3.8 Configuring the Local Controller

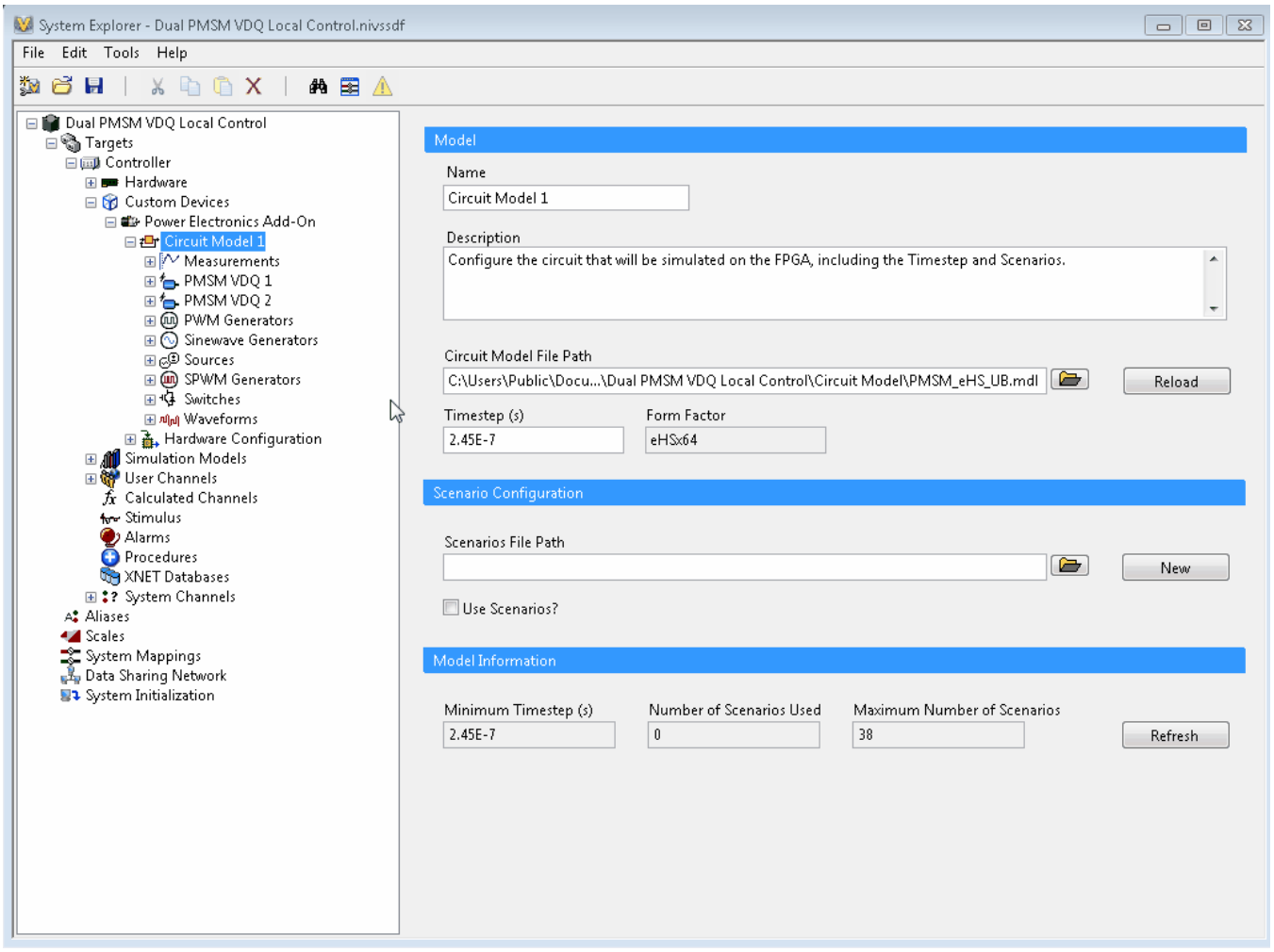
A simple closed-loop SPWM frequency and modulation index controller model was developed for this example. It has been compiled into *.dll* and *.so* files for execution on both Phar Lap and Linux RT targets, respectively. The controller model is configured to run on the Real Time CPU and will be executed by the **VeriStand Engine**. Data is routed from the CPU simulation (local controller) to the FPGA simulation (eHS, SCIM model) using the VeriStand System Configuration Mappings.

Local Controller File Path: *<Public Documents>\National Instruments\<NI VeriStand 20xx>\Examples\OPAL-RT\Power Electronics Add-On\Dual PMSM VDQ Local Control\Local Controller*

Exploring the Local Controller

- In the Configuration Tree, expand **Controller >> Simulation Models >> Models**. Click **Volts Per Hertz Controller** to view the model file information.
- By default, the *.so* file is loaded for execution on a **Linux RT** target. To run the example on a **Phar Lap** target, click the **Browse Simulation Model** button in the Menu bar and select *VoltPerHertzController.dll* from the **Local Controller File Path** listed above.
- In the Configuration Tree, expand **Volt Per Hertz Controller >> Inports** and confirm that all inports have been configured as shown in the table below.
- Click **Save.**

Inport	Default Value
Dclink	0
DesiredSpeedRPM	0
ModIndexLookup	20 110 100 298 200 525 500 25 800 830 1000 830 1500 830 2000 830 2500 830 3000 830
Poles	3
RampRate	10

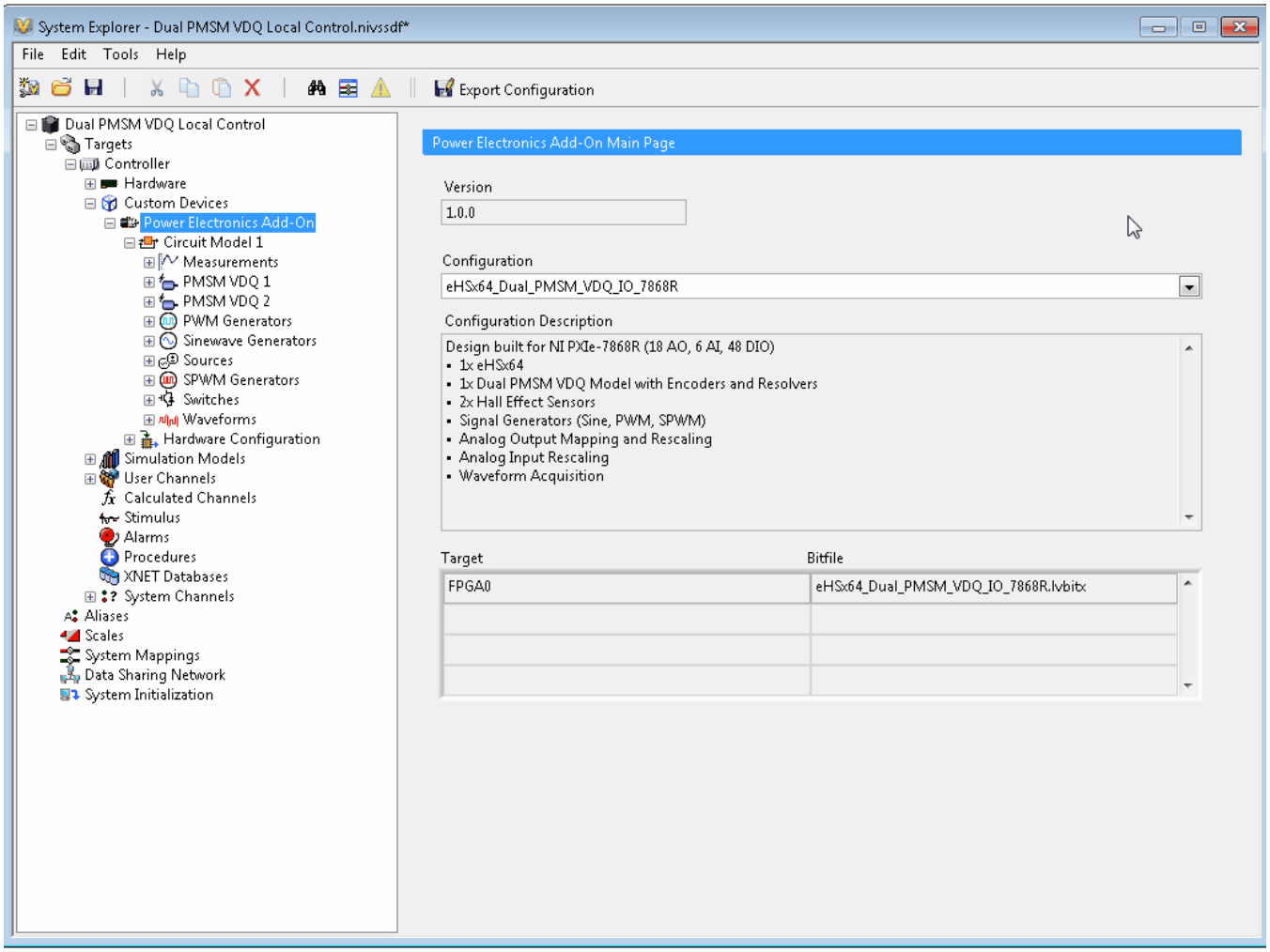


Exploring the Mappings to and from the Local Controller

The VeriStand System Configuration Mappings are used to route signals between the local controller and the other model components simulated on the FPGA.

- In the VeriStand System Explorer window, navigate to **Tools >> Edit Mappings** in the Menu bar.
- Confirm that the mappings are configured as shown in the table below.

Sources	Destinations
*Controller/User Channels/**Supply Voltage***	*Controller/Custom Device/OPAL-RT Power Electronics Module/Circuit Model/Signal Generators /Sinewave Generators/SWG Generator 0/**Amplitude***
*Controller/User Channels/**Supply Voltage***	*Controller/Custom Device/OPAL-RT Power Electronics Module/Circuit Model/Signal Generators /Sinewave Generators/SWG Generator 1/**Amplitude***
*Controller/User Channels/**Supply Voltage***	*Controller/Custom Device/OPAL-RT Power Electronics Module/Circuit Model/Signal Generators /Sinewave Generators/SWG Generator 2/**Amplitude***
*Controller/Simulation Models/Models /Speed_Control/Outports/**InverterFrequencyRef***	*Controller/Custom Device/OPAL-RT Power Electronics Module/Circuit Model/Signal Generators /SPWM Generators/SPWM Frequency Generators/**SPWM Frequency Engine 0***
*Controller/Simulation Models/Models /Speed_Control/Outports/**ModIndexRef***	*Controller/Custom Device/OPAL-RT Power Electronics Module/Circuit Model/Signal Generators /SPWM Generators/SPWM Carrier/**Modulation Index***


System Explorer - Dual PMSM VDQ Local Control.nivssdf*
File
Edit
Tools
Help
Export Configuration
Dual PMSM VDQ Local Control
Targets
Controller
Hardware
Custom Devices
Power Electronics Add-On
Circuit Model 1
Measurements
PMSM VDQ 1
PMSM VDQ 2
PWM Generators
Sinewave Generators
Sources
SPWM Generators
Switches
Waveforms
Hardware Configuration
Simulation Models
User Channels
Calculated Channels
Stimulus
Alarms
Procedures
XNET Databases
System Channels
Aliases
Scales
System Mappings
Data Sharing Network
System Initialization
Power Electronics Add-On Main Page
Version
1.0.0
Configuration
eHSx64_Dual_PMSM_VDQ_IO_7868R
Configuration Description
Design built for NI PXIe-7868R (18 AO, 6 AI, 48 DIO)
• 1x eHSx64
• 1x Dual PMSM VDQ Model with Encoders and Resolvers
• 2x Hall Effect Sensors
• Signal Generators (Sine, PWM, SPWM)
• Analog Output Mapping and Rescaling
• Analog Input Rescaling
• Waveform Acquisition
Target
Bitfile
FPGA0
eHSx64_Dual_PMSM_VDQ_IO_7868R.lvbitx