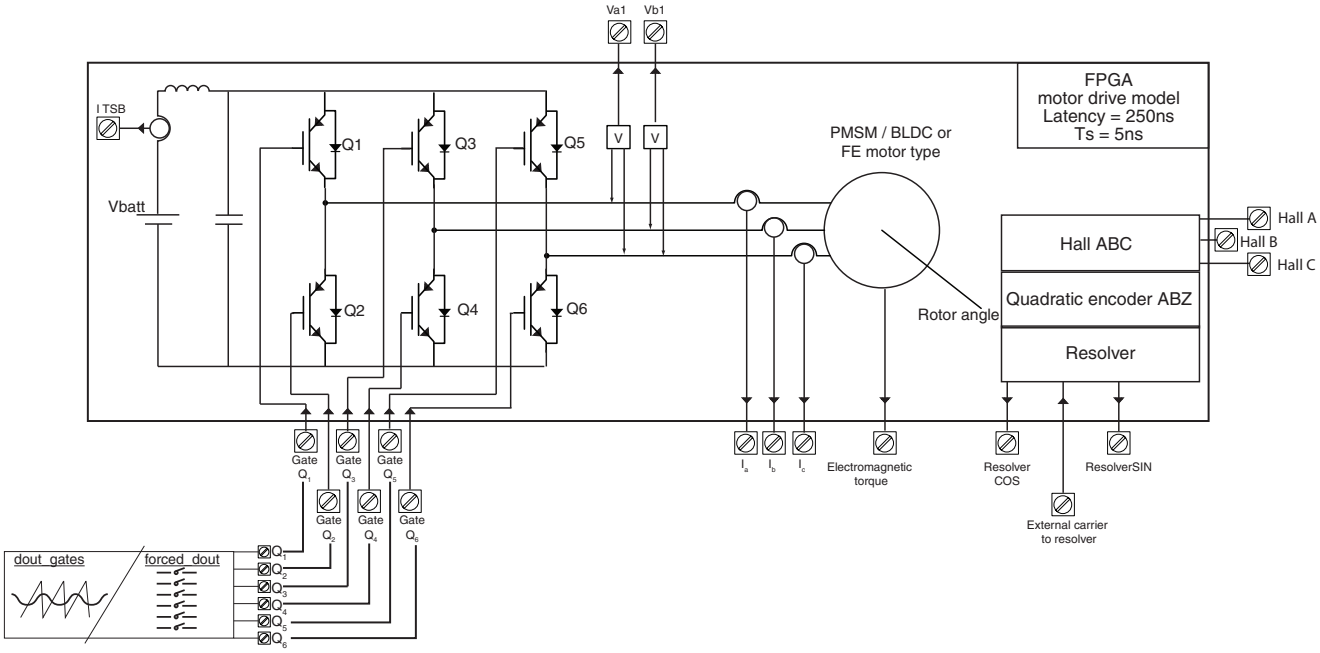


Figure 41: PMSM drive with test PWM modulator implemented on the FPGA

WORKING WITH THE ELECTRIC MOTOR CONTROLLER TEST MODEL

The 'Motor Test Block' of the RT-LAB Console of the FPGA PMSM Drive was made to help you:

- Verify the functionalities of the FPGA-PMSM drive model.
- Verify and adjust the various model sensors.
- Connect an external controller to the model.

It is important for you to understand that the PMSM motor can be tested without a machine controller. Because we are working with virtual components, it is very easy to make the motor back-EMF and stator voltage turn in full synchronism. In real-life, this is in fact exactly what the controller tries to do to produce a useful electric torque and make the machine turn. In our case, the rotor speed (and therefore the backEMF) and the stator voltage (sinusoidal or PWM) are simply computed to be synchronous when in test mode.

VERIFYING THE ELECTRIC MOTOR CONTROLLER MODEL FUNCTIONALITIES

The Electric Motor Controller should give the following reading upon loading in the default mode (sinusoidal internal test source 100 Hz, 30 degree difference between backEMF and terminal voltage).

Figure 43 shows some key data coming from the PMSM-FPGA model. The top curve displays the motor currents, which are sinusoidal. The 2nd curve outputs the electrical and mechanical angles of the rotor, which can help determine the motor speed and direction. The 3rd curve is the motor torque (computed on both CPU and FPGA). This last curve indicates one phase of terminal voltage along with the corresponding back-EMF. The default model is made so the terminal voltage amplitude is always nearly equal to the Back-EMF. Then, by changing the angle of the terminal voltage one can observe the phase shift on the 4th trace and at the same time the torque variation. For example, with terminal voltage and back-EMF voltage amplitude being equal, the torque becomes almost zero when the angle between terminal voltage and the back-EMF is also null. This is normal and one can observe the similarity with fundamental power system equation for power transmission:

$$P = \frac{V_{\text{term}} * V_{\text{backEMF}}}{X_{\text{machine}}} \sin(\theta)$$

Figure 42: Fundamental power system equation for power transmission

Where θ is the angle between the terminal voltage and the Back-EMF voltage.

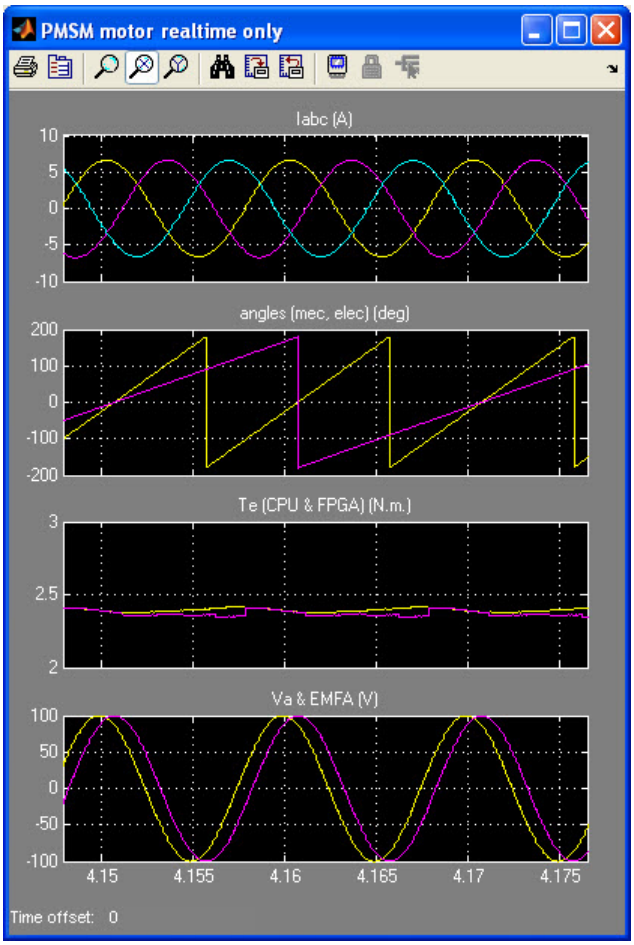


Figure 43: Openloop test , sinusoidal source 100 Hz, 30 deg.

Selecting the PWM source should display very similar waveforms for the currents and average torque (Figure 43) if the selected modulation index (mi) is set so ½*Vdc*mi equal the sinusoidal source amplitude. The PWM application can be viewed on the terminal voltage curves. Be aware that the relatively slow sampling of the simulator Console (typ: 50 µs) may cause strong aliasing effect on the PWM modulated terminal voltage.

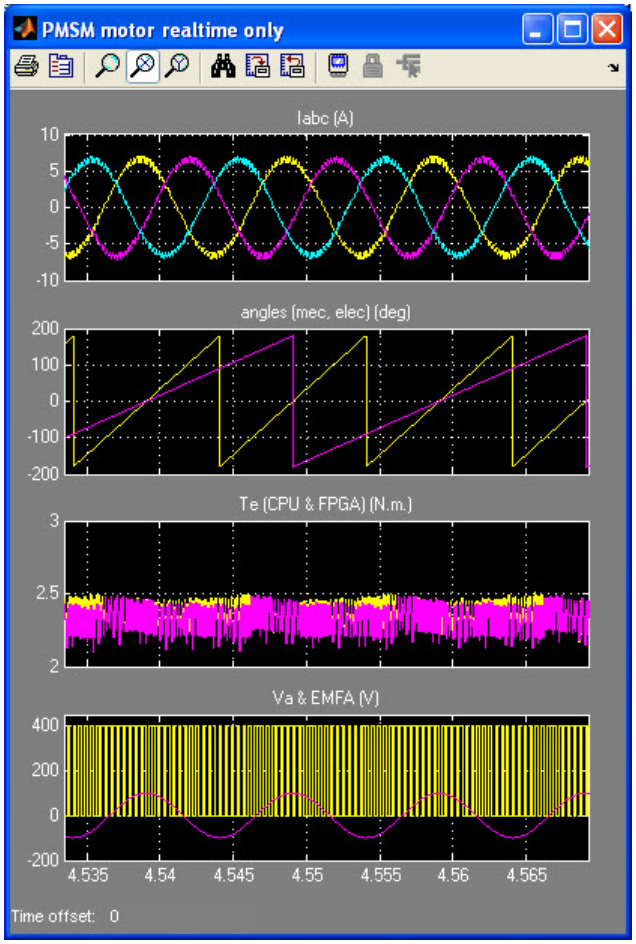


Figure 44: Openloop test (100 Hz 30 deg PWM=2.5 kHz, 1 µs dead-time)

It is possible to turn off the IGBT pulse in the PWM mode by setting the "**Force all IGBT gate signals OFF**" in the 'Motor Test Block'. In this mode, the IGBT inverter will act as a diode rectifier when the backEMF becomes higher than the terminal voltage (Figure 45).

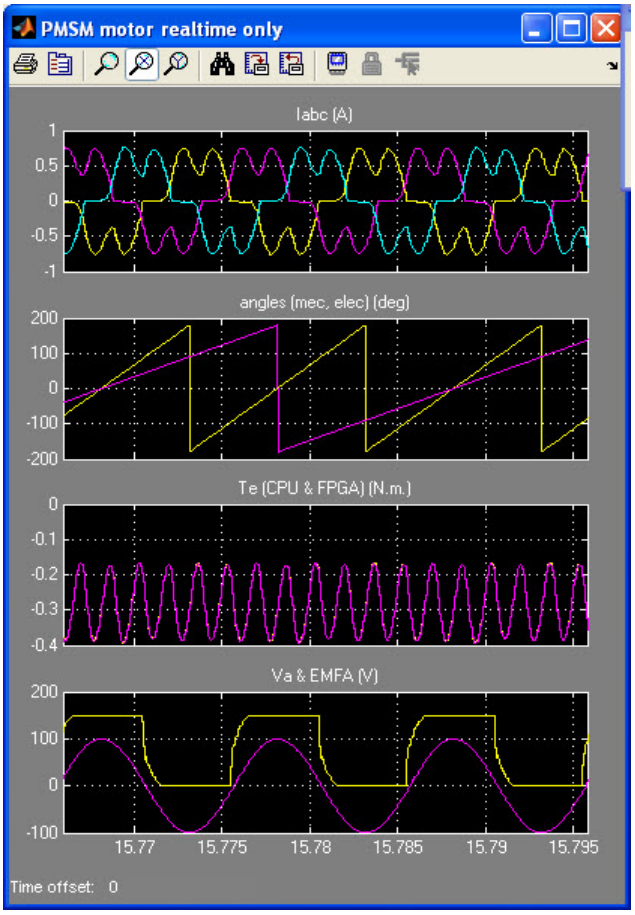


Figure 45: PWM mode, pulse disabled (100 Hz 30 deg, Vdc-link =150V)

CLOSING THE LOOP ON AN EXTERNAL CONTROLLER

This section explains how to close the loop on the FPGA-PMSM model with an external controller. The external controller reads the motor currents through the analog outputs of the FPGA-PMSM model and will connect its own FPGA pulse to the Electric Motor Controller through the digital inputs.

The rotor angle can be read in several ways. The electric motor controller sends unmodulated sin-cos (optional), resolver-modulated sin-cos (with and without external excitation) through the analog outputs and quadrature encoder signals through the digital outputs.

Closing the loop will be done progressively, step-by-step, to allow testing the different components necessary to its proper functioning.

- Verify that the user motor parameter are correctly read by the model (scaling, bitstreams, etc..).
- Verify the motor speed reading on the controller.
- Verify the scaling of the motor currents as read in the external controller.
- Verify the motor dq axis frame referential.
- Verify the current control in closed-loop.
- Verify the speed control.

SPEED, ANGLE SENSORS ILLUSTRATIONS

The images provided below are graphic representations of the Encoder, Resolver and Hall effect signals.

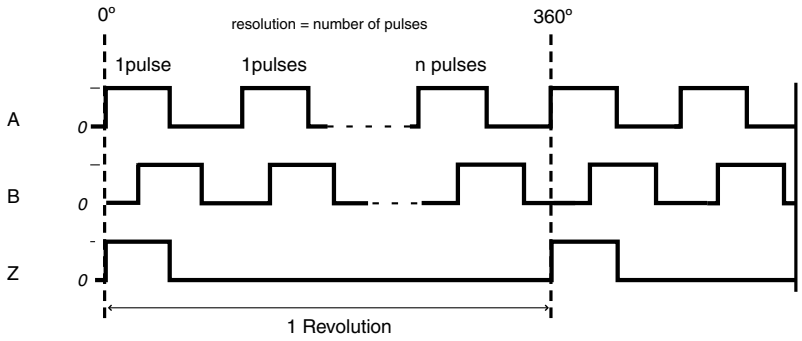


Figure 46: Quadrature encoder signal

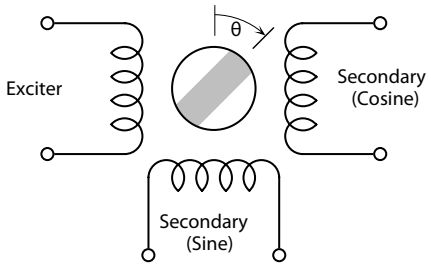


Figure 47: Resolver configuration

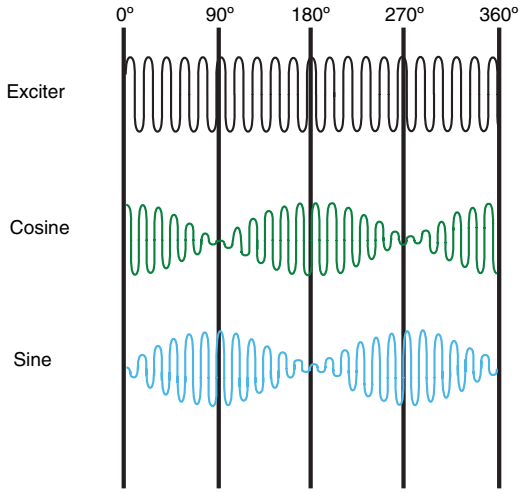


Figure 48: Resolver signal

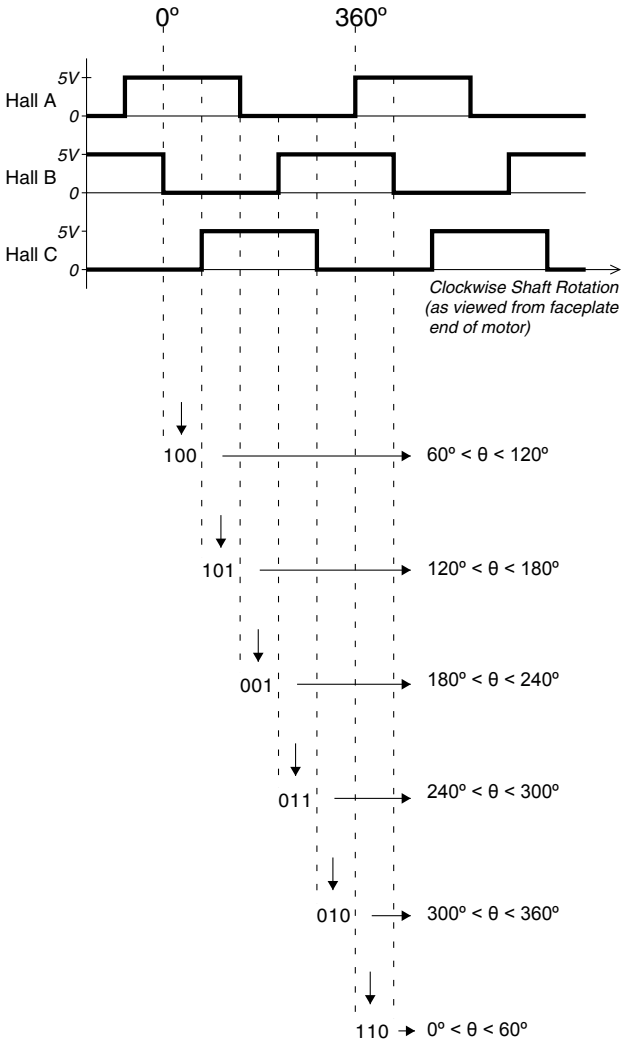


Figure 49: Hall effects signal

VERIFYING THAT THE SIMULATOR ACCEPTS USER MOTOR PARAMETERS CORRECTLY

The first step in using the FPGA-PMSM model is to make sure that the user parameters have been correctly incorporated in the model.

After loading the user model, the following items should be checked:

BackEMF amplitude: In Fixed-speed (user nominal speed) and internal sinusoidal source mode, the BackEMF amplitude should be verified to be equal to $\Phi^*\omega e$, where Φ is the specified machine flux (in the case of PMSM) and ωe is the electric frequency in rad/s.

Current and torque: The FPGA Electric Motor Controller model can be compared with SimPowerSystems reference models for example.

Short-circuit test: By setting the sinusoidal source equal to 0 or by setting Vdc link voltge to 0, one can compare the obtained short-circuit currents with theory. This short-circuit current only depends on the backEMF, and the Ld Lq inductances and the stator resistance.

Overflow verification: All curves should be free of discontinuities. If some sharp discontinuities appear, it probably means that some fixed-point overflow occurs in the FPGA model and that the model base have not been properly chosen. The inner FPGA calculation are implemented with a security factor of 4 with regards to the nominal working mode of the motor, which is supposed to be equal to the chosen bases for the model. This means that the model is designed to work correctly even if motor quantities are 4 times higher than the nominal (or base) values.

FIRST CONNECTION OF THE EXTERNAL CONTROLLER AND VERIFICATION OF SPEED READING.

The first step toward closing the loop is to connect the external controller and verify that it can properly read the motor speed.

This step is done in fixed-speed mode with sinusoidal source. The user needs to set the motor at its nominal speed and verify that the controller reads it correctly in positive and negative speed.

If the controller read speed is not correct but stable, you must check for sin-cos I/O signal reversal or a problem with the number of poles.

Verification of Sensor Null Angle

With the "*resolver and quad encoder elec angle offset pu (0-1)*" block of the model console, you can compare the sensors null angle with the back-EMF position and adjust it accordingly to the controller specifications.

Verification of Resolver Excitation Input

The resolver excitation on the analog input can be verified by monitoring the analog output where it is connected (for monitoring purposes). Internally, the Resolver excitation should be set at 1 pu of amplitude.

VERIFICATION OF THE SCALING OF THE MOTOR CURRENTS READ ON THE EXTERNAL CONTROLLER

An important step in the closing of a control loop is to verify the scaling of the quantities transmitted between the plant and the controller. The motor currents are such important quantities that it is imperative to verify their scaling.

The test can be made again with the sinusoidal or PWM mode. At the end of this step, the same currents should be readable inside the external controller and inside the model (in the console).

FPGA-based Analog-Output Scaling

It is important to consider that all FPGA motor values are in per-unit (pu) on the FPGA model. Additionally, the motor phase currents are outputted with the equivalent [-4/+4] on the Opal-RT Analog Output block, which expects a format of [-16/+16]. Consequently, there is a factor 4 implicit scaling, i.e. a 1 pu current corresponding to 4 volts on the Analog Output in the standard motor configuration.

VERIFICATION OF THE DQ AXIS REFERENTIAL OF THE CONTROLLER

The next step is to verify that the external controller has the proper dq-axis referential; it is still made in fixed-speed mode with sinusoidal source. The user must:

- Set the motor so it produces a positive torque using the "Motor Test Block" **Frequency** and **Angle** parameters.
- Verify, in its controller, that the Iq values read is stable and positive (most common Park-based controller definition). The motor torque and Iq values should also vary together.

VERIFYING THE CURRENT CONTROL

Now, close the current loop. The step is still done in Fixed-speed mode (like in a bench). You must:

- Set the external controller in current mode (or torque control mode).
- Enable the PWM by unselecting the Internal **Test Source Enabled** parameter of the Motor Test Block.

You should now be able to control the motor current with the user command in the external controller.

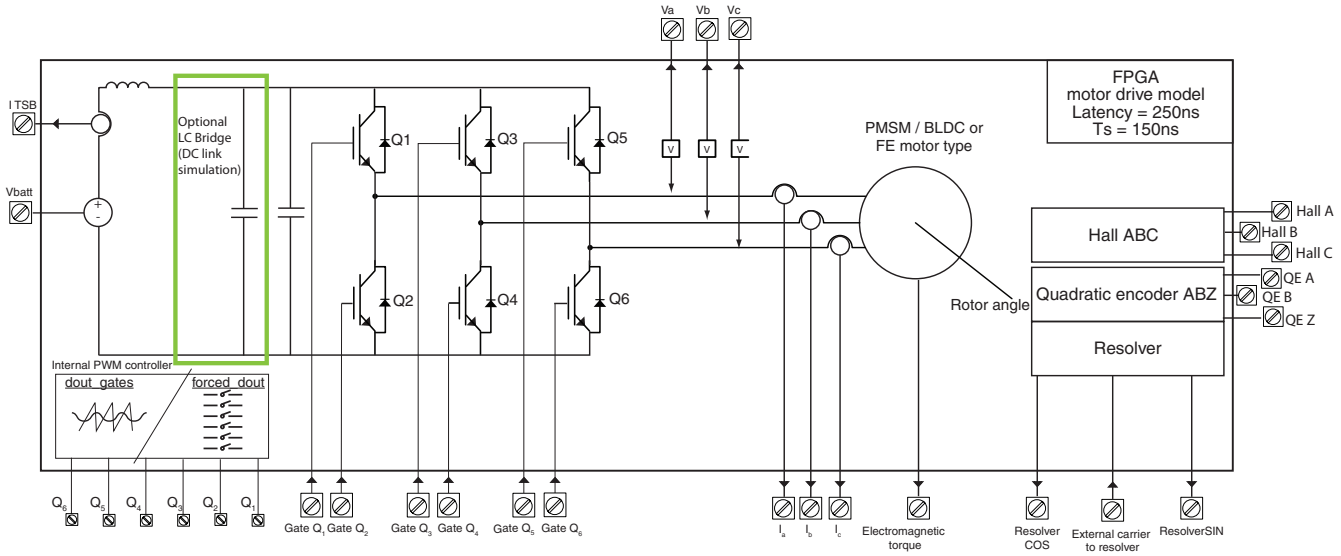
If problems occur, such as controller protection that triggers because of transients, you can try to open the motor stator phase (‘Open Stator’ of the Motor Test Block). This is like opening motor phase breakers. With the stator open, you should be able to verify that the controller commanded Iq value goes in the right direction. This commanded value, again located inside the external controller, will probably saturate because the stator is open but this enables you to verify that the closed loop has the correct polarity.

SPEED CONTROL

Speed control can be implemented after the current (i.e. torque control) has been verified by connecting the motor Electric Torque signal to the user mechanical model and feed the resulting motor speed back into the motor model.

PIN ASSIGNMENTS

Each input and output shown in the image is described in the Pin Assignments table.



Signal Type	Channel	Signal name	Description	Scaling
Digital Output	0	Test Q1	Gate firing for loopback testing	
	1	Test Q2	Gate firing for loopback testing	
	2	Test Q3	Gate firing for loopback testing	
	3	Test Q4	Gate firing for loopback testing	
	4	Test Q5	Gate firing for loopback testing	
	5	Test Q6	Gate firing for loopback testing	
		Hall ABC A	Hall effect sensor Phase A of rotor position	
		Hall ABC B	Hall effect sensor Phase B of rotor position	
		Hall ABC Z	Hall effect sensor Phase A of rotor position	
	6	A	Quadrature encoder A signal	
	7	B	Quadrature encoder B signal	
	8	Z	Quadrature encoder Z signal	
	9-31	Unused		

Signal Type	Channel	Signal name	Description	Scaling
Digital Input	00	Q1	Firing pulse applied to inverters	
	01	Q2	Firing pulse applied to inverters	
	02	Q3	Firing pulse applied to inverters	
	03	Q4	Firing pulse applied to inverters	
	04	Q5	Firing pulse applied to inverters	
	05	Q6	Firing pulse applied to inverters	
	06	Fault Trigger 1	External Trigger for Fault number 1	
	07	Fault Trigger 2	External Trigger for Fault number 2	
	08	Short Trigger 1	External trigger for AC/DC Short 1	
	09	Short Trigger 2	External trigger for AC/DC Short 2	
	10-31	Not used		
Analog Input	00	External resolver carrier	External resolver carrier	1v = 1pu*console_gain
	01	External_Vdc	Voltage of the voltage source	1v=1pu*console_gain
Analog Output	00	Ia	Current in motor phase A	pu multiplied by current gain from CPU
	01	Ib	Current in motor phase B	pu multiplied by current gain from CPU
	02	Ic	Current in motor phase C	pu multiplied by current gain from CPU
	03	Vplus		Between -1 and 1
	04	I igbt SUM low	Sum of Q2-4-6 igbts current	Between -1 and 1
	05	I igbt A low	Q2 Igbt's current	pu
	06	I igbt B low	Q4 Igbt's current	pu / between -1 and 1
	07	I igbt C low	Q6 Igbt's current	
	08	Va	Motor Phase A voltage	
	09	Vb	Motor Phase B voltage	
	10	Vc	Motor Phase C voltage	
	11	Vcap filter	Right-most voltage capacity	
	12	Vcap Cable	Voltage between two coils of the left-most capacity	
	13	I igbt A up	Current in Q1 IGBT	
	14	I igbt B up	Current in Q3 IGBT	
	15	I igbt C up	Current in Q5 IGBT	

ANNEX C - CPU FPGA INTERFACE SPECIFICATIONS

The objective of this document is to provide the various parameters for the FPGA-PMSM model running on the Opal-RT ML605 FPGA board. These parameters are enumerated at their connection with OpFcnML605EX1Send/Receive blocks, which interfaces with the FPGA-PMSM model.

WARNING	The user should never tamper with the OpFcnML605EX1Send/Receive data used by the JMAG/PMSM/BLDC FPGA model. If the user wants to transmit/receive other data to/from the FPGA they should add additional channels OpFcnML605EX1Send/Receive blocks as well as in the FPGA model (DataIn and DataOUT blocks)

LIST OF OPFCNML605EX1SEND/RECEIVE INPUTS

Indice	Source block	Description	Format
1	PackCPUData/data_ram_f	Inductance data	internal
2	PackCPUData/en_write_ram_f	Inductance data	internal
3	PackCPUData/wm_pu	Mecanical motor speed in PU	F18.15 (see note 1)
4	PackCPUData/Params	General test parameter	Time multiplexed data (see Table 2)
5	PackCPUData/pm_params	Machine parameters	Time multiplexed data (see Table 2)
6	PackCPUData/ram_EMF_f	BackEMf data	Internal (see note 2)

Table 2: OpFcnML605EX1Send/Receive input data interface description for the PMSM-FPGA model

Data Index	Name	Format	
General test parameter			NOTE: TDM using 12 bit index block
100	PWM max count	U20.0	
101	Deadtime count	U12.0	
102	Carrier scaling	U17.9	
103	Faults and setting	U20.0	Bit0: phase A to gnd fault Bit1: phase B to gnd fault Bit2: phase C to gnd fault Bit3: internal_source_selected Bit4: nodal_inverter_selected Bit5: set Fixed rotor angle Bit6: set Fixed Vstator Bit7:reserved Bit8: reserved Bit9: reserved Bit10: set No_Pulse_on_IGBT Bit11: reserved Bit12: reserved Bit13: reserved Bit 14: set external_resolver_carrier_select Bit 15: set_Open_stator Bit: 16-17: reservedBit 18: 2nd I/O config selected
104	Vstator angle (pu)	U18.18	This is the angle between the back-EMF and stator voltage
105	Modulation index in PWM modulation with sinusoidal source	F18.15	
106	DC voltage amplitude	F18.15	
107	Test source selection	U2.0	=1: sinusoidal stator voltage =2 PWM voltage =3 PWM voltage with I/O loopback
108	Inverse of pair of pole number	F20.20	
110	Analog output motor Current Gain	F18.15	
111	Analog output Idc current gain	F18.15	
112	Analog output Resolver gain	F18.15	
113	Analog output batter current gain	F18.15	
119	Reserved	U18.0	
125	Speed integration factor	F18.15	Contains a 2^18 scaling factor
130	Nodal inverter #1 faults (part 1) + AC and DC faults	U20.0	Bit 0: DC short Bit 1 : AC short Bit-2-7: IGBT OFF Bit 8-13 Diode Off Bit 14-19: IGBT & Diode OFF
131	Nodal inverter #1 faults (part 2)+ open_phase faults + trigger	U6.0	Bit 0-5: IGBT & Diode ON Bit 6-8: Open phase A,B,C Bit 9 : disable all faults for both inverters Bit 10: Enable all fault for both inverter Bit 11: I/O Trigger fault for inverter #1 Bit 12: I/O Trigger fault for inverter #2

Data Index	Name	Format	
132	Nodal inverter #2 faults (part 1)	U20.0	Bit 0: reserved Bit 1 : reserved Bit-2-7: IGBT OFF Bit 8-13 Diode Off Bit 14-19: IGBT & Diode OFF
133	Nodal inverter #2 faults (part 2)	U6.0	Bit 0-5: IGBT & Diode ON
NOTE 130-133			Bit order with respect to phases is: Bit0 Upper IGBT/Diode phase A Bit1 Lower IGBT/Diode phase A Bit2 Upper IGBT/Diode phase B Bit3 Lower IGBT/Diode phase B Bit4 Upper IGBT/Diode phase C Bit5 Lower IGBT/Diode phase C
126	Sensor_offset angle (electric angle)	U18.18	Used to offset resolver
127	Sensor mechanical angle offset	U18.18	Used to offset the Quad-Encoder
PM parameters			NOTE: TDM using 6 bit index block
1	Stator resistance phase A (pu)	F26.18	
2	Stator resistance phase B (pu)	F26.18	
3	Stator resistance phase C (pu)	F26.18	
4,5,6	reserved		
7,8	Integration factors	F26.0	This is the flux integration factor which is put on two F26 numbers w_base*5e-9*2 (integration in PU made at 10ns)
9	High impedance value (pu) ex: open phase cases	F26.18	Nominal 24 PU but can be increased sometimes

Table 3: Table 2: Multiplexed data specification

Port#	Destination block	Description	format
1	ExtractFPGAData/dout_1	Motor current (phase a and b) in PU	[-4/+4] (x2)
2	ExtractFPGAData/dout_2	Motor current (phase c) in PU and electric angle in pu [0-1]	[-4/+4] [0/1]
3	ExtractFPGAData/dout_3	DC-link current in PU Not used	[-8/+8] --
4	ExtractFPGAData/dout_4	Motor input voltage (phase a b) in PU electric angle	[-4/+4] [-4/+4]
5	ExtractFPGAData/dout_5	Motor input voltage (phase c) in PU	[-4/+4]
6	ExtractFPGAData/dout_6	reserved Mechanical angle (pu)	[0/1] [0/1]
7	ExtractFPGAData/dout_7	Hall sensors	binary
8	ExtractFPGAData/dout_8	Motor backEMF voltage (phase a b) in PU electric angle	[-4/+4] [-4/+4]
9	ExtractFPGAData/dout_9	Motor backEMF voltage (phase c) in PU	[-4/+4]
10	ExtractFPGAData/dout_10	Digital input reading	binary
11	ExtractFPGAData/dout_12	24V and 12V readings from A/D converters	[-32/+32] [-32/+32]
12	ExtractFPGAData/dout_13	GND and 12V readings from A/D converters	[-32/+32] [-32/+32]

Table 4: OpFcnML605EX1Send/Receive output data interface description for the PMSM-FPGA model

	Each 32 bit output channel is divided into two 16 bit channels

Note 1: XSG fixed-point format specification.

TX.Y number format refers to the Xilinx notation when T={U:unsigned or F:signed}, X is the total number of bits of the number and Y is the location of the binary point. Ex: F18.15 refers to a number expressed in signed format with 18 bits of length with binary point in the 15th place. This format can therefore represent numbers between ±4 with a resolution of 2^-15

FPGA MODEL SIGNAL SPECIFICATIONS

The following signals are located in the 'PMSM_Drive' block in the FPGA model.

'extract_input' block signals		
we_pu	Electrical speed in PU	F18.15
ExtCarSel	Resolver excitation selection (if=1, then the resolver excitation comes from A/D)	binary
w0_Vabc	If=1, Stator test voltage source frequency becomes 0	binary
w0	If=1, rotor speed is set to 0	binary
VOffset	Offset of stator test source angle (12 bit table offset)	U12.0
Sine Gain pu	Gain of stator test voltage source	F18.15
Deadtime	Dead-time of internal PWM test source (not used)	
Max count	Count number for the internal PWM source triangular carrier (not used)	
Car Scaling	Scaling of the triangular carrier to obtain 0-1 amplitude (not used)	
V plus	DC bus voltage in Pu	F18.15
faults	Inverters outputs faults Bit0: phase neutral fault Bit1: A-B (phase-phase) fault	U2.0
noIGBTpulse	Blocks all IGBT pulse from the PWM test source only. (not used)	
ABC Sin fault	Put sinusoidal test source to 0 (phase A B and/or C)	U3.0
JMAG_type	obsolete	
BLDC_type	obsolete	
I_over_P	Number of pair of pole (max 15 pair of poles)	U4.0
DA_gains	Analog outputs gains (Currents, TSB, resolver, battery)	F18.15 (4 times)
Fail_sig	Status bits	U3.0
Tempsig	Temperature status	U16.0 (4 times)
Gates_CPU	Overrides test gate signals (not used)	U6.0
sel	Overrides test gate signals selection bit (not used)	binary
TSB&short-circuit fault block		
Vabc	3-ph Sinusoidal test source (PU)	F18.15
Rmach_HighImp	Put machine stator in high-impedance model if=1	U3.0
Ip	DC-link current (PU)	F18.14
Permanent-Magnet Machine		
Iabc	Machine currents (PU)	F18.15
Vind_ABC	BackEMF induced voltage (pu)	F18.15
V_neutral	Stator neutral point voltage (pu)	F18.15
IambdaABC		

PWM_and_resolvers		
thetaXSG	Effective rotor angle(pu)	U18.18
sin and cos	Resolver sin-cos signal	F18.16
HallABC	Hall sensor outputs	U3.0
abz	Quandrature encoder output	U3.0
car	Resolver carrier	F18.16
sinRES	Resolver sin-modulated output	F18.16
cosRES	Resolver cos-modulated output	F18.16
gates_gen	PWM test source IGBT gate signals (not used)	
sinABC	3-ph sinusoidal test source (pu)	F18.15
we_pu_XSG	Rotor speed in pu	F18.15

The following signals are located in the ‘JMAG Inductance Table’ block in the FPGA model.

JMAG Inductance Table ML605		
DDR1 block signals		
L1, L2, L3 L4	Inverse of inductance matrix in pu (this matrix is 2x2)	F32.25
dL1 dL2 dL3 dL4	Derivative of inductance matrix in pu (this matrix is 2x2)	F32.25
Note: 2x2 matrix format is		
[L1 L2]		
[L3 L4]		

LIMITED WARRANTY

Limited Warranty

Opal-RT Technologies Inc. warrants to the original purchaser and/or ultimate customer (“Purchaser”) of Opal-RT products (“Product”) that if any part thereof proves to be defective in material or workmanship within one (1) year, such defective part will be repaired or replaced, free of charge, at Opal-RT Technologies’ discretion, if shipped prepaid to Opal-RT Technologies Inc. at 1751 Richardson, suite 2525, Montreal, Quebec, Canada, H3K 3G6, in a package equal to or in the original container. The Product will be returned freight prepaid and repaired or replaced if it is determined by Opal-RT Technologies Inc. that the part failed due to defective materials or workmanship. Otherwise, the fees will be charged to the client (see article “Warranty Limitation and Exclusion”). The repair or replacement of any such defective part shall be Opal-RT Technologies’ sole and exclusive responsibility and liability under this limited warranty.

Purchaser must request an RMA number before shipping any Product for repair:

1. Access the Opal-RT website (www.opal-rt.com/support/ return-merchandise-authorization-rma-request), click on support and select Return Merchandise (RMA).
2. Fill out the online form and submit.
3. Opal-RT’s Support department will evalute the return and either issue an RMA number via email
   - If the Product is returned for repair more than 12 months after purchase, the Purchaser is responsible for the cost of repair. Opal-RT will assess the repair and prepare a quote. The RMA number will be sent with the quote.
4. Only when the Purchaser receives the RMA number, may they ship the Product, prepaid, to Opal-RT.

RETURN POLICY

The following fees will apply when customers return products for credit:

A full credit, less a 15% fee and less return fee will only be issued if the product is in perfect working condition and if the product is returned within 1 month following the shipping date. If repairs are required on the returned product, the cost of these repairs will be deducted from the credit to be issued.

No credits will be issued beyond the one month period.

EXCLUSIONS

If third party products are part of the Product, Opal-RT will honor the original manufacturer’s warranty.

This limited warranty does not cover consumable items, such as batteries, or items subject to wear or periodic replacement, including lamps, fuses or filter elements.

Warranty Limitation and Exclusion

Opal-RT Technologies will have no further obligation under this limited warranty. All warranty obligations of Opal-RT Technologies are void if the Product has been subject to abuse, misuse, negligence, or accident or if the Purchaser fails to perform any of the duties set forth in this limited warranty or if the Product has not been operated in accordance with instructions, or if the Product serial number has been removed or altered.

Disclaimer of Unstated Warranties

The warranty printed above is the only warranty applicable to this purchase. All other warranties, express or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose are hereby disclaimed.

Limitation of Liability

It is understood and agreed that Opal-RT Technologies' liability, whether in contract, in tort, under any warranty, in negligence or otherwise shall not exceed the amount of the purchase price paid by the purchaser for the product and under no circumstances shall Opal-RT Technologies be liable for special, indirect, or consequential damages. The price stated for the product is a consideration limiting Opal-RT Technologies' liability. No action, regardless of form, arising out of the transactions under this warranty may be brought by the purchaser more than one year after the cause of actions has occurred.

CONTACT

Opal-RT Corporate Headquarters

1751 Richardson, Suite 2525
Montréal, Québec, Canada
H3K 1G6
Tel.: 514-935-2323
Toll free: 1-877-935-2323

Technical Services
www.opal-rt.com/support

Note:

While every effort has been made to ensure accuracy in this publication, no responsibility can be accepted for errors or omissions. Data may change, as well as legislation, and you are strongly advised to obtain copies of the most recently issued regulations, standards, and guidelines.
This publication is not intended to form the basis of a contract.

UG12-04016-OP1
01/2012