

EXHIBIT 1037

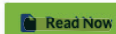
Simulation and Closed-Loop Testing of Camera, Radar, and Lidar Sensors for Highly Automated Verification and Validation of Data Fusion Systems

Jace L. Allen and Benjamin Hager

AIAA 2020-0895

Session: Modeling- and Simulation-Based Software Development and Verification II (Invited)

Published Online: 5 Jan 2020 • <https://doi.org/10.2514/6.2020-0895>



Abstract:

Aerospace environments consist of both static and dynamic objects both in the air and on the ground. Dynamic objects include aircraft, UAVs, ground vehicles, birds, and people. Static objects involve runways, runway markers, buildings, and any other scenery/objects. Detecting, classifying, identifying and tracking these objects is needed for collision avoidance algorithms deployed in current aircraft systems. Multiple sensors, such as cameras, radar, LIDAR, etc., need to be used for capturing the object's characteristics and location. Data fusion is used to combine data from these sensors. Advanced algorithm techniques are then employed for object identification. Detect-and-Avoid (DAA) systems are highly dependent on the performance of communications, navigation and surveillance (CNS) systems for detection, tracking and avoiding (DTA) tasks and maneuvers. To effectively detect objects, multiple high-performance, reliable and accurate avionics sensors and systems are used. This includes: non-cooperative sensors (e.g., visual and thermal cameras, Laser radar (LIDAR) and acoustic sensors) and cooperative systems [e.g., Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Collision Avoidance System (TCAS)]. It is important to utilize this sensor technology for object detection and identification to improve the integrity of these systems in all sorts of potential scenario situations and environmental conditions. This paper will explore how to enhance autonomous sensor testing capabilities and system quality assurance using completely automated Hardware-in-the-loop (HIL) and/or Software in the Loop (SIL) simulation environments that integrate autonomous sensor technology, such as cameras, RADAR, LIDAR, and other key technologies such as GNSS/maps and bus communications, such as ARINC or Ethernet. The key to performing such real-time testing is the ability to stimulate the various sensor Electronic Control Units (ECUs) through closed-loop simulation of the vehicle, its environment, traffic, surroundings, along with synchronization with vehicle bus and application data. These techniques are necessary for developing and implementing a complete System Verification and Validation (V&V) solution that can handle diverse types of perception systems in the loop. In order to handle the testing necessary for these types of systems, new technologies and effective techniques for optimization of test processes and certification relative to DO178C will be shown, along with examples of specific use-cases for development of aero-applications utilizing simulated scenarios. A cornerstone for testing quality and process certification is traceability, which is necessary in order to ensure the compliance of the development and testing processes. A system integration environment is introduced that can tie together the large amounts of common data, models, and test facets that a company faces when implementing these types of test systems. In order to effectively use the MBD tools in the development process, it is necessary to be able to manage this data and metadata in an efficient manner that relates directly to the actual tools and methods used. This test management capability also is critical when it comes to managing the scale of test environments – such as cloud-based testing, which can encompass hundreds to thousands of simulations/tests. The tools and system proposed in this paper is used for test process management and advanced test methodologies, as well as handling the traceability process for certification needs.



AIAA Scitech 2020 Forum

6-10 January 2020

Orlando, FL

<https://doi.org/10.2514/6.2020-0895>

Crossmark



Information

Copyright © 2020 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

Topics

- Aerodromes
- Aerodynamics
- Aeronautics
- Aerospace Sciences
- Air Traffic Management
- Aircraft Collision Avoidance Systems
- Airport Infrastructure
- Airports
- Automatic Dependent Surveillance
- Aviation
- Aviation Authorities and Organizations
- Aviation Meteorology
- Aviation Safety
- Avionics
- Runways
- Sensors
- Transducers

 INVITED PAPER[Meeting Paper Home](#) [For Authors](#) [AIAA SciTech Forum Home](#)

Radar, and Lidar Sensors for Highly Automated Verification and Validation of Data Fusion Systems

Jace L. Allen and Benjamin Hager

AIAA 2020-0895

Session: Modeling- and Simulation-Based Software Development and Verification II (Invited)

Published Online: 5 Jan 2020 • <https://doi.org/10.2514/6.2020-0895>

[ABOUT](#)

Abstract:

Aerospace environments consist of both static and dynamic objects both in the air and on the ground. Dynamic objects include aircraft, UAVs, ground vehicles, birds, and people. Static objects involve runways, runway markers, buildings, and any other scenery/objects. Detecting, classifying, identifying and tracking these objects is needed for collision avoidance algorithms deployed in current aircraft systems. Multiple sensors, such as cameras, radar, LIDAR, etc., need to be used for capturing the object's characteristics and location. Data fusion is used to combine data from these sensors. Advanced algorithm techniques are then employed for object identification. Detect-and-Avoid (DAA) systems are highly dependent on the performance of communications, navigation and surveillance (CNS) systems for detection, tracking and avoiding (DTA) tasks and maneuvers. To effectively detect objects, multiple high-performance, reliable and accurate avionics sensors and systems are used. This includes: non-cooperative sensors (e.g., visual and thermal cameras, Laser radar (LIDAR) and acoustic sensors) and cooperative systems [e.g., Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Collision Avoidance System (TCAS)]. It is important to utilize this sensor technology for object detection and identification to improve the integrity of these systems in all sorts of potential scenario situations and environmental conditions. This paper will explore how to enhance autonomous sensor testing capabilities and system quality assurance using completely automated Hardware-in-the-loop (HIL) and/or Software in the Loop (SIL) simulation environments that integrate autonomous sensor technology, such as cameras, RADAR, LIDAR, and other key technologies such as GNSS/maps and bus



Simulation and Closed-Loop Testing of Camera, Radar, and Lidar Sensors for Highly Automated Verification and Validation of Data Fusion Systems

Jace L. Allen¹ and Ben Hager²
dSPACE, Inc., Wixom, MI, 48393, USA

Aerospace environments consist of both static and dynamic objects both in the air and on the ground. Dynamic objects include aircraft, UAVs, ground vehicles, birds, and people. Static objects involve runways, runway markers, buildings, and any other scenery/objects. Detecting, classifying, identifying and tracking these objects is needed for collision avoidance algorithms deployed in current aircraft systems. Multiple sensors, such as cameras, radar, LIDAR, etc., need to be used for capturing the object's characteristics and location. Data fusion is used to combine data from these sensors. Advanced algorithm techniques are then employed for object identification. Detect-and-Avoid (DAA) systems are highly dependent on the performance of communications, navigation and surveillance (CNS) systems for detection, tracking and avoiding (DTA) tasks and maneuvers. To effectively detect objects, multiple high-performance, reliable and accurate avionics sensors and systems are used. This includes: non-cooperative sensors (e.g., visual and thermal cameras, Laser radar (LIDAR) and acoustic sensors) and cooperative systems [e.g., Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Collision Avoidance System (TCAS)]. It is important to utilize this sensor technology for object detection and identification to improve the integrity of these systems in all sorts of potential scenario situations and environmental conditions. This paper will explore how to enhance autonomous sensor testing capabilities and system quality assurance using completely automated Hardware-in-the-loop (HIL) and/or Software in the Loop (SIL) simulation environments that integrate autonomous sensor technology, such as cameras, RADAR, LIDAR, and other key technologies such as GNSS/maps and bus communications, such as ARINC or Ethernet. The key to performing such real-time testing is the ability to stimulate the various sensor Electronic Control Units (ECUs) through closed-loop simulation of the vehicle, its environment, traffic, surroundings, along with synchronization with vehicle bus and application data. These techniques are necessary for developing and implementing a complete System Verification and Validation (V&V) solution that can handle diverse types of perception systems in the loop. In order to handle the testing necessary for these types of systems, new technologies and effective techniques for optimization of test processes and certification relative to DO178C will be shown, along with examples of specific use-cases for development of aero-applications utilizing simulated scenarios. A cornerstone for testing quality and process certification is traceability, which is necessary in order to ensure the compliance of the development and testing processes. A system integration environment is introduced that can tie together the large amounts of common data, models, and test facets that a company faces when implementing these types of test systems. In order to effectively use the MBD tools in the development process, it is necessary to be able to manage this data and metadata in an efficient manner that relates directly to the actual tools and methods used. This test management capability also is critical when it comes to managing the scale of test environments – such as cloud-based testing, which can encompass hundreds to thousands of simulations/tests. The tools and system proposed in this paper is used for test process management and advanced test methodologies, as well as handling the traceability process for certification needs.

¹ Business Development Manager - Simulation and Test Systems, dSPACE, Inc. AIAA Member

² Project Manager, Autonomous Simulation and Test Systems, dSPACE, Inc. AIAA Member

I. Nomenclature

<i>MBD</i>	=	Model-Based Development
<i>AV</i>	=	Autonomous Vehicle
<i>HIL</i>	=	Hardware-in-the-Loop
<i>MIL</i>	=	Model-in-the-Loop
<i>SIL</i>	=	Software-in-the-Loop
<i>ECU</i>	=	Electronic Control Unit
<i>V-ECU</i>	=	Virtual ECU
<i>LRU</i>	=	Line Replaceable Unit
<i>GPU</i>	=	Graphics Processing Unit
<i>HPC</i>	=	High-performance Computing system
<i>SUT</i>	=	System under test
<i>PLM</i>	=	Product Lifecycle Management
<i>ALM</i>	=	Application Lifecycle Management
<i>XIL-API</i>	=	ASAM XIL-API Interface Standard
<i>ARINC</i>	=	Avionics Communication Bus Standard
<i>I/O</i>	=	Physical input/output channels of electronic control units, devices, etc.
<i>UAV</i>	=	Unmanned Aerial Vehicle
<i>OEM</i>	=	Original Equipment Manufacturer
<i>OS</i>	=	Operating System
<i>ROS</i>	=	Robot Operating System
<i>RTOS</i>	=	Real-time Operating System
<i>ESI</i>	=	Environment Sensor Interface
<i>SOTIF</i>	=	Safety of the Intended Functionality (ISO/PAS 21448)

II. Introduction

Validating highly complex automated vehicle systems is a big challenge. Dependency on environment situations and scenarios for testing, infrastructure and resources required for testing can be quite expensive, time consuming, and rarely effective enough to test the majority of key scenarios in a regressive manner. HIL systems are the accepted standard for an embedded system validation process, but even with these test systems, the limited availability of resources and project time constraints leave us looking for more efficient alternatives. The adoption of the same process for sensor simulation when doing SIL testing provides a more-efficient alternative that is also scalable to address large-scale testing processes on high-performance clusters (HPCs) or the cloud. It is also very applicable to the individual software developer that can use this technology to debug their sensor-based algorithms during the development phases. This entails implementing a modular test system that provides the proper technologies to be implemented in diverse environments and flexibly handling various ways users implement their system algorithms, whether model-based or code-based.

Key technologies are available which allow for direct sensor data injection into ECUs for test interaction and stimulus, in addition to dynamic on-the-fly modification of sensor data streams. It will be shown how these techniques are integrated with current HIL and SIL, or xIL systems, in a manner that addresses the need of users for sensor perception development and sensor system V&V. This entails handling diverse development environments, such as CUDA, C++, etc. and complex system frameworks such as ROS, DDS, etc. It also entails supporting complex hardware acceleration needed via GPUs for sensor processing and simulation. Finally the concepts presented can be extended to address system integration for complete data fusion environments, from single to multiple sensor ECUs, supervisory controllers, and also virtual controllers or hybrid integration of real/virtual ECUs. With new testing technologies, a logical flight/path scenario can be run through the XIL test system with sensor distortions or faults that can help improve the test coverage (e.g. varying levels of pixel errors, lens shadings, and color gains for a video stream). Key test scenarios or edge-cases can also be analyzed by test algorithms that allow advanced test processes (e.g. stochastic permutation) to be used. These advanced test management approaches provide a strategy for an exponential increase in the coverage of test scenarios that can be utilized during validation of the system, via automation and statistical processing techniques.

In order to achieve validation of Autonomous control systems, it is critical to utilize state of the art techniques such as virtual validation to get the full bandwidth required for System V&V. This approach works well in the framework of MIL (model-in-the-loop), SIL, and HIL simulation to establish a testing process geared to provide

greater coverage for V&V of autonomous systems. The technologies discussed here also provide the benefit of automation and test asset reusability, giving further advantages to expand V&V capabilities and coverage.

New technologies and effective techniques for optimization of test processes and certification relative to DO178C can be used today, and these allow for effective verification and validation (V&V) of aero-applications utilizing simulated scenarios. The development of key initiatives in the automotive industry such as SOTIF/ISO26262 for development and certification of Autonomous vehicles can be correlated to DO178C processes and practices, enabling risk mitigation of data fusion systems. The necessities of V&V process traceability and coverage analysis need to be considered when implementing a chain of V&V processes, in order to manage the reusability and automate the processes of these V&V environments. A critical process for safety is fault testing in order to minimize risk and the sensor simulation processes described here can be used for this type of failure testing.

The key for testing quality and process certification is traceability. This has become difficult today with these complex V&V environments, given the many components and artifacts that need to be managed for testing in the MBD and xIL processes. Traceability is also necessary in order to ensure the compliance of the development and testing processes. A system integration tool and process is needed to tie together the large amounts of common data, models, and test facets that every company faces when implementing these types of test systems. The MBD/SW process leads to generation of large volumes of data artifacts and work-products throughout the V-Cycle of development. The various components of these environments, from models to parameters to tests, can be inundating, and variants and versions of these artifacts lead to even larger amounts of data. These artifacts have traditionally been managed with Configuration Management systems and Product/Application Lifecycle Management (PLM/ALM) tools, but these can be limited when considering the integration of specific tooling. In order to effectively use the MBD tools in the development process, it is necessary to be able to manage this data and metadata in an efficient manner that relates directly to the actual tools and methods used.

III. Aerospace Applications for Sensor Fusion

Aerospace environments consist of both static and dynamic objects both in the air and on the ground. Dynamic air objects include aircraft, UAVs, and birds. Dynamic ground objects include aircraft, ground vehicles and people. Detecting, classifying, identifying and tracking these objects is done in collision avoidance algorithms deployed in current systems. Multiple sensors, such as cameras, radar, LIDAR, etc., need to be used for capturing the object's characteristics and location. Data fusion is used to combine data from these sensors. Fusion is then subsequently used for object identification [1]. Detect-and-Avoid (DAA) systems are highly dependent on the performance of communications, navigation and surveillance (CNS) systems for detection, tracking and avoiding (DTA) tasks and maneuvers. In order to perform effective detection of objects, a number of high-performance, reliable and accurate avionics sensors and systems are integrated and used. This includes: non-cooperative sensors (e.g., visual and thermal cameras, Laser radar (LIDAR) and acoustic sensors) and cooperative systems (e.g., Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Collision Avoidance System (TCAS) [5]). It is critical to utilize this sensor technology for object detection and identification to ensure the integrity of these systems in all sorts of potential scenario situations and environmental conditions.

The current process for validating the sensor localization and fusion algorithms employed in these types of systems involves using HIL and SIL testing with the System-Under-Test (SUT = system LRUs/ECUs or SW). In order to accomplish this, compatible interfaces must be employed to simulate and bypass the interfaces of the sensors used in the application. The following sections will cover these sensor test interfaces and tools and processes that are used to validate sensor fusion algorithms. It will also cover using advances scenario-based testing processes that can be leveraged for greater coverage of the scope of testing needed for these types of automated systems.

IV. Validation of Autonomous Application Systems

Autonomous vehicle systems are highly-complex, given the degree of integration of various sensor systems and other external data sources, such as GPS, communication systems, etc. The process for validation is even more complex, given the need to exercise these systems through a nearly infinite potential set of possible scenarios, assuming that these scenarios can be defined and used for testing. The automotive industry and ISO are defining a process called SOTIF (ISO/PAS 21448) to help mitigate the risk of these systems by utilizing scenario-based testing that is constrained within the Operational Design Domain (ODD) of the specific vehicle. This approach allows for evaluation of the unknown risk of the system, using advanced statistical testing and analysis to help mitigate this

risk. However, even this approach may need to utilize a huge amount of scenarios and tests to achieve this level of safety or certification [7].

The standard process for system V&V is to utilize the MIL-SIL-HIL back-to-back testing approach and then vehicle testing and this is addressed by both ISO26262/SOTIF (Automotive safety) and DO178C (Aerospace). Vehicle testing is expensive and does not provide good regression possibilities. HIL systems are more powerful for testing regressively through extreme/dangerous conditions and provide automation capability, but with resource constraints for the amount of testing needed, more efficient alternatives such as SIL testing and scalable HPC simulation systems are being used. The difficulty with these approaches is the need to integrate testing process for advanced sensing systems into the loop. In order to use these test systems they must also support various SUT implementation frameworks/environments and flexibly handle the users' methods to implement their system algorithms, whether model-based or code-based.

An autonomous control system is used to define motion planning for the vehicle under test. Actuation simulation is needed to control the system in longitudinal and lateral directions, including braking and steering systems. The vehicle actuation directly feeds back to the sensing process in the environment, necessitating accurate vehicle dynamics simulation, so the system will properly react to scenario maneuvers and feedback this into actual vehicle performance to close the loop. The autonomous system testing is not usually only on the main control algorithm, the perception and processing functions of the sensors may have to be taken into account as well. A general sequence of process for autonomous controllers and test models is shown in Figure 1.

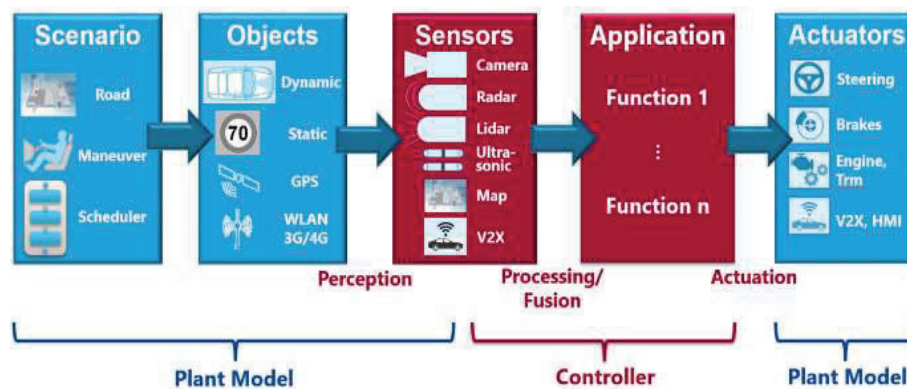


Figure 1. Autonomous controller architecture [6]

There are 2 approaches for validating algorithms based on sensor data, either open-loop data playback of captured real sensor data/scenarios or closed-loop simulation of a vehicle and traffic/environment and feeding the generated sensor data to the SUT for testing. With open-loop testing, the SUT is subjected to more realistic inputs with all sensor effects, like environmental conditions, actual bus (ARINC, Ethernet) traffic, and actual scenarios. The control algorithm (and SUT) cannot freely move in replay environment, as the stimulus data is defined for a fixed trajectory only, keeping the actuation and vehicle dynamics control out of the test. This could also change the system sensing process, as the moving vehicle relative to the environment changes the sensing area. To address this limitation, a full simulation of platform, sensors, and environment allows closed-loop and scenario-based testing to be done, as shown in Figure 2.

The sensors are stimulated by model interaction with a virtual environment, including dynamic objects (traffic) and static ones (e.g., signs, barriers, buildings). GPS signals and Wi-Fi and mobile communication signals, can also be simultaneously simulated. To close the loop and complete the simulation process, the control actuators are also modeled, allowing for steering, braking, taxiing, flying, and communication processes to be tested. This allows for not only functional tests, but also provides a method for failure mode testing to help ensure proper system operation of critical functions. [6]

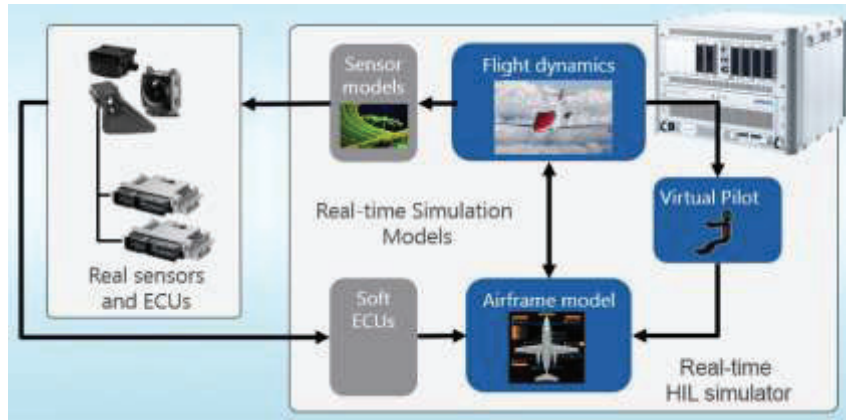


Figure 2. Closed-loop scenario-based test process

Scenarios for testing are used to define a system simulation through specific SUT interaction with their environment. The state of the art in the industry recommends this testing approach in order to minimize risk with more comprehensive testing. This process has been supported by the Pegasus Consortium and also is recommended by the PAS 21999 (SOTIF) standard. The following diagram shows the functional decomposition of the scenario into six layers for semantic representation by Pegasus [9].

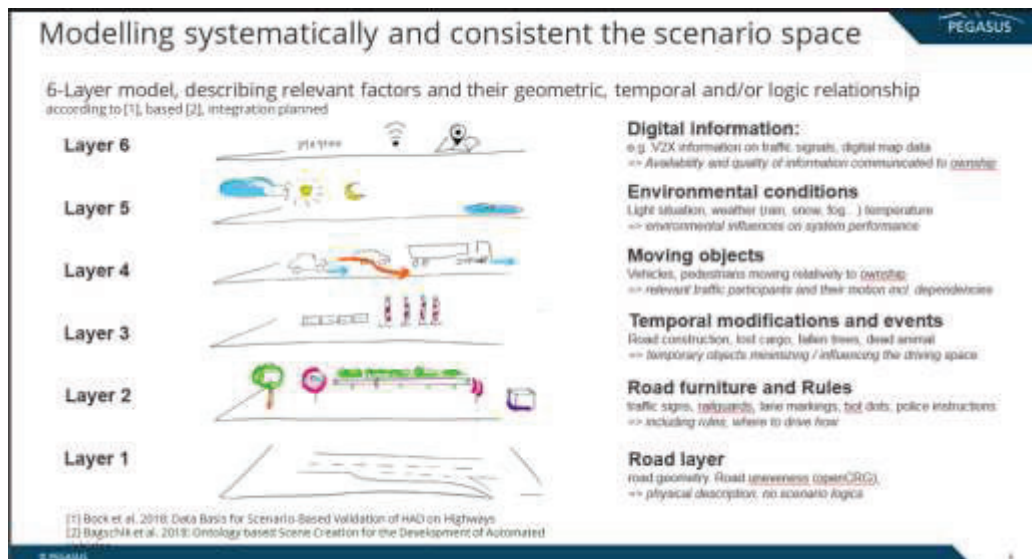


Figure 3. PEGASUS 6-layer model for Scenario Definition [8]

A scenario can be defined in a logical sense and is composed of the following components:

- Static Objects/Obstacles/Infrastructure (may come in via map or measured data)
- Traffic/Dynamic Objects (Objects in motion)
- Environment/Weather
- Temporal modifications/events (construction, signs, situational info, etc.)
- Road/Runway/Maps (HD Maps, GNSS data)
- Digital Information (V2X data, information received, etc.)

A scenario is defined by the trajectories of all the dynamic objects: e.g. ego, fellows, pedestrians, etc. – this refers to a road, which contains the static objects. The road is the geometrical road network and topology, including lanes, intersections, height profiles, and this could include infrastructure elements, like traffic signs, traffic lights, and scenery. The road could also have temporal changes, due to construction, weather, etc. Scenery is composed of 3D assets such as buildings, trees, houses, etc.

A scenario is created to replicate specific test situations using high-fidelity models and simulation tools. A logical scenario is modelled based on general conditions and this can be used to generate different test cases with parameter variations. These parameter variations can take the form of changing kinematics of traffic participants, faulting the sensor data for failure-mode handling, and also changing environmental conditions, etc. Once a test scenario has been modelled, the real benefit is that it can be translated into endless different test cases by managing parameter variations. These parameter variations can take the form of changing the sensor simulation properties, faulting the sensor data for failure-mode handling, etc. This approach also facilitates implementing statistical-based testing for more robust validation [6].

V. Testing advanced sensors and ECU interfaces

Automated pilot and control systems are based on environmental perception using camera, radar and LIDAR sensors. In the automotive industry, autonomous cars integrate multiple of these sensors simultaneously. Whether one uses Open-loop or Closed-loop testing, different interface levels can be used to inject sensor data within an ECU/LRU (HIL testing) or AV software stack (SIL testing). These insertion options are shown in Figure 4 and explained in further detail below with various sensors.

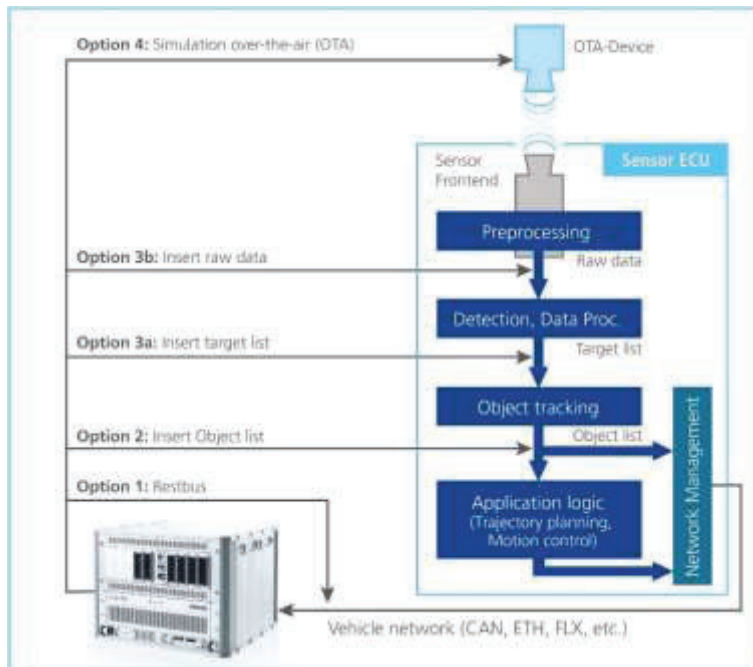


Figure 4. Different options for sensor data connection to an ECU

‘Over-the-Air’ (OTA) stimulation is a long-used approach for connecting simulated or real sensor data into a camera system (Figure 4, option 4). For a camera system, the environment simulation is rendered by a 3D engine and visualized by a monitor. This is used as an input for the camera ECU by placing it in front of a monitor and employing a lens for correct perspective. The complete sensor hardware / software (HW / SW) processing chain is tested, and the physics are considered in the analysis. The limitations of this approach are the following:

1. difficulty with simulation of the complete environments for radar and LIDAR
2. trying to integrate multiple sensor setups in synchronization
3. Taking into account variances in the conditions of the simulated sensor environment and other lighting and sensor-related conditions (e.g., glares, exposure, etc.)

These shortcomings lead to other approaches that bypass sensor processing layer(s) and inject simulated sensor data into ECUs to validate and develop autonomous algorithms [2].

Another common testing approach is the injection of target lists (Figure 4, option 3a) or object lists (Figure 4, option 2). The ground truth sensor information of a simulation can be used to provide these detected lists, which can also be injected into the application HW/SW Logic at the proper point. For example, a detected target can be fed

into the tracking algorithm for a camera and/or a complete set of objects can be used to generate a LIDAR 3-D point-cloud for testing. This V&V approach is viable if proper functional decomposition is done for system testing and it is assumed that the sensor functionality is also tested separately. It is necessary to be able to connect to the ECU hardware or AV software stack in order to do this data injection. Similarly, with Restbus simulation (Figure 4-option 1), the majority of the sensor software stack can be bypassed for distributed sensor and processing systems. This is accomplished by using the simulator to publish the object and tracking data directly to the airframe bus system, provided that the interface exists to feed this data at the proper point in the processing toolchain. These simple approaches are used to test higher-level applications such as trajectory planning or motion control of the vehicle [6].

In order to test the complete sensor processing toolchain, new advanced technologies are available for bypassing the hardware ECU sensing front-end (e.g., camera's optics and imager chip) or also feeding the sensor processing software in a SIL simulation. This allows injection of raw data (voltage level or sensor protocols) by the simulator (Figure 4, option 3b). This powerful sensor simulation capability is modular and extensible, since the raw data can be generated for each sensor setup individually. This is possible for all types of sensors, providing you have a real-time sensor model for simulation. This also allows for more advanced camera simulation, including infrared and stereo cameras, as well as different lens technology and data rates.

Using a similar process for sensor simulation when doing SIL testing is scalable to address large-scale testing processes on high-performance clusters (HPCs) or the cloud. This approach also allows the individual software developer to debug their sensor-based algorithms during the development phases. Sensor simulation requires a high performance PC for both HIL and SIL. In the SIL use case, it is needed to calculate sensor data faster than real time to test many test scenarios as fast as possible. In the HIL use case, however, new data samples have to arrive within hard, real-time constraints and be handled synchronously [6].

VI. HIL Testing and Sensor Simulation

HIL testing is used for validating the data fusion process and algorithms that are based on the autonomous test vehicle's sensor systems. The advantages of HIL testing (deterministic, repeatable, cost effective, etc.) make it viable for certification processes and safety requirement verification. HIL testing is also used for true "real time" evaluation of all software and hardware aspects of the system, a basis of safety standards like DO-178C and is an established standard methodology for ECU software validation in the aerospace and automotive industries [3].

Simulation requires creating models to replicate the sensor behavior (e.g., cameras, radar, LIDAR, ultrasound, etc.) on the autonomous vehicle. The localization algorithms in AV-systems combine this data with GNSS and maps must to determine a complete map of the vehicle's environment. The models of the sensor systems are function blocks, and sensing models are based on environment detection points that are determined by physics, such as optics, wave propagation, etc. The fidelity of the sensor models used depends on the need for simulation, ranging from ideal ground-truth sensors to physics-based raw-data and finally actual sensors in the loop (Figure 5). Figure 5 also relates the types of sensor models to the different levels of sensor injection which can be used for testing. Ground truth sensor models yield object lists and minimal compute time is required. Probability can be applied to these models to generate target lists with effects of weather and sensor performance. When the sensing process component is to be tested, a more physically accurate model is needed to validate the sensing and object detection capabilities. This may also entail such aspects as material properties and ray-tracing based simulation, in order to model the physics of the sensor waveforms properly. Finally, real sensor interfaces are stimulated with models to create test bench systems [6].

Simulation tools like X-Plane and FlightGear can be used to visualize and animate the truth data being generated by models like a 6-DOF simulation. This animated scenery, however it is generated, is used for raw data stimulation (Figure 4, option 3b) and Over-The-Air' stimulation (Figure 4, option 4). In order to handle the rendering process, high-performance graphic processing units (GPUs) are used. For camera simulation, a shader-based approach can be used, as well as 3D raytracing engines like Unreal. Raytracing is definitely needed for LIDAR and radar simulation to simulate the multiple reflections in the environment [6].

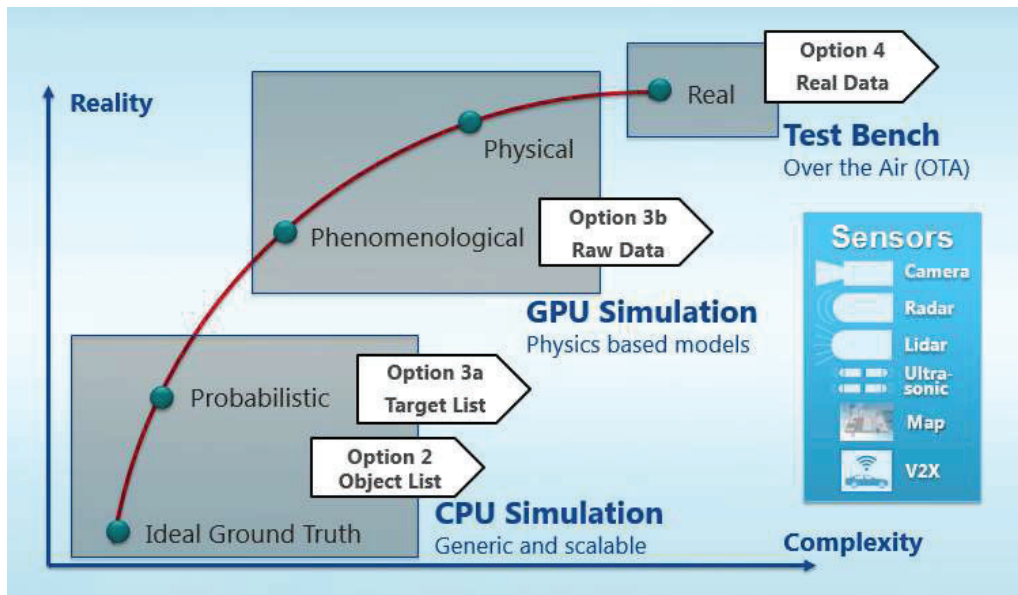


Figure 5. Sensor simulation levels

In order to handle high data rates and challenging synchronization requirements of raw data stimulation, the dSPACE Environment Sensor Interface Unit (ESI Unit) can be used (Figure 6). This FPGA-based unit uses the rendered sensor data as an input and converts this to the raw data protocol for various ECU sensor-specific interfaces. A pod can be used to handle very-short-range protocols on the ECU, if needed. The ESI unit can synchronously feed multiple sensor ECUs with low latency and has a very high total bandwidth (> 50 MB/s). This allows multiple sensors to be simulated and injected in real-time, and ESI units can be run in parallel to handle larger numbers of sensors [2]. The ESI also allows for dynamically changing the sensor signals on the fly, allowing for error-injection (e.g., pixilation errors, color ranging, etc.). Integration of the ESI within a dSPACE Scalexio HIL environment provides a means to synchronize the various simulation signals to handle latencies and ECU interaction mechanisms, allowing for synchronization and control of the simulated sensor front-ends and communication busses (e.g. ARINC, Ethernet, etc.) from the ECUs in real time [3]. In order to simulate many sensors, dedicated real-time GPU computing nodes can be used for the sensor calculation (Figure 6). These can also be employed in a modular, parallel implementation which provides the necessary synchronized calculation capability for the system to handle many sensors and provides the simulation horsepower for complete integration testing.

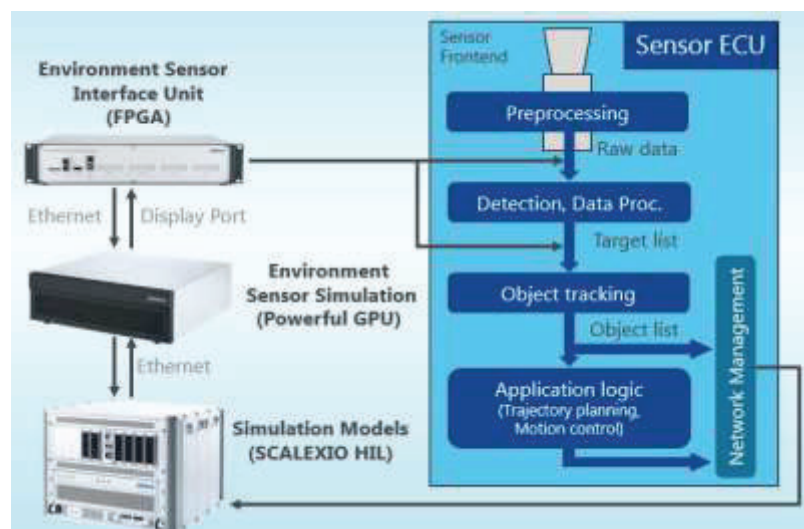


Figure 6. Environment Sensor Interface (ESI) unit and GPU-based environment sensor simulation

VII. Virtual Validation – SIL Testing with advanced sensor technology

As V&V activities are moved to the left in the V-cycle, clear benefits can be realized in cost and efficiency. This is a standard process in terms of using MIL and SIL testing in order to validate systems at earlier stages in the process. MIL and SIL testing is based on connecting controls models or software algorithms to realistic plant models, providing a simulation process to test the control concept being developed. SIL testing can provide more coverage by simulating the complete software system services architecture, including the simulated network interfaces of multiple ECUs. This testing process can evaluate overall system behavior as well as test diagnostics and interaction faults between and inside subsystems, prior to actual hardware being available for testing [3].

In order to achieve greater synergy in the development cycle, it can be very effective to use the same plant models and test tools/processes from HIL testing with the simulations in a workstation environment [3]. In order to do this, it is important to utilize a common toolset that allows for reuse of the models, tests and data across the multiple development stages. The VEOS simulation platform from dSPACE provides this type of common toolset usage across the MIL, SIL, and HIL stages of the development process. Figure 7 depicts the tool usage scenario for both the non-real-time PC-based SIL testing as well as real-time HIL testing. VEOS provides an open workstation simulation environment where models and software from multiple environments can be integrated and executed, allowing the user to design, execute, and validate the target software/models, with advanced debugging capability. VEOS uses the same open standards (e.g. XIL-API) as HIL for tuning/parameterization, user interfaces, data capture/stimulus playback, and testing, providing a seamless way to front-load testing and prove out system interfaces prior to executing on a HIL system. It also supports open model interfaces via the Functional Mockup Interface (FMI) standard, allowing models and software from different toolchains to be integrated. VEOS can also generate virtual ECUs for integration of real and virtual components in a HIL integration rig for ring-out or test of an Avionics ECU system/network prior to ECU availability [4].

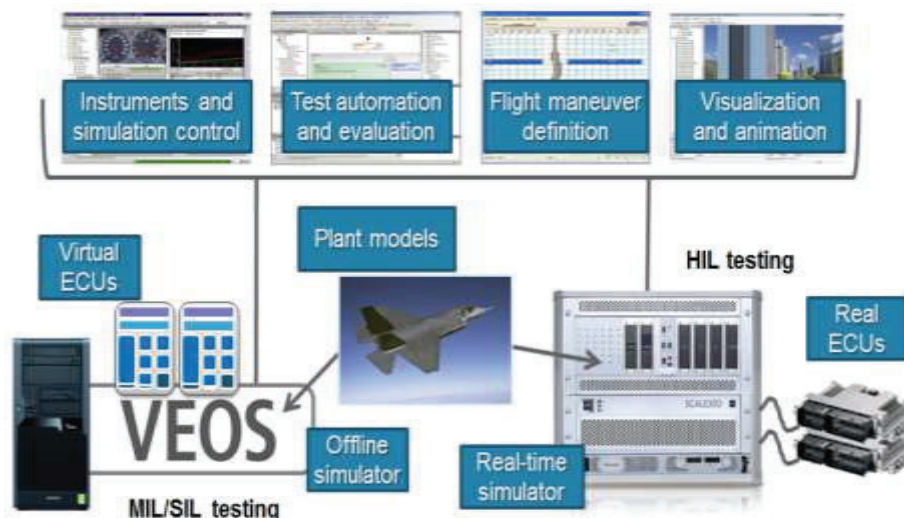


Figure 7: HIL/SIL Test environment and tools [4]

An important aspect of this type of MIL/SIL testing for autonomous systems is that VEOS also provides the same open interfaces used for Scenario-based HIL testing, such as model, scenario, map data, and sensor interface standards. This synergy and reuse can help with the huge task of testing that must be done for autonomous system validation. Most virtual configurations can execute tests faster than in real time, enabling high test throughput possibilities. Massive efficiency improvements are possible with a large number of simulations running in parallel either in a high-performance cluster or a cloud-based environment. The following section will detail a toolchain that supports the testing process and management needed for such systems. You can also run scenario tests with the models and single step or debug the code in action [6]

To provide for testing in the autonomous sensor domain, VEOS can provide for streaming sensor data to the ECU software in a similar way as with HIL testing. This is enabled through a virtual ESI interface, which provides a flexible API to map sensor data to the AV software stack and has mechanisms built into the VEOS environment to synchronize the data flow. This simulation environment is shown in Figure 8. The sensor models run on a CPU for ground truth sensors and also GPU computing node for physics-based realism. The Virtual-ESI translates the sensor

model data to the proper format and uses an API to connect the data to the algorithms. This type of interface is modular and extensible, providing for the ability to deploy multiple AV software stacks (ECUs) and also any number of sensors needed for testing (multiple V-ESI/Sensor simulations in parallel). The modular implementation of the components in this environment support component execution across single or multiple workstations (connected via real or virtual Ethernet) and also support different AV-stack frameworks such as ROS, ADTF, etc. A feedback mechanism to the simulation models allows the loop to be closed on the system and provides the determinism and control of the simulation process.

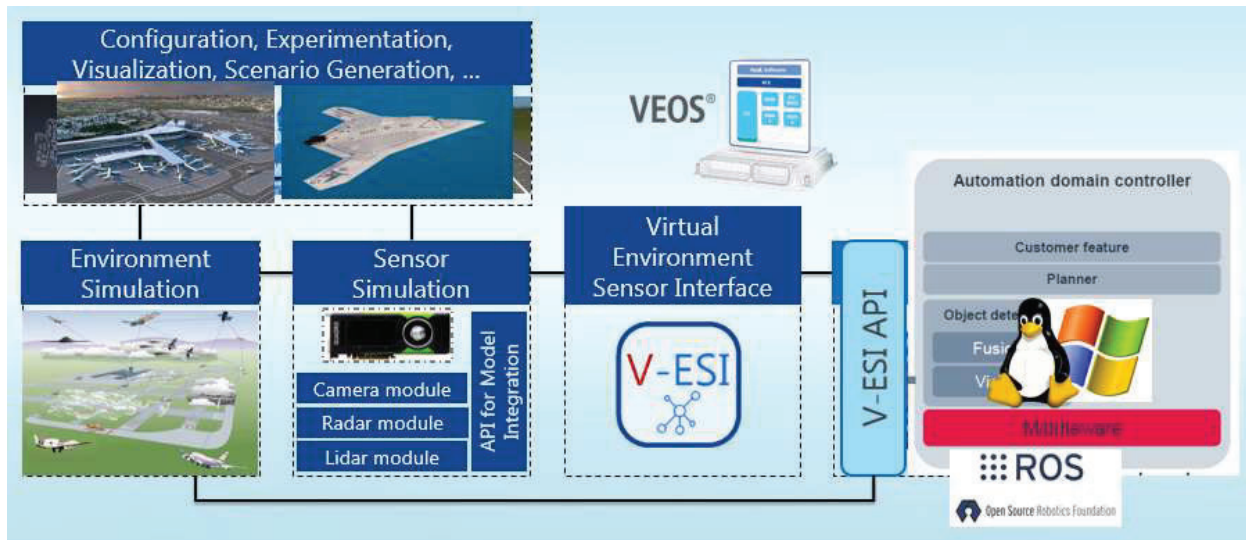


Figure 8. Virtual validation of advanced sensor AV software stacks

VIII. Automated Testing and Test Management

Requirements-based testing is the standard for V&V processes, characterized as back-to-back testing where controls are validated against plant models from MIL to SIL to HIL test systems. For test coverage and cost reasons, this testing should be automated wherever possible. “Lights-out” testing capability through fully automated regression testing on HILs is common, based on different test development environments ranging from script-based languages, such as python, to high-end graphical tools, such as AutomationDesk. AutomationDesk also supports an advanced real-time testing capability, which supports running tests directly on the HIL hardware in real-time, providing requirements validation and observer systems for more comprehensive testing capability [3].

Automated testing needs are much greater with autonomous systems and scenario-based testing. It may be necessary to run thousands or millions of scenario-based tests. Even more important is to determine those edge-case scenarios which are the critical test situations, then permute the test parameters to cover more test space relative to the requirements. Since many MIL/SIL tests are being performed via cluster and/or Cloud testing, the need to manage thousands of test cases and processes seamlessly is even more critical, along with doing smarter testing to minimize resource usage [6].

Test management is a broad scope of processes, ranging from functionality that covers requirements traceability, test execution planning and management, and test platform management. All of these advanced automated test processes depend on efficient test management and execution capabilities. It is necessary to manage the testing assets used throughout the test process, and to provide traceability to the system requirements of all tests performed [3]. Bi-directional requirements traceability is a cornerstone of standards such as DO-178C, ISO26262, and other functional safety processes. The RTCA DO-331 Model-Based Development and Verification Supplement documents the needs for this traceability, used to manage V&V processes [6].

When cluster/cloud testing is performed, execution management is needed to handle the test configuration and execution process control. Another area for optimization is the separation of test execution and analysis and utilization of different resources for handling more efficient testing. The dSPACE Synect test management system has been used to do this for automotive OEM testing processes, helping manage the testing processes and integrating the specific OEM PLM/ALM toolchain [3]. There has also been considerable automation of MIL and SIL simulations, performing testing based on DOE/Taguchi and Monte-Carlo methods. Another interesting approach is using a higher-level observer process to analyze test results and guide the testing processes, based on statistical

approaches (Figure 9). These observers can be used to instrument tests with Key Performance Indicators (KPIs), which are used to evaluate test results with minimal time-consuming post-test analysis [6].

V&V process data encompasses test cases and attributes related to tests that multiply as we go from component to integration testing and includes the continually expanding test processes for SIL up to cloud testing. This generates a large test data infrastructure that includes complex MIL/SIL/HIL test attributes, models, parameters, target hardware/software attributes, signal interfaces, etc. There are many efforts to manage signals and interfaces within aerospace organizations for product lifecycle needs, such as an Interface Control Document (ICD) database, which defines interfaces between avionics systems and airframe systems. This signal information can be used to help manage scalability of subsystem to system test automation and reconfiguration based on virtual or HIL testing. A strong test process will utilize test management systems like Synect to enable optimization of testing systems while providing a comprehensive data management approach to testing [3].

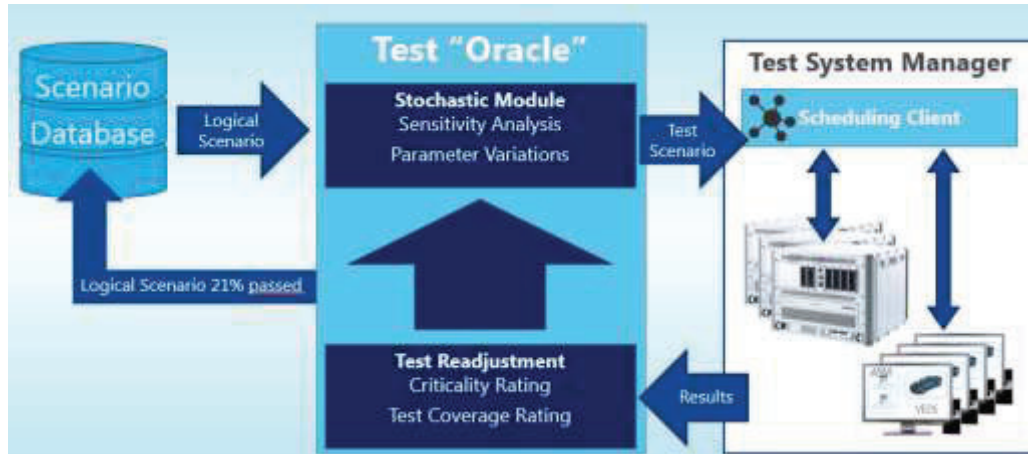


Figure 9. Test enhancement via intelligent monitoring

IX. Application Use Cases

There are many applications for this automated raw data testing methodology with autonomous systems. A few examples are shared here as potential case studies:

- Airport Traffic Monitoring Systems that incorporate data from various ground based sensors, such as; cameras (vision and infrared), LIDAR, radar, etc. These systems can be tested with a near infinite number of generated scenarios and also benefit from testing of real-world scenarios that are recorded and played back in the lab [1].
- UAVs in combat and non-combat environments are often tasked with identifying objects on the ground and in their airspace. The AI/ML systems that they employ can be trained and tested using these advanced testing solutions and the HIL/SIL approach presented above. New technologies would allow direct training of the Image Signal Processor (ISP) and the data fusion algorithms currently in use for these applications. LIDAR is also finding an expanded role in small, low cost civilian UAV platforms for terrain mapping, disaster rescue, urban planning, environmental monitoring, resource exploration, and so on [11].
- LIDAR can be used in ground based systems around airports or other sensitive areas to detect and tracking small UAVs that may pose a risk to normal operations. The automated test methodologies outlined above, can be use both in the training of these detection algorithms, but also the testing of them [10].
- Companies are investigating new uses for airborne LIDAR in the civilian aircraft market. Detection of weather, winds aloft, and turbulence that is nearly 11 miles away are some of the applications being researched. Testing and training of these systems will require advanced modeling and testing tools similar to those discussed in this paper. The data from the LIDAR can also be compared to information being processed by other aircraft sensors to garner more information about the environment the aircraft is headed towards.

X. Conclusion

Given the challenging requirements to validate autonomous control systems in the aerospace industry, it is imperative to implement new test technologies and environments. A toolchain to inject sensor data into autonomous sensor ECUs has been presented that provides a new way to do HIL testing with either data playback or full closed-loop simulation systems. This same process can also be employed with virtual ECU SIL testing of AV software stacks, in order to test system software and interfaces prior to the HIL testing stage. The toolchain is based on standards for data-fusion systems involving camera, LIDAR, and radar hardware, but is also based on deployment processes that allow for flexible SUT integration in different frameworks.

A scenario-based testing process is used in the automotive industry and by using advanced statistical processes, appears to have great potential for mitigating risk in the deployment of autonomous systems. The testing process allows for stochastic parameter variation for more complete test coverage and utilizes test automation and high scalability to be viable, given the potentially huge amount of test cases required to ensure system robustness. This has become especially evident in the automotive autonomous system testing process, where the trend is to increase virtualization and simulation-based testing. These new technologies enable multi-sensor systems to be tested at a full-integration level for HIL, using true sensor and simulation synchronization for test data stimulation.

References

1. Thupakula, K., "Data Fusion Techniques for Object Identification in Airport Environment," SAE Technical Paper 2017-01-2109, 2017, <https://doi.org/10.4271/2017-01-2109>.
2. Sievers, G., Seiger, C., Peperhowe, M., Krumm, H., Graf, S., "Driving Simulation Technologies for Sensor Simulation in SIL and HIL Environments", DSC 2018 Europe, 2018
3. Allen, J.; An overview of Model-Based Development Verification/Validation Processes and Technologies in the Aerospace Industry, AIAA SciTech 2016, AIAA 2016-1922, San Diego, USA <https://doi.org/10.2514/6.2016-1922>
4. Himmler, A., Allen, J., and Moudgal, V., "Flexible Avionics Testing - From Virtual ECU Testing to HIL Testing," SAE Technical Paper 2013-01-2242, 2013.
5. Cappello, F., Sabatini, R., and Ramasamy, S., "Multi-Sensor Data Fusion Techniques for RPAS Detect, Track and Avoid," SAE Technical Paper 2015-01-2475, 2015, <https://doi.org/10.4271/2015-01-2475>.
6. Hager, B. and Allen, J., "Raw Data Injection and Failure Testing of Camera, Radar, and Lidar for Highly Automated Systems," SAE Technical Paper 2019-01-1378, 2019, <https://doi.org/10.4271/2019-01-1378>
7. Becker, Nicolas, "The Safety of the Intended Functionality Report on ISO/TC22/SC32/WG8 activities Geneva", ISO21448 project, January 31, 2019
8. G. Bagschik, T. Menzel, and M. Maurer, "Ontology based Scene Creation for the Development of Automated Vehicles", 2018 IEEE Intelligent Vehicle Symposium, Changshu, 2018
9. Eberle, Ulrich, "Perspectives on Automated Driving, Development Processes and Beyond," Pegasus Symposium Aachen November 2017
10. Hammer M., Hebel M., Laurenzis M., and Arens M., "Lidar-based detection and tracking of small UAVs", Proc. SPIE 10799, Emerging Imaging and Sensing Technologies for Security and Defence III; and Unmanned Sensors, Systems, and Countermeasures, 107990S (4 October 2018); <https://doi.org/10.1117/12.2325702>
11. Zhou, G., Yang, J., Li, X., and Yang, X.: ADVANCES OF FLASH LIDAR DEVELOPMENT ONBOARD UAV, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B3, 193-198, <https://doi.org/10.5194/isprsarchives-XXXIX-B3-193-2012>, 2012.

Journal Acceptance Procedure

The Institute publishes eight archival journals. Authors should review the scopes of the journals to determine the most appropriate venue for their manuscript submission. If an Editor-in-Chief of a journal believes that a paper should be submitted to a different AIAA journal of more suitable scope, a transfer will be recommended. In the journals, material may be published in one of the following forms: Survey Paper (broad review of literature in specific field), Regular Article or Full-Length Paper (including abstract), Technical or Engineering Note (short paper disclosing new, significant data or developments of limited scope), or Technical Comment (discussion related to a paper previously published by AIAA). *AIAA Journal* also includes the option of Express Articles for rapid publication. For other, less frequently published article types, see the complete list of **Types of Journal Content (/publications/journals/Journal-Scopes-and-Content)**.

The following sections explain the acceptance procedure for submitted material.

Evaluation of Manuscripts (click to show)



1. The first step is an examination of submitted papers by the Editor-in-Chief. He or she first tests the manuscript according to the several criteria of subject scope, archival editorial style, apparent technical validity, topical importance, timeliness, relationship to prior publication, conciseness, appropriate references, and length. Papers that vary widely from the prescribed archival style (those written as speeches, ill-defined manuscripts, progress reports or news releases, or those strongly flavored with advertising or sales nuances) will not be considered for publication. If the English is difficult to understand, the author may be requested to rewrite the paper prior to technical evaluation. If the subject scope makes the paper more suitable to one of the other AIAA journals, the Editor-in-Chief may recommend a transfer. If, in his or her judgment, the paper obviously fails in the areas of technical validity and/or advance over prior publication, the Editor-in-Chief can reject it, giving the author the reasons for rejection, or request that the author revise it as a Note.
2. If it survives the tests at this point, the paper will be assigned to the Associate Editor for that journal who has most direct knowledge of the subject matter and of expert

reviewers in the field. The Associate Editor then evaluates the paper according to the same criteria and, in most cases, has the paper sent to one or more reviewers in the field (usually two) for confidential review. The Associate Editor may, however, at his or her discretion, accept the paper without review, reject it giving explicit reason, or request that the author prepare it in a different format.

3. Considerable significance is attached to the review reports. Each reviewer is asked to judge the technical validity of the manuscript and the extent of its advance over work previously published. The reviewer is asked also for advice as to whether the paper merits publication in an archive journal. However, the decision to publish, to require major revision before publication, or to reject for reasons cited lies first with the Associate Editor and ultimately with the Editor-in-Chief. Reviewers often disagree with each other in their advice, and some reviewers tend to be too lenient (or too severe) in light of the technical faults they have found in a paper. Hence the publication decision is made by the technical editors.

The Editorial Decision to Accept or Reject *(click to show)*



4. It may take six months or more after receipt of the manuscript to accomplish the evaluation and review steps discussed above. The Editors will inform the author of their decision (acceptance, conditional acceptance, or rejection) as soon as possible. In the case of rejection, the author will be given specific reasons related to the criteria enumerated above. In the case of conditional acceptance, the required revisions will be clearly indicated. On some occasions, the Editors may anticipate a need for further reviews after revision; if so, the author will be notified.
5. The next step is up to the author. If the paper has been rejected or if extensive revisions have been requested that the author believes are incorrect or unwarranted, then he or she is entitled to submit a point-by-point rebuttal to the Editor's statement of reasons and the reviewers' comments. The rebuttal then is analyzed by the Editors, and a decision is made. In rare cases of a complex point of dispute, the Editors, at their discretion, may mandate additional reviews. In no case shall a paper go through more than two reviewing cycles before a decision is given. If the dispute still remains unresolved, then the decision of the Editor-in-Chief is final and overrides all other considerations.
6. It is the policy of the Editors to make sure that no unconventional hypothesis or original idea is throttled if there is a chance that such a paper might stimulate either progress or constructive controversy on a technical point. It also is a basic principle of the publications program of the Institute that all authors shall have an equal opportunity to have their papers published, provided they can meet certain objectively defined standards. However, the primary responsibility of the Editors is to maintain these high technical standards for the archive journals of the Institute. This commitment takes

precedence over all other factors. Within the bounds of that objective, the following working policy has been adopted by the Editors.

Fair Procedure for Rebuttal by Author *(click to show)*



7. In the confrontation between the rejection statement and the rebuttal statement, the decision goes in favor of the author if the dissenting reviewer's case is not clearly convincing.
8. Because promptness is the essence of fairness, an author who has received no decision on his or her rebuttal within three months is welcome to request an immediate explanation and status report on the paper. If the forthcoming report is unsatisfactory to the author, then he or she may request a prompt decision regarding either rejection or conditional acceptance. The Editor should make every effort to respond to this request in a maximum of four weeks.
9. By the same token, authors who are requested by Editors to revise their papers must make an effort to accomplish the requested revisions in the stated period, which normally is four weeks for major revisions, three weeks for minor revisions. If the author does not respond to our subsequent inquiries, the paper will be regarded as withdrawn. Normally, an author who has good reason to request a time extension will be granted such an extension.
10. A reviewer who feels strongly that a particular paper should not be published may be given the opportunity, if the Editor decides nevertheless to accept it, to write the criticism as a Technical Comment. The author then is allowed to write a closing response for publication in the same issue as the Comment.

Formal Acceptance and Publication *(click to hide)*



11. Formal acceptance will not occur until the author has complied with all of the revision requests (if any) made by the Associate Editor or the Associate Editor has accepted the author's rebuttal, and the author has prepared the paper in the AIAA archival style as described under Manuscript Style and Format.
12. Accepted papers will be published as soon as possible in Articles in Advance and then scheduled for publication in a forthcoming issue. Survey Papers and Full-Length Papers normally will be assigned to issues in the order in which they initially were received. Depending upon the number of papers awaiting publication and the projected size of issues, this may require that papers be scheduled several issues ahead. Items for the Notes and Comments sections will be scheduled for the earliest available issue. The Editor-in-Chief also may designate certain special-category papers for immediate publication. The online publication date is the official date of publication for the paper.

13. Page proofs will be made available to authors for correction and release prior to publication online. Authors should update their records in ScholarOne and inform the Journals Production Team of any anticipated change of e-mail address between acceptance and page proof time. Authors are expected to read and release their proofs within seven days.
14. To allow for late or non-release of proofs by authors and to provide the flexibility to meet issue-length and topic-mix constraints, issues will be overscheduled by about 25%. Thus, there will always be a certain number of papers held over for the next issue. Papers not published in the issue for which they were originally scheduled will have first priority for publication in the following issue. Authors will be informed of the final issue assignment.

This policy was first approved on 16 October 1964 by the Editors-in-Chief of the archive journals and by the Vice President–Publications, and has been revised and updated as needed.

American Institute of Aeronautics and Astronautics

12700 Sunrise Valley Drive, Suite 200

Reston, VA 20191-5807

800-639-AIAA (2422)

© 2024 American Institute of Aeronautics and Astronautics