

Current Approaches in HiL-Based ADAS Testing

Marius Feilhauer and Juergen Haering
ETAS GmbH

Sean Wyatt
ETAS Inc.

ABSTRACT

The way to autonomous driving is closely connected to the capability of verifying and validating Advanced Driver Assistance Systems (ADAS), as it is one of the main challenges to achieve secure, reliable and thereby socially accepted self-driving cars. Hardware-in-the-Loop (HiL) based testing methods offer the great advantage of validating components and systems in an early stage of the development cycle, and they are established in automotive industry.

When validating ADAS using HiL test benches, engineers face different barriers and conceptual difficulties: How to pipe simulated signals into multiple sensors including radar, ultrasonic, video, or lidar? How to combine classical physical simulations, e.g. vehicle dynamics, with sophisticated three-dimensional, GPU-based environmental simulations?

In this article, we present current approaches of how to master these challenges and provide guidance by showing the advantages and drawbacks of each approach. Therefore, we discuss different ADAS setups and show ways of how to implement HiL test benches for these. We discuss two categories: 1) Hardware level: we focus on the communication structure between the simulated plant model and the Unit under Test (UuT). We show possible interfaces into the sensor units and involved bus systems. 2) Software level: we focus on how to provide the data the UuT expects. This results in rendering images, creating data lists or providing ray-tracing based point clouds.

This article provides solutions for current and up-coming challenges when dealing with HiL-based validation of ADAS and presents an overview of current test-approaches.

CITATION: Feilhauer, M., Haering, J., and Wyatt, S., "Current Approaches in HiL-Based ADAS Testing," *SAE Int. J. Commer. Veh.* 9(2):2016, doi:10.4271/2016-01-8013.

INTRODUCTION

The development process in automotive industry is dominated by the V-model. Here, all constructive stages on the left hand side have a corresponding counterpart testing its outcomes on the right hand side. Different testing methods to verify and validate functionalities along all stages are applied and are summarized to X-in-the-Loop (XiL) methods. These testing methods enable short feedback cycles and [figure 1](#) visualizes their usage along the V-Model.

Model-in-the-Loop (MiL) settings test a model of the functionality to be developed. MiL is applied in early stages of the development process to verify basic architectural decisions, design choices or elementary implementations. In a later step, Software-in-the-Loop (SiL) setups test the functionality of the program code, mostly compiled for the target hardware, but without including real

hardware. Hardware-in-the-Loop (HiL) methods verify and validate the software implemented in the physical target Electronic Control Units (ECUs). Therefore, HiL test benches need to provide realistic real-time stimuli and load simulation to the ECUs to let them perform as if they were built into a real vehicle. Multiple ECUs are tested additionally in system and integration tests, e.g. network HiL setups. Thereby, tests of the complete electrical setup of a vehicle including its bus systems, sensors and actuators are possible. The more recent approach of Vehicle-in-the-Loop (ViL) replaces the vehicle simulation (and therewith most of the virtual ECUs, too) by a physical car. This approach enables e.g. safety critical ADAS tests by simulating and crashing virtual obstacles while still driving the real vehicle. The simulated parts of the environment are injected into the vehicle's sensors or ECUs. [1]

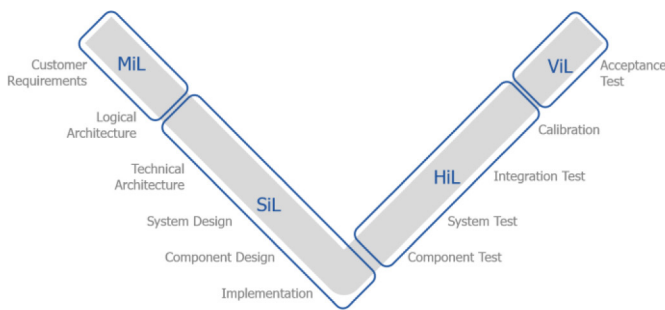


Figure 1. X-in-the-Loop methods in the V-Model [1]

All mentioned test methods have “*in-the-Loop*” in common, which means that the reaction of the UuT influences the plant simulation. In respect to ADAS, this feedback is necessary when e.g. testing a Lane Change Assist (LCA): Here the influence of the ECU’s control signals on the simulated vehicle dynamics is in focus.

Another way of testing ADAS are Open-Loop tests. In the following, the concept is illustrated using the example of regression testing. If a vehicle project is shortly before start of production, meaning all hardware sensor layouts and their locations within the vehicle are fixed, testing the car in real life scenarios is inevitable. Many aspects of these test runs are random, as e.g. weather conditions or the behavior of other traffic participants are neither foreseeable nor reproducible. Nevertheless, these tests are necessary in the scope of Advanced Driver Assistance Systems development. In reality, during testing, all sensor data has to be recorded. If an unwanted behavior occurs, involved algorithms can be adjusted and afterwards be tested using the recorded data as regression tests. This is done by piping the recorded data through the algorithms again and observing the effects of the change of the algorithms in all recorded scenarios again. The Automotive Data and Time-Triggered Framework (ADTF, [7]) is a software framework often used within the automotive industry for such tasks. As an example, Figure 2 visualizes these Open-Loop regression tests for the example of a road-sign recognition function schematically.

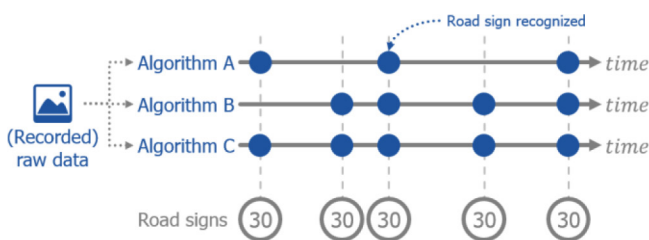


Figure 2. Open-Loop Algorithm Regression Testing

The illustrations shows how sensor data is piped through *Algorithm A*, which is not recognizing all road signs. *Algorithm A* might have been used within a test run in reality. With *Algorithm B* the engineers want to improve the road sign recognition, mainly focusing on the not recognized scenes. When now using the recorded raw sensor data to test *Algorithm B*, it becomes obvious that it is not perfect, too. Within another iteration, the data processing is further improved, resulting in *Algorithm C* recognizing all road signs on the available data.

The main benefit of such an Open-Loop approach is the realistic data it is based on. On the other hand, Closed-Loop setups are advantageous when investigating a function’s feedback loop. Here the power of Closed-Loop setups lies in the possibility to modify the stimuli data in real-time and therewith enabling to evaluate how the function affects the overall ADAS system. For example, the impact of starting a lane change earlier can be investigated in a Closed-Loop setup, which is not possible when using immutable recorded data in an Open-Loop setup.

To realize Closed-Loop ADAS HiL test benches, the simulated plant models consists of three components:

- *Vehicle Simulation including a Driver Simulation*
This simulation component consist of a driver able to follow a road or predefined trajectory, a powertrain simulation representing engine and drivetrain, and a three dimensional vehicle dynamics simulation to characterize the vehicle’s (truck and trailer) movement.
- *Environment and Sensor Channel Simulation*
The perception of the vehicle’s environment is a core component in autonomous driving functionalities. Therefore, the environment model is part of the simulation. The model contains a three-dimensional representation of the vehicle’s surrounding including e.g. a road network, traffic participants or buildings. This virtual world enables to simulate the sensor channel to provide inputs to the sensors, e.g. a video picture to a video camera.
- *Traffic Simulation*
Many ADAS functions interact with other traffic participants. This is why these entities are part of the simulation. Here, simplified simulation models, e.g. for vehicle dynamics, can be used as long as a realistic and deterministic behavior of all entities is ensured.

When integrating these plant models to an overall plant simulation, connecting it to the sensor interfaces (i.e. stimuli) of the ADAS hardware, and feedback the ECU reaction to the simulation, the loop is closed. Figure 3 visualizes this generic ADAS HiL architecture.

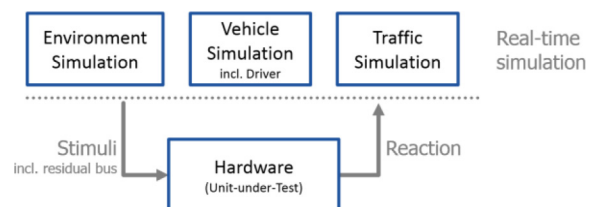


Figure 3. Generic HiL architecture for ADAS testing

This setting will be used in the next section to explain how to integrate ADAS hardware in various test setups.

ADAS HiL ARCHITECTURE

To evaluate ADAS within a HiL context one of the first questions arising is what the Unit-under-Test and its interfaces are. There might be a single ECU covering the core functionality, e.g. a Radar-based

Emergency Brake Assist (EBA). Another setup might include a dedicated and centralized ADAS ECU, implementing sensor fusion and control algorithms and receiving sensor-information from additional control units. Depending on the setup, multiple real ECUs have to be included in the test bench and connected via automotive network systems, like CAN, FlexRay, or Automotive Ethernet. Additional ECUs have to be available virtually to provide relevant supplementary information.

Once the basic test bench setup was identified, a suitable simulation environment for the Unit-under-Test has to be defined. In addition, one of the main challenges when testing ADAS in the context of HiL has to be solved, i.e. **how to inject the stimuli into the ADAS ECU**. A core component of autonomous driving functionality is the perception of the vehicle's environment via internal sensors. If the HiL test bench includes real sensor systems, the simulated environmental data, like e.g. video images, has to be injected into the hardware components. There are many different possibilities where data gets inserted into the ECU, which results in testing different parts of the UuT.

Figure 4 visualize a simplified high-level ADAS architecture ("ADAS Pipeline") that can be found in many ADAS systems. This architecture defines potential interfaces to inject data into the ECU.

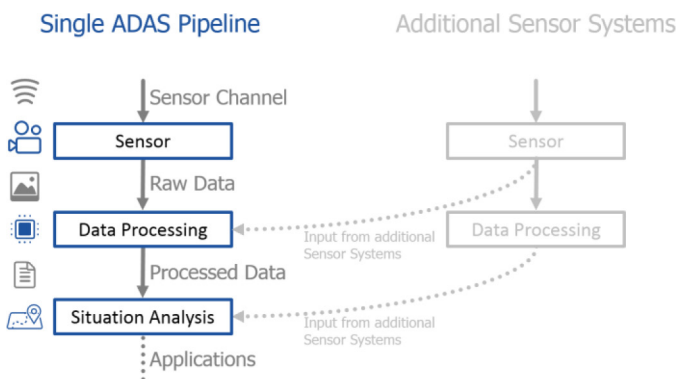


Figure 4. High-level function pipeline of advanced driver assistance systems

The main interface to the environment are the vehicle's sensors. Stereo and mono video cameras, ultra-sonic, radar or lidar sensor systems are the most relevant sensor concepts used for ADAS. Combining different sensor concepts with different characteristics helps to gather as many aspects of the surrounding as required to safely analyze the current traffic scenario. The sensor captures information from its sensor channel. The raw data is then further processed by e.g. feature and object recognition algorithms, resulting in a processed data set, like a grid based map representation of the environment or a list of recognized objects. The situation analysis combines this processed information and estimates the current traffic situation, forwarding the data to the ADAS application. The application finally decides which actions to take, like e.g. triggering an emergency brake.

In a HiL test bench the accessible layers for injecting virtual data in the Unit-under-Test might vary. Based on the presented pipeline, the following sections discuss different possibilities to inject the data.

Hardware Level

Injecting Data into the Sensor Channel Layer

The most complete and complex test bench includes multiple real sensors and their connection to a (centralized) ADAS ECU. This requires the simultaneous injection of multiple physical signals in each sensor channel. For a camera sensor, this results in injecting light in the sensor's spectrum, i.e. a sequence of images of the actual scenery. Figure 5 visualizes the principle of injecting a synthetically generated image into a camera sensor.

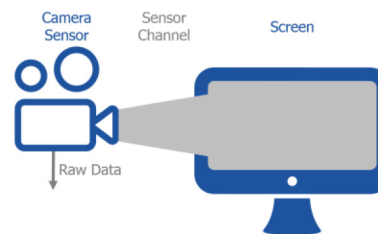


Figure 5. Schematic visualization of injecting a light signal

A screen or projector emits light and injects it directly into the camera's optics. By using this approach, the sensor is stimulated via its physical sensor channel. This approach is already well established.

A similar setup can be realized using a radar sensor. The measurement principle of radar is based on the echo-effect: Within a HiL setup, the radar signal does not have to be generated but the emitted signal must be modified. The available methods to modify these signals are not yet as sophisticated as in the spectrum of light, but the principle technology is already available (see [2]). Figure 6 visualizes this approach.

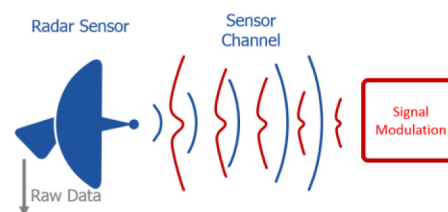


Figure 6. Schematic visualization of modulating a radar signal

The radar unit emits radars waves, which are modified in a way that the reflected signal represents the simulated scenery, e.g. a moving target object. As the functionality of ultrasonic and lidar sensor systems are based on the echo-effect, too, the same sensor-channel-based testing approach is conceivable.

When using the physical sensor channels in an ADAS HiL test bench there are several advantages and drawbacks: One of the main benefits is that there is no need to modify the involved UuT for testing purposes at all. The injected signal is based upon the same physical principles as the signal the vehicle's sensor system will be exposed to in reality. This is why the original housing and the standard data output of the ECU, e.g. via FlexRay, can be used for testing the complete unit. However, using the physical sensor channel to stimulate the UuT involves great challenges, too. A major problem is interference from the test surrounding. This is why such test benches must be setup within a sealed environment. No stray light must