Hardware-in-the-Loop Simulation

Introduction

What is Hardware-in-the-Loop (HIL) Simulation or What is Hardware-in-the-Loop (HIL) Test?

The Hardware-in-the-Loop process has existed for no more than 15 to 20 years. Its roots are found in the Aviation industry. The reason the use of a HIL process is becoming more prevalent in all industries is driven by two major factors: time to market and complexity.

Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex process systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. These mathematical representations are referred to as the “plant simulation.”

Hardware-In-the-Loop is a form of real-time simulation. Hardware-In-the-Loop differs from real-time simulation by the addition of a real component in the loop. This component may be an “Electronic Control Unit” (ECU).

The purpose of a Hardware-In-the-Loop system is to provide all of the electrical stimuli needed to fully exercise the ECU. In this way you “fool” the ECU into thinking that it is indeed connected to a real plant.

The HIL simulation includes a mathematical model of the process and a hardware device/ECU you want to test, e.g. an industrial PID controller we will use in our example. The hardware device is normally an embedded system.

Why use HIL simulation?

This question is an important part of understanding real-time technology. To restate the question using a control systems term: Why not connect the embedded system under test to the “real plant”, that is the dynamic system being controlled, to perform development and testing? In many cases, the most effective way to develop an embedded system is to connect the embedded system to the real plant, if such a plant exists. Increasingly however, HIL simulation is more efficient and or required.

The main purpose with the HIL Simulation is to test the hardware device on a simulator before we implement it on the real process.

http://home.hit.no/~hansha
The metric of development and test efficiency is typically a formula that includes the following factors:

- Cost
- Duration
- Safety

You may want to test the different part of the system individually to make sure it works as planned and HIL simulation is important in design and testing of the different systems.

It may be very useful, e.g., to test a controller function with a simulated process before the controller is applied to the real (physical) process. If the mathematical model used in the simulator is an accurate representation of the real process, you may even tune the controller parameters (e.g. the PID parameters) using the simulator.

It is also very useful for training purposes, i.e., the process operator may learn how the system works and operate by using the hardware-in-the-loop simulation.

Another benefit of Hardware-In-the-Loop is that testing can be done without damaging equipment or endangering lives. For instance, potentially damaging conditions in an engine, such as over-temperature, can be simulated to test if the ECU can detect and report it. Another instance would be an anti-lock braking (ABS) simulation at performance extremes. If simulated, the performance of the ABS system can be evaluated without risk to the vehicle or operator.

HIL should be an integrated part of the design and testing cycle.

As the complexity of the hardware being controlled increases, so too does the complexity of the embedded system that is designed to control the hardware. Hardware-in-the-Loop (HIL) simulation is a technique that is used increasingly in the development and test of complex real-time embedded systems.

The purpose of HIL simulation is to provide an effective platform for developing and testing real-time embedded systems, often in close parallel with the development of the hardware. Software development no longer needs to wait for a physical plant in order to write and test code.

HIL simulation provides an effective platform by adding the complexity of the plant under control to the development and test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation (model) of all related dynamic systems. These mathematical representations are referred to as the “plant simulation.”

**Challenges**
When testing, we have lots of challenges:

- Cost to test
- Cost of failure
- Availability
- System variation
- Repeatability

In these situations, is HIL simulation a powerful technique. With HIL Testing we will reduce cost and risk.

With HIL Testing cost and risk will be reduced:

- Increased reliability and quality
- More efficient development
- Lower cost to innovate

Applications

Embedded Control Systems

HIL simulation is widely used in developing Embedded Control Systems, such as:

- Medical Devices
- Industrial machines
- Power Generation Systems
- White Goods
- Aerospace
- Automotive
- Process Control

Procedure

The main steps in HIL Simulation are as follows:

1. **Develop a mathematical model.** Create a mathematical model of the real environment where the hardware device is meant to be used.
2. **HIL Simulation** (Software + Hardware). Test your device on a simulated process (mathematical model).
3. **Implement your hardware on the Real Process** (Hardware only). If everything is OK, you may want to implement your hardware device in the real environment where it meant to be used.

These tasks follow the main idea with a HIL simulation. First step is to simulate your system in software. Next is to test your hardware on the simulated process. Finally you implement your hardware on the real system.
**Practical Example**

**Introduction**

It may be very useful to test a controller function with a simulated process before the controller is applied to the real (physical) process. If the mathematical model used in the simulator is an accurate representation of the real process, you may even tune the controller parameters (e.g. the PID parameters) using the simulator.

If the controller to be tested is implemented in the controller hardware, often denoted the electronic control unit (ECU), and the simulator has to run in real time, i.e. the simulation time develops as real time. This real time simulation is obtained by setting the simulation algorithm cycle time equal to the simulation time step.

Typically, the simulator communicates with the ECU via ordinary I/O (current, voltage, digital). Such a system - where the real controller is controlling a simulated process - is denoted Hardware-in-the-loop (HIL) simulation. HIL-simulation is used in many industries, e.g. automotive industry for testing clutch automation systems and in marine and aircraft industry to test autopilots of vessels.

The Figure below illustrates the principle of testing a control system by replacing the physical system (or process) to be controlled by a simulated system. The controller is assumed to be a PID controller, but the figure applies to any controller function.

![Diagram of Hardware-in-the-Loop Simulation](image)

**Simulated Process**

In this example a mathematical model of the following small-scale process is used (“Air Heater”):
The mathematical model is:

\[ \dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u (t - \theta_d) + T_{env}] \} \]

Where:

- \( u \) [V] is the control signal to the heater.
- \( \theta_t \) [s] is time-constant.
- \( K_h \) [\( \text{deg} \, \text{C} / \text{V} \)] is the heater gain.
- \( \theta_d \) [s] is the time-delay representing air transportation and sluggishness in the heater.
- \( T_{env} \) is the environmental (room) temperature. It is the temperature in the outlet air of the air tube when the control signal to the heater has been set to zero for relatively long time (some minutes).

**Hardware**

The main purpose with the HIL Simulation is to test the hardware device on a simulator before we implement it on the real process.

In this we use an ordinary industrial PID controller, such as Fuji PGX5.
We will test the Fuji PGX5 PID controller on a model, and if everything is OK we will implement the controller on the real system.

We will use LabVIEW in order to implement the HIL Simulation. LabVIEW is a graphical programming language from Nation Instruments, and it is well suited for such implementation.

**Procedure**

The procedure is as follows:

4. **PID Control and Simulation in LabVIEW (Software only).** Simulate the model and implement the built-in PID controller in LabVIEW. No hardware involved.

5. **Configure the Fuji PGX5 PID controller (Hardware only).** Configure and be familiar with the industrial Fuji PGX5 PID controller.

6. **HIL Simulation in LabVIEW (Software + Hardware).** Test your industrial Fuji PGX5 PID controller on your simulated process.

7. **PID Tuning (Software + Hardware).** Find proper PID parameters, etc. for the controller based on the model.

8. **Implement your hardware, i.e., the Fuji PGX5 PID controller on the Real Process (Hardware only).** Now that you have tested your Fuji PGX5 PID controller on the
simulated process, it’s time to implement it on the real process. Fine-tune PID parameters if necessary.

These tasks follow the main idea with a HIL simulation. First step is to simulate your system in software. Next is to test your hardware on the simulated process. Finally you implement your hardware on the real system.

Below we see the difference between a traditional process system using a software program for implementing the control system and a HIL simulation.

**Traditional process system using a software program for implementing the control system:**

![Diagram of traditional process system](image)

In this case you need to scale the voltage signal you get from the process and the DAQ to a temperature value (1 − 5V → 20 − 50°C).

**HIL Simulation:**

![Diagram of HIL simulation](image)

In this case you need to scale the temperature value you get from the simulated process before you send the value to the Fuji PGX5 PID controller (20 − 50°C → 1 − 5V).

**HIL Simulation in LabVIEW**
Below we see an excerpt of the program created in LabVIEW:

In the example we have used a “Simulation Loop” in LabVIEW, but an ordinary While Loop may also be used. The model is implemented in a Simulation Subsystem.

**PXG5 PID.vi:**

Inside the SubVI “PXG5 PID.vi” is the I/O from and to the PXG5 PID controller implemented using an ordinary DAQ device (NI USB-6008 USB DAQ device), i.e., the simulated process value needs to be sent to the controller and the manipulated value from the controller need to be sent to the simulated process. Scaling is also implemented in this SubVI.

Below we see the “PXG5 PID.vi”:

**Mathematical Model:**

In the Model – Air Heater.vi simulation subsystem is the mathematical model implemented as shown below:
Results:

The simulation results become:

The Set Point (SP) is set on the PXG5 PID controller (in this case 30°C at time $t = 2s$). The simulation is based on PID parameters set on the PXG5 PID controller using the built-in Auto-tuning functionality that the PXG5 PID controller has.
References

The following references have been used in this document:

- PrecisionMBA (2011) - http://www.precisionmba.com/hardware_in_the_loop.htm
Hans-Petter Halvorsen, M.Sc.

E-mail: hans.p.halvorsen@hit.no
Blog: http://home.hit.no/~hansha/

University College of Southeast Norway
www.usn.no