A Framework for Real Time Hardware in the Loop Simulation

Abhishek Shukla, Pandit Ishan

Abstract

Hardware in Loop Simulation is a technique which is increasingly used in automotive industry for testing and development of complex real time embedded systems. Not only validation of correct behaviour when the entire electric system is fault free must be performed, but also validation during failure. How shall the electrical system behave with sensor broken or some other fault present? Fail-safe is a term used to describe that something is safe during failure. To maintain a failsafe vehicle extensive testing has to be done to verify that functionality is never becoming dangerous.

Keywords: **Hardware in Loop (HIL), Embedded Systems, Simulation, MATLAB/SIMULINK, dSPACE, ECU.**

Introduction

Hardware-in-Loop is a form of real time simulation. Hardware-in-Loop differs from pure real time simulation by the addition of a "real" component- ECU in the loop. The current definition of Hardware-in-Loop says that the plant is simulated and ECU is real. ECU is a physical unit with number of sensors and actuators attached to it. The purpose of HIL system is to provide all of the electrical stimuli needed to fully exercise the ECU. In effect, "fooling" the ECU into thinking that it is indeed connected to real plant.

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."

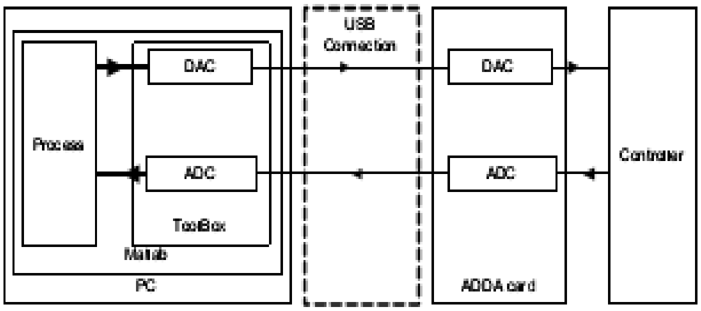

Figure 1.Hardware in the Loop Simulation Diagram

HIL testers provide the engineer with the ability to develop test cases that exercise the ECU as if it were in the end product, whether that product is an automobile engine or a medical device. The HIL system provides programmable analog and digital I/O along with a real-time deterministic platform that ensures the ECU reacts as if it were in the end. In addition, these systems include the ability to provide real life loads or simulated loads, and the ability to insert faults in the system programmatically. HIL systems also offer testing of failure modes that are unsafe or impractical to test in the real world.

How HIL works?

An HIL simulation must include electrical emulation of sensors and actuators. These electrical emulations act as the interface between the plant simulation and the embedded system under test. The value of each electrically emulated sensor is controlled by the plant simulation and is read by the embedded system under test. Likewise, the embedded system under test implements its control algorithms by outputting actuator control signals. Changes in the control signals result in changes to variable values in the plant simulation. The ECUs are connected to FIU (Failure Insertion Unit) which simulates short circuits or open circuits on sensors and actuators.

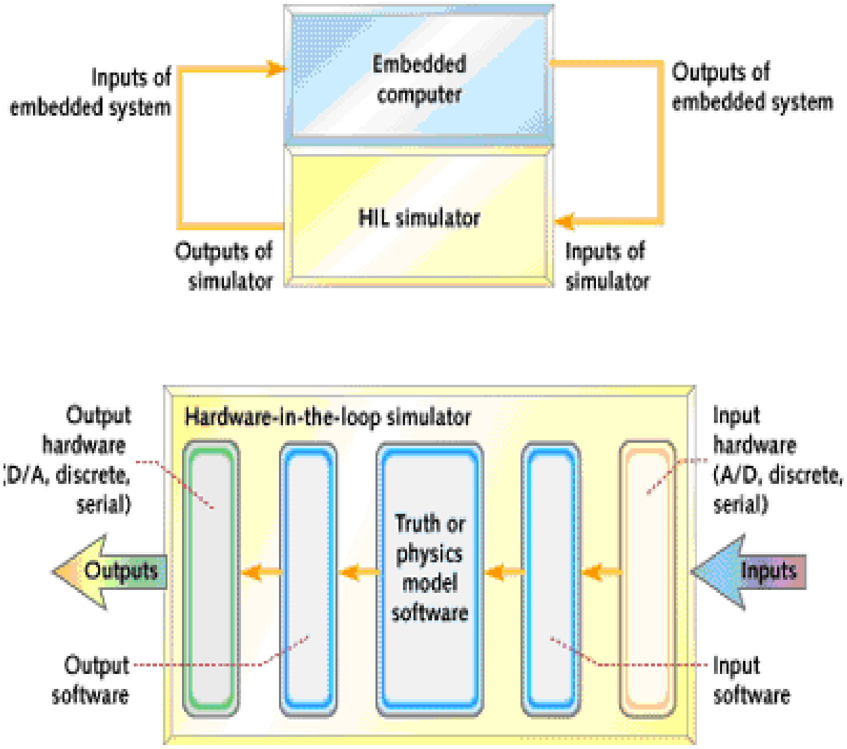
Inputs of embedded system
Embedded computer
Outputs of embedded system
HIL simulator
Outputs of simulator
Inputs of simulator
Hardware-in-the-loop simulator
Output hardware (D/A, discrete, serial)
Input hardware (A/D, discrete, serial)
Outputs
Truth or physics model software
Inputs
Output software
Input software

Figure 2.HIL Simulator

Advantage is that everything can be run from a computer, meaning that complete vehicle and functionality can be tested using a control panel in a computer. Since everything can be controlled, automated test can be run using test script written in programming language like python.

Why do we use Hardware in Loop simulation?

This topology was devised to manage the ever-increasing complexity of ECUs. Open loop or stimulus boxes were no longer suited to testing ECUs. Reason is the ECUs were closing dynamic controls loops to manage, say, fuel mixture. Open loop stimulus-boxes cannot test dynamic closed loops. Second, automated methods were needed to test and verify all the features of the ECU. Hardware-In-the-Loop systems typically have the ability to automatically run through tests automatically.

Block Diagram of HIL

The Major components of HIL are HIL simulator, ECU under test and Host computer on which application is running. Host computer and HIL simulator are connected through CAN bus and optical serial connector.

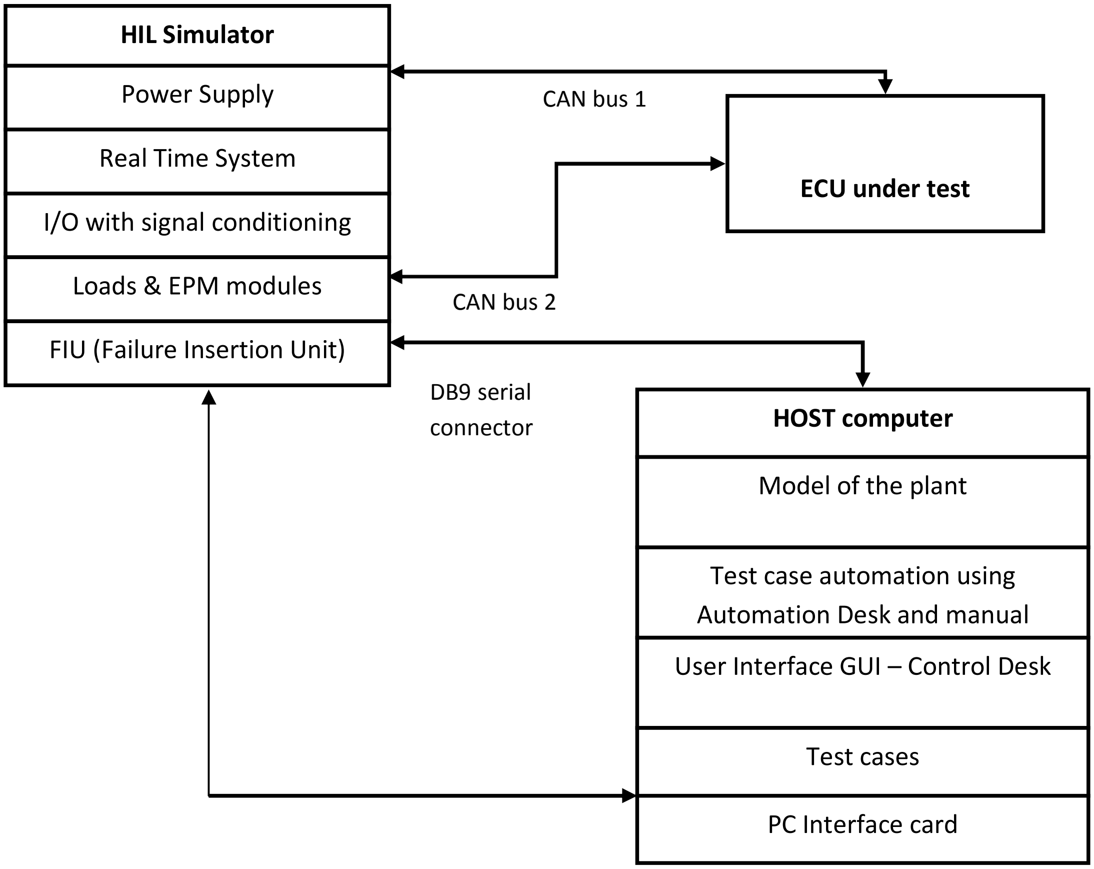

Figure 3.HIL Setup Block Diagram

ECU is Electronic Control Unit and EPM is Electronic Power Module.

Host Computer

The dSPACE Full-Size Simulator is hosted by a PC. The PC needs model of the plant created using SIMULINK which serves as a basis for real time application. In this model all inputs/ outputs/CAN messages required from other controllers are modeled to test prototype ECU. The MATLAB/SIMULINK model will access the base boards and I/O boards of simulator to stimulate I/O of other ECUs. This real time interface between Simulator and Matlab-Simulink development software is done using few proprietary Real time Drivers/modules. The real time library is used to configure all the base boards and I/O boards of the simulator. Depending upon the complexity and number of ECUs in the loop, the HIL simulator boards may have a multiprocessor/multicore system with abundant memory in it.

The Real Time Workshop (RTW) module of Matlab/Simulink generate,

1. An optimized ‘C’ code.
2. Invokes the complier to create an object file from generated ‘C’ code.
3. Loads the application to Simulator hardware board (C code generated from model) automatically.

Test cases are made in excel sheet following a standard format. Test cases contain inputs to the ECU, outputs from the ECU and CAN parameters sent across the bus from different controller. Tools, dSPACE Automation Desk and dSPACE Control Desk can access model’s input/output variables. In Automation desk, a script is made in python language to run bunch of the test cases made in excel sheet. Control desk provides GUI panels which is helpful for manual testing.

HIL Simulator

The real-time computation system is comprised of a PowerPC based processor board. The I/O boards are connected to the processor board using a Peripheral High Speed bus. HIL simulator has its own power supply which provides power all on board modules. E.g. FIU modules, on board processor, signal conditioning circuits. Real time system is present along with processor which takes care of real time simulation. Various signal conditioning modules are used to interface the I/O to external signals from and to the ECU. These signal conditioning modules provide protection for the I/O Boards. The Load and Failure Insertion System is comprised of Load Modules and FIU Modules, along with dedicated heat sinks to mount high-power loads on. These FIUs are nothing but input/output replay switches and various resistances in it. Integrated FIU module allows the following faults for all ECU outputs and inputs (simulator inputs and outputs): Open wire, short to GND, short to Battery and short between ECU pins. Current measurement is also included to monitor PWM Driver outputs of the ECUs. From model developed on host computer, code is generated which is dumped into the HIL Simulator. HIL Simulator is behaving as a plant and an ECU for the target ECU. It emulates input to the target ECU and output from the target ECU. HIL Simulator also emulates EPM. In physical system each ECU contains EPM to deal with high current input/output requirements.

CAN Network

There are two can networks. HIL Simulator and target ECU communicates with each other through CAN bus 1. Target ECU and EMP block of the HIL Simulator communicates with each other through CAN bus 2.

Host PC Interface

The Host PC connects to the dSPACE Simulator using a PC interface card. The dSPACE software running on the Host PC uses this interface to configure and control the dSPACE Simulator Hardware and also to exchange data between the Host PC and the Simulator. The only exception to this interface is the Fault Insertion Unit (FIU). Control of the FIU is done through the PC’s serial port. With the help of FIU all the Diagnostic Trouble Codes (DTC) are tested.

Importance of HIL simulation

To meet challenges of future Automotive applications, Vehicle manufacturers are adopting ECU architectures because they are more reliable, more efficient, add less weight to a vehicle, and can offer more functionality.

The increased complexity inherent in these systems works against business pressures to reduce time-to-market and development cost. Fortunately, HIL simulation has proven to be a practical solution to these diverging challenges. While HIL simulation does not replace the need for physical testing, it does help engineers accomplish the following by enabling tests earlier in the development cycle and eliminating the need for a physical system during test.

Noticeable advantages of using HIL

- Follow tight development schedules: The tight development schedules associated with most new automotive, aerospace and defense programs do not allow embedded system testing to wait for a prototype to be available. In fact, most new development schedules assume that HIL simulation will be used in parallel with the development of the plant. For example, by the time a new automobile engine prototype is made available for control system testing, 95% of the engine controller testing will have been completed using HIL simulation.
- High burden rate plant: In many cases, the plant is more expensive than a high fidelity, real-time simulator and therefore has a higher-burden rate. Therefore, it is more economical to develop and test while connected to an HIL simulator than the real plant. For example Jet engine manufacturers, HIL simulation is a fundamental part of engine development. Each jet engine can cost millions of dollars. In contrast, an HIL simulator designed to test a jet engine manufacturer’s complete line of engines may demand merely a tenth of the cost of single engine.
- Make educated design decisions: Evaluate specific design alternatives early in the development process based on test data.
- Test earlier in the development process: Identify design errors earlier when they are less expensive to correct and have smaller impact on time to market.
- Reduce testing cost and possibility of risk: Testing can be done without damaging equipment or endangering lives. Extreme conditions can be simulated and tested which in turn detect and report errors. HIL reduces capital, repair, and maintenance expenses for test fixtures without the need for a physical system.
- Achieve greater testing capacity: It enables "lights-out" testing of systems where unmonitored physical testing is not possible.

- Increase test coverage: Test ECUs under extreme conditions that might not be practical for physical testing due to safety or equipment damage concerns. Testing of all possible input combinations is easy with writing test cases.
- Increase test flexibility: Simulate winter road conditions for a vehicle under test even in the heat of summer.
- Increase test repeatability: Isolate deficiencies in an ECU even if they occur only under certain circumstances.

References

1. Fernando Henrique Ataide and Fabiano Costa Carvalho and Carlos Eduardo Pereira and Max Mauro Dias Santos. An overview of Hardware-In-the-LoopTesting Systems at Visteon. SAE International, 1 edition, 2005. 2005-01-4144.

2. Mauro Velardocchia and Aldo Sorniotti. Hardware-In-the-Loop to EvaluateActive Braking System Performance. SAE International, 1 edition, 2005.2005-01-1580.

3. Chiu-Feng Lin, Chyuan-Yow Tseng and Tsai-Wen Tseng, "A hardware-in-the-loop dynamics simulator for motorcycle rapid controller prototyping", Control Engineering Practice, Volume 14, Issue 12, December 2006, Pages 1467-1476

4. Guenter Randolf and Robert M. Moore, “Test system design for Hardware-in-Loop evaluation of PEM fuel cells and auxiliaries” Journal of Power Sources, Volume 158, Issue 1, 14 July 2006, Pages 392-396.

5. Zijad Lemeš, Andreas Vath, Th. Hartkopf and H. Mäncher, “Dynamic fuel cell models and their application in hardware in the loop simulation", Journal of Power Sources, Volume 154, Issue 2, 21 March 2006, Pages 386-393

6. Darcy Bullock, Brian Johnson, Richard B. Wells, Michael Kyte and Zhen Li, "A hardware-in-the-loop simulation", Transportation Research Part C: Emerging Technologies, Volume 12, Issue 1, February 2004, Pages 73-89