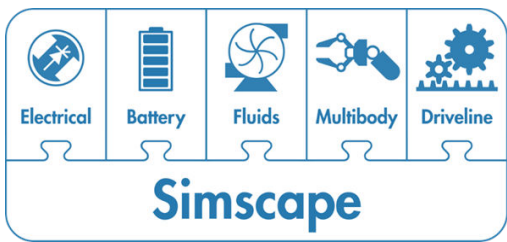# Modeling and Hardware-in-the-Loop (HIL) Testing of Physical Systems using Simscape

Duration: 2.5 hours

Version: MATLAB 2025b

Table of Contents

Introduction to Simscape (10 minutes)

Enable physical modeling of multidomain physical systems

- Assemble a schematic
- Equations derived automatically
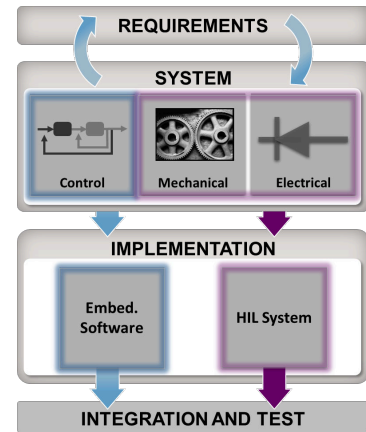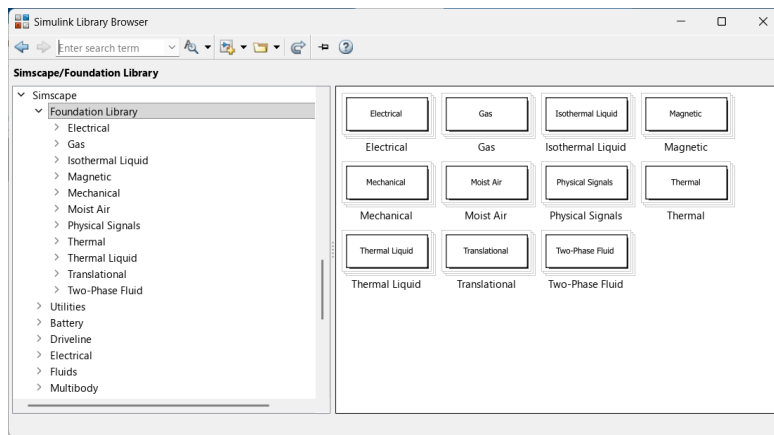- Leverage MATLAB and Simulink

With Simscape, you can:

- Refine requirements for system
- Discover integration issues early
- Design control systems and logic
- Optimize system-level performance
- Test embedded software without hardware prototypes

$$i = (C_0 + C_v v)\frac{dv}{dt} + \frac{v}{r_d}$$

```
Editor - C:\+MyComponents\LossyUltraCapacitor.ssc
40    equations
41        i == (C0 + Cv*vc)*vc.der + vc/Rd;
42        v == vc + i*R;
43    end
```

This library contains blocks that model systems from many physical domains: electrical, mechanical, fluid, thermal, and more.
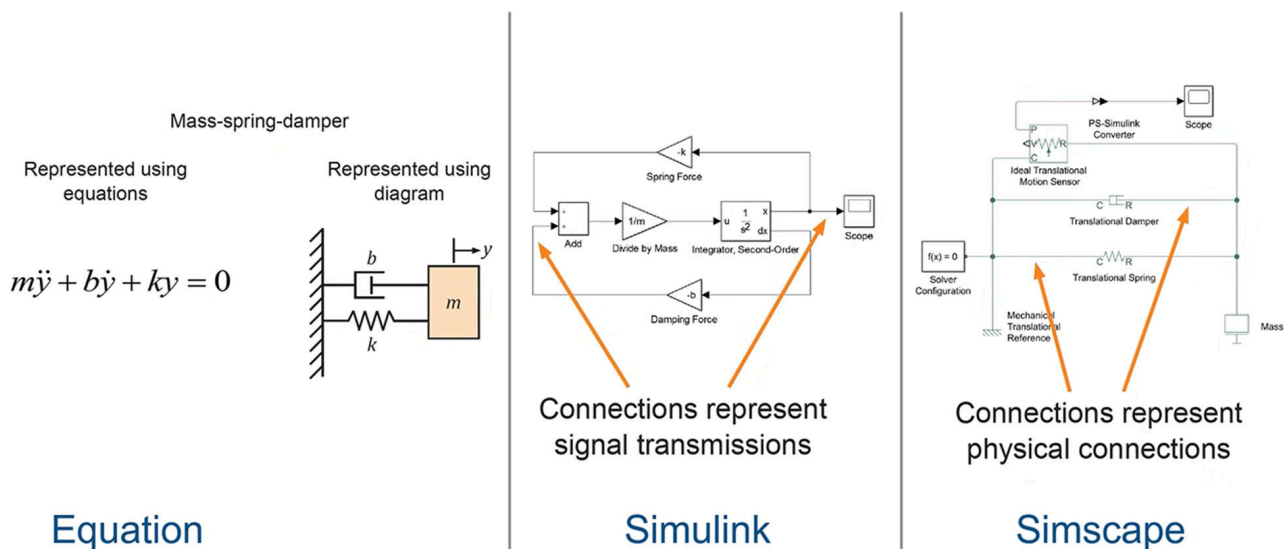
```
slLibraryBrowser
```

Simscape offers a comprehensive environment for physical modeling, providing a wide array of foundation blocks that span over ten different physical domains. Beyond the standard blocks, Simscape allows users to create custom components through its dedicated Simscape language.

The diagram above shows an electrohydraulic servo-valve created from Simscape components. For applications requiring real-time performance, Simscape provides specialized solvers optimized for this purpose.
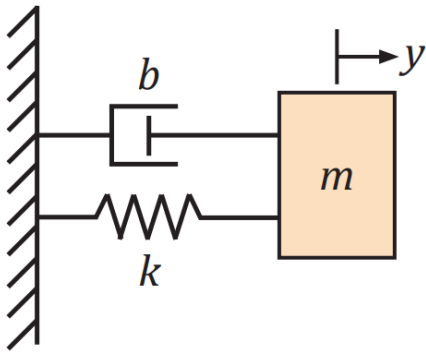
Simscape supports C-code generation, enabling users to test embedded controllers in a simulated environment.

Building your First Simscape Model (30 minutes)

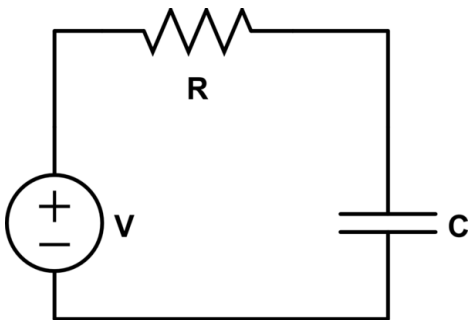Modeling Differences Between Simulink® and Simscape™

In the first exercise, you will use blocks from Simscape Foundation Library to model physical systems, specifically to build a mass spring damper.

```
open("E01_Mass_spring_damper.slx")
```

In this exercise, you will model physical systems specifically using elements from the electrical domain to build an RC circuit.


```
open("E02_RC_circuit.slx")
```

Solver Configuration

https://www.mathworks.com/help/releases/R2025b/simscape/ref/solverconfiguration.html?searchHighlight=Solver%20Configuration&searchResultIndex=1
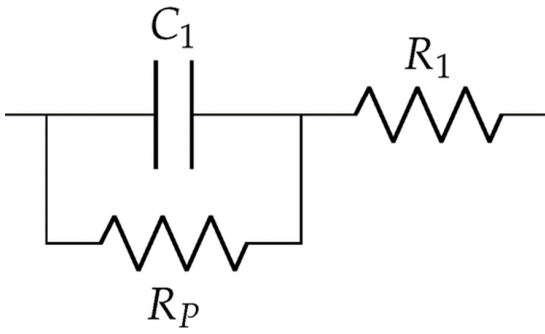
Parameterization and Real-World Data Integration (20 minutes)

In this example, we will model real capacitor (ECA-1CM100) which can be found in the datasheet:

https://industrial.panasonic.com/cdbs/www-data/pdf/RDF0000/ABA0000C1218.pdf

The extracted parameters are:

Parameters	Values
Capacitance	$10\mu F \pm 20\%$
Rated Voltage	$16V$
ESR (at $100Hz$)	2.6Ω (typ)
Leakage current	$3\mu A$ (max)
Operating temp	$-40°C$ to $+85°C$

3

```matlab
% Real component values
C = 10e-6;
ESR = 2.6;
V_rated = 16;
I_leakage = 3e-6;
R_leakage = V_rated/I_leakage;

% Simulation time
stop_time = 0.01;

open("E03_Real_RC_circuit.slx")
```

Simulation Data Inspector

https://www.mathworks.com/help/releases/R2025b/simulink/slref/simulationdatainspector.html?searchHighlight=data%20inspector&searchResultIndex=1

Multidomain Modeling Exercise (30 minutes)

This model is based on a Faulhaber Series 0615 DC-Micromotor. The parameters values are set to match the 1.5V variant of this motor. The model uses these parameters to verify manufacturer-quoted no-load speed, no-load current, and stall torque.

When running the simulation, for the first 0.1 seconds the motor has no external load, and the speed builds up to the no-load value. Then at 0.1 seconds the stall torque is applied as a load to the motor shaft. The simulation results shows a good level of agreement with manufacturer data.

```matlab
open("E04_PermanentMagnetDCMotor.slx")
```
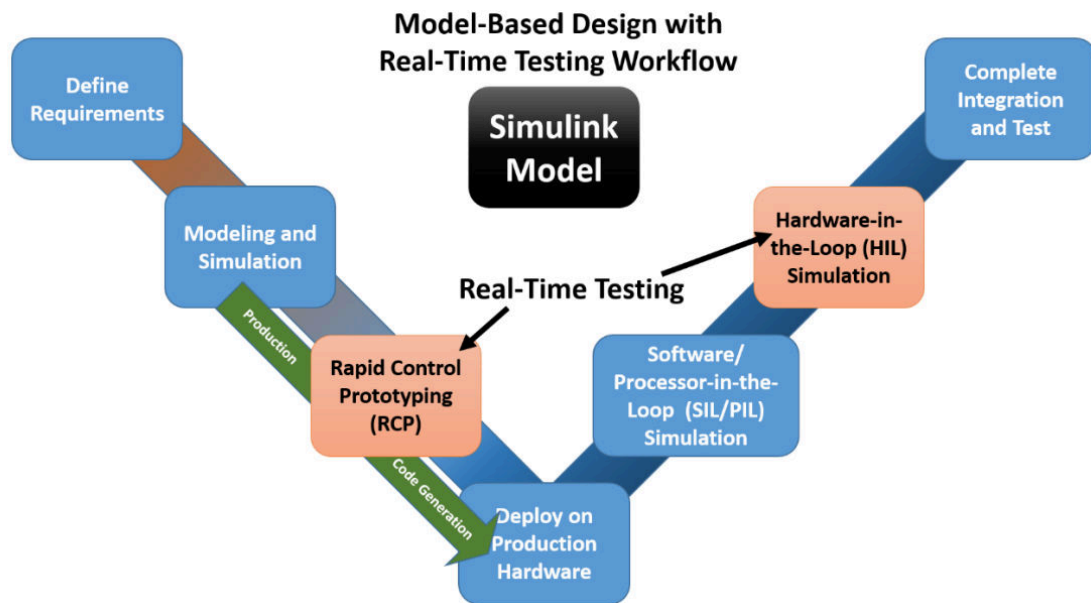
Simscape Result Explorer

https://www.mathworks.com/help/releases/R2025b/simscape/ref/simscaperesultsexplorer.html?searchHighlight=simscape%20result%20explorer&searchResultIndex=1

Integrating with Control Algorithms (15 minutes)

Using the model we developed previously, we will be integrating control algorithms to control the motor speed.

```matlab
open("E05_PMDC_PID.slx")
```

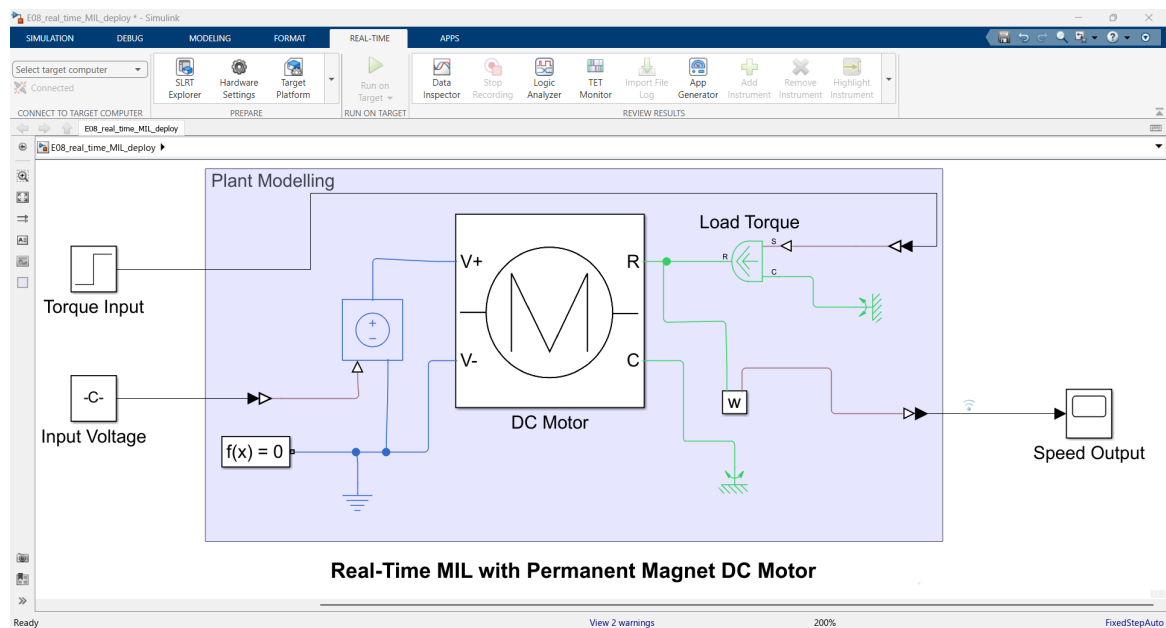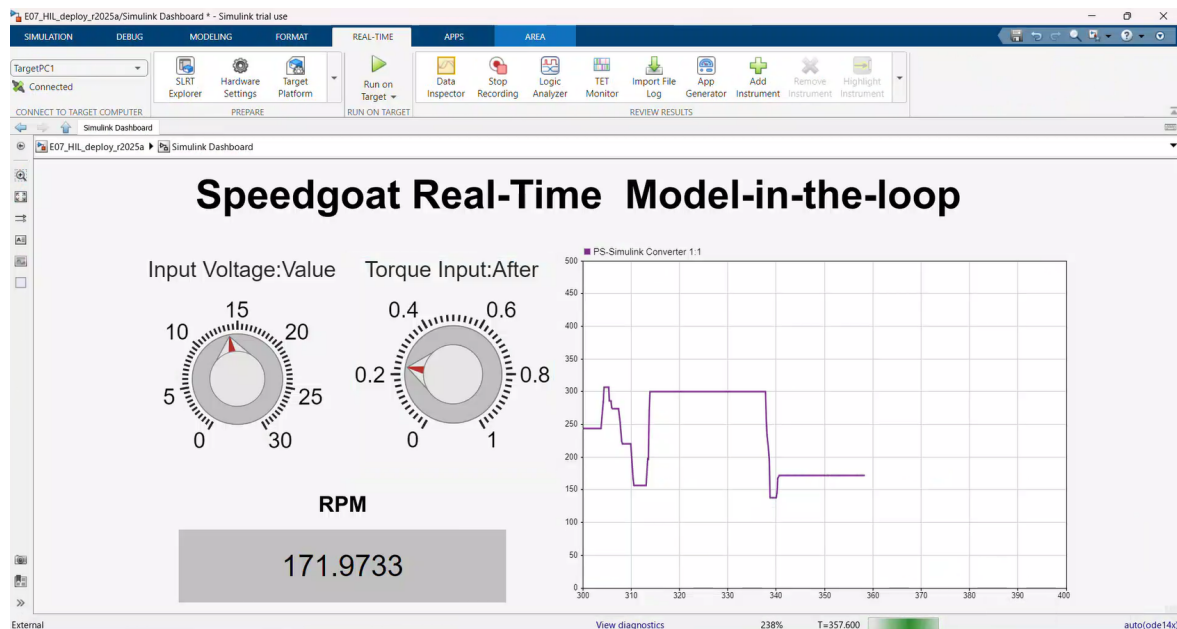# Demonstration - Speedgoat Hardware-in-the-Loop (HIL) (25 minutes)

Model-Based Design with Real-Time Testing Workflow

Speedgoat

https://www.speedgoat.com/

Speedgoat Real-Time MIL

MIL Plant in Speedgoat:

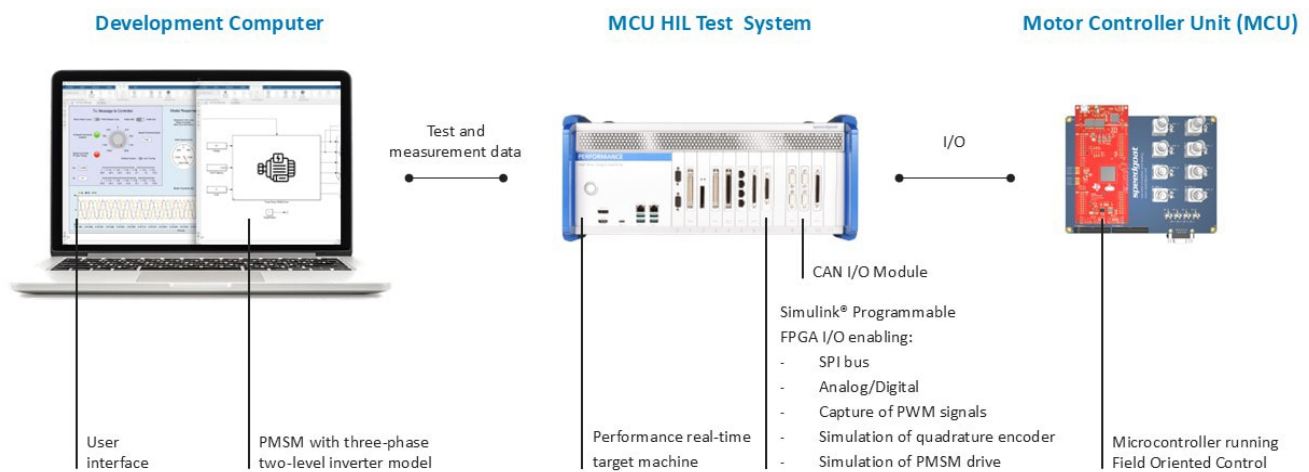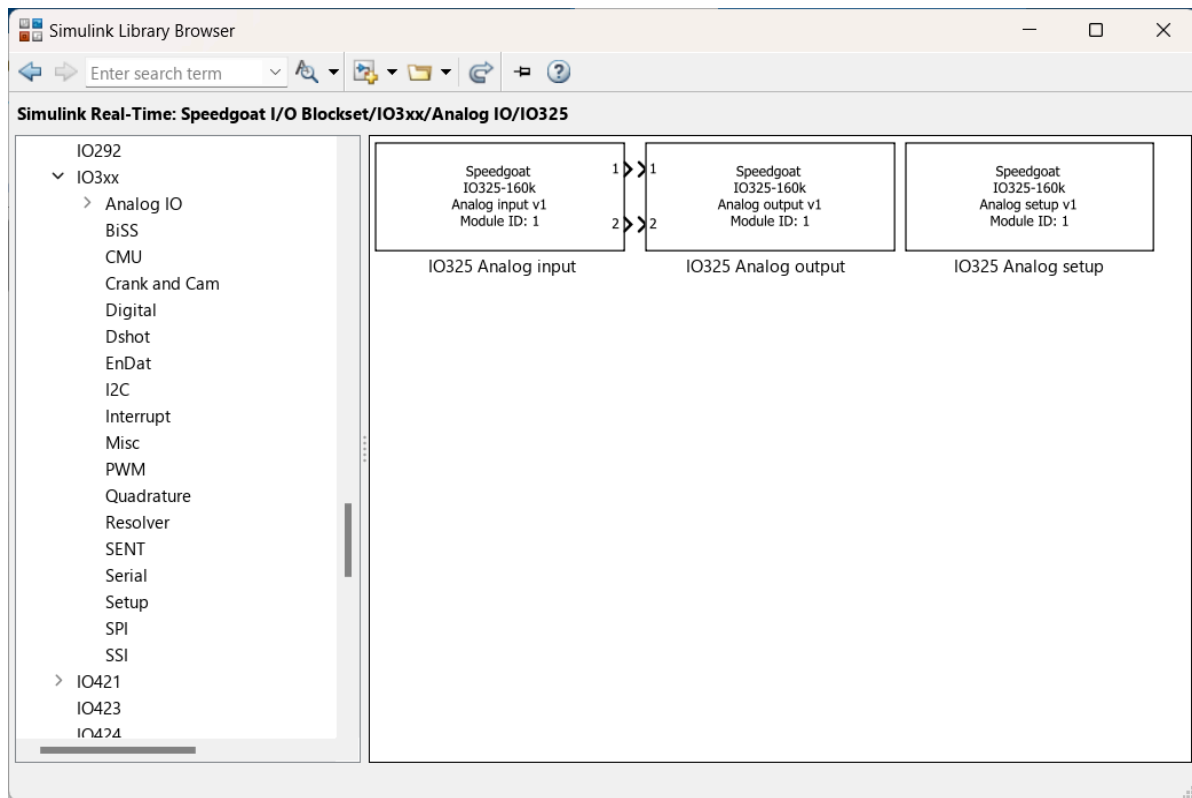Real-Time MIL with Permanent Magnet DC Motor

```
open("E08_real_time_MIL_deploy.slx") % for plant
```
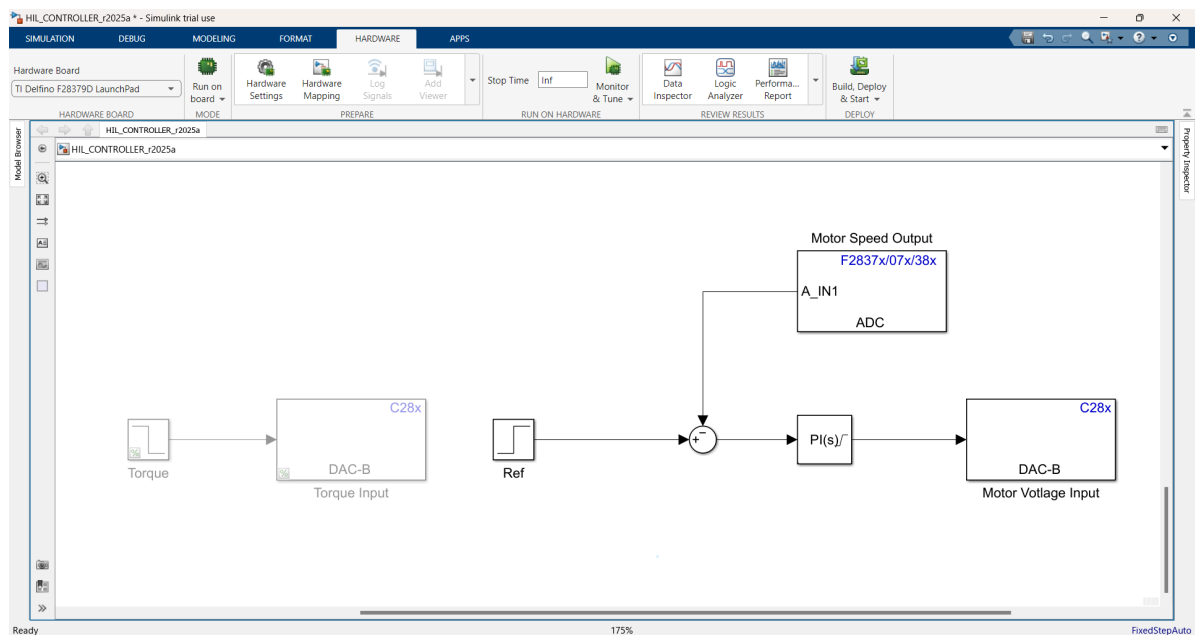
Speedgoat Real-Time HIL Workflow

```
open("E06_HIL_partition.slx")
open("E07_HIL_deploy.slx")
```


Development Computer   **MCU HIL Test System**   **Motor Controller Unit (MCU)**

Test and measurement data

I/O

CAN I/O Module

Simulink® Programmable
FPGA I/O enabling:

- SPI bus
- Analog/Digital
- Capture of PWM signals
- Simulation of quadrature encoder
- Simulation of PMSM drive

User interface

PMSM with three-phase two-level inverter model

Performance real-time target machine

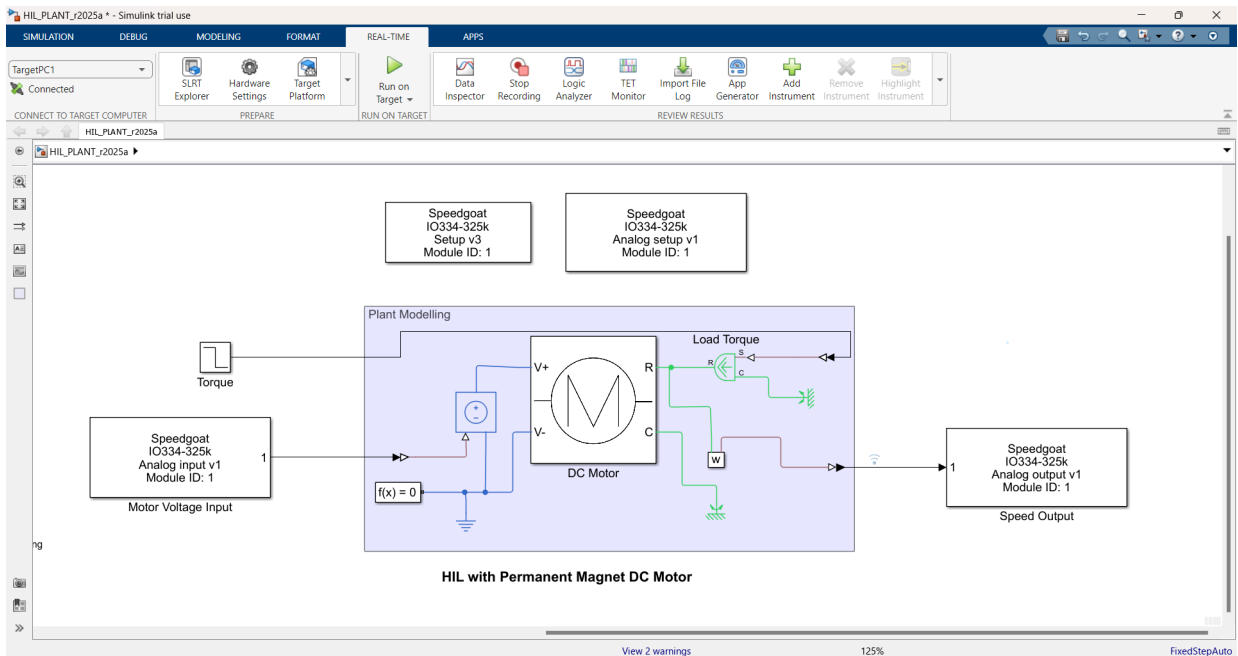Microcontroller running Field Oriented Control

Simulink Real-Time: Speedgoat I/O Blockset:
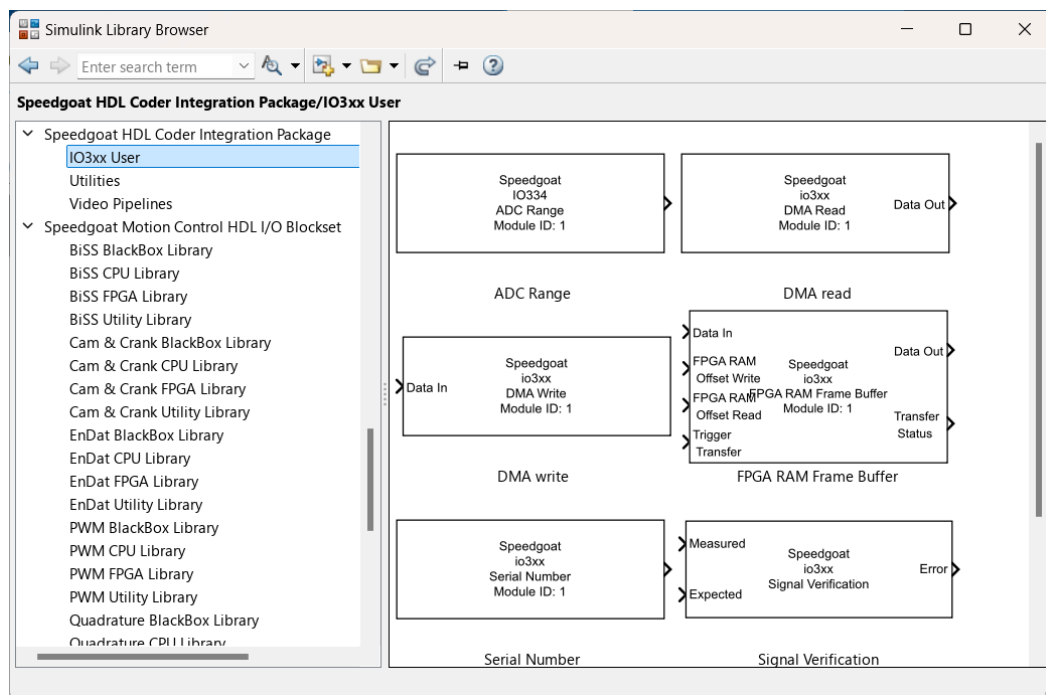
HIL Controller in Device under Test (Controller):
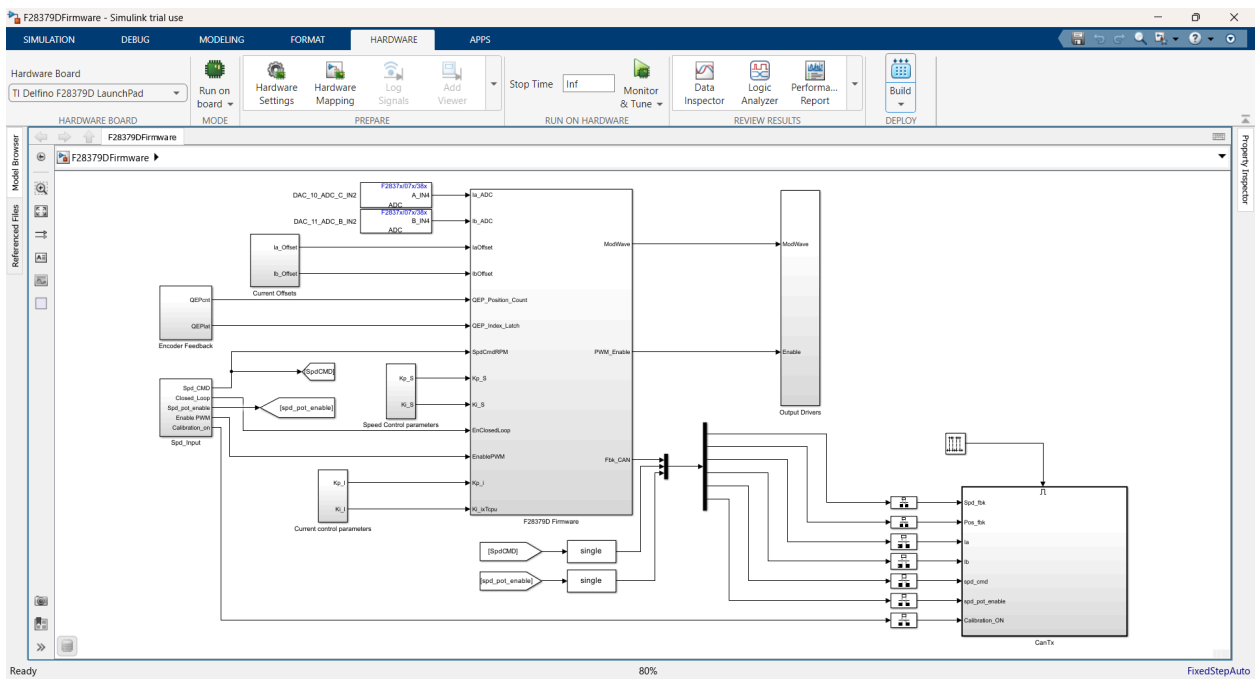

HIL Plant in Speedgoat:

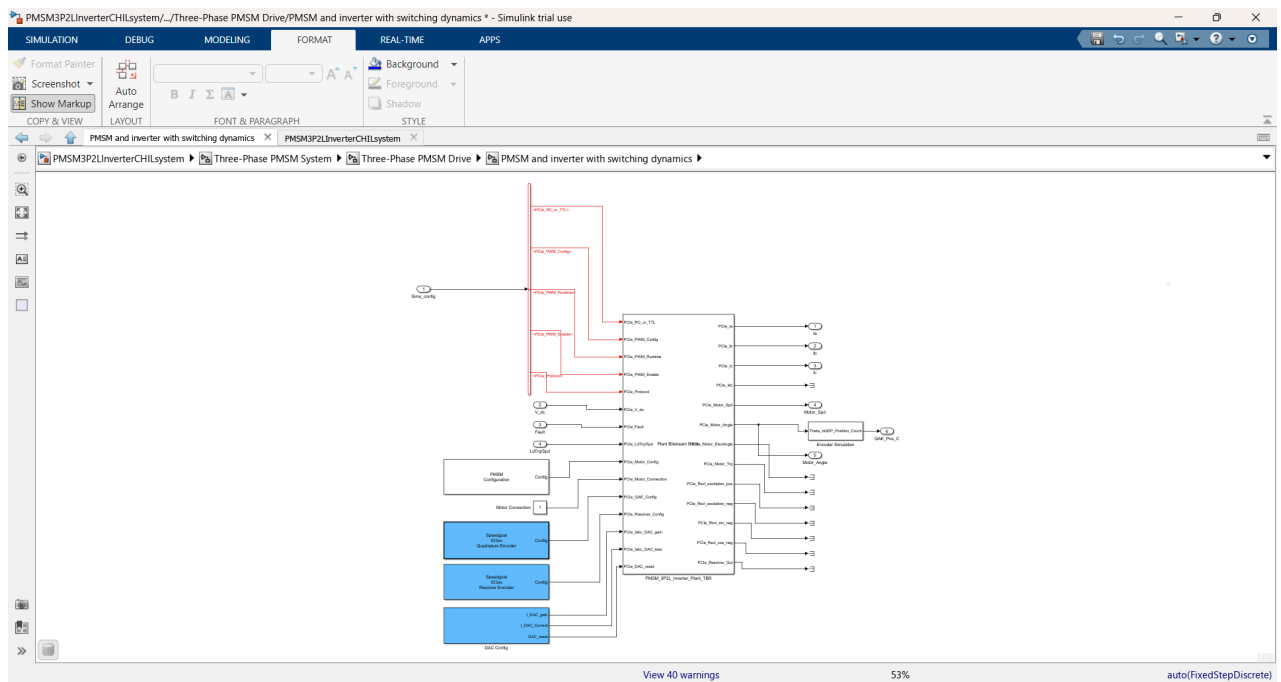## HIL Demo with HDL Workflow - Three-Phase PMSM with Two-Level Inverter

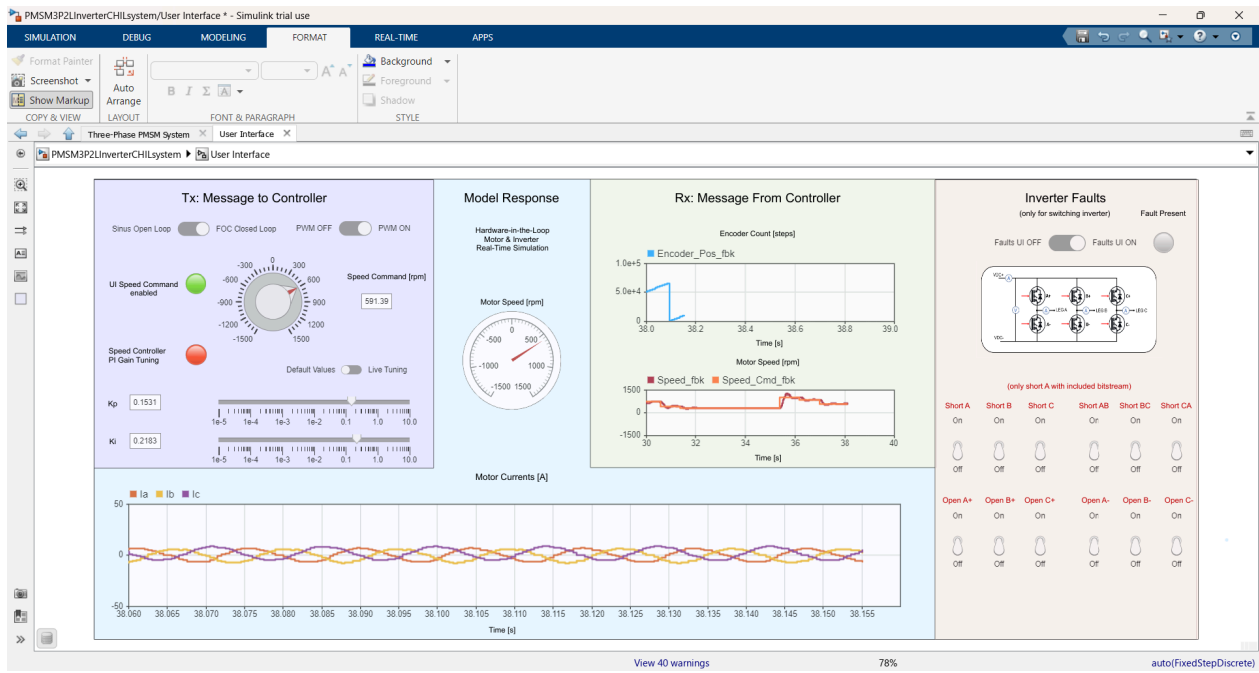Simulink Real-Time: Speedgoat HDL Blockset:

HIL Controller in Device under Test (Controller):

HIL Plant in Speedgoat:

HIL with Simulink UI:

Simulink Real-Time Target Support Package

https://www.mathworks.com/matlabcentral/fileexchange/76387-simulink-real-time-target-support-package