

# Review on Hardware-in-Loop Simulation used to Advance Design Efficiency and Test Competency

Prajakta Patil<sup>1</sup>, Snehal Bhosale<sup>2</sup>

<sup>1,2</sup> Department of E & TC, R.M.D Sinhgad School of Engineering, Pune University, Pune 411052, India

**Abstract:** *In the Automotive Industry, product management has never been an easy task. It is becoming more complex because of increasingly intricate vehicle functionalities; lack of fool-proof verification and validation processes; inability to perform extensive validation of features versus actual vehicle performance; non access to actual vehicle for verifying performance characteristics. The challenges are going to increase with hardware obsolescence / migration. Automotive electronic systems are becoming truly distributed to achieve increased system functionality by tightly coupling Electronic Control Units (ECUs). These changes encourage a significantly revised approach to automotive system test. The fundamental functional design approach has been modular and ECU-centric, but the ECU count has steadily increased. The next big shift is to achieve functionality through the integration of multiple ECUs. Nowadays, model-based formal verification approaches are mostly used for benchmarking automotive electronic systems. However, these approaches unable to check the system for bus errors, faulty hardware configurations and compatibility problems efficient. Solution for this is the Hardware-in-the-Loop (HiL) test. HiL allows testing embedded software or prototypes or final electronic components, to test design against a real time dynamic environment without the real plant. This paper provides a brief overview of the HiL simulation and its applications.*

**Keywords:** Electronic Control Unit (ECU), Hardware in the Loop (HiL), Model in the Loop (MiL), Software in the Loop (SiL), Driver-Vehicle-Environment (DVE).

## 1. Introduction

The automotive industry is highly competitive and vehicle manufacturers are facing the constant challenge of meeting increasingly short development times whilst maintaining a high standard of quality and safety. In addition to this, the complexity of modern vehicles has increased greatly in recent years with electrical/electronic systems replacing many traditionally mechanical and pneumatic systems and providing new functionality for the user. A typical vehicle contains more than 100 Electronic Control Units [7]. Now, the design and implementation of control algorithms is a critical element in the development of automotive embedded systems [8].

Therefore integration of testing, debugging, and process simulation tools is necessary for software verification against vehicle models [2]. Solution for such optimization of the test process with respect to testing depth and incurred costs is application of automated testing in model in-the-loop, software in-the-loop, and hardware in-the-loop simulations.

The idea behind Hardware-in-the-Loop testing is to simulate a control system in use as close as possible to its real environment without actually installing it on hardware. Hardware-in-the-loop (HiL) testing is the combination of physical and virtual prototyping (modeling and simulation). Hardware-in-the-Loop (HiL) testing systems involve an essential contribution to quality assurance during the early phases of ECU development. By simulating the model components known as Driver-Vehicle- Environment (DVE), they simplify the testing of the functions or diagnostic behavior of ECUs in the laboratory. This makes it possible to run reproducible and wide-ranging tests of virtually any driving situation with the added benefit of extensive test coverage through automation entirely without hazard to

driver and vehicle. HiL contributes prospects to test functionality, performance, robustness and other important features in the system and allows checking of software errors, bad configurations and other problems with the code. To uncover any such faults is the main goal of the HiL system. We can do this in safe and real time environment. HiL Testing is possible without prototypes and also in early development cycle. Automated testing can be done with HiL testing.

## 2. Software Challenges

Today almost all advanced control systems are designed and run on a computer, and parts like as the controller and dynamic allocation are implemented as advanced algorithms. This leads to large and complex computer programs that can be hard to interpret and understand without proper documentation.

A complete control system will usually include software from a wide range of different vendors; each designed to support a specific component, but not necessarily made with thought of optimal communication with the other components in the system. Faulty communications between different software and hardware might lead to safety- and operation dangerous situations and the risk of getting such errors rise as the software grows more complex and are subject to frequent updates, often performed by the individual vendors without much knowledge of the other parts of the system.

Software is extremely susceptible to faults. In a mechanical system small variations might occur without degrading the performance of the system in a noticeable degree, but this is not the case with software, where even the slightest error might cause entire systems to crash. Making a software system reliable and knowing how it will interact with other

software and hardware is always difficult.

There are listed several reasons why software failure occurs:

- Logical / semantic errors in the code;
- Not a clear standards i.e. Tailor made systems;
- Configured on tight time schedule late in the projects;
- Complex control systems;
- Problems with integration of systems from different vendors;
- Frequent updates;
- Unaccounted for scenarios;

Redundancy in software is also a challenge. It seems quite clear that as an increasing number of critical systems rely on non-redundant software. The best way to prevent any potentially harmful failures in the software is to test it thoroughly in its supposed environment. This is what HiL testing is striving to achieve with a simulated environment providing full feedback and environmental forces to test the software on its designated hardware.

### 3. Difference between the HiL and Standard Software testing process

Testing is one of the fundamentals in modern software development, and such programming methods rely on writing test before writing the actual code to highlight the importance of testing the code's agreement with the desired results. This kind of testing only concerns the functionality of the isolated code. Testing the code of a number of different vendors together in conventional white-box testing will be difficult as most of the source code is closed source and secret.

HiL testing relies on black-box testing, meaning the tester knows nothing about the internal states of the system, but is only concerned with the inputs and generated outputs. This means it is tremendously important for the tester to do a thorough job feeding the test system all possible input values as this is the only way to know the response from the control system. As practically no knowledge of the inside system is required to carry out black-box testing it is ideal for checking the functionality of a multifaceted system from different vendors.

Another problem with common software testing and code analysis only uncovers errors in the topmost application layer of the software, while HiL also tests fulfillment with other aspects of the software such as I/O, drivers, memory and CPU management and hardware compatibility.

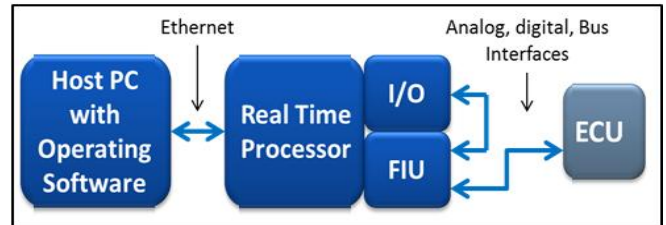
### 4. Basic HiL Architecture

Mainly the Hardware-in-the-loop test system consists of three primary components as per following:

1. Real-Time Processor is the heart of the HiL test system. With this target highly dynamic physical plant models and control processes can be precisely simulated in real-time. It provides execution of most of the HiL test system modules such as hardware I/O communication, data recording and stimulus generation.
2. I/O Interfaces are mainly used for signal generation and

calibrating. Numerous automotive-capable I/O cards are available (Ex. Analog I/O, PWM, angle synchronous signal generation and bus systems such as CAN, FlexRay, LIN, etc.) which interact with unit under test.

3. Operating software provides interface for user to communicate with the real-time processor which provides test commands and visualization. They also provide configuration management, test automation, analysis and reporting tasks.

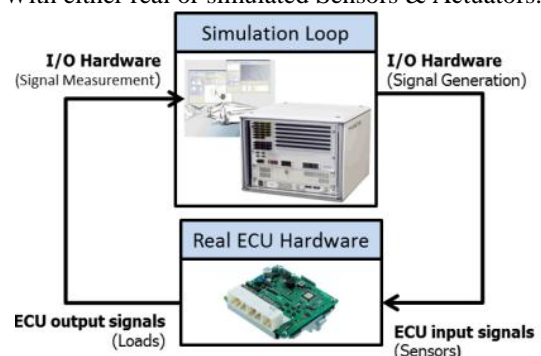


**Figure 1:** Basic HiL Architecture

In hardware-in-the-loop simulation (HiL simulation), the behavior of the vehicle is simulated by software and hardware models. Real vehicle components (real parts) are then connected, via their electrical interfaces, to a simulator, which reproduces the behavior of the real-time environment. Fig. 1 shows the basic design of HiL systems. Instead of being connected to an actual vehicle, the ECU to be tested is connected to a simulation system. This runs a model of the vehicle process and associated sensors and actuators that will usually have been developed and implemented with suitable modeling tools such as MATLAB / Simulink.

Using the HiL simulation we can do following tests,

1. Hardware Test: Current/voltage range tests, overvoltage & under-voltage tests, Standby Current, Endurance test, EMV test, Electrical Failure Simulation, etc.
2. Basic Software Test: Communication test in CAN, FlexRay, LIN buses, Wakeup, Sleep mode, Response tests, Diagnostic tests, Runtime tests (like Delays & Timing), Performance tests and Stress tests.
3. Component Test: Single ECU Functionality test (Ex. Power Train Engine ECU test, Closed loop test with either Real or Simulated Loads, Sensors and Actuators)
4. Integration Test: In Distributed ECU Functionality test we can analyze entire ECU network or its subparts including bus simulation –
  - With either real or simulated Loads
  - With either real or simulated Sensors & Actuators.



**Figure 2:** Basic HiL System

## 5. Advantages of HiL Simulation

The growing popularity of HiL simulation shows many advantages, including:

1. Open, Modular architecture: Scalable for easy, project-specific functional adaptation.
2. Control algorithms can be tested in the early stages of development, even before vehicle prototyping stage. Therefore, electronics system reaches a high degree of maturity very early, provides a better-quality starting point for future development stages.
3. Integrated signal conditioning for easy reuse of existing configurations across different projects.
4. Simultaneous development of ECU function and real vehicle model is possible with HiL simulation only which cuts the time needed for development.
5. Simulation speed of HiL simulations is faster than purely virtual simulations of the same phenomena.
6. With HiL it is possible to test embedded controllers under extreme conditions that might not be practical for physical testing due to safety or equipment harm concerns. Thus typical winter test drives under low- $\mu$  conditions (snow and ice) can be carried out in lab and Cold-Start tests can be executed repetitively.
7. HiL simulation has non-destructive nature. Some errors and failures that might have devastating effects in a real system (sensor failures, line breaks, ground errors, error frames on the CAN bus etc.) can be simulated and tested systematically.

## 6. Applications of HiL Simulation

HiL testing is used in an extensive range of industries today such as Aerospace, Avionics, Automotive & Nuclear with growing interests, it outcomes new uses. HiL testing was first introduced by the safety critical software systems of aerospace industry, encouraged by the growing pressure to develop safer systems in shorter time duration while at the same time reducing costs; this is noted in [7]. According to [8], 90% of all faults in the automotive industry found using a traditional test drive were also found using HiL-simulations while spending much less time and money than the physical tests. One of the advantages with HiL testing is its usability in all stages of a system's life-cycle. Many other testing techniques are appropriate for specific time, while HiL tests easily can be performed to test new functionality and analyze problems in an aging system.

## 7. Summary and Conclusion

In Automotive industry, Hardware-in-the-Loop simulation has been effectively developed and verified to be very efficient and effective in several autonomous driving circumstances in features, functions and algorithm development, testing and verification. It also offers the realistic and efficient environment to debug and validate system architecture and communication hardware and software for control processors, sensors and actuators.

With virtual world sensors, the HiL simulation can be extended to simulating multiple vehicles networking with one

another under traffic, and with surrounding environment. This makes the simulation feasible for various dangerous road and traffic conditions, software and hardware failures, and fault estimation.

Under the same architecture as implemented in prototype vehicle, the HiL simulation, with the same processors, actuators, sensors and other hardware devices in the loop, has proved to be especially beneficial for developing, testing, debugging and verifying a large and complex system under lab setup. Thus it will shorten the control development time and reduce the cost.

## References

- [1] Alberto Sangiovanni-Vincentelli, Marco Di Natale, "Embedded System Design for Automotive Applications", Published by the IEEE Computer Society, p.p. 0018-9162/07, Oct. 2007 (Journals style)
- [2] Klaus Lamberg, Peter Wältermann's "Using HiL Simulation to Test Mechatronic Components in Automotive Engineering," 2nd congress on 'Mechatronic in Automobile', 15/16 Nov. 2000. (Technical Report style)
- [3] N. Schetinin, N. Moriz, B. Kumar, A. Maier, S. Faltinski, and O. Nigge-mann, "Why do verification approaches in automation rarely use HiL-test?" Industrial Technology (ICIT), 2013 IEEE International Conference on, 1:1428-1433, 2013.
- [4] D. Maclay, "Simulation gets into the loop", IEE Review, 43:109-112, 1997.
- [5] Chris Washington, Santiago Delgado, "Improve Design Efficiency and Test capabilities with HiL Simulation", IEEE AUTOTESTCON, p.p. 978-1-4244-2226-5, 2008
- [6] Hosam K. Fathy, Zoran S. Filipi, Jonathan Hagena and Jeffrey L. Stein, "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area", Modeling and Simulation for Military Applications, Proc. of SPIE Vol. 6228 62280E-1, 2006
- [7] Praveen Kumar and Jegan.A, "Systematic Approach in V-Model Development Cycle for an Automotive Embedded Control System" Advance in Electronic and Electric Engineering, ISSN 2231-1297, Volume 3, Number 4 (2013), pp. 465-470
- [8] Mark Dooner, Jihong Wang, Alexandros Mouzakitis, "Development of a Simulation Model of a Windshield Wiper System for Hardware in the Loop Simulation", 19th International Conference on Automation & Computing, Brunel University, Uxbridge, UK, 13-14 September 2013