Developing Advanced Driver Assistance Systems (ADAS) and Functions for Autonomous Driving
### Contents

**Advanced Driver Assistance Systems (ADAS) and Functions for Autonomous Driving**

- dSPACE – Your Partner for Autonomous Driving 3
- Chain of Effects in Autonomous Driving 4
- Sensors in the Simulation Environment 5
- Use Cases 6
- Overview of Tools 24
- Videos 25
dSPACE – Your Partner for Autonomous Driving

The topic of highly automated driving is the focus of many automobile manufacturers’ development activities. Requirements such as 360° redundant surround view with numerous heterogeneous sensors, high-precision positioning or car connectivity are also challenging topics for tool suppliers.

The dSPACE response to this is an end-to-end tool chain for autonomous driving from a single source. Unique rapid prototyping solutions of high-performance platforms and a tailored software environment allow for the development of complete multisensor applications in the vehicle, from perception and fusion algorithms to real-time controls. The significant increase in testing effort can be managed only by moving the tests forward to software-in-the-loop (SIL) simulation. PC clusters enable a high test throughput by means of greatly parallelizing computer nodes and simulations while at the same time maximizing scalability. For release tests, the hardware-in-the-loop (HIL) simulation remains indispensable. One of the greatest challenges for this is integrating real environment sensors such as camera, radar or lidar and the sensor fusion.

dSPACE offers a complete range of integration options, from simple restbus simulation and raw data feed to over-the-air stimulation. dSPACE supports the SIL and HIL testing with realistic Graphics Processing Unit (GPU)-based sensor models and detailed real-time vehicle and environment simulations. The dSPACE tool chain also ensures that the data volumes generated during the test drives can be recorded in a time-correlated manner and played back synchronously and jitter-free in the test lab later.
The chain of effects in autonomous driving generally consists of different processing stages. First, the sensor’s raw data has to be preprocessed (perception). The goal is to detect features and static or dynamic objects as well as free spaces in the environment of the vehicle on the basis of single images or reflection points. During the subsequent stage, the results are merged and collated to a consistent environment model (data fusion). For this, time synchronization and correlation of sensor data is important. In addition, it is necessary to know the exact location and lane position of the vehicle based on a high-definition map (localization).

Based on the environment model, the situation around the vehicle is analyzed, the potential driving trajectories are planned, the decision for a certain maneuver is made, and the longitudinal and lateral control is executed.
Sensors in the Simulation Environment

A detailed and comprehensive simulation of the real world is the basis for a successful validation. Using suitable sensor models and the integration of real sensors with the test environment plays an important role. The range of sensor models extends from technology-independent variants, which generate object lists directly from information provided by the environmental model, to phenomenological or physical models, which are typically calculated on a high-performance GPU and feed raw data to the connected real sensors such as camera or radar. There are different integration options for sensors depending on the type of data and the layer to stimulate. These options can range as far as direct stimulation of the sensor front end, either over-the-air, such as radar, or via HF cable with GNSS (Global Navigation Satellite System) or V2X (Vehicle-to-X) signals. Using the real sensors in the test environment is often indispensable since the signal preprocessing, the sensor data fusion, and creating the environment model in the sensor’s control unit have a deep impact on the chain of effects.

<table>
<thead>
<tr>
<th>Option</th>
<th>Camera</th>
<th>Radar</th>
<th>Lidar</th>
<th>Ultrasound</th>
<th>GNSS</th>
<th>V2X</th>
<th>Electronic horizon</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td></td>
<td></td>
<td>n/a</td>
<td>n/a</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Camera box</td>
<td>ARSG</td>
<td>–</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓, GNSS RF signal</td>
</tr>
</tbody>
</table>

ARSG: Automotive Real-Time Radar Scene Generator; n/a: not applicable; RF: radio frequency

dSPACE offers tailor-made solutions to integrate real sensors with the test environment.
Use Cases

On the following pages, you will find a number of different use cases for developing functions for autonomous driving with dSPACE hardware and software tools. The dSPACE tools form a well-coordinated tool chain for function development, virtual validation, hardware-in-the-loop simulation, and data logging in the vehicle. So you benefit from perfectly matched tools that interact smoothly throughout all the development steps. No matter whether you are developing software functions, modeling vehicles, environment sensors and traffic scenarios, or running virtual test drives on PC clusters. The following overview table lets you easily find the use case you are interested in.

<table>
<thead>
<tr>
<th>Development Step</th>
<th>Use Case</th>
<th>Sensors or Sensor Models Included in the Use Case</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Camera</td>
<td>Radar</td>
</tr>
<tr>
<td>Rapid prototyping</td>
<td>Typical setup for prototyping functions for ADAS and automated driving</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Prototyping functions for automated driving on embedded platforms</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>High-performance sensor data processing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Prototyping complex functions for automated driving in real traffic</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Developing V2X applications based on wireless ad hoc communication</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Prototyping electronic horizon applications</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MIL/SIL simulation</td>
<td>Testing AEB functions according to Euro NCAP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Testing automated driving in urban areas and on highways</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Testing functions for highly automated driving on PC clusters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIL simulation</td>
<td>Release tests for camera and radar applications using closed-loop HIL system</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Open-loop HIL system for testing image processing ECUs</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Closed-loop HIL testing of camera-based systems</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Closed-loop HIL testing of multisensor systems</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>HIL test bench for over-the-air radar-in-the-loop simulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Closed-loop HIL testing of V2X applications</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>HIL setup for GNSS-based driving functions</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time correlated data recording with separate data logging unit</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Typical Setup of Prototyping Functions for ADAS and Automated Driving

- Shock- and vibration-proof MicroAutoBox II with Embedded PC extension for in-vehicle use
- RTMaps for developing perception and sensor fusion algorithms in C/C++, OpenCV, or Python under Linux or MS Windows® on Embedded PC
- MicroAutoBox II for rapid prototyping applications and control functions in Simulink® or according to the AUTOSAR standard.
- RTMaps Interface Blockset to exchange data between the Simulink model on the MicroAutoBox II and RTMaps

Typically, functions for ADAS and automated driving, such as adaptive cruise control (ACC), autonomous emergency braking (AEB), lane keeping assist (LKA), or the intersection assistant, consist of different processing stages. RTMaps, a prototyping tool for multisensor applications, is typically used on the MicroAutoBox Embedded PC hardware platform for developing perception and sensor fusion algorithms in C++, OpenCV, or Python under Linux or MS Windows. The calculation result is then transmitted to the application and control functions on the dSPACE MicroAutoBox II. The MicroAutoBox II performs real-time processing, ensures functional safety, integrates Simulink® and AUTOSAR code, and serves as the interface to the vehicle network, so it can interact with the braking and steering ECUs, for example. The data exchange between RTMaps and the Simulink model on the MicroAutoBox II is established via the RTMaps Interface Blockset.
Prototyping Functions for Automated Driving on Embedded Platforms

- Development, execution, and debugging of algorithms directly on embedded platforms from PC
- Use of components already compiled for target platforms
- Consideration of runtime behavior and hardware acceleration in an early development stage
- Support for MicroAutoBox Embedded SPU\(^1\), NVIDIA\(^\text{®}\) DRIVE\(^\text{™}\) PX 2, NXP BlueBox, and Renesas HAD Solution Kit

When developing functions for embedded platforms, a high number of time-consuming intermediate steps have to be planned. The algorithms are created in a development environment on the PC, cross-compiled and often manually transferred to the target platform. Code debugging can also be very laborious and may require the connection of monitor and keyboard to the embedded system. The innovative approach of Remote Studio Connector in RTMaps aims to avoid unnecessary work steps and thus simplify and accelerate the development of algorithms.

The Remote Studio Connector allows the developer to use RTMaps Studio to create, execute, and debug diagrams on the target platform directly from a PC via an SSL-protected TCP/IP connection. Several precompiled component libraries are available for this purpose, e.g., for the connection of environmental sensors. As the algorithms are executed on the embedded system, many effects such as runtime behavior or hardware acceleration can be taken into account at an early stage of development. RTMaps supports the current development platforms for autonomous driving.

\(^{1}\) MicroAutoBox Embedded SPU is still under development. The information about MicroAutoBox Embedded SPU is subject to change without notice. For more information, please contact dSPACE.
High-Performance Sensor Data Processing

- **MicroAutoBox Embedded SPU**\(^1\) with Multicore ARM\(^\text{®}\) CPU and NVIDIA\(^\text{®}\) GPU for prototyping multisensor applications in the vehicle
- Various interfaces to environment sensors, such as camera, radar and lidar, GNSS positioning, and wireless communication

**RTMaps** Developer version on standard PCs to develop algorithms in C/C++, Python, NVIDIA CUDA or Open Computing Language (OpenCL)

**RTMaps** Runtime version to execute algorithms on MicroAutoBox Embedded SPU

High-performance sensor data processing is essential for automated driving and self-driving vehicles. Developing these kinds of applications requires a dedicated hardware and software environment that can process and merge data from various sensors, such as cameras, radars, lidars, and GNSS receivers, calculate application algorithms, and connect to actuators or HMIIs. In addition, sensor and vehicle network data has to be recorded and played back time-synchronously for testing purposes.

To prototype associated algorithms in the vehicle and to process the large data volumes, a compact and robust prototyping unit with high processing power and an intuitive software development environment is required. The MicroAutoBox Embedded SPU in combination with RTMaps is perfectly tailored to this use case.

---

\(^1\) MicroAutoBox Embedded SPU is still under development. The information about MicroAutoBox Embedded SPU is subject to change without notice. For more information, please contact dSPACE.
Prototyping Complex Functions for Automated Driving in Real Traffic

- Compact and robust prototyping system for in-vehicle use
- MicroAutoBox Embedded SPU\(^1\) with various interfaces to sensors (camera, radar, lidar, etc.) for 360° redundant sensing. Up to six raw data cameras can be connected.
- Six-core ARM®v8 CPU and an integrated NVIDIA® GPU with 256 cores for high-performance sensor data processing and deep learning (artificial intelligence)
- Multi-GNSS receiver with integrated inertia measurement unit and Dead Reckoning. Optional Cat. 6 LTE module
- MicroAutoBox II with real-time processing unit and safety monitoring mechanisms, such as a multistage watchdog, challenge-response, and memory monitoring
- Multiple cascaded MicroAutoBox Embedded SPUs to easily extend data processing capabilities and increase the number of sensor interfaces.

Reliable environment recognition, precise localization, and a well-coordinated vehicle-driver interaction are essential prerequisites for highly automated and autonomous driving. The large volume of sensor and communication data are often gathered in a central control unit. Here, they are preprocessed and classified using complex algorithms, e.g., from the field of artificial intelligence, and then merged into a unified environment model that is used in subsequent steps to determine the driving trajectory as well as vehicle longitudinal and lateral control in real time. The combination of MicroAutoBox II and Embedded SPU provides more than sufficient resources for this. With various sensor and communication interfaces, a high-performance CPU/GPU combination for perception and fusion algorithms as well as a real-time unit for vehicle control and function monitoring, it is ideally suited for rapid prototyping of autonomous driving functions in the vehicle.

\(^1\) MicroAutoBox Embedded SPU is still under development. The information about MicroAutoBox Embedded SPU is subject to change without notice. For more information, please contact dSPACE.
Developing V2X Applications Based on Wireless ad hoc Communication

Rapid prototyping V2X applications on MicroAutoBox
Dedicated blockset to access V2X communication from Simulink via a suitable hardware adapter, such as MK5-OBU by Cohda Wireless
Support for vehicle- and infrastructure-related messages according to standards in Europe and the USA
Map instrument for analyzing traffic and V2X-specific information in ControlDesk

Vehicle-to-X (V2X) technology enables the exchange of data between vehicles and between vehicles and the infrastructure via wireless ad-hoc networks according to the IEEE 802.11p standard. The network participants send messages that include information about details such as their position, speed, and driving direction or events such as traffic jams. In the development of V2X applications, various standards such as ITS-G5 in Europe or DSRC in the USA as well as a growing number of application protocols such as CAM, DENM, or BSM must be taken into account. They are constantly being adapted and further developed. The V2X Solution, consisting of a blockset for Simulink and a map instrument for ControlDesk, facilitates rapid prototyping of applications via easy access to the V2X communication channel and allows for the graphical analysis of V2X-specific data. The blockset provides blocks for handling V2X messages, access to network and transport layers as well as GNSS encoding and decoding. It also enables developers to import adapted application protocols and to expand the blockset with new message types. The communication and physical layer as well as the security of an external V2X hardware adapter are used for connecting to the V2X channel. For this purpose, a flexible interface was developed that in principle allows for connecting different HW types via Ethernet (UDP/IP). The incoming V2X messages are stored in the Local Dynamic Map (LDM). The LDM saves platform resources by centrally storing and automatically managing data, and allows applications flexible and context-specific access to V2X content. The Map Instrument graphically displays the contents of the LDM in ControlDesk and facilitates its analysis and plausibility check.
Prototyping Electronic Horizon Applications

- dSPACE ADASIS v2 HR Blockset to provide an electronic horizon reconstructor on MicroAutoBox II
- Accessing electronic horizon data from Simulink according to the ADASIS v2 Protocol

To help users develop map-based driver assistance systems in fast iteration cycles and directly experience the effects in a vehicle, dSPACE offers the ADASIS v2 Horizon Reconstructor (HR) Blockset for the PC-based simulation platform VEOS and the rapid control prototyping systems MicroAutoBox and AutoBox. This blockset provides access from within a Simulink® model to electronic horizon data that is transmitted via the ADASIS v2 Protocol. Only a few mouse clicks are necessary to select the predictive road data, connect it to the actual driver assistance function in the model, and implement the application on the development system. The support for ADASIS v3 is currently under development.
Testing Autonomous Emergency Braking (AEB) Functions According to Euro NCAP

- Tool chain for testing controller software for AEB systems in a PC-based simulation with VEOS
- Automotive Simulation Models (ASM) for simulating environment sensors, vehicle dynamics, and traffic
- Ready-to-use and fully automated test scenarios based on the Euro NCAP test protocols AEB City, Inter-Urban and Vulnerable Road User (VRU)
- Realistic 3-D scenario visualization in MotionDesk

The European New Car Assessment Programme (Euro NCAP) uses a five-star system to assess the safety of new vehicle models from different manufacturers, giving consumers the opportunity to compare objectively. In addition to classic passive safety systems such as airbags or belt tensioners, the effectiveness of what are known as active safety systems, for example, of autonomous emergency braking (AEB), has also been investigated in extensive scenario-based tests for some years now. To meet this challenge as early as possible, many vehicle manufacturers relied on tests accompanying the development process. dSPACE offers a coordinated tool chain for virtual validation of AEB according to Euro NCAP. The test suite is based on ASM with a vehicle dynamics model and an AEB Soft-ECU as well as a corresponding environment model. It includes test scenario descriptions in ModelDesk and fully automated test execution, evaluation, and report generation in AutomationDesk. In MotionDesk, the scenarios can be visualized realistically and in real time.
Testing Automated Driving in Urban Areas and on Highways

- Closed-loop tests of functions for automated driving on the PC using VEOS
- Automotive Simulation Models (ASM) with open, customizable, and real-time-capable Simulink models for the vehicle and traffic environment
- ModelDesk for configuring complex urban and highway scenarios with an unlimited number of detectable static and movable objects
- MotionDesk for 3-D visualization of virtual test drives, e.g., in highway scenarios with realistic construction sites
- SYNECT for central data and workflow management, test scheduling and execution

Even though hardware-in-the-loop (HIL) simulation remains indispensable for release tests, the significant increase in testing effort can be managed only by moving the tests forward to software-in-the-loop (SIL) simulation, especially in connection with autonomous driving. dSPACE offers a well-tuned tool chain that supports the seamless transition from SIL to HIL. The realistic simulation is the basis of the virtual validation, and an essential component are the real-time capable ASM sensor models. Modeling the environment with roads, roadside development, traffic signs, etc. as well as displaying the road users also becomes more detailed, as these components interact directly with the sensor models in the simulation. In addition, ASM’s multi-agent simulation enables many full-fledged vehicles to drive autonomously in the same environment. ModelDesk supports generating road networks from navigation data and a comfortable import of scenario descriptions, for example, from real driving tests. MotionDesk provides realistic visualization, with advanced lighting simulation even virtual driving at dusk and night becomes easy. Data artifacts, such as simulation models, test scenarios, and parameters, are centrally managed in SYNECT. Test runs can also be planned and automated from here.
Testing Functions for Highly Automated Driving on PC Clusters

- Scalable PC clusters with a high number of VEOS instances and virtual ECUs for the parallel execution of virtual test drives
- SYNECT for data management and scheduling the execution of the simulation jobs and test cases
- Testing faster than real time and driving many test kilometers every day during the early development phase
- Deterministic and reproducible test execution
- Comprehensive integration tests during development

In order to validate functions for highly automated driving, a large number of tests must be performed during the development phase and for the final release. In addition to the established methods, this effort can only be accomplished with the help of software-in-the-loop (SIL) simulation. For maximum test throughput in the shortest possible time, the virtual validation tool chain with VEOS clusters relies on greatly parallelizing execution nodes and simulations in combination with creating high scalability. The driving scenarios to be tested are fed into a central manager node. This node distributes the individual tests to a network of execution nodes that are integrated either as PCs or as virtual machines. Here, the tests on VEOS are performed in batch mode. Furthermore, ECU prototypes often do not exist in early development phases, so that virtual ECUs (V-ECUs) are used. To avoid manual integration of a large number of V-ECUs during extensive chains-of-effects tests, the continuous integration approach is becoming increasingly important, in which the V-ECUs are generated fully automatically from current software versions. The SYNECT test and data management software manages the cluster and allows for an easy integration of the existing tests and continuous-integration processes.
Release Tests for Camera- and Radar-Based Applications Using Closed-Loop HIL System

- Typical HIL set up for testing intelligent adaptive cruise control (ACC) and autonomous emergency braking (AEB) systems
- MotionDesk for stimulating the camera in the loop by means of 3-D visualization
- dSPACE Automotive Simulation Models (ASM) for vehicle and traffic simulation, and a radar sensor model to insert detected objects into the radar ECU
- ModelDesk for parameterizing the virtual test drives
- Real-time HIL system to ensure time correlation between objects detected by the radar and camera sensors

Nowadays, ADAS applications are often based on reliable environment detection by camera or radar. In order to test corresponding ECUs by means of simulation, the real sensors must be stimulated in a suitable way at the HIL. A typical approach for release tests is using a camera box. Enclosed in a chamber, the camera sensor is stimulated by a monitor with 3-D scenarios. The pictures are generated on a PC with a graphics card on the basis of data from a vehicle and environment model. This has to be done in real time with a high frame rate and sufficient detail accuracy, so the camera control unit can recognize and classify objects correctly. For applications that require other types of sensors such as radar, sensor fusion takes place in the control unit. An algorithm combines the objects detected by the camera and radar to form a uniform environment model. It requires that the data streams of both sensors reach the control unit correlated in time. In a HIL test environment, this is done by real-time simulation. It ensures that stimulating the camera from MotionDesk and feeding object lists to the radar control unit are time-synchronized.
Open-Loop HIL System for Testing Image Processing ECUs

- RTMaps to play back and time-correlate video frames and CAN data as well as to compensate different delay times
- dSPACE Environment Sensor Interface Unit (see info box below) to insert data into the image sensor output
- dSPACE HIL system to interface the ECU and replay CAN data in real time without jitter
- ControlDesk to monitor and analyze data and to remote-control RTMaps

Often there is a need to replay recorded data during test drives in the laboratory. It must be ensured that the heterogeneous data streams are reproduced time-coherently. In an open-loop test setup, RTMaps in conjunction with the Environment Sensor Interface Unit and a HIL system ensures realistic real-time stimulation of the device under test via a time-correlated and jitter-free transmission of video and vehicle bus data.

The Environment Sensor Interface Unit is the ideal solution if you have to overcome typical problems of testing camera-based systems, such as limited contrast of monitors, distortion caused by lenses or a complex setup with stereo or multiple cameras. It supports the stimulation of one or more camera sensors with raw image data. For long-range video interfaces, it can be connected directly to the image processing unit of the camera sensor, in case of short-range interfaces via fiber-optic cable and an optional dSPACE plug-on device (ESI-POD). The modular approach supports all common interface types such as GMSL, FPD-Link III or CSI2 and is individually configured according to customer requirements. The powerful FPGA based system handles the data preparation (e.g., image manipulation) and ensures time-correlated data transmission with a high number of simultaneously connected camera ECUs. Optionally, it can be mounted as an additional unit in a HIL system (19-inch rack).
Use Cases / HIL Simulation

Closed-Loop HIL System for Testing Camera-Based Systems by Inserting Data into the Image Sensor Output

- HIL system with Automotive Simulation Models (ASM) and ModelDesk for the real-time simulation of the vehicle and traffic
- GPU-based camera sensor models in MotionDesk for generating image raw data taking camera-specific effects such as lens, pincushion, and barrel distortions, vignetting, and chromatic aberration into account.
- dSPACE Environment Sensor Interface Unit (p. 17) to insert raw data into the image processing unit
- Plug-on device (POD) to adjust to specific interfaces and connectors with sensor ECUs featuring imager and processing unit in one housing (one-box)

The camera box is a widely used method when testing camera-based ADAS applications. However, with the increasing complexity of sensors, issues such as limited contrast of monitors or distortions caused by optical lenses become more evident. In addition, conventional HIL setups with stereo or multi-camera system often reach the limits of economic efficiency or technical feasibility. A very effective and space-saving alternative is feeding raw image data directly into the camera’s image processing unit. The basis for raw data generation is a GPU-based camera sensor model in MotionDesk that takes into account not only scenarios from the ASM simulation model but also effects typically caused by camera lens and image sensor when generating images.

The data stream is then transmitted via HDMI to an Environment Sensor Interface Unit (ESI Unit, p. 17), where it can be preprocessed in real time, i.e., provided with pixel errors or soiled, until it is finally fed to the camera via a long-range (e.g., GMSL) or short-range interface of the image processing unit. For short-range connection (e.g., CSI2), the ESI-POD is also used. The device is connected to the ESI unit via an optical fibre cable and via a short data line to the camera. For a test setup with a stereo front, four surround view, and an in-cabin (driver) camera, a single ESI unit is sufficient; it can be conveniently placed in the 19”rack of the HIL.
Closed-loop HIL Testing of Multisensor Systems

- GPU-based sensor models in MotionDesk to generate raw data for multiple sensors and stream all data to the Environment Sensor Interface Unit (p. 17)
- Environment Sensor Interface Unit to separate the incoming data according to the individual sensors and to insert time-correlated data behind the respective sensor front ends

Autonomous driving systems will be using multiple camera, radar, and lidar sensors. To integrate all these sensors in a HIL set up for release testing, using several camera boxes, over-the-air radar test benches, for example, might go beyond the scope of the project. In addition, it is essential to accurately synchronize the stimulation of the individual sensors. The solution is the Environment Sensor Interface Unit (p. 17) that supports the time-correlated feeding of sensor raw data to one or more camera or lidar ECUs, for example.
HIL Test Bench for Over-the-Air Radar-in-the-Loop Simulation

- HIL system with Automotive Simulation Models (ASM) to simulate the vehicle and traffic environment, and to control the Automotive Real-Time Radar Scene Generator (ARSG) by miro•sys
- Over-the-air stimulation of the radar sensor front end to test the complete chain of effects
- Testing the radar ECU with all software and hardware layers in the original mounting position

Aside from cameras, radar sensors are the most important sensors in advanced driver assistance systems (ADAS). They are essential in the chain of effects of an ADAS function. Therefore, reliable test methods are necessary. Over-the-air radar-in-the-loop simulation allows test engineers to validate the complete chain of effects including the sensor’s radio frequency (RF) components in the laboratory.

Technical Details

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radar objects</td>
<td>- Four fully independent objects with the following manipulation parameters:</td>
</tr>
<tr>
<td></td>
<td>- Distance</td>
</tr>
<tr>
<td></td>
<td>- Velocity</td>
</tr>
<tr>
<td></td>
<td>- Radar cross section (RCS)</td>
</tr>
<tr>
<td></td>
<td>- Azimuth angle</td>
</tr>
<tr>
<td>Update rate</td>
<td>- 1 ms</td>
</tr>
<tr>
<td>Distance range</td>
<td>- 2.0(^{1}) ... 1000 m</td>
</tr>
<tr>
<td>Increment</td>
<td>- 5.6 cm (digital)</td>
</tr>
<tr>
<td>Speed range</td>
<td>- ±700 km/h</td>
</tr>
<tr>
<td>Increment</td>
<td>- 4 mm/s</td>
</tr>
<tr>
<td>Azimuth angle range</td>
<td>- ±90°</td>
</tr>
<tr>
<td>Accuracy</td>
<td>- 0.05°</td>
</tr>
<tr>
<td>Azimuth angular speed</td>
<td>- Max. 200°/s</td>
</tr>
<tr>
<td>Supported radar frequencies</td>
<td>23-26 and 75-82 GHz</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>- 1000 ... 4000 MHz(^{1})</td>
</tr>
</tbody>
</table>

\(^{1}\) Under development.
Closed-Loop HIL Testing of V2X-Based Applications

- Real-time simulator with Automotive Simulation Models (ASM) for simulating the vehicle and traffic environment
- dSPACE V2X Blockset connected to a V2X hardware adapter to provide messages of surrounding vehicles and infrastructure via radio channels according to wireless ad hoc network standards in Europe and the USA
- dSPACE GNSS Simulator Interface blockset connected to a Global Navigation Satellite System (GNSS) signal generator to provide vehicle positions as radio frequency signals
- ControlDesk for controlling the simulation, analyzing V2X data, and manipulating test parameters

The simulation environment is suitable for testing typical V2X applications that are suggested by the Car2Car Communication Consortium for the introduction phase of V2X (day 1), for example. Typical applications are intersection collision warning, emergency vehicle warning, dangerous situation warning, or adverse weather warning. In addition, the test bench addresses the communication with roadside units (RSU, e.g., smart traffic lights and road works) and the relevant applications, such as construction sites warning and green light optimized speed advisory (GLOSA) by supporting the corresponding application protocols. To test the communication with multiple vehicles or RSUs and to take the WLAN channel behavior into consideration, additional V2X radio adapters as well as a radio channel emulator can be added.
HIL Setup for GNSS-Based Driving Functions

- HIL system with Automotive Simulation Models (ASM) for simulating the vehicle and traffic
- Dedicated GNSS Simulator Interface blockset for controlling a Global Navigation Satellite System (GNSS) signal generator, such as GSG-5 by Spectracom
- Selecting specific GNSS scenarios with desired signal conditions according to the current satellite constellation, geographic location, etc.
- Providing vehicle position data as radio frequency signals

Many applications, such as the electronic horizon (eHorizon), but also V2X communication and autonomous driving functions require satellite-supported detection of the vehicle position. Since the availability of satellite systems varies worldwide and the quality of the Global Navigation Satellite System (GNSS) signals depends on many factors such as deep street canyons, flyovers etc., the robustness and suitability for the everyday use of the applications in various test scenarios must be ensured. With the GNSS Simulator Interface Blockset this can be done virtually. For this purpose, a GNSS signal generator is connected to the hardware-in-the-loop (HIL) simulator. The blockset allows you to select predefined GNSS scenarios and control the signal generator. In a typical test, the start position, date, route, and driving maneuver of the vehicle are first parameterized in the ASM model. Then the desired GNSS scenario, for example, a specific satellite constellation (GPS, Galileo, Glonass, Beido, etc.) and signal attenuation are selected. When executing the test, the HIL simulator continuously sends the vehicle’s position data to the signal generator, which then prepares it according to the GNSS scenario and provides it to the device under test as a real radio frequency signal.
Time-Correlated Data Recording During Real Test Drives

- Time stamping, tagging, recording, and play back sensor and vehicle bus data using RTMaps
- Support for dSPACE MicroAutoBox Embedded SPU\(^1\) (Sensor Processing Unit) or Embedded PC, as stand-alone system or integrated in MicroAutoBox II
- Synchronized clocks on all platforms to precisely time correlate the recorded data
- Fast data storing via SATA III interface on MicroAutoBox Embedded DSU\(^2\) with up to four 2.5” SSDs
- Easy configuration of data logger via RTMaps on separate PC

The reliable 360° field detection, which is required for autonomous driving, is performed with a variety of different environmental sensors, such as a camera, radar, or lidar. Every second, these sensors generate high amounts of data, which must be stored safely during a test drive for future use. By using the MicroAutoBox Embedded SPU or Embedded PC in conjunction with the MicroAutoBox Embedded DSU, the heterogeneous sensor data can be stored in a time-correlated manner and with high data throughput on high-speed SSDs to be played back in the laboratory at a later stage.

---

1) *MicroAutoBox Embedded SPU is still under development. The information about MicroAutoBox Embedded SPU is subject to change without notice. For more information, please contact dSPACE.*

2) *MicroAutoBox Embedded DSU is still under development. The information about MicroAutoBox Embedded DSU is subject to change without notice. For more information, please contact dSPACE.*
# Overview of Tools (Selection)

<table>
<thead>
<tr>
<th>Product</th>
<th>Details</th>
<th>Use Cases</th>
</tr>
</thead>
</table>
| MicroAutoBox Hardware                        | - MicroAutoBox II – compact and robust real-time prototyping system with various I/O interfaces and functional safety mechanisms for in-vehicle use  
- MicroAutoBox Embedded PC – shock- and vibration-proof standalone-system or MicroAutoBox II extension with Intel® Core i7 processor and Windows® or Linux  
- MicroAutoBox Embedded SPU 1) (Sensor Processing Unit) – stand-alone prototyping system or MicroAutoBox II extension with six-core ARM® CPU and embedded NVIDIA® GPU for developing perception and sensor fusion algorithms  
- MicroAutoBox Embedded DSU 2) (Data Storage Unit) with up to 64 TB flash memory – for fast storing of big amounts of data | see page 7, 9-12, 23 |
| VEOS – PC-based simulation platform          | - Validation of ADAS and functions for automated driving by means of open-loop and closed-loop simulation on standard PCs  
- Support of any number of virtual ECUs and Simulink® models and of the AUTOSAR, FMI, ASAM XCP, and ASAM XIL interface standard  
- Simulation faster than real time and on scalable PC clusters | see page 13-15 |
| dSPACE HIL simulation technologies           | - Wide range of hardware-in-the-loop (HIL) simulators, specific hardware components and related software for ECU testing  
- Integration options for camera, radar, lidar, ultrasonic, V2X and GNSS (Global Navigation Satellite System) sensors  
- Testing of complete chain of effects of ADAS and automated driving systems in the laboratory | see page 16-22 |
| Simulink blocksets for vehicle networks and ADAS | - Connecting CAN, CAN FD, LIN, FlexRay, and Ethernet to dSPACE real-time platforms  
- Access to electronic horizon (ADASIS v2, ADAS RP/HERE) and IEEE 802.11p-based V2X communication  
- Interfaces to RTMaps from Intempora, EB Assist ADTF, and Spectracom’s GNSS signal generator | see page 7, 11, 12, 21, 22 |
| Automotive Simulation Models (ASM), ModelDesk and MotionDesk | - Open and real-time-capable Simulink models including configuration software for vehicle dynamics, driving maneuvers, road networks, and traffic environment  
- 3-D online visualization of traffic scenarios and generation of video frames to test mono, stereo, and fish eye cameras  
- Sensor models providing object lists of radar, camera, and ultrasonic sensors as well as 3-D point cloud data | see page 13, 14, 16, 18-22 |
| RTMaps – Real-time Multisensor Applications  | - Diagram-based development of multisensor applications under Windows® and Linux on x86 or ARM® platforms  
- Comprehensive component libraries for sensors, vehicle networks, and data processing  
- Easy integration of C++, Python, and Simulink code | see page 7, 8, 9, 17, 23 |
| SYNECT – Data management software            | - Infrastructure to support validation of functions for autonomous driving on MIL, SIL, and HIL  
- Central management of test scenarios and related data such as simulation models, signals, and parameters  
- Integrated variant management  
- Support for PC clusters | see page 14, 15 |

1) MicroAutoBox Embedded SPU is still under development. The information about MicroAutoBox Embedded SPU is subject to change without notice. For more information, please contact dSPACE.

2) MicroAutoBox Embedded DSU is still under development. The information about MicroAutoBox Embedded DSU is subject to change without notice. For more information, please contact dSPACE.
On the dSPACE Website (www.dspace.com/adas) you can find a collection of dSPACE videos that demonstrate the interaction of the various dSPACE tools when developing functions for advanced driver assistance systems (ADAS) and autonomous driving. Typical examples are:

- Over-the-air tests of radar ECUs
- Highly automated driving on highways (including lane detection, construction sites, etc.)
- Testing V2X applications
- Testing Euro NCAP scenarios
- Autonomous parking assistant, intersection assistant

... and many more