

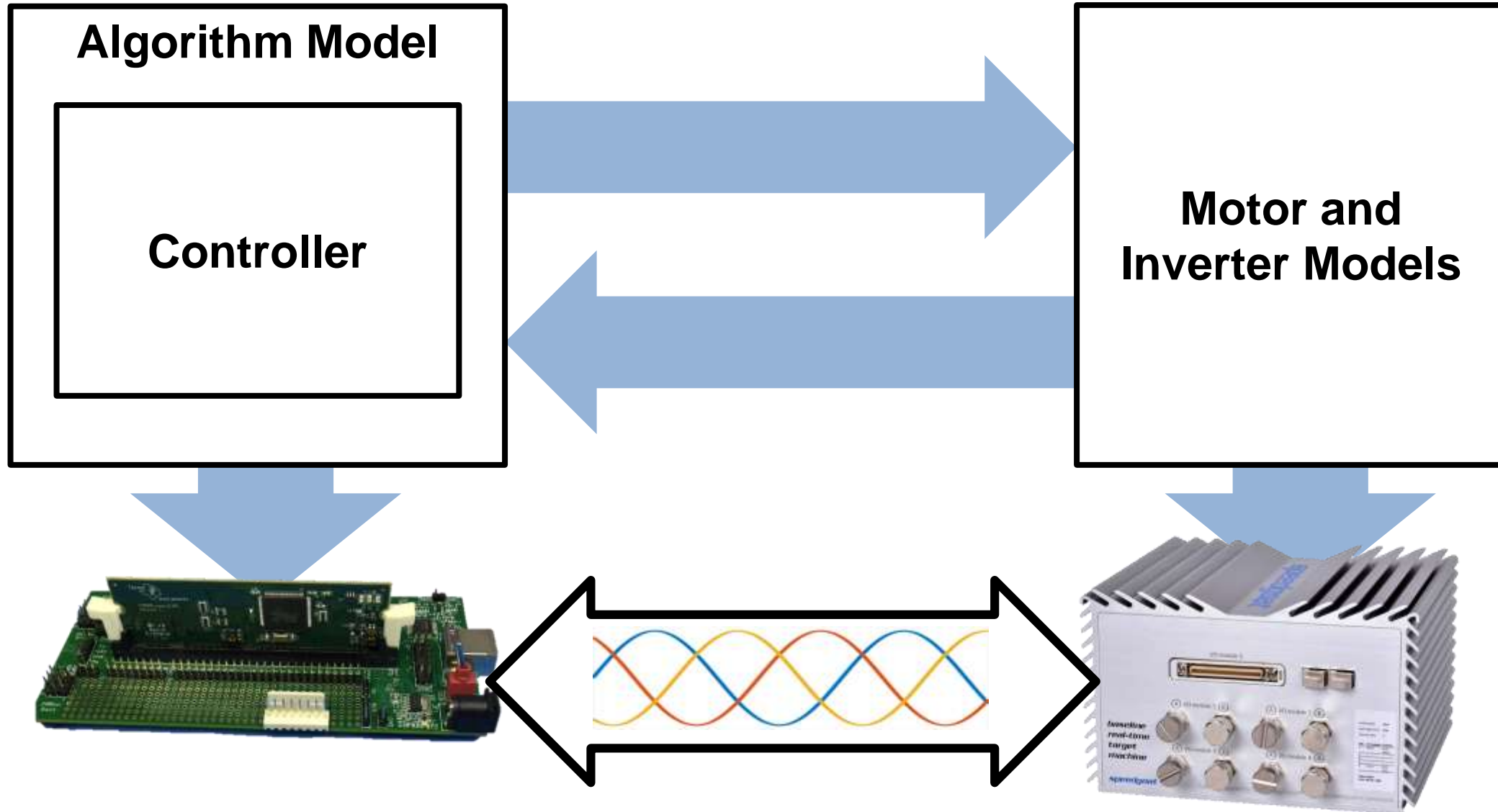
Accurate Simulation of EV/HEV Power Electronics Switching Events for HIL Testing

Joel Van Sickel
Application Engineer

Overview

1. Hardware-in-the-loop (HIL) overview
2. Review of a system level model of a motor and inverter in Simulink
3. Use of HDL Coder to generate floating-point HDL from the Simulink model to achieve 1 MHz simulation (1 μ s time-step)
4. HIL simulation using Simulink Real-Time and Speedgoat target hardware

What is HIL

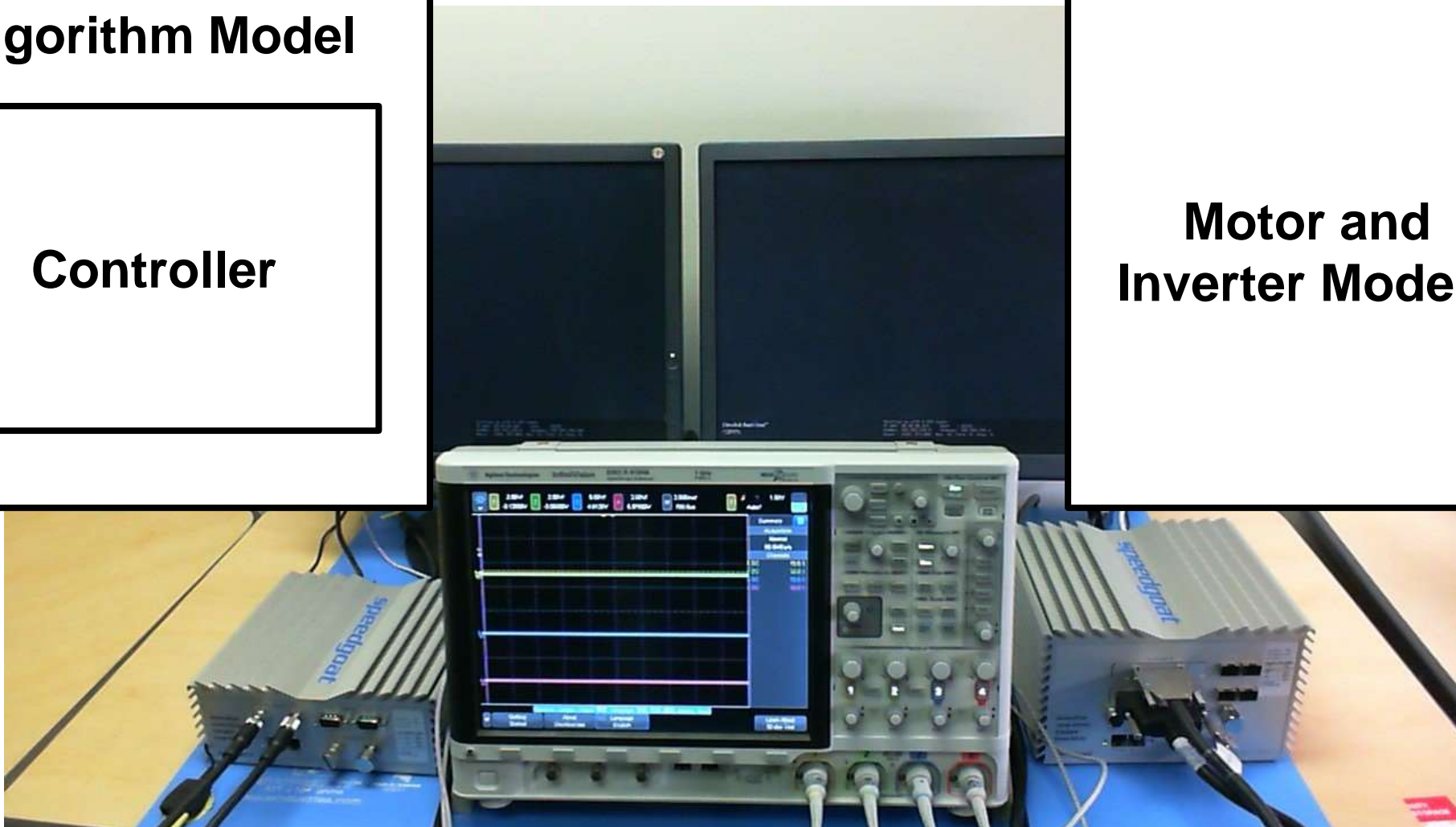


Demo

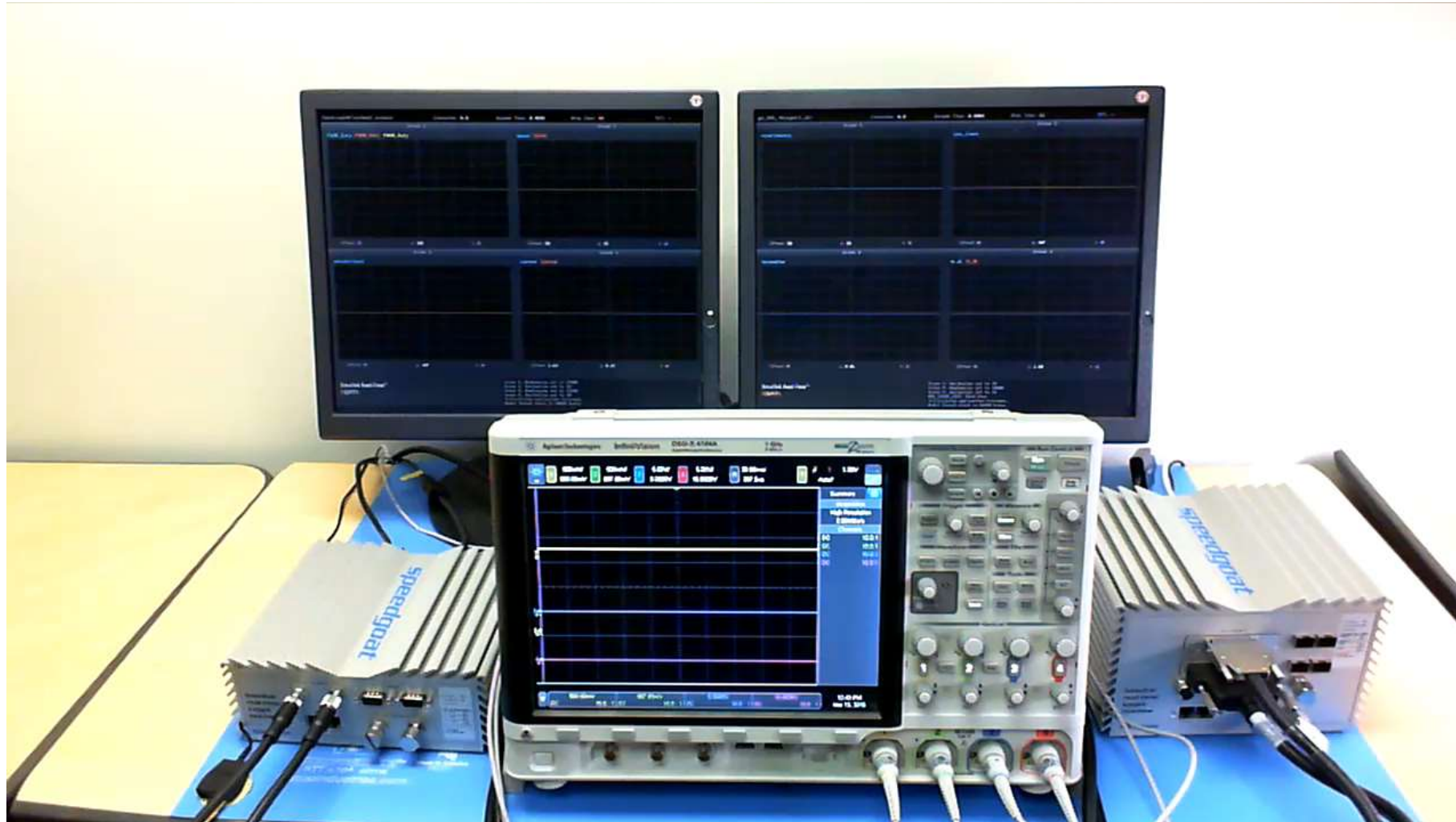
Algorithm Model

Controller

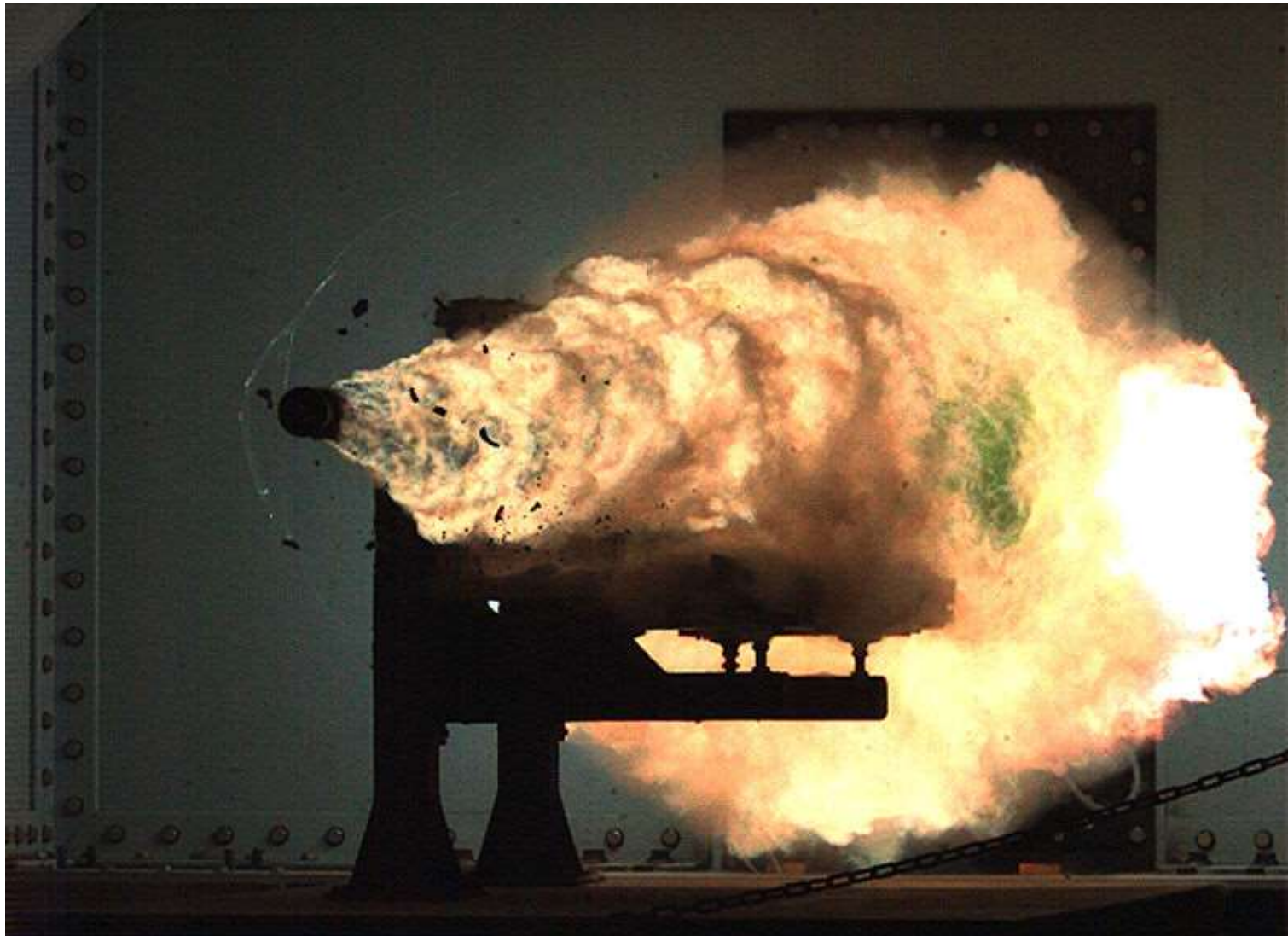
**Motor and
Inverter Models**



Demo



Why HIL?

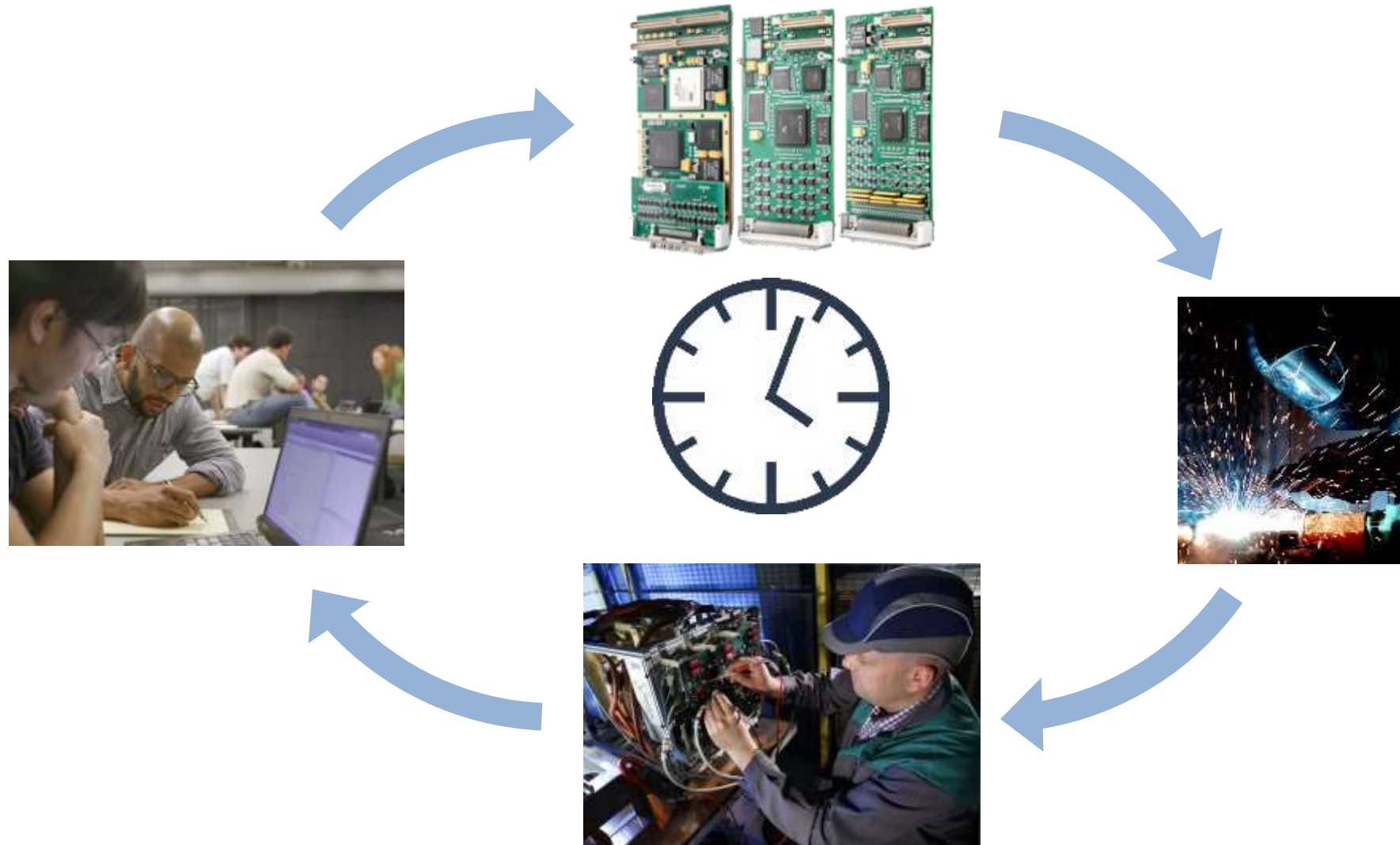


Why HIL?



By Marshelec - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=16585159>

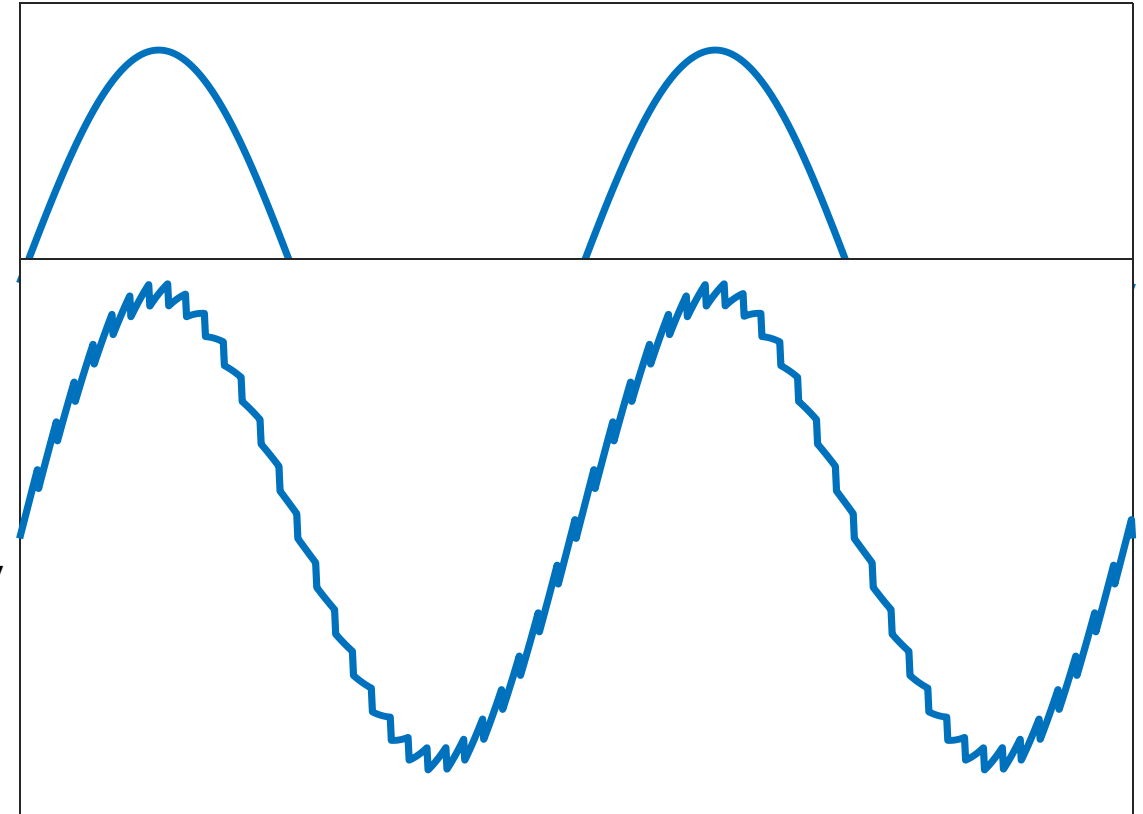
Why HIL?



Power Electronics and Motor Control - Switching

2 ways to simulate power electronics

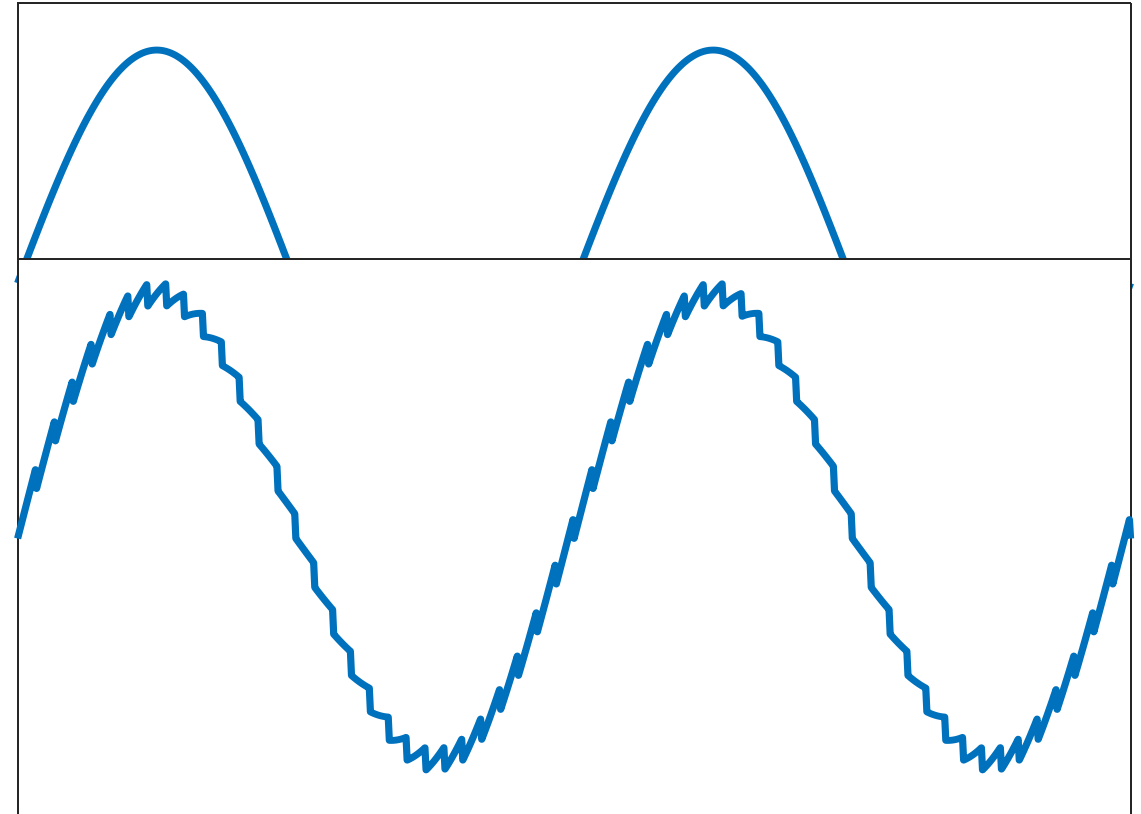
- **Average**
 - Easy to implement in real time
 - Ignores dynamics of switching devices
 - Good enough for some types of analysis
- **Switching**
 - Captures switching events
 - Model faults and analyze switching efficiency
 - Requires simulation 100 times faster than switching frequency
 - Motor at 10 kHz requires 1 MHz simulation



CPU vs FPGA Simulations

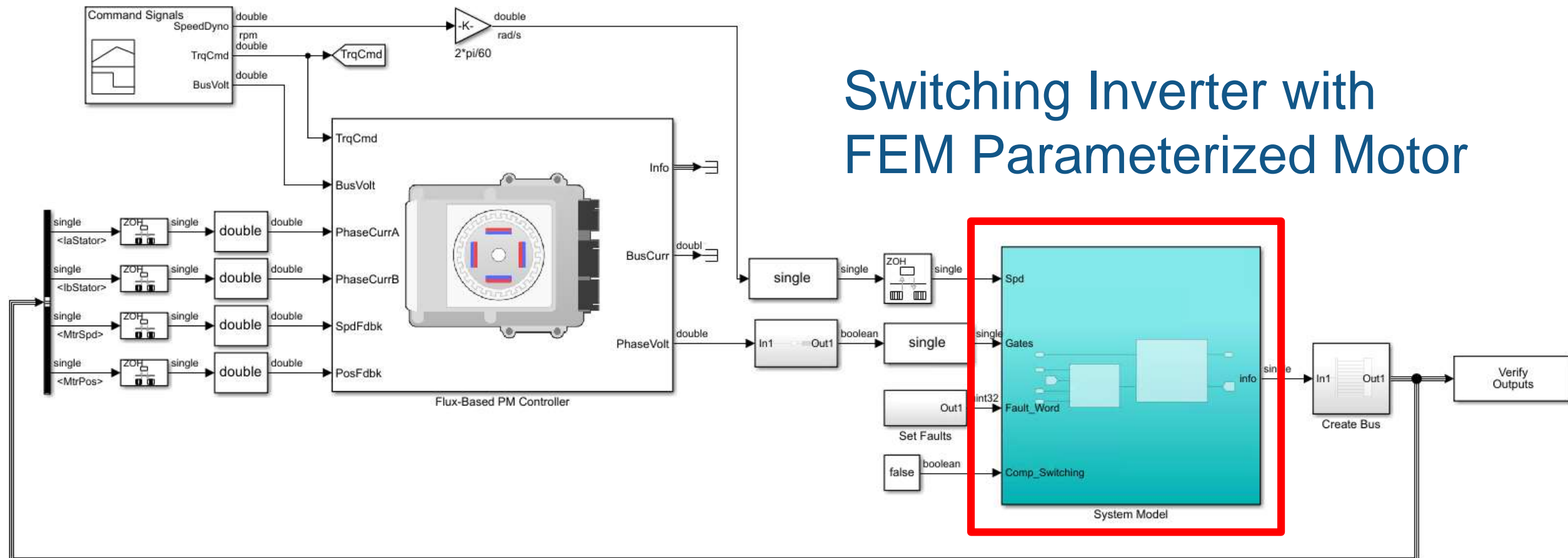
2 ways to simulate power electronics

- CPU
 - Cheaper hardware
 - Can run continuous domain simulation
- FPGA
 - Can run any code gen compatible block
 - Multiple orders of magnitude faster
 - Requires discrete domain simulation
 - Uses single floating point values
 - Simulation
 - e



Review of a System Level Model of a Motor and Inverter in Simulink

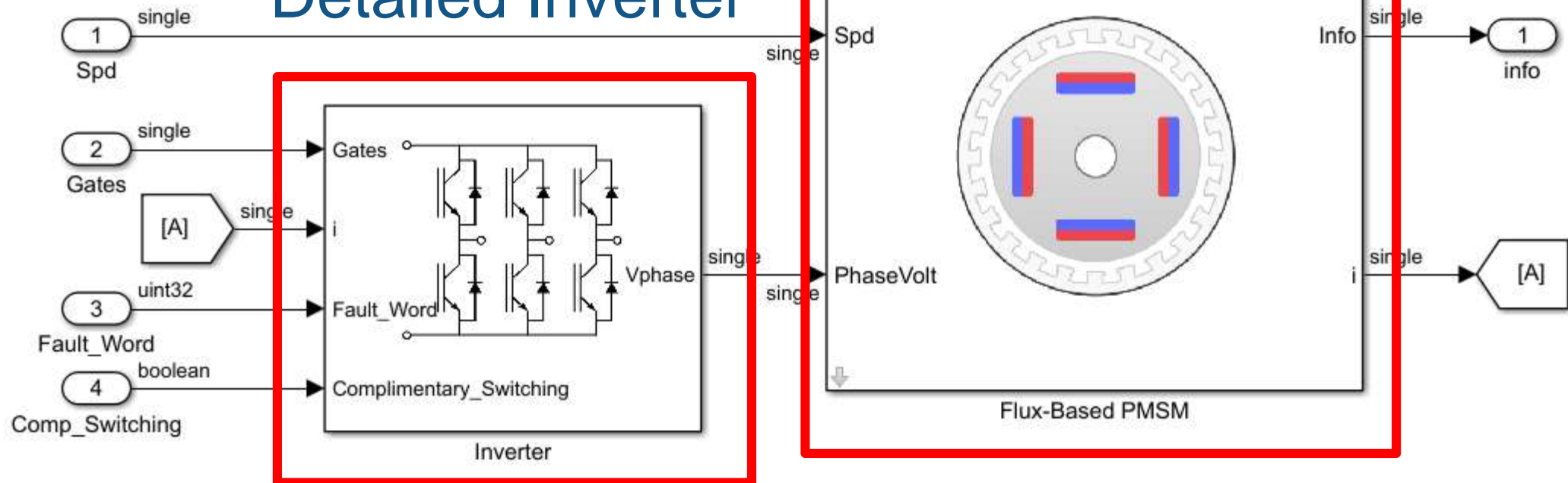
Switching Inverter with FEM Parameterized Motor



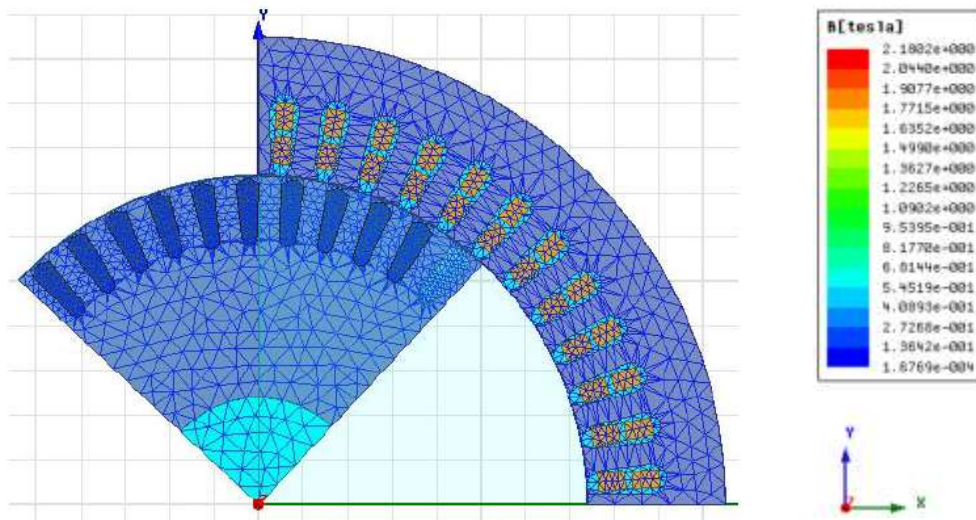
Review of a System Level Model of a Motor and Inverter in Simulink

FEM Based Motor Model

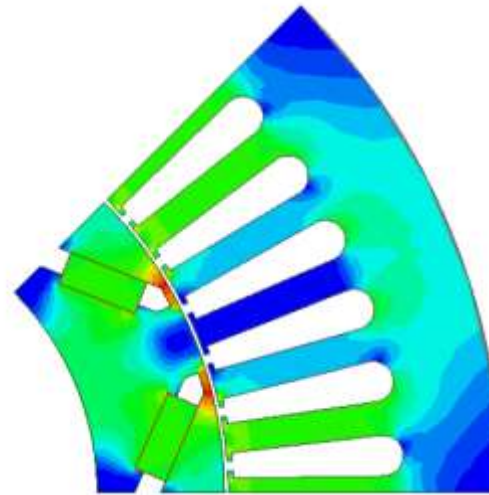
Detailed Inverter



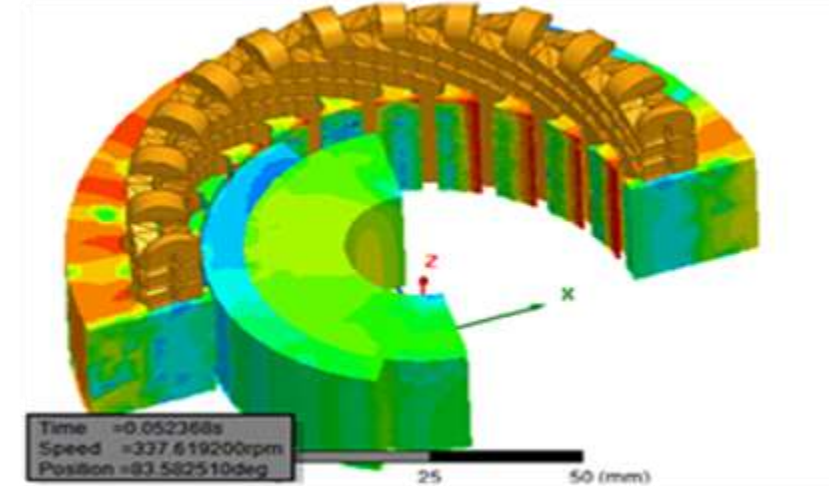
Can Extract Lookup Table from FEM Tool



FEA Meshing for Induction Machine

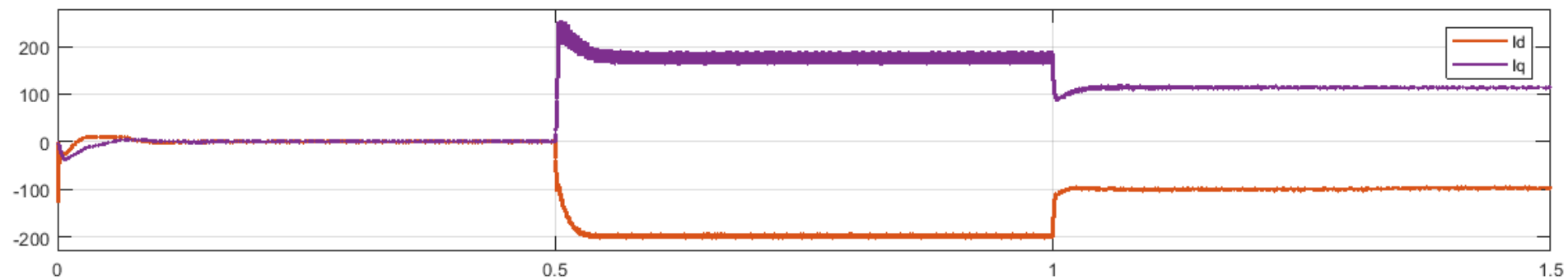
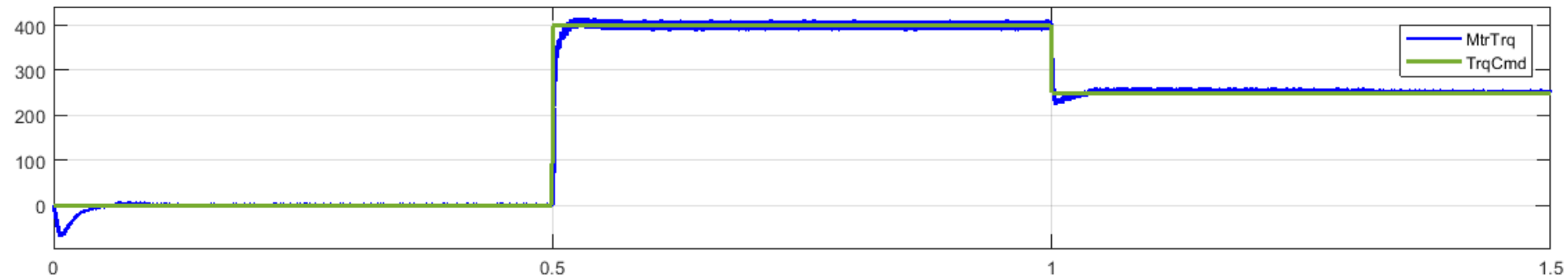


2D FEA PMSM Model



3D FEA PMSM Model

Simulation Results

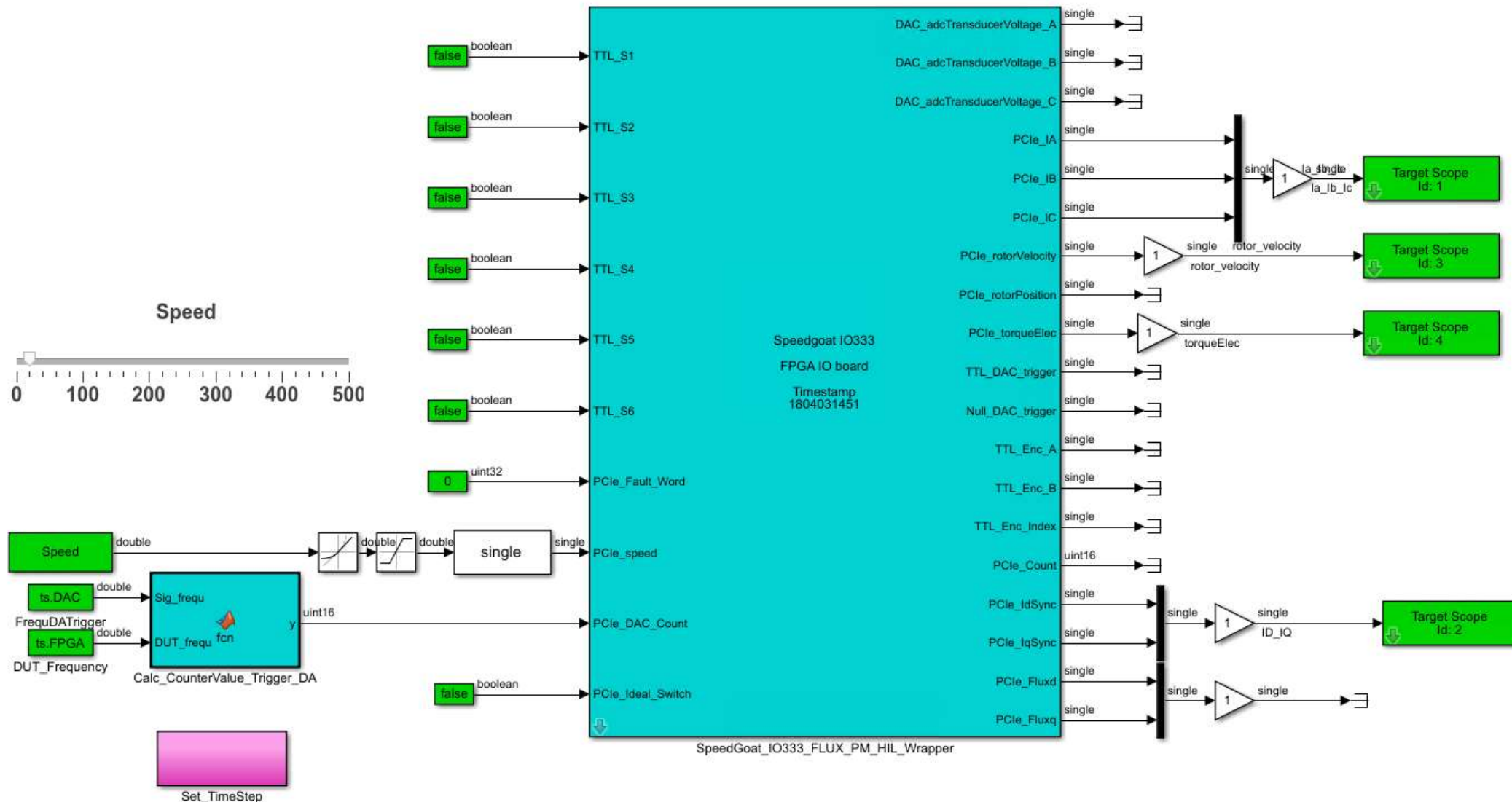


Time (seconds)

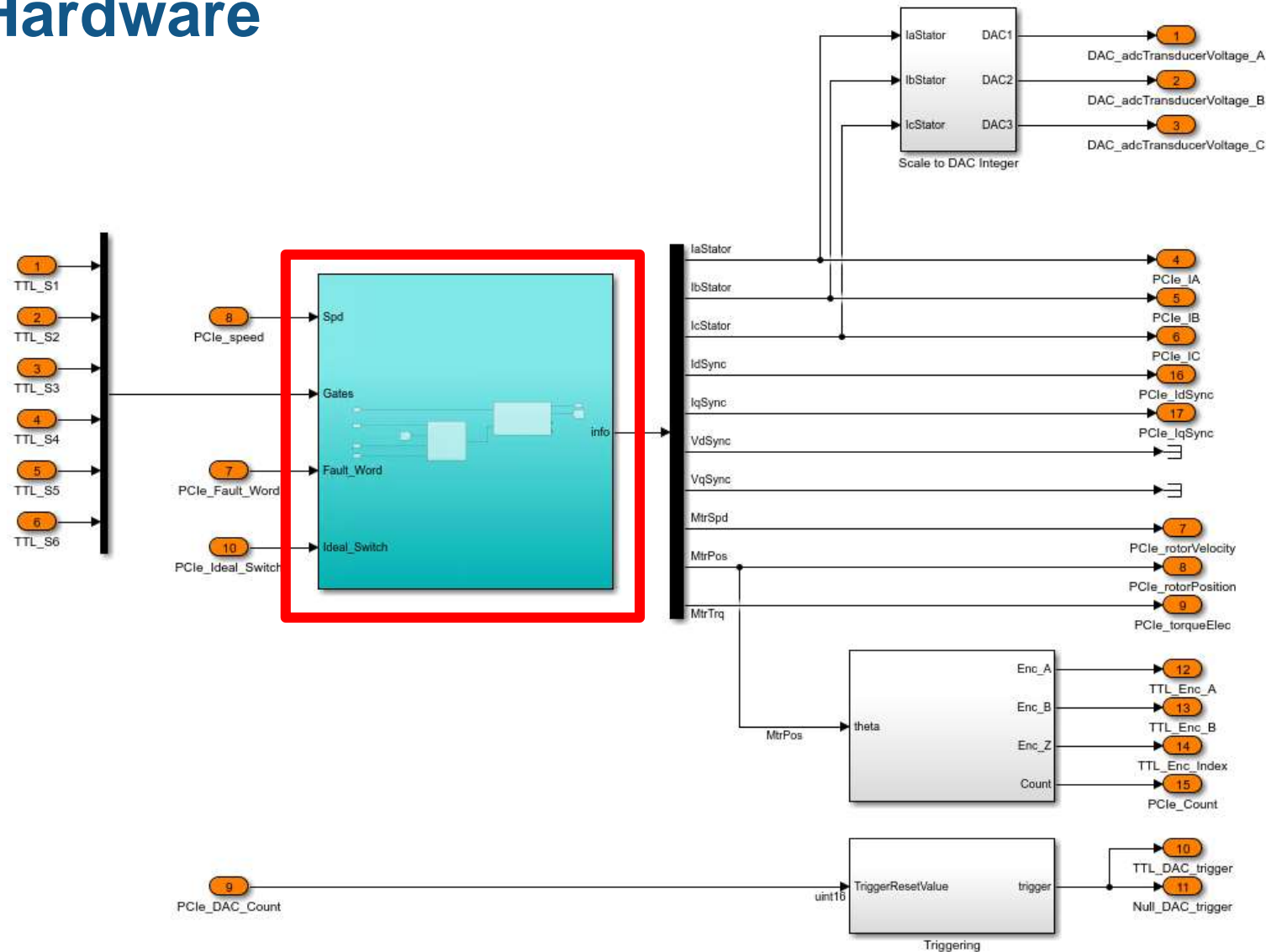
High Level Process for Deploying Model to FPGA

1. Create high level subsystem for defining I/O
2. Convert model to discrete time
3. Convert to single precision
4. Use HDL workflow advisor to setup model settings
5. Deploy model to the Speedgoat real-time machine

HIL Simulation Using Simulink Real-Time and Speedgoat Target Hardware



HIL Simulation Using Simulink Real-Time and Speedgoat Target Hardware



Use of HDL Coder to Generate Floating-Point HDL From the Simulink Model to Achieve 1 MHz Time-Steps

Example walkthrough available through webinar:

<https://www.mathworks.com/videos/hardware-in-the-loop-hil-testing-for-power-electronics-systems-modeled-in-simulink-1522417541924.html>

The screenshot displays the HDL Workflow Advisor interface for the project 'SpeedGoat_attempt/HDL_GM_2D_Model'. The left pane shows a tree view of the workflow steps, with '5.1. Generate Simulink Real-Time interface' selected and marked with a green checkmark. The right pane shows the details for this task, including a 'Run This Task' button, a 'Result: Passed' status, and a message: 'Passed Generate Simulink Real-Time Interface. Generating new Simulink Real-Time Interface model: gm SpeedGoat_attempt slrt. Simulink Real-Time Interface model generation complete.'

HDL Workflow Advisor - SpeedGoat_attempt/HDL_GM_2D_Model

File Edit Run Help

Find: [] [] []

HDL Workflow Advisor

- 1. Set Target
 - ^1.1. Set Target Device and Synthesis Tool
 - ^1.2. Set Target Reference Design
 - ^1.3. Set Target Interface
 - 1.4. Set Target Frequency
- 2. Prepare Model For HDL Code Generation
 - 2.1. Check Global Settings
 - ^2.2. Check Algebraic Loops
 - ^2.3. Check Block Compatibility
 - ^2.4. Check Sample Times
- 3. HDL Code Generation
 - > 3.1. Set Code Generation Options
 - ^3.2. Generate RTL Code and IP Core
- 4. Embedded System Integration
 - 4.1. Create Project
 - 4.2. Build FPGA Bitstream
- 5. Download to Target
 - 5.1. Generate Simulink Real-Time interface

5.1. Generate Simulink Real-Time interface

Analysis

Generate Simulink Real-Time interface

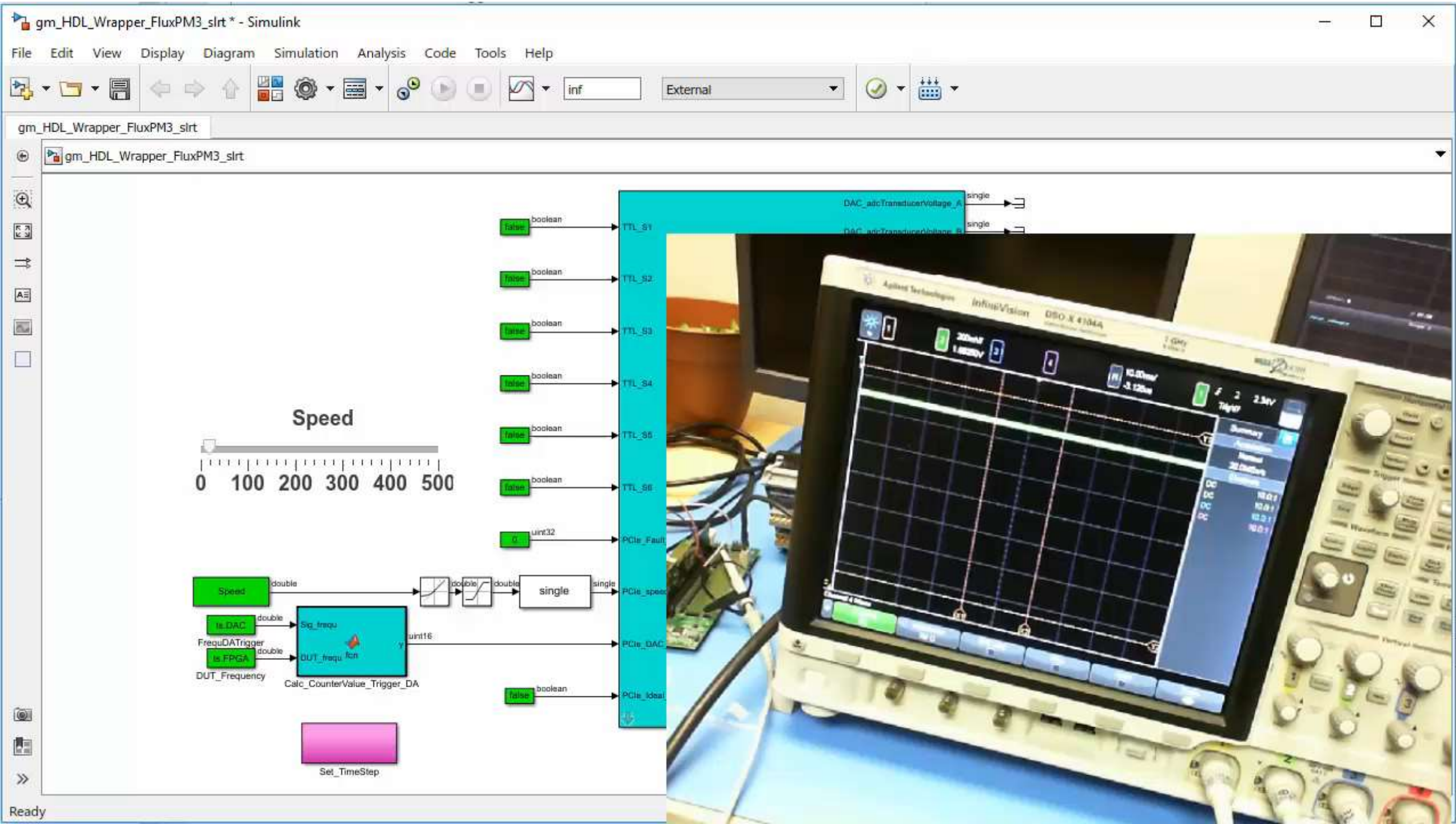
Run This Task

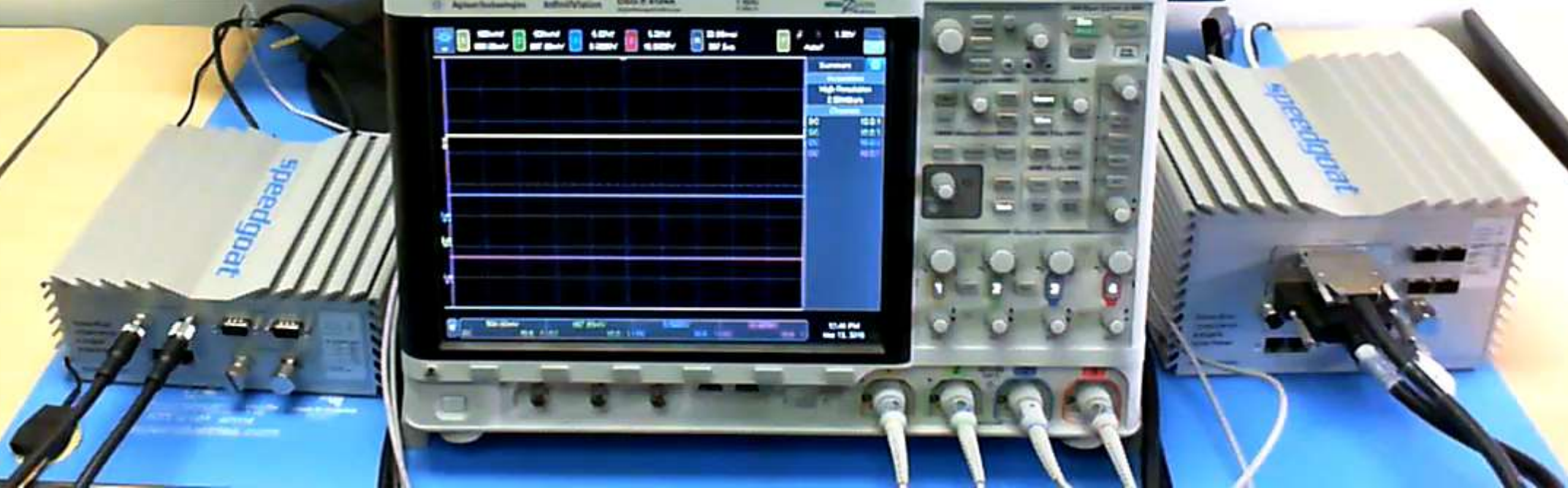
Result: ✔ Passed

Passed Generate Simulink Real-Time Interface.

Generating new Simulink Real-Time Interface model: [gm SpeedGoat_attempt slrt](#)

Simulink Real-Time Interface model generation complete.





SDI

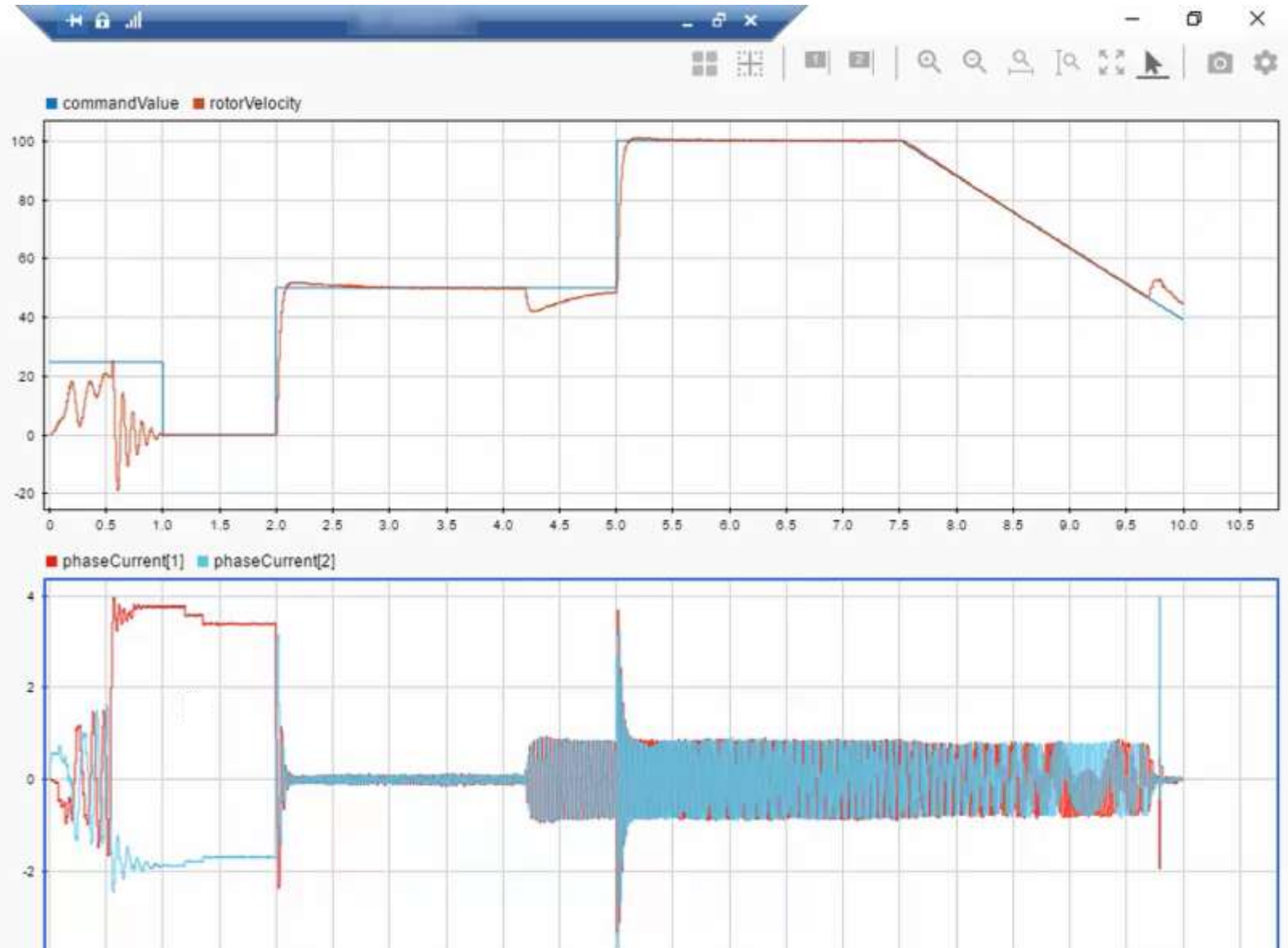
Simulation Data Inspector - untitled*

Inspect Compare

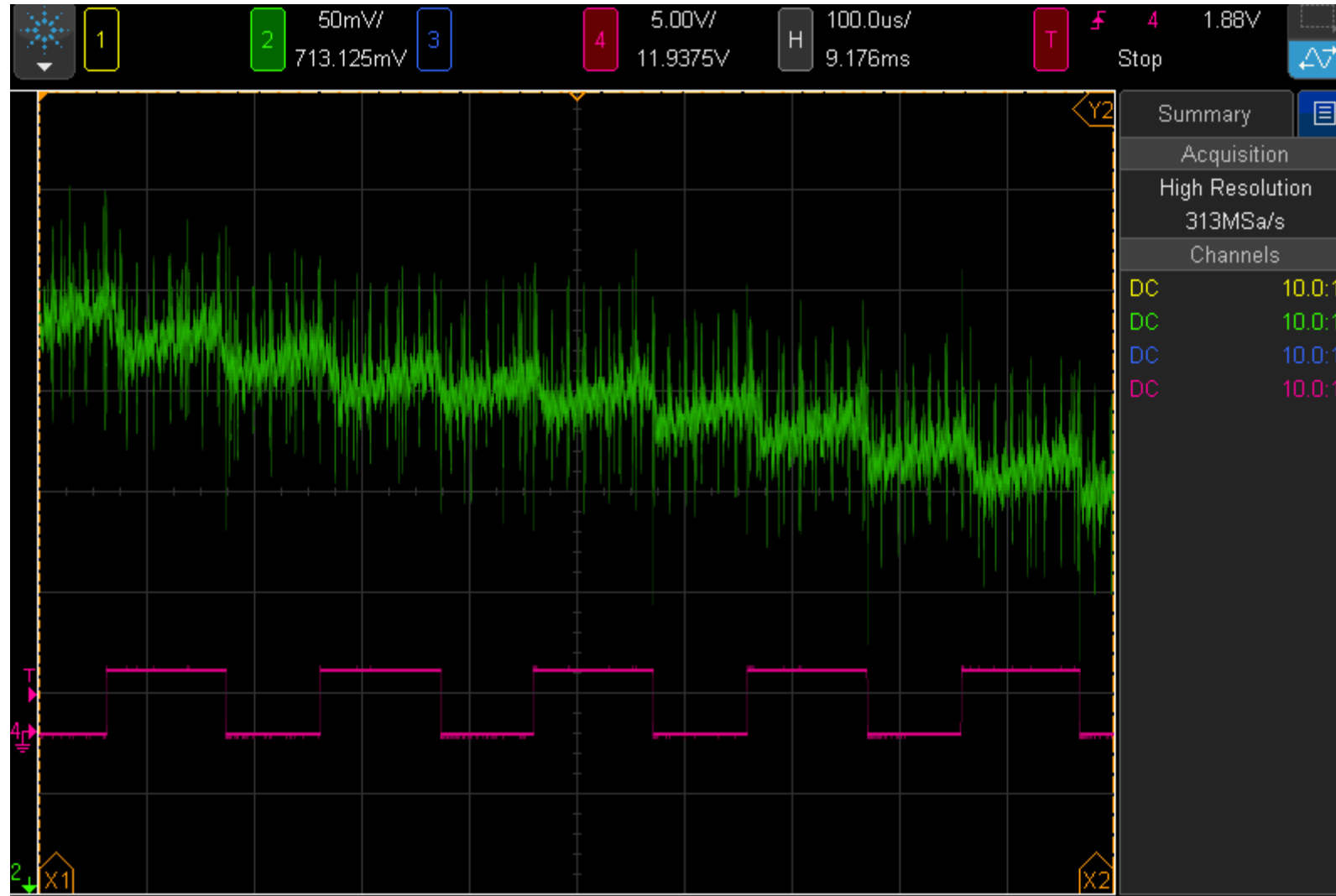
Filter Signals

NAME	LINE
<input type="checkbox"/> TET.BaseRate.maxTET	---
<input type="checkbox"/> TET.BaseRate.TET	---
<input type="checkbox"/> TET.SubRate1.minTET	---
<input type="checkbox"/> TET.SubRate1.maxTET	---
<input type="checkbox"/> TET.SubRate1.TET	---
<input type="checkbox"/> dCurrent	---
<input type="checkbox"/> qCurrent	---
<input type="checkbox"/> qCurrentCommand	---
<input checked="" type="checkbox"/> phaseCurrent[1]	---
<input type="checkbox"/> encoderOffset	---

PROPERTIES	VALUES
Name	phaseCurrent[1]
Line	---
Units	
Data Type	single
Sample Time	0.0001
Model	OpenLoopHWTestBe...
Block Name	ADC_Transducer_Vol...
Block Path	OpenLoopHWTestBe...
Port	1
Dimensions	[2]
Channel	[1]
Run	Run 1: OpenLoopHW...



Current Waveforms



High Level Process for Deploying Model to FPGA

1. Create high level subsystem for defining I/O
2. Convert model to discrete time
3. Convert to single precision
4. Use HDL workflow advisor to setup model settings
5. Deploy model to the Speedgoat real-time machine

Conclusion

1. Integrate desktop simulation with HIL simulation
2. Native Floating Point support for FPGA
3. Workflow advisor creates a seamless transition from desktop simulation to FPGA implementation

For more Information:

Joel Van Sickle

Joel.Vansickle@MathWorks.com

248-596-7939

**Want to know more?
Visit us at the booth.**

For more Information:

Joel Van Sickle

Joel.Vansickle@MathWorks.com

248-596-7939

Simulink Programmable FPGA I/O modules

Optimized for Power Electronics HIL and RCP

The IO334 I/O module is optimized for HIL simulation of real power stages. The card combines fast, low-latency analog and digital I/O capabilities, and is optimized for use with HDL Coder Workflow Advisor from MathWorks.

Analog connectivity:

16 x 5 MHz ADC, +/-10V, ENOB > 13-bit at 5 MHz

16 x 2 MHz DAC, +/-10V, settling time <1us

Multi-Gigabit Transceivers:

4 x MGT for inter-board communication

Enables scalability - I/O and computational resources

Selectable rear plug-ins add:

Digital TTL/RS422 I/O support for PWM / Encoder

Front SFP cages to access MGT at the out side of the enclosure

