

Perform Hardware-in-the-Loop Simulation with MATLAB and Simulink to Test and Validate Control Algorithms

When you design a complex control system for industrial automation production equipment, one of the chief concerns is catching design errors before integration testing. Errors that aren't caught can cause shipping delays or expensive downtime in production, as well as equipment damage or operator injury. These sophisticated control systems require extensive validation and testing before use in the field.

Developing a control system for industrial equipment is time consuming and complex. Traditionally, this includes the equipment developed by mechanical and electrical engineers, the algorithms developed by a control engineer, and integration and performance testing performed on the actual equipment or a hardware prototype. This process presents some common problems that can add significant delays to projects:

- Integration mistakes are often found late in the project
- Errors introduced in design can damage prototype equipment

Testing cannot exercise all potential failure modes when the hardware is damaged or inaccessible (such as a heating oven or a turbine).

These factors can end up delaying a project and also cost extra time and money while putting a strain on resources. You also want to test the control system in an environment that is close to the real system before deploying the control algorithm in the field.

An Approach to Preventing Failure

What if there is a possibility to simulate a model of the controller and machine in real time before you introduce the control system to run your oil rig or a wind turbine in the field?

What if you can validate your controller design by testing it against a model that virtually responds and behaves like the actual hardware?

Hardware-in-the-loop (HIL) simulation helps you test and validate your control algorithm in real time against your machine simulation even before connecting your controller to the actual plant and equipment (Figure 1).

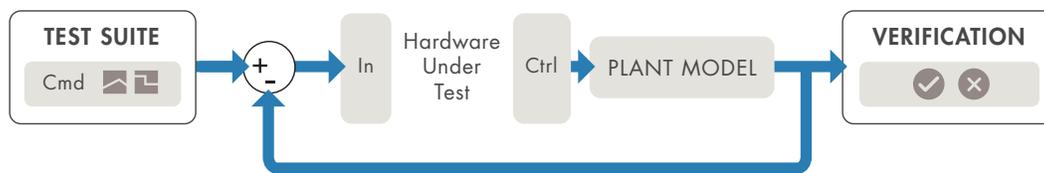


Figure 1. HIL simulation.

Why Perform HIL Simulation?

Typical testing involves using actual hardware connected to the controller to test real-use situations or environmental proxies (for example, a pressure chamber). In HIL simulation, you do not have to use the actual hardware or environmental test setup. HIL simulation allows you to use your model to represent the equipment and environment, offering benefits in cost and practicality. The goal is to perform HIL simulation to work out the errors and then move to final hardware testing, where you validate to prove good design, not test to find bad design.

There are several areas in which HIL simulation offers cost savings over traditional validation testing. HIL simulation tends to be less expensive when it comes to design changes because it is low cost to change a simulation model compared to modifying or replacing actual hardware. You can perform HIL simulation earlier than hardware validation in your development workflow, providing more time to identify problems and begin the redesign process. Detecting issues early is beneficial for several reasons:

- Your team is more likely to approve changes
- Design changes are less costly to implement
- You can catch errors early and spend more time on design
- You reduce the probability of breaking expensive hardware or equipment while testing control algorithms
- HIL simulation complements your existing workflow by testing critical faults and scenarios that are hard to test otherwise

HIL simulation is more practical than validation for testing your controller's response to unusual events. For example, you can model extreme conditions (such as vibration, temperature, or failures of an oil rig or packaging machine). You can also test how your controller responds to stimuli occurring in inaccessible environments (such as deep sea or deep space).

HIL Simulation Setup and Workflow

In HIL simulation, you use a separate and dedicated target computer as a virtual representation of your equipment model and a real version of your controller. Figure 2 shows a typical HIL simulation setup. On the desktop computer (development computer), you design the real-time capable model of the equipment. The development computer also contains an interface with which to control the virtual input to the model. The controller hardware contains the control algorithm software, which can be developed from the controller model or by any other independent workflow. The real-time target hardware runs code for the physical system that is generated from the hardware model.

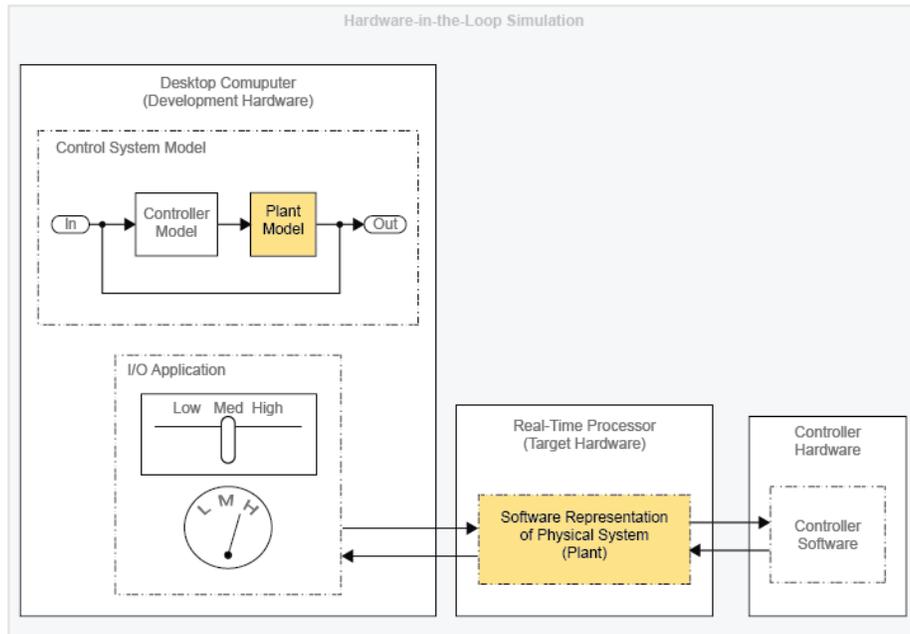


Figure 2. Controller-plant HIL setup.

MathWorks Solution for HIL Simulation

MathWorks provides a real-time simulation and testing solution with Simulink Real-Time and Speedgoat target hardware; this allows engineers to start with desktop simulation and proceed to design, build, and test the real-time application. Simulink Real-Time and Speedgoat target computer hardware are expressly designed to work together to create real-time systems for desktop, lab, and field environments.

One such HIL example is shown in Figure 3. This HIL simulation setup has a PLC (programmable logic controller) connected to a Speedgoat target computer. Both the controller and plant simulation (on the target computer) run in real time and communicate via a high speed bus. Using Simulink, you can design the plant simulation on your desktop computer by using physical modeling tools such as Simscape. You then run the control algorithm on a PLC and the plant simulation on the Speedgoat real-time target computer.

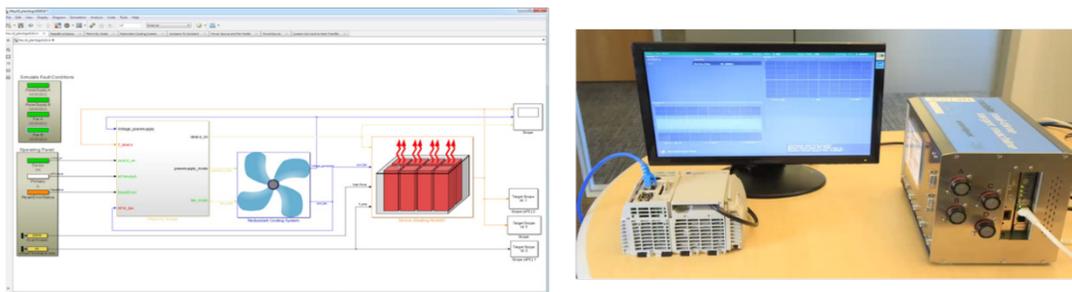


Figure 3. Left: Simulink model on the desktop computer with the control algorithm and the plant simulation of the cooling fan system. Right: Industrial supervisory control cooling fan system HIL simulation setup with the plant simulation running in real time on Speedgoat target hardware.

In this setup, the control algorithm and plant simulation are automatically generated from Simulink to target the PLC and Speedgoat target computer.

- The control algorithm is deployed to the PLC using Simulink PLC Coder
- The plant simulation is run in real time on a Speedgoat target computer using Simulink Real-Time and Simulink Coder

Communication between the PLC and Speedgoat target computer is via an industrial communication bus (see Figure 4).

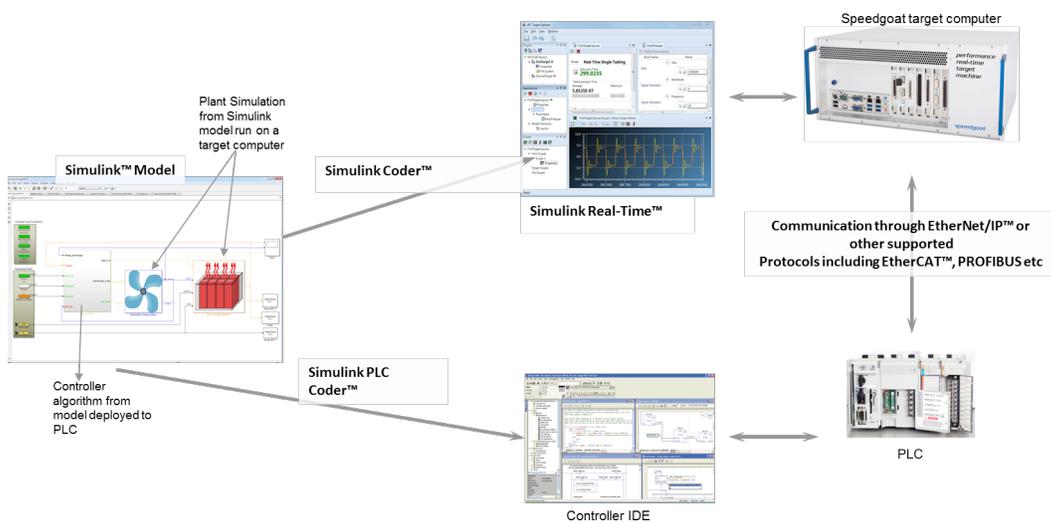


Figure 4. HIL simulation setup with industrial PLC and Speedgoat target computer using Simulink Real-Time.

At this point, you are validating the control algorithm by running it against a real-time equipment simulation, even before testing the controller on a real machine. This is a faster, safer, and more cost efficient approach to catch errors earlier in the development process. Once you have completed your HIL simulation, your controller can interface with the real plant.

Summary

You have seen how you can perform HIL simulation to test and validate your control algorithm before you run it using actual equipment. This approach saves costs, finds errors, and prevents damage to expensive equipment. You can run multiple iterations of your design quickly and prevent downtime of your actual plant in the event design errors are found on a production system.

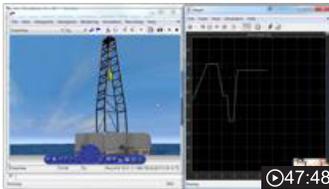
Explore How Engineers Use Simulink Real-Time



manroland Develops High-Precision Commercial Printing Press Controller



Hardware-in-the-Loop Simulation with Simulink Real-Time Using PLCs and Ethernet/IP Protocol for Validating Industrial Automation Control Algorithm



HIL Simulation and Testing with Simulink Real-Time and Speedgoat Target Hardware



Protocol Support - Industrial automation and machinery protocol support from Simulink Real-Time