

HARDWARE IN THE LOOP SIMULATION FOR EV/HEV DRIVETRAIN DEVELOPMENT

MODEL-BASED TECHNIQUES AID DEVELOPMENT OF VEHICLES AND OTHER COMPLEX ELECTROMECHANICAL SYSTEMS

By Lee Goldberg, Editor

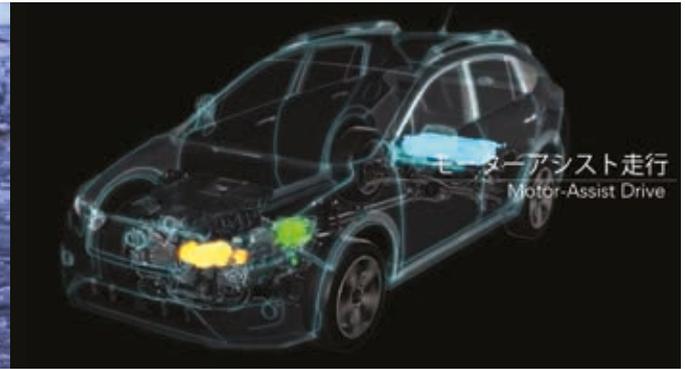


Figure 1: HIL simulation and test enabled Subaru to insure the hybrid drivetrain of its XV Crosstrek SUV would remain efficient, reliable, and safe under even the most severe road conditions. Images courtesy of Subaru.

Hardware in the Loop Simulation (HILS) and Real-Time Control Prototyping (RCP) are both model-based development techniques that substitute a portion of the target system with a mathematical model, implemented in either software running on one or more processors, or on hardware logic (typically an FPGA). In addition to enabling parallel development of hardware and software, HILS systems provide a straightforward method for analyzing the complex interactions between electric, electronic, and mechanical components which can create problems that are difficult to identify or analyze using traditional testing methodologies.

This article provides an introduction to the capabilities that gave HILS/RCP an early foothold in the automotive industry and made them indispensable tools creating today's complex conventional, electric, and hybrid-electric vehicle drivetrains. Although the examples in this story involve automotive applications, HILS is becoming increasingly popular for the development of aircraft, agricultural equipment, defense systems, large wind generators, and other applications that involve complex electro-mechanical systems.

HIL IN A NUTSHELL

HIL simulation presents an embedded system controller with artificial stimulus that closely mimics the actual behavior of the system's physical elements and logs the controller's

responses. When used in automotive applications, it can be used to exercise an electronic control unit (ECU), simulating the behavior of a motor and/or its drivetrain components interacting with a virtual environment as well as any associated sensor/actuator interactions.

At its simplest, generating stimulus for an ECU controlling an electric motor might entail creating signals representing a series of desired speeds (to simulate pushing the accelerator pedal) and vehicle loads that result from a set of simulated road conditions. The system's responses are analyzed to verify all elements are exhibiting proper function and performance (Figure 2). More complex HILS systems can also accurately model drivetrain loads created by the interactions between the vehicle body, and the roadway.

As shown in Figure 3a, an HIL test system consists of three primary components: a real-time processor, I/O interfaces, and an operator interface. The real-time processor (RTP) is responsible for providing deterministic execution of most of the HIL test system components. This includes model execution and stimulus generation, as well as acquiring and logging data for analysis by the operator interface. The RTP communicates with its target system via its I/O interface that contains analog, digital, and bus signals that interact with the unit under test.

The simple HILS architecture shown in Figure 3a uses software to simulate fault conditions. While this technique is sufficient for many types of simulation, it is not always accurate, especially systems which must produce fine-

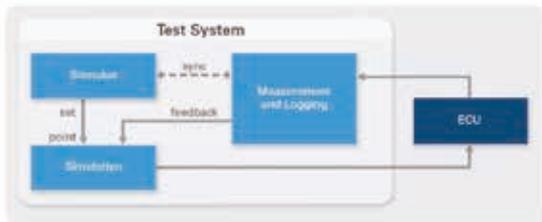


Figure 2: A Hardware-in-the-Loop test system contains several key functions when simulating the world around an ECU. Image courtesy of National Instruments

grained responses to rapidly changing conditions. For these applications, a technique known as Hardware Fault Insertion is used to characterize, or validate the behavior of the device (Figure 3b). To accomplish this, fault insertion units (FIUs) are placed between the I/O interfaces and the ECU, thereby enabling the HIL test system to switch the interface signals between normal operation and fault conditions such as a short-to-ground or open circuit.

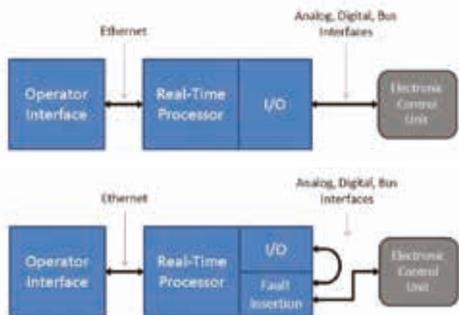


Figure 3: a) A high-level block diagram of a HILS test system used to evaluate an automotive electronic control unit (ECU). b) A HILS system with hardware fault insertion. Images courtesy of National Instruments.

KEEPING IT REAL

When implementing an HILS test system, one of the first issues to address is making sure that the real-time simulator's response rate is fast enough to faithfully reproduce the drivetrain's actual behavior. Since the RTP is primarily a digital system, the simulations it creates are performed in discrete time steps. The speed at which it samples inputs from the ECU and produces the corresponding stimulus is referred to as Step Time or Sample Time (T_s). The delay between the sampled input from the article under test and the corresponding output from the RTP and I/O signal chain defines the minimum T_s the HIL system can support.

As shown in Figure 4, care must be taken to insure that the RTP's step time is fast enough to accurately model the behavior of the drivetrain elements it's simulating. For applications involving relatively low-frequency phenomena, the algorithms that define them can be implemented in software running on a dedicated processor or processor array. When simulating complex systems, the model's algorithms can be broken up and run on two or more CPUs to keep T_s within

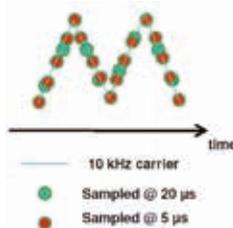


Figure 4: The RTP's step time (T_s) must be fast enough to accurately represent the signals of the system it is simulating. Source: Opal-RT

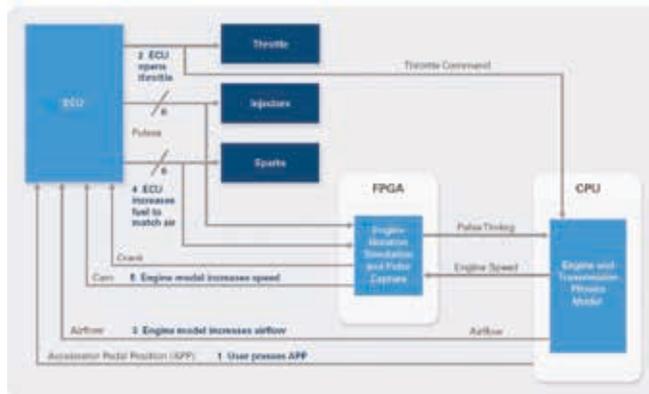


Figure 5: A hybrid HIL system uses a combination of software and programmable hardware logic to achieve higher fidelity modeling.

acceptable limits.

But system bus latency and several other factors limit the maximum sampling speeds CPU-based HIL systems to only 50-100 kHz ($T_s=10-20 \mu s$). As a result, even multiple CPUs may not be sufficient to accurately model an advanced electric drivetrain. For these applications, the RTP's CPU(s) can be offloaded by implementing some of the model's algorithms and functions in hardware, typically using one or more field-programmable gate array (FPGA) devices. The system's FPGAs can be configured using a variety of tools that translate models written in MATLAB, Simulink and, in some cases, VHDL, into device-specific programming commands that are downloaded into the target platform (Figure 5).

TESTING HYBRID DRIVETRAINS

The capabilities we've covered here make HILS systems an excellent tool for developing and testing the complex electric drivetrains found in today's EVs and hybrid vehicles where the ECU must manage the interaction between the conventional engine and the electric motor, along with its power systems. In these applications, HIL simulation is especially valuable for software robustness testing, where the test engineers use fault injection, corner case conditions, diagnostics, and automated testing to document the system's performance across a larger range of possible operating scenarios and fault conditions than would be possible with field testing alone.

One of the tasks where this technique can be invaluable is simulating the drivetrain to icy driving conditions where one or more wheels can suddenly lose some or all traction. Because the drivetrain's gasoline and electric motors have very different time constants and response rates, unexpected wheel slippage could cause them to interact in unexpected ways. Complex control algorithms for specific safety conditions need to be developed and verified, but these slip and skid conditions cannot be physically reproduced on a dynamometer and are difficult or impossible to reproduce on a test track.

HIL IN ACTION

HILS played a leading role in the development of Subaru's first hybrid vehicle, the XV Crosstrek, first introduced in 2013 (Figure 1). The company opted to use a HIL-based

development platform early in the program after an earlier prototype ECU developed in a conventional manner proved unable to meet the full gamut of safety and performance requirements expected of a production vehicle.

Subaru engineers used HILS to begin software development and verification early in the design process, long before a prototype or pre-production vehicle became available. They connected the ECU to a real-time electric motor simulation that would methodically test and verify its operation under a carefully defined set of conditions. The conditions included extreme corner case events that had the potential to break the system if a traditional mechanical testing regime was employed. In order to make sure the software-based approach would accurately model the real-world drivetrain, Subaru developed a set of three primary goals for their test program:

- The tests should verify ECU functionality across the widest possible range of conditions, including extreme environments not easily created or replicated.
- Create a detailed matrix for mapping test cases to requirements, and a methodical test plan to ensure complete test coverage.
- The test platform should allow design iterations to be quickly validated using regression testing.

To achieve these goals, they used a phased methodology for embedded software design and deployment validation. The plan included test points at each stage where the team could use the HIL system to verify the motor ECU's behavior against a real-time simulation of the actual vehicle motor. They also wanted to use the HIL System to meet traceability requirements by recording test results automatically and performing automated regression tests whenever a change was made to the ECU software.

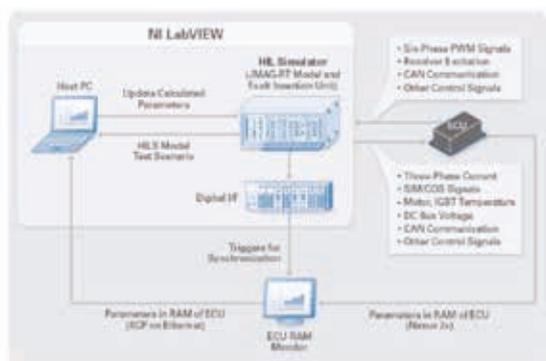


Figure 6 – Subaru's verification environment. Source: National Instruments

Subaru's verification system consists of a real motor ECU and a HIL system from National Instruments that simulates motor operations (Figure 6). It can represent any operating condition of the motor by setting physical parameters such as inductances or resistances. The simulator can also represent the characteristics of the drivetrain's power electronics for test scenarios involving nearly any combination of load torque and desired rotating speed, including fault conditions.

By simply changing a parameter in the middle of the test, the HIL system can easily simulate complex test scenarios like the previous loss of traction example or even a power electronics

fault in the inverter that would destroy physical hardware.

Because the computational performance required to simulate some drivetrain elements was so high, they employed a mix of software and hardware simulation. The core system is based on NI's FlexRIO FPGA modules, which are PXI-based controllers with FPGA chips. The modules executed a model representing the simulated operation of the motors, with all deployed programs using NI LabVIEW system design software.

To insure the software was exercised across the full range of simulated operating conditions, Subaru's test engineers created a chronological test matrix using an Excel spreadsheet. The matrix described how the test conditions (torque, rotating speed, etc.) would vary at each 1 millisecond interval. During the test, the motor ECU would react to those conditions, sending out signals, such as a pulse-width modulation signal, to the HIL system that it used to as inputs for the motor simulation algorithm. The resulting signals representing the motor's torque and tri-phase current were returned to the motor ECU.

The test engineers wrote a program in Visual Basic that would read and run each test scenario by stepping through its associated matrix of test conditions and writing the test results into an Excel spreadsheet. By automating the test procedures, Subaru was able to cut test and verification cycles dramatically while insuring that the ECU software was thoroughly exercised under every possible operating scenario.

HILS ACCELERATES THE DESIGN CYCLE

By providing the controller with high-fidelity models of a drivetrain, robotic component, or other mechatronic system, HILS enables software engineers to begin debugging their control algorithms and evaluating hardware designs long before the physical hardware required to perform actual end-to-end testing becomes available.

HILS was originally used late in the design flow of automotive systems, where software developers would finish their ECU firmware and pass it to the test department to perform HIL testing. Now, however, HILS role is expanding, according to Marcus Monroe, Senior Technical Marketing Specialist at National Instruments. Monroe explains, "The exponential growth of embedded software and the extreme technical challenges of EV/HEV testing are driving test earlier into the development process. If firmware developers can test their algorithms on the desktop before they pass their code on to the test teams, they can eliminate bugs earlier and reduce the overall cost to test."

REFERENCES

- "Real-Time Simulation Technologies to Improve the Development of Electric Motor Drives" - Opal-RT, <http://www.opal-rt.com/sites/default/files/MotorDrive2013.pdf>
- "HIL Testing for Power Electronics Systems" National Instruments, <http://www.ni.com/white-paper/13778/en/>
- "Key Considerations for Powertrain HIL Test" – National Instruments, <http://www.ni.com/pdf/app-note/HIL-AppNotes-IA.pdf>
- "Advancing Subaru Hybrid Vehicle Testing Through Hardware-in-the-Loop Simulation" – National Instruments, <http://sine.ni.com/cs/app/doc/p/id/cs-15982# PDD>