Delft University of Technology

Delft Center for Systems and Control

Technical report 09-042

Development of a Driver Information and Warning System with Vehicle Hardware-in-the-Loop Simulations*

O. J. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen

To cite this work, please refer to the published version:

O. J. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-the-loop simulations," *Mechatronics,* Special Issue on Hardware-in-the-Loop Simulation, vol. 19, no. 7, pp. 1091–1104, Oct. 2009. doi:10.1016/j.mechatronics.2009.04.012

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://dspub.eu/09-042

Development of a driver information and warning system with vehicle hardware-in-the-loop simulations ⋆

O.J. Gietelink [a,b], J. Ploeg [a,*], B. De Schutter [b,c], M. Verhaegen [b]

[a] *TNO Science and Industry, Business Unit Automotive, P.O. Box 756, 5700 AT, Helmond, The Netherlands*

[b] *Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands*

[c] *Delft University of Technology, Marine & Transport Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*

Abstract

This paper presents a new method for the design and validation of advanced driver assistance systems (ADASs). With *vehicle hardware-in-the-loop* (VeHIL) simulations the development process, and more specifically the validation phase, of intelligent vehicles is carried out safer, cheaper, and more manageable. In the VeHIL laboratory a full-scale ADAS-equipped vehicle is set up in a hardware-in-the-loop simulation environment, where a chassis dynamometer is used to emulate the road interaction and where robot vehicles are used to represent other traffic. In this controlled environment the performance and dependability of an ADAS is tested to great accuracy and reliability. The working principle and the added value of VeHIL are demonstrated with test results of a driver information and warning system. Based on the 'V' diagram, the position of VeHIL in the development process of ADASs is illustrated.

Key words: hardware-in-the-loop simulation, advanced driver assistance systems, controller validation

Preprint submitted to Elsevier

1 Introduction

Every year in Europe alone, more than 40 000 casualties and 1.4 million injuries are caused by vehicle-related accidents [1]. Although advances in *passive safety*, as illustrated in Fig. 1, have made passenger cars ever safer, the safety potential of further improvements in passive safety features is limited. However, *active safety* systems like ABS and ESP offer possibilities for improving traffic safety by assisting the driver in his driving task. In addition, advanced driver assistance systems (ADASs) have the potential to significantly reduce the number of road accidents. An ADAS is a vehicle control system that uses environment sensors (*e.g.*, radar, laser, vision) to improve driving comfort and traffic safety by assisting the driver in recognising and reacting to potentially dangerous traffic situations. Since an ADAS can even autonomously intervene, an ADAS-equipped vehicle is popularly referred to as an 'intelligent vehicle'. As explained in more detail in several surveys [2] [3], the following types of intelligent vehicle systems can be distinguished:

- *Driver information systems* aim to support the driver on the *strategic level* of the driving task, such as advanced route navigation.
- *Driver warning systems* support the driver on the *maneuvering* level of the driving task and actively warn the driver of a potential danger. The driver can then take appropriate actions in order to mitigate or to completely avoid the dangerous event. Examples are parking assistant, lane departure warning assistant, blind spot warning, and forward collision warning (FCW) systems.
- *Intervening systems* provide active support to the driver on the *control* level of the driving task, such as lane keeping and adaptive cruise control (ACC) [7].
- *Integrated passive and active safety systems* include all systems (including some of the above) that work towards vehicle safety in a cooperative manner [8].

In the remainder of this paper we will focus on the development of driver warning systems. Collision warning algorithms typically issue a warning when

⋆ Research supported by the Netherlands Organisation for Applied Scientific Research TNO, the Netherlands Research School for Transport, Infrastructure and Logistics TRAIL, the TNO TRAIL Transport Research T3 program, the Transport Research Centre Delft of Delft University of Technology, the European 6th Framework Network of Excellence "HYbrid CONtrol: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)", contract number FP6-IST-511368, and the subproject SASPENCE of the Integrated Project PReVENT, co-funded by the European Commission under the Sixth Framework Programme.
* Corresponding author:
Email address: jeroen.ploeg@tno.nl (J. Ploeg).

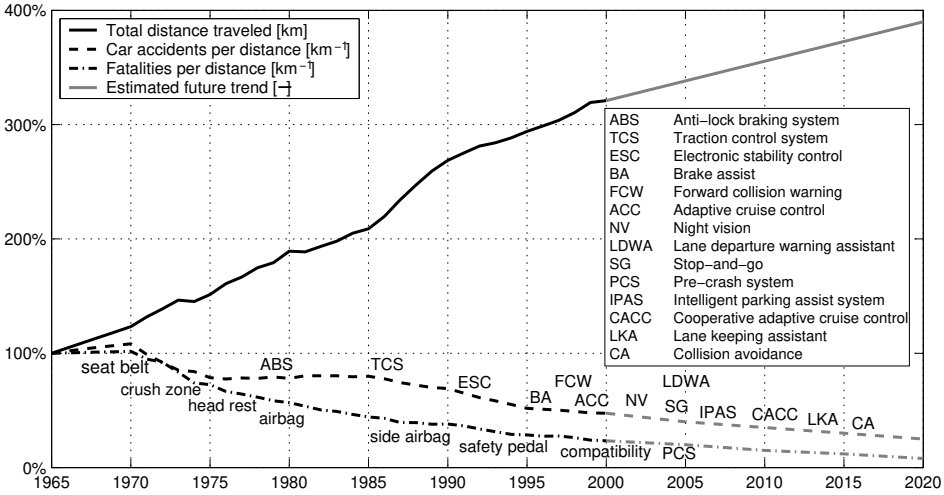


Fig. 1. Total number of road accidents and fatalities per total distance travelled, indexed on 1965 data for the EU [1]. In addition, the graph shows when passive safety systems (which reduce fatalities in case of an accident) and active safety systems (which assist in avoiding an accident) have first been introduced (or are expected to be introduced) to the market, as well as the expected safety potential of these systems [4] [5] [6].

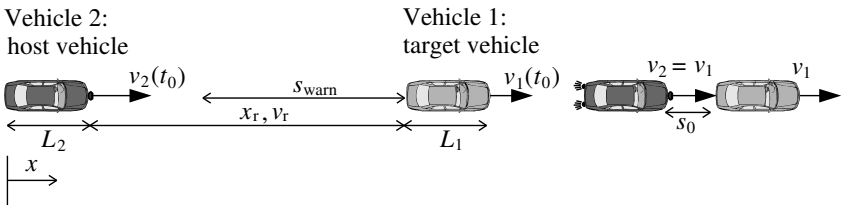


Fig. 2. Development over time of a typical scenario for forward collision warning, including a target vehicle (light) and the FCW-equipped host vehicle (dark). The length of the bold arrows gives an indication of the absolute vehicle speed.

the current range to an object (the headway) x_{r} is less than the *critical warning distance* s_{warn} [9] [10] [4]. The warning then allows the driver to stop or approach no closer than a designated distance s_0 behind a stopped or decelerating target vehicle, as illustrated in Fig. 2. This figure shows a host vehicle i and target vehicle $i-1$, each with state $\mathbf{x}_i = \begin{bmatrix} x_i & v_i & a_i \end{bmatrix}^{\text{T}}$, where x_i is position, v_i velocity, and a_i acceleration of vehicle i. The figure also indicates the vehicle length L_i, the headway $x_{\text{r}} = x_{i-1} - x_i - L_i$, and the relative velocity $v_{\text{r}} = v_{i-1} - v_i$.

Using field-operational test drives with subject drivers, warning algorithms have been developed for use in several commercial FCW systems [11] [12] to give warnings corresponding with natural driver behaviour. Unfortunately, these algorithms will also warn drivers when they intend to perform a late lane-change maneuver, since the algorithm only considers longitudinal vehicle motion. As a result, drivers may find the system conservative and they

may become less sensitive to future warnings. This illustrates the need for appropriate warnings to the driver. In this respect, drivers expect an ADAS to meet high requirements in terms of (subjective) performance, reliability (low rate of false alarms), and safety (low rate of missed detections). Therefore, the ADAS must be tested for the wide variety of complex traffic situations that the system should be able to recognise and to handle [10]. Unfortunately, exhaustive testing of an ADAS prototype is usually impossible due to constraints in costs and time-to-market. Not only the design, but especially the validation of ADASs, thus requires a growing effort in the development process. To address these issues, efficient methods are required for the design of ADAS controllers and for the validation of their safety and performance.

The objective of this paper is to present a new method for the development of ADASs that complements the existing development process. This method consists of vehicle hardware-in-the-loop (VeHIL) simulations that allow to efficiently and accurately test full-scale ADAS-equipped vehicles in an indoor laboratory environment.

The remainder of this paper is organised as follows. The problem statement is further defined in Section 2 by reviewing the development process of ADASs and state-of-the-art test methods. In Section 3 we then present the working principle and added value of the VeHIL concept, and discuss the position of the VeHIL laboratory in the ADAS development process. This is demonstrated in Section 4, where VeHIL test results for a new driver information and warning system are presented. Finally, Section 5 presents the conclusions, and discusses ongoing research activities.

2 Tools in the design and validation process

2.1 Challenges in the ADAS development process

In the automotive industry the different phases in the development process of safety-critical mechatronic systems are often connected using the 'V' diagram [13]. As depicted in Fig. 3, this diagram uses a 'top-down' approach for *design* and a 'bottom-up' approach for *validation*, although in practice the development process does not strictly follow all phases in this sequence and goes through several iteration loops. For relatively simple mechatronic systems, the design process is quite surveyable, as formalised in various generic methodologies for the design of mechatronic systems (*e.g.*, [14]). However, the various development phases for complex mechatronic systems, such as an ADAS-equipped vehicle, face some specific challenges that are addressed in this section.

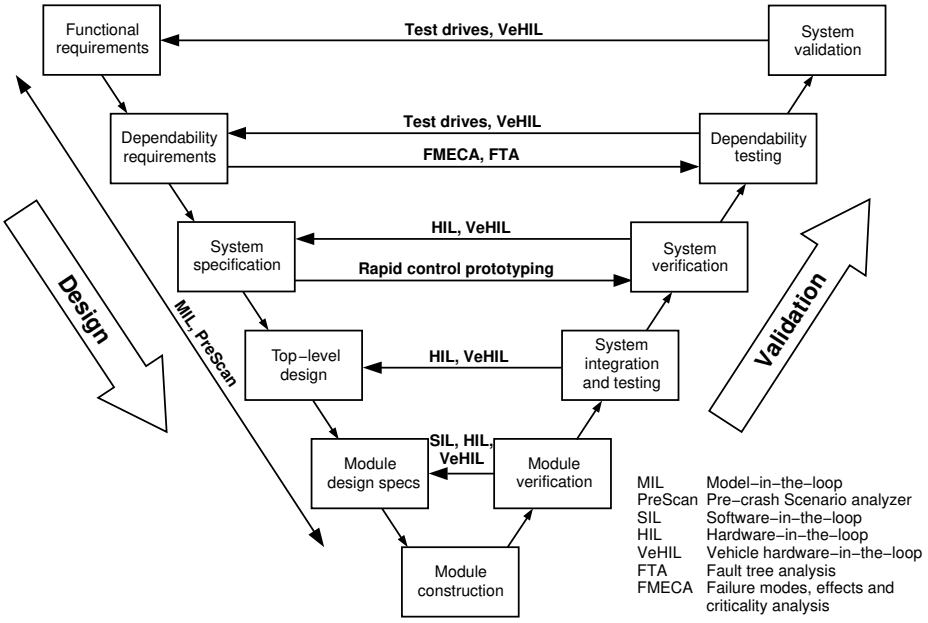


Fig. 3. The 'V' diagram represents the sequential design and validation phases in the development of automotive safety-critical systems. The horizontal arrows indicate the mapping of design phases onto the corresponding validation phases (or vice versa), using the appropriate test tools.

2.1.1 Requirements and specification phase

There are guidelines and procedures available for ADAS development, such as the ADAS Code of Practice [15] and the ISO standard for FCW [16]. Unfortunately, these can only be applied at a high abstraction level in the development process, and do not provide objective requirements and evaluation criteria for ADAS validation and benchmarking, nor do they prescribe the use of specific tools and methods in the validation process.

Quantitative *functional requirements* are therefore required in terms of the desired performance, driver comfort, and operational constraints. In addition, ADASs are safety-critical systems that require a high level of dependability. The system designer should therefore perform hazard and risk analyses (such as FMECA and FTA) to identify the *dependability requirements* in terms of the required level of reliability, safety, and fault tolerance [13].

Reliability is usually defined in terms of the rate of false alarms (when an ADAS takes unnecessary action) and missed detections (when it fails to correctly detect a dangerous situation). The NHTSA field-operational test has demonstrated a false alarm rate around $2 \cdot 10^{-3}$/km for an FCW system, and around 10^{-5}/km for an ACC [9], but this is still considered too high.

From the functional and safety requirements a *system specification* is produced to define the precise operation of the system. However, in practice dependabil-

ity requirements are often difficult to define and subject to ambiguity, which may lead to an incomplete or incorrect specification.

Subsequently, the system specification is used as the basis for the top-level design of the system architecture, followed by detailed module design (environment sensor, controller, actuator, human-machine interface). After implementation of the individual hardware and software modules, system integration takes place by assembling the complete system from its component modules.

2.1.2 Verification and validation

In every integration phase *verification* takes place to determine whether the output of a phase meets its specification, as illustrated by the horizontal arrows in Fig. 3. On the component level this could mean, for example, testing the range, accuracy, and tracking capabilities of the environment sensor [17]. On a higher level, verification must assure that integration with other subsystems does not have any negative side-effect.

Since verification only confirms compliance with the specification, errors in the specification may result in a faulty product. Furthermore, faults must be identified that have not yet been found during the design process. It is therefore important to perform *validation* of the integrated system against its requirements, especially for type approval and certification purposes.

Usually, the development process involves several iterations, where the results of verification and validation are used to modify the system specification and design, after which another test cycle takes place. Consequently, manufacturers are facing longer development times, whereas they have an increasing desire for a shorter time-to-market of their products. Likewise, the costs for the validation process increases. It is estimated that verification and validation of an automotive control system may take up to 50 % of the total development costs [18]. Obviously, there is a need to reduce the number of iterations and speed up this process. Because of the need for fast, flexible, and repeatable test results, various 'in-the-loop' simulation tools are increasingly being used for design and validation of ADAS controllers, as indicated in Fig. 3. After a review of these tools, the position of the new VeHIL simulation tool in this development process will be clarified in Section 3.

2.2 Model-in-the-loop simulations

The initial design of the ADAS controller is supported by so-called *model-in-the-loop* (MIL) simulations, where the controller logic is simulated in closed-loop with models of vehicle dynamics, sensors, actuators, and the traffic envi-

ronment. Unfortunately, current simulation tools lack the possibility for testing the complete ADAS in a reliable way with full integration of operating conditions, sensor characteristics, vehicle dynamics, and complex traffic scenarios. The new simulation concept PRESCAN was therefore developed in [19]. PRESCAN allows reliable MIL simulation of ADASs, using validated physical sensor models for radar, lidar, and camera vision in a virtual environment, illustrated in Fig. 6. The simulation of traffic scenarios is based on a multi-agent approach, as will be explained in Section 3.

2.3 Software-in-the-loop simulations

When MIL simulations have provided sufficient results, software code can be compiled from the simulation model of the control system. The real code can then be verified with software-in-the-loop (SIL) simulations, where the remaining hardware components, vehicle dynamics, and environment are simulated in real-time.

2.4 Hardware-in-the-loop simulations

Similar to testing the real software in a SIL simulation, the real hardware can be tested in a real-time hardware-in-the-loop (HIL) simulation. HIL simulations consist of a combination of *simulated* and *real* components, see Fig. 4. Alternatively, a real component can be *emulated*, *i.e.*, replaced by an *artificial* component that has the same input and output characteristics. Ideally, every component should be unable to distinguish between real, simulated or emulated components that it is connected to in the closed-loop configuration. Therefore, HIL offers the flexibility of a simulation, where the use of real hardware offers a high level of reliability.

The main advantage of a HIL simulation is that it provides a repeatable laboratory environment for safe, flexible, and reliable controller validation. Controller performance and stability can be systematically tested without disturbances from other unrelated systems, and dependability can be tested by controlled injection of disturbances and faults. HIL also allows validation of the real hardware in an early development phase without the need for a prototype vehicle, since any missing vehicle components can be simulated. For these reasons, HIL simulations are more efficient and cheaper than test drives, and are extensively used for the development of vehicle control systems, such as ABS [20], engine control systems [21], and semi-active suspension systems [22]. ADASs can also be tested in several HIL configurations, as will be discussed next.

As indicated in Fig. 3, in an early stage *rapid control prototyping* is carried

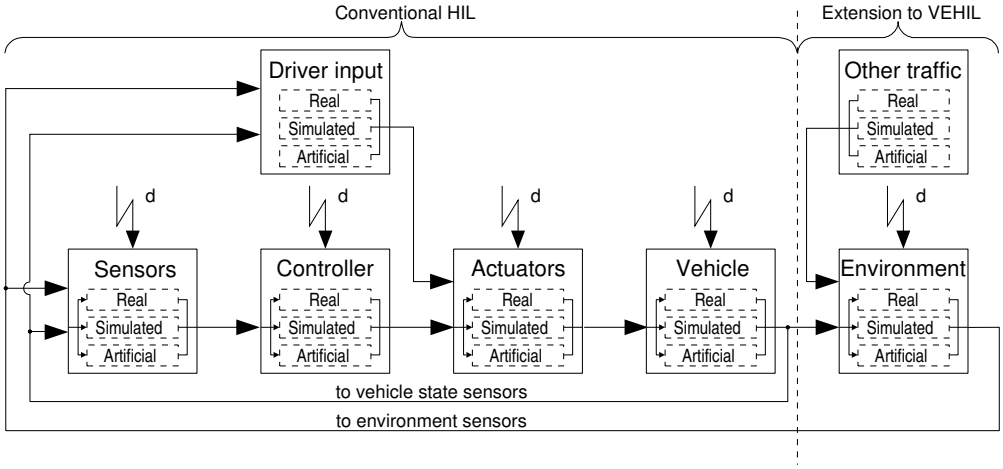


Fig. 4. Possible configurations for HIL simulations, where parts of the system may be real, simulated or artificial. Feedback of signals from environment sensors and vehicle state sensors provides a closed-loop simulation. Additional disturbances **d** can be injected by the operator to test the system's dependability.

out with emulated control functions. This involves implementing a model of the desired controller in a prototype vehicle for the purpose of rapid proof-of-concept, controller testing, and parameter adjustments. Next, the hardware controller can be tested in a HIL simulation for its real-time behaviour [23]. This limited HIL setup can gradually be extended to include other modules, as the integration of the vehicle progresses. For instance, ADAS controllers can be tested in a HIL simulation with real actuators [23] and real sensors [24], where all other components are simulated. However, a complex interface between the *simulated* environment and the *real* sensor is necessary to generate a sensor signal. Yet another type of HIL simulation is a driving simulator, which creates an artificial environment for an 'in-the-loop' human driver [25]. Driving simulators are useful for subjective evaluation of the ADAS and for fine-tuning ADAS controller settings.

Finally, the complete system can be real, including sensor, controller, actuator, and vehicle dynamics. This complete vehicle system is in interaction with the road surface (through its actuators), as well as with the traffic environment that is formed by other objects in the world (through its sensors). Since environment sensors should receive a real input, an *artificial* traffic environment must be created to test an ADAS-equipped vehicle in a HIL simulation. Up to now, no such HIL environment has been available for testing complete intelligent vehicles.

2.5 Test drives

Test drives with prototype vehicles are always the final link in the validation chain to evaluate the system's performance in the real world environment that it will finally be used in. However, the value of test drives for control system design is limited, because test results are hard to reproduce and often inaccurate, due to the lack of 'ground truth' knowledge on the exact state (*e.g.*, obstacle position) of the vehicles involved in the test. In addition, these tests are often expensive, unsafe, time consuming, and heavily dependent on weather conditions [10] [23]. In the next section we therefore propose a solution to combine the advantages of HIL simulations with the representativeness of test drives, by extending the HIL environment from vehicle level to the traffic level, as indicated in Fig. 4.

3 Vehicle hardware-in-the-loop (VeHIL) simulations

To address the challenges mentioned in the previous section, we present a new method for the design and validation of intelligent vehicle systems: vehicle hardware-in-the-loop (VeHIL) simulations. VeHIL provides a solution for testing a full-scale intelligent vehicle in a HIL environment [26]. This paper summarises the VeHIL working principle and discusses the added value of this new type of HIL simulations in the ADAS development process based on new test results with an novel driver information and warning system.

3.1 Working principle of the VeHIL simulation

VeHIL constitutes a multi-agent simulator for intelligent vehicle systems, in which some of the simulated vehicles are replaced by real vehicles. These vehicles operate in an indoor laboratory that forms an artificial HIL environment for the intelligent vehicle. The environment sensors that are used in ADASs (radar, lidar, vision), collect relative position data in the absolute traffic environment. VeHIL therefore makes a transformation from the *absolute* motion of the objects in a traffic scenario to *relative* motion between those objects, as illustrated in Figures 5(a) and (b). Using only the relative motion between a fixed intelligent vehicle and target vehicles allows to have a controlled and space-efficient environment.

The software architecture of VeHIL is based on the multi-agent real-time simulator developed by Papp *et al.* [27], as illustrated in the lower-right part of Fig. 7. This multi-agent framework consists of a collection of autonomous

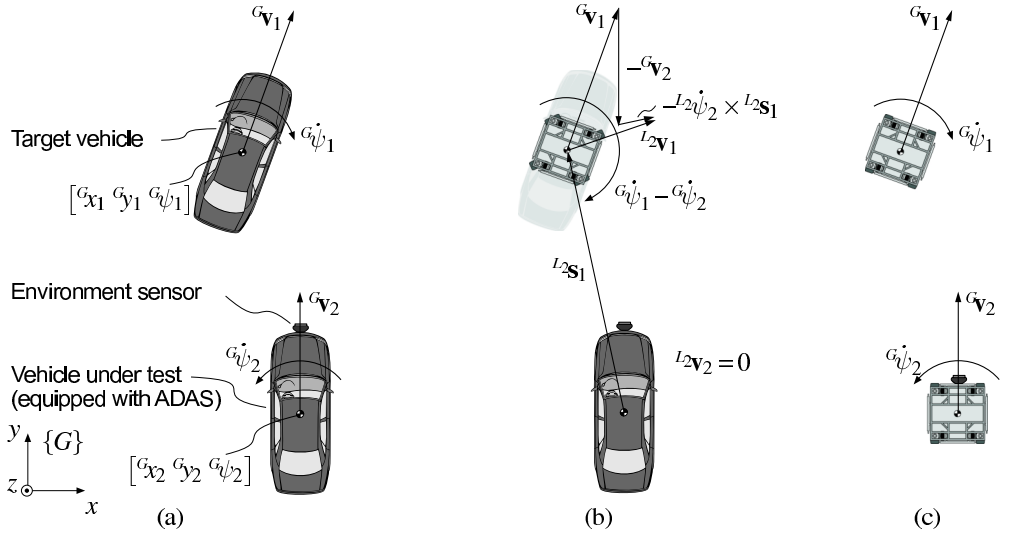


Fig. 5. Transformation of coordinate frames: (a) absolute motion in the real world; (b) relative motion in VeHIL; and (c) absolute motion with two moving bases in VeHIL.

entities E (vehicles, other road users, or any other dynamical component), each controlled by its own internal dynamics (*e.g.*, a vehicle model). An entity has an absolute state $\mathbf{x}$ in the global coordinate frame $\{G\}$, denoted as ${}^{G}\mathbf{x} = \begin{bmatrix} \mathbf{s}^{\mathrm{T}} & \mathbf{\Phi}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}} & \dot{\mathbf{\Phi}}^{\mathrm{T}} & \mathbf{a}^{\mathrm{T}} & \ddot{\mathbf{\Phi}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, where ${}^{G}\mathbf{s} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$ represents the position and ${}^{G}\mathbf{\Phi} = \begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^{\mathrm{T}}$ the orientation in Euler angles (roll, pitch, and yaw) of the entity. The corresponding velocity and acceleration components are represented by ${}^{G}\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^{\mathrm{T}}$, ${}^{G}\dot{\mathbf{\Phi}} = \begin{bmatrix} \dot{\varphi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^{\mathrm{T}}$, ${}^{G}\mathbf{a} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^{\mathrm{T}}$, and ${}^{G}\ddot{\mathbf{\Phi}} = \begin{bmatrix} \ddot{\varphi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^{\mathrm{T}}$.

Furthermore, a *virtual world* is defined that serves as a formal representation of the environment relevant to these entities. Entities are typically represented in the virtual world by *objects* O that interact with other objects (vehicles, bicyclists, pedestrians, infrastructure, traffic lights). Objects are *not* simulation models, but are merely the virtual representation of the simulation entities E. A visual representation of this virtual world with objects is shown in Fig. 6, and is provided by the same visualisation module of PreScan. After every integration time step of this multi-agent simulation, the internal dynamics of an entity (*e.g.*, E_2, representing vehicle 2) result in a state $\mathbf{x}_2$ in the *global* coordinate frame $\{G\}$, denoted as ${}^{G}\mathbf{x}_2$. Through the link between the simulation entity E and its virtual object O, the entity updates the representation ${}^{G}\mathbf{x}_2$ of the associated object O_2 in the virtual world. This link between entity and object is indicated by the dashed lines in Fig. 7.

An important feature of the modelling concept of the multi-agent real-time simulator is that an entity (*e.g.*, a vehicle model) uses abstract sensors S and actuators A to interface with other objects in the virtual world. Through its

Fig. 6. Visual representation of a cut-in scenario in the virtual world: an ACC-equipped vehicle drives on the middle lane, when suddenly one of the two preceding vehicles cuts in from the adjacent lane with a lower velocity.

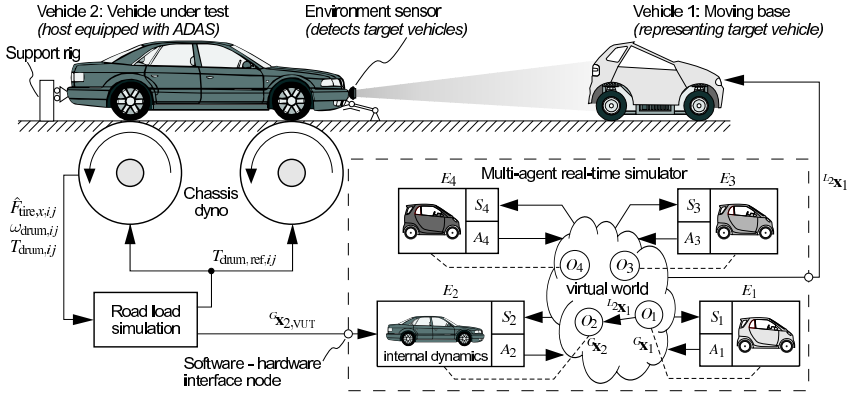


Fig. 7. Schematic representation of the VeHIL closed-loop working principle. Every integration time step the simulation loop runs counterclockwise via the vehicle under test, the chassis dyno, the multi-agent real-time simulator, and the moving base, whose motion is detected by the sensor of the vehicle under test.

abstract sensor S_2 the entity E_2 can collect information about the state ${}^{G}\mathbf{x}_1$ of another object O_1 (*i.e.*, vehicle 1, associated with E_1) in the virtual world. Vice versa, the entity E_2 has an abstract actuator A_2 to act on the state ${}^{G}\mathbf{x}_1$ of O_1. Note that these sensors and actuators are handled in an *abstract* way: they have no dynamics and data processing features. Instead they can be interpreted as queries and actions on the virtual world. *Real* sensors and actuators are modelled as part of the entity's internal dynamics [19].

Using this simulation principle, the relative motion between vehicles 1 and 2 (entities E_1 and E_2) from the viewpoint of vehicle 2 is obtained by a coordinate transformation, where the state ${}^{G}\mathbf{x}_1$ of vehicle 1 is represented in the *local* coordinate frame $\{L_2\}$ of vehicle 2, *i.e.*, ${}^{L_2}\mathbf{x}_1$. For the transformation to relative

position and orientation, we then get

$$ {}^{L_2}\mathbf{s}_1 = {}^{L_2}_{G}\mathbf{R}({}^{G}\mathbf{s}_1 - {}^{G}\mathbf{s}_2), \tag{1} $$
$$ {}^{L_2}\mathbf{t}_1 = {}^{L_2}_{G}\mathbf{t}\,{}^{G}\mathbf{t}_1, \tag{2} $$

where ${}^{L_2}_{G}\mathbf{R}$ is the rotation matrix from frame $\{G\}$ to $\{L_2\}$ and $\mathbf{t}$ represents the orientation in Euler parameters (also known as quaternions) [28]. If we neglect vertical vehicle dynamics ($z,\ \varphi,\ \theta$) and only consider relative motion in the horizontal plane ($x,\ y,\ \psi$), the coordinate transformation in (1) and (2) simplifies to

$$ \begin{bmatrix} {}^{L_2}x_1 \\ {}^{L_2}y_1 \end{bmatrix} = \begin{bmatrix} \cos {}^{G}\psi_2 & \sin {}^{G}\psi_2 \\ -\sin {}^{G}\psi_2 & \cos {}^{G}\psi_2 \end{bmatrix} \left(\begin{bmatrix} {}^{G}x_1 \\ {}^{G}y_1 \end{bmatrix} - \begin{bmatrix} {}^{G}x_2 \\ {}^{G}y_2 \end{bmatrix} \right), \tag{3} $$
$$ {}^{L_2}\psi_1 = {}^{G}\psi_1 - {}^{G}\psi_2. \tag{4} $$

Please refer to Fig. 5 for a visual representation of this transformation. In a similar way, the transformations to relative velocity (${}^{L_2}\mathbf{v}_1,\ {}^{L_2}\dot{\psi}_1$) and relative acceleration (${}^{L_2}\mathbf{a}_1,\ {}^{L_2}\ddot{\psi}_1$) are derived [28]. For brevity, these derivations are omitted here.

The simulation is run by execution of entities on computing nodes, which are connected via a local area network. Each node has its own runtime environment, which also contains a representation of the virtual world. Entities communicate with this virtual world via their abstract sensors and actuators. The 'engine' of the entity simulation is an integrator (numerical solver), which invokes the entity's code (*i.e.*, the vehicle model) in timely manner (synchronised with other entities in real-time). The implementation of the system architecture is Java-based with time-critical parts in C/C++. An interface is established to MATLAB/SIMULINK: C code compiled from SIMULINK models can be embedded into the runtime environment as entities. More details on this modelling concept and the runtime environment are described in [27].

The multi-agent simulator provides the framework, in which *any* type of vehicle model can be simulated. The model complexity depends on the type of ADAS and the objective of the simulation. Based on the scenario categorisation in [26], a scenario library is available that contains traffic scenarios, such as car-following, tailgating, cut-ins, and lane-changes. The PreScan simulation tool, described in Section 2.2, is used for scenario definition and simulation before the actual VeHIL test takes place, based on the same multi-agent approach.

Table 1
Specifications of the chassis dyno.

Wheelbase	1.8 m to 4.0 m
Track width	1.2 m to 2.4 m
Drum configuration	4-wheel independent drive
Drum diameter	1592 mm
Total peak power	832 kW
Traction force	24 kN
Response time	< 10 ms
Maximum velocity	250 km/h
Dynamic range passenger cars (500 to 3500 kg)	full dynamics, $-10\,\mathrm{m/s^2}$ to $+10\,\mathrm{m/s^2}$
Dynamic range commercial vehicles ($\leq$12 000 kg)	reduced dynamics

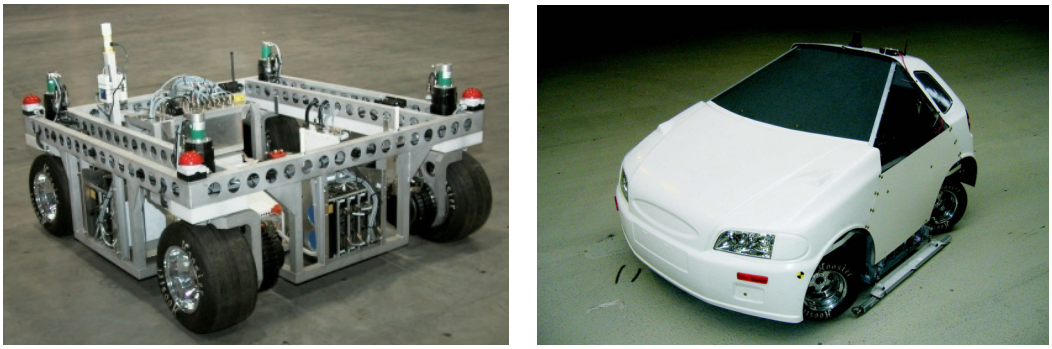
3.2 Substitution of a vehicle dynamics model by a vehicle under test

With the ADAS-equipped vehicle and other road users modelled, the real-time simulation could run as a MIL simulation only, *i.e.*, a PreScan simulation without hardware. However, a vehicle model is usually not sufficient to accurately represent the ADAS-equipped vehicle. In order to test a *real* intelligent vehicle in a HIL configuration, the vehicle model of entity E_2 is substituted by the real vehicle under test (VUT), hence the term '*vehicle* hardware in-the-loop'. The ADAS-equipped VUT is therefore placed on a chassis dynamometer that provides a realistic load for the vehicle's actuators (engine, brake system) and sensors (*e.g.*, wheel speed sensors).

The dynamic response of the chassis dyno, depicted in Fig. 8, to driving actions of the VUT must be representative of real road conditions, especially in terms of delay time and phase lag. The operating frequency of the multi-agent real-time simulator is 100 Hz, which means that the delay time is at the most an acceptable 10 ms. The dynamics of a passenger vehicle typically has a bandwidth in the 1 Hz frequency range. This implies that the chassis dyno must at least have a bandwidth of about 5 Hz in order to minimise positioning phase lag. Furthermore, an emergency stop of a passenger vehicle can cause a maximum deceleration of around $10\,\mathrm{m/s^2}$. Consequently, the chassis dyno must be able to achieve this as well.

These real-time requirements are met by a setup with four individual electric motor driven drums. The chassis dyno can fully simulate a vehicle mass between 500 and 3500 kg up to a maximum velocity of 250 km/h. The adjustable

Fig. 8. Vehicle under test with radar sensor at the front bumper, driving on the chassis dyno in VeHIL and supported by a rig at the front and back bumper.



(a) (b)

Fig. 9. Moving base: (a) without body and (b) with body.

wheelbase accommodates a wide range of vehicle types: apart from passenger vehicles, also trucks, busses, and other automated guided vehicles can be tested. Table 1 summarises the main specifications.

Note that the VUT itself replaces the vehicle model and the chassis dyno only needs to emulate the tire forces $F_{\text{tire},x,ij}$ that the VUT would encounter on the road. Each force $F_{\text{tire},x,ij}$ is emulated by the drum moment of inertia I_{drum} and the electric motor torque $T_{\text{drum},ij}$ as

$$F_{\text{tire},x,ij} = \frac{C_0 + C_1\omega_{\text{drum},ij} + C_2\omega_{\text{drum},ij}^2 + I_{\text{drum}}\dot{\omega}_{\text{drum},ij} - T_{\text{drum},ij}}{R_{\text{drum}}}, \qquad (5)$$

where the first three terms in the numerator represent friction losses in the chassis dyno, $\omega_{\text{drum},ij}$ is the measured drum speed, and R_{drum} the drum radius. From (5) the reference signals for the required motor torque are then calculated

as

$$T_{\mathrm{drum,\,ref},ij} = I_{\mathrm{drum}}\dot{\omega}_{\mathrm{drum},ij} + C_0 + C_1\omega_{\mathrm{drum},ij} + C_2\omega^2_{\mathrm{drum},ij} - \hat{F}_{\mathrm{tire},x,ij}R_{\mathrm{drum}}, \quad (6)$$

where $\hat{F}_{\mathrm{tire},x,ij}$ are observer estimated tire forces.

This setup also emulates the correct correlation between the individual drum speeds:

$$\omega_{\mathrm{drum},ij} = \frac{v_{\mathrm{wheel},x,ij}}{R_{\mathrm{drum}}}, \quad (7)$$

to enable simulation of different wheel speeds when driving through curves, where $v_{\mathrm{wheel},x,ij}$ are the velocity components at the individual wheels. In addition, a special restraint system that keeps the vehicle on top of the drums allows realistic heave and pitch motions of the vehicle body, as depicted in Figures 8 and 12(a). This rig produces a realistic dynamic vertical load transfer between rear and front axle during acceleration and deceleration.

Finally, a road load simulation model estimates the VUT state vector ${}^{G}\mathbf{x}_{2,\mathrm{VUT}}$ using the chassis dyno measurements and updates the state ${}^{G}\mathbf{x}_2$ of the associated object in the virtual world. No further interfacing between the real VUT and the simulation environment is necessary, such that the VUT can be tested as a black box system in a genuine HIL setup.

3.3 Substitution of a simulated target by a moving base

Similar to incorporation of the real VUT in a HIL simulation, surrounding road users can be represented by a so-called moving base, depicted in Fig. 9(a). The moving base is a 4-wheel driven, 4-wheel steered robot vehicle that responds to position commands of the multi-agent real-time simulator and emulates the motion ${}^{L_2}\mathbf{x}_1$ of other road users *relative* to the VUT, such that this motion is detected by the VUT's environment sensor. For this purpose, the soft real-time simulator (Ethernet network) and the hard real-time chassis dyno and moving bases (CAN bus) are linked through dedicated interfaces, indicated in Fig. 7. In order to carry out the desired relative maneuvers, the moving base must be able to perform motions that are not possible with a standard car (*e.g.*, sideways), as illustrated by the resulting velocity vector ${}^{L_2}\mathbf{v}_1$ in Fig. 5(b). For this reason the individual wheels can be steered in a range of $-350°$ to $+350°$.

Like the chassis dyno, the moving base should also have a control bandwidth of about 5 Hz in order to minimise positioning phase lag. In addition, the moving base should be capable of accelerating with $10\,\mathrm{m/s^2}$ in order to emulate the relative motion resulting from an emergency stop of the VUT. Finally, the top speed, which in view of the relative VeHIL world corresponds to the maximum

Table 2
Specifications of a moving base.

Vehicle mass (including body)	650 kg
Wheelbase	1.4 m
Track width	1.4 m
Chassis configuration	4-wheel independent drive/steer from -350° to $+350^\circ$
Installed power	52 kW
Battery pack	288 NiMH D-cells, 375 V, 100 kg
Maximum velocity	50 km/h
Maximum longitudinal acceleration	$10\,\mathrm{m/s^2}$
Maximum longitudinal deceleration	$-10\,\mathrm{m/s^2}$
Acceleration from 0 to 50 km/h	2.1 s
Maximum centripetal acceleration	$12\,\mathrm{m/s^2}$

relative velocity, should at least be equal to 50 km/h. This covers almost all highway scenarios [26].

These requirements are met by a vehicle platform equipped with independent all-wheel steering and all-wheel drive, using battery-powered DC servomotors. The trajectory controller of the moving base realises the desired trajectory $\mathbf{x}_{\mathrm{MB,ref}}(t)$, defined by the relative motion ${}^{L_2}\mathbf{x}_1(t)$ of the target vehicle in the horizontal plane. The only conditions are that the trajectory ${}^{L_2}\mathbf{x}_1(t)$ fits within the dimensions of the VeHIL laboratory (200 m by 40 m) and meets the specifications of Table 2. The moving base controller determines the drive torques and steering torques so as to minimise the difference between the actual and desired moving base position, such that a repeatable trajectory is achieved within a position error ε of at most 0.10 m, depending on the dynamics of the scenario. The moving base navigation system uses a combination of a magnet grid and odometry with a measurement accuracy of 0.04 m, resulting in a total positioning accuracy of (0.10 ± 0.04) m. For more information on the design and control of the moving base, the reader is referred to the work by Ploeg *et al.* [29] [30].

In order for the VUT to obtain realistic sensor data, the moving base is equipped with a vehicle body that represents similar target characteristics as a real vehicle, see Fig. 9(b). Its radar cross section is similar to that of a standard passenger car, and the body has a similar shape and reflection properties for testing vision and lidar systems.

Subsequently, the ADAS controller receives realistic input signals through its

vehicle state sensors and environment sensors, and outputs command signals to the vehicle actuators (engine, brake) with a realistic actuator load, just as if the VUT was driving on the road. It must be emphasised that the actual moving base motion in VeHIL is *not* known *a priori*, but is the real-time equivalent of the resulting relative motion between an autonomously simulated target vehicle and an ADAS-controlled VUT. For example, when the VUT makes an emergency stop with deceleration $a_{2,\mathrm{VUT}}$, the moving base accelerates forward with $a_{1,\mathrm{MB}} = -a_{2,\mathrm{VUT}}$. In this way a *closed-loop* HIL simulation is obtained, such that the ADAS is validated in an artificial traffic environment, including real vehicle dynamics and real sensor input.

3.4 Representativeness of VeHIL

A fundamental aspect of a HIL test environment is that it provides a *repeatable* and *representative* testing environment. As we have demonstrated, the error variance and bandwidth of the moving base and chassis dyno are within the noise levels of environment sensor systems, such that VeHIL can be regarded as a repeatable testing environment.

Furthermore, the input from the artificial VeHIL environment into the VUT must be representative for the actual driving conditions on the road. A restriction of the VeHIL simulation in this respect is that vehicle-based inertial sensors (accelerometers and yaw rate sensors) do not give a representative signal, since the VUT is held at a stationary position. Another restriction is that the chassis dyno does not produce correct lateral tire forces $F_{\mathrm{tire},y,ij}$ during steering actions of the VUT, since the slip angles of the front wheels α_{1j} equal the wheel angles δ_{1j}. However, the resulting relative lateral and yaw motion can still be correctly emulated, as shown in Fig. 5(b). On the road, environment sensors can be perturbed by obstacles outside the relevant area (*e.g.*, infrastructure elements outside the path of motion). Much of the effort in sensor post-processing is associated with filtering out these disturbances. In VeHIL these disturbances can be different from the real world or even absent, although the absence of these disturbances does not affect the basic operation of the ADAS.

To solve these issues, the HIL concept allows to feed the ADAS in real-time with a 'mixture' of real and virtual sensor signals. Any missing sensor signal can be generated from the real-time simulation of the vehicle model in the multi-agent real-time simulator (the internal dynamics of entity E_2). This signal then replaces the real sensor signal and is subsequently fed into the ADAS controller, as schematically presented in Fig. 4. For example, imaging sensors and detection algorithms could experience problems due to the absence of moving lane markings. Instead, simulated road scenarios can be projected on

a screen in front of the camera in real-time, as will be illustrated in Section 4.

Alternatively, inertial and environment sensors can be installed on a moving base that executes a traffic scenario as if it were a standard road vehicle, while another moving base represents a target vehicle, as shown in Fig. 5(c). This setup also allows to obtain a relative velocity of up to 100 km/h, when two moving bases drive towards each other.

Due to the absence of a realistic driving environment, VeHIL is not intended to serve as a driving simulator, although it has potential to include driver interaction, as will be illustrated in Section 4. VeHIL is therefore not meant to replace test drives, but focusses on repeatable and accurate testing of the ADAS performance and dependability before 'human-in-the-loop' test drives take place. In addition, VeHIL tests are used for those scenarios that are too difficult or dangerous to perform on the road.

3.5 Added value of VeHIL in the development process of ADASs

By providing a world-wide unique HIL environment for intelligent vehicle systems, the VeHIL laboratory offers a number of distinct advantages:

- Tests are performed in a repeatable and flexible way with high accuracy, since the moving bases are operated from a computer-controlled environment. This allows precise variation of test parameters to assess the influence of specific parameters and failure modes on the ADAS performance.
- Tests are safer, due to the absence of high absolute velocities. Furthermore, traffic scenarios are monitored by a supervisory safety system, which prevents any real collisions. This allows to test ADASs in safety-critical scenarios.
- The costs of the validation process are reduced, because many tests are performed in a short time frame. The VUT can drive for hours and be continuously tested, which is not possible during test drives. Depending on the complexity of the scenarios, up to 20 tests per hour can be performed, including scenario compilation, trial runs, test execution, and data acquisition. A test cycle is therefore significantly faster than is possible with test drives [10]. This will also be demonstrated in the case studies.
- Due to the high accuracy of tests, a high success rate for testing is obtained.

Because of these advantages, VeHIL complements the existing development process of ADASs in many phases and on many of the levels of the 'V'-diagram of Fig. 3:

- Rapid control prototyping in VeHIL can help to define the system specifications in an early development stage. In addition, based on safety-critical

maneuvers and fault injection, potential hazards can be analysed.

- The flexible transition from MIL simulation in PreScan to HIL simulation in VeHIL allows a model-based development of the controller. Critical scenarios that are identified with MIL simulations can be quickly uploaded in VeHIL for experimental testing. Test results can then be compared with the simulation results for validation of sensor and vehicle models.
- On the module level the ability to combine high position accuracy with high and accurate relative speeds makes VeHIL an efficient tool in verification and benchmarking of the exact performance of environment sensors (*e.g.*, sensor calibration).
- On the system level VeHIL especially facilitates the *functional* validation of the performance and dependability of complex black-box controllers against objective measures. Algorithm evaluation, fine-tuning, and benchmarking can be done efficiently.
- For production sign-off and certification purposes the high repeatability and ability to deal with safety-critical applications make VeHIL a strong tool.
- Finally, VeHIL facilitates the transition from simulations to outdoor test drives that are used to evaluate the real performance and dependability on the road. These test drives can be performed with a much higher confidence and less risk, when the ADAS has already been thoroughly tested in VeHIL.

We will demonstrate the suitability and added value of VeHIL in the next section with a case study.

4 Case study: Validation of a driver information and warning system

This section demonstrates the application of the validation methodology that was developed in [26] using VeHIL simulations. The subject in the present case study is a novel driver information and warning system for safe speed and safe distance (SASPENCE), which has been developed within the Integrated Project PReVENT (Preventive and Active Safety Applications), co-funded by the European Commission under the Sixth Framework Programme [31]. The goal of the PReVENT subproject SASPENCE is the development and evaluation of a 'Safe Speed and Safe Distance' application that supports the driver in avoiding potentially dangerous situations and that improves driver comfort [32]. The SASPENCE system should cooperate unobtrusively with the driver by suggesting a safe speed and safe distance to keep, relative to the vehicle in front [33]. In addition, the system gives a speed advice, taking into account speed limits, road infrastructure, and weather conditions.

Table 3
Traffic scenarios and operating conditions for the SASPENCE system.

No.[a]	Scenario description	Warning level w
1	Host vehicle breaks speed limit	1
2	Critical weather conditions present (*e.g.*, fog, heavy rain, snow)	1
3	Obstacle appears ahead, but not on the host path	0
4	Obstacle appears ahead on the host path, without being dangerous	0
5	Obstacle appears ahead and could become dangerous	2
6	Obstacle appears ahead and is dangerous	3
7	On-coming vehicles approaching (on one-way rural road)	1
8	Host vehicle approaches hazardous infrastructure too fast (*e.g.*, sharp bend, traffic light, or pedestrian crossing)	1

[a]See Fig. 10 for an illustration of these scenarios (except scenario 7).

4.1 A system for safe speed and safe distance

Since speeding and tailgating are widespread traffic hazards, the SASPENCE system considers a wide range of traffic scenarios and operating conditions, as listed in Table 3. Some of these scenarios are illustrated in Fig. 10. The table also indicates the corresponding system response in terms of the warning level w. Warning level 0 means that no warning is given, level 1 is an information mode, level 2 a mild warning, and level 3 an emergency warning.

Since the safety potential that is expected from a system capable to appropriately warn the driver in case of excessive speed and small headway looks very promising [34], it is of paramount importance to accelerate the deployment of such an ADAS. In order to have a system ready for the short-term market, the SASPENCE project aims to develop a low-cost system by combining ADAS components that are already available in modern passenger cars (such as components for ACC, lane departure warning, and satellite navigation). The corresponding system architecture based on existing hardware components is therefore presented next.

The SASPENCE system is installed in a Fiat Stilo Multiwagon that serves as one of the two demonstrator vehicles for this project. In the system architecture of Fig. 11 several modules can be distinguished. The sensor array of the SASPENCE system consists of a long-range ACC radar for obstacle

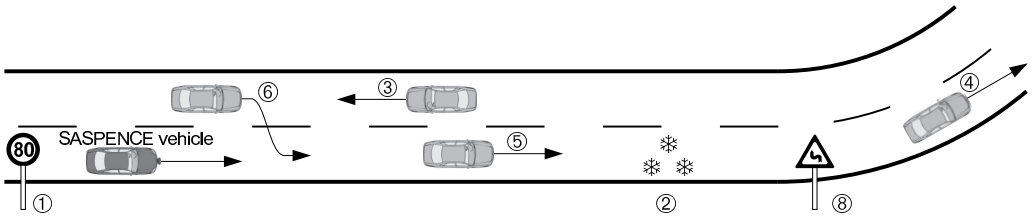


Fig. 10. Overview of scenarios for the SASPENCE system.

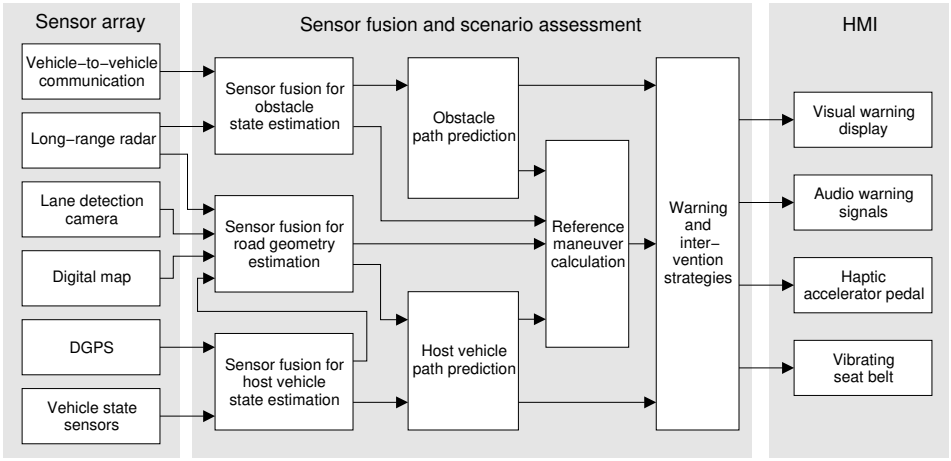


Fig. 11. System architecture of the SASPENCE system.

detection, mounted on the front of the vehicle. Lane recognition by video image processing is used to distinguish potentially dangerous obstacles from objects in adjacent lanes and on the side of the road. In addition, vehicle-to-vehicle communication enhances the selection of relevant targets. Differential GPS (DGPS) combined with digital map navigation is used for global state estimation and for providing information on speed limits and relevant infrastructure [35]. Several human-machine interface (HMI) channels are available to provide information and warnings to the driver: a haptic accelerator pedal (trough force feedback and pedal vibration), a visual warning display, seat belt vibration, and audio signals. In-vehicle networking between the sensor array, the signal processing modules, and the HMI is primarily provided by a dedicated CAN bus.

Sensor data is fused at multiple levels to provide an enhanced view of the environment. Sensor fusion of DGPS and vehicle state sensors based on extended Kalman filtering provides an estimate of the host vehicle's global state ${}^{G}\mathbf{x}$. A precise estimation of the road course ahead is created by fusion of navigational map points and lane detection information [36]. All detected vehicles are projected into the estimated road geometry to determine their relative positions to the host vehicle and their predicted paths [37].

The output of the sensor fusion and path prediction modules is then used to

compute an optimal reference maneuver by solving the optimisation problem

$$J = \frac{1}{S} \int_0^S f\Big(\mathbf{x}(s), \mathbf{u}(s)\Big) ds, \tag{8}$$

subject to a given set of inequality constraints (used to impose trajectory constraints) and equality constraints (includes the vehicle model with $\mathbf{x}(s)$ the vehicle state vector). The functional J has to be minimised over the planning distance interval $[0,\, S\,]$ by finding the control functions $\mathbf{u}(s)$ (steering, throttle, and brake input). The penalty function $f\,(\mathbf{x}(s), \mathbf{u}(s))$ is used to define the driving style. In addition, the penalty function can be considered a risk performance measure, where the integral is a measure for the risk level of the maneuver. Apart from these safety considerations, requirements for user acceptance and mobility are included in the penalty function. This allows the SASPENCE system to compute an appropriate speed and safe distance to the preceding vehicle, as well as to consider speed limits and weather conditions.

The optimal reference maneuver is then compared to the predicted path of the host vehicle. In case the difference crosses a threshold, the system intervenes by giving information and/or warnings to the driver. The warning and intervention module computes the appropriate warning type and warning level, and directs this to the available HMI channels. For more information on the calculation of the reference maneuver for the SASPENCE system, we refer to the work by Biral *et al.* [38]. In the remainder of this section we focus on the validation of the system's capability for giving appropriate and reliable warnings in response to potentially dangerous obstacles ahead.

4.2 Definition of the validation objectives

The first step in the validation process is to define suitable evaluation criteria from the system requirements, with emphasis on measures that relate to an appropriate interaction with the driver. The dependability is assessed through the missed alarm rate p_{FN} and false alarm rate p_{FP}. For each individual scenario j, these performance measures are calculated by comparison of the output of the SASPENCE system w_j with a reference system $w_{\text{ref},j}$, which gives information on the warning level and when it *should* be given from the perspective of an average attentive driver. This reference warning is based on empirical data from the CAMP project [39], that indicates the most appropriate warning time and warning level of a forward collision warning system for a representative set of drivers.

The associated dependability can then be estimated according to the methodology of Gietelink [26]. Furthermore, the timeliness of a scenario j indicates

to what extent the warning is given too soon or too late:

$$e_{\text{time},j} = t_{\text{warn},j} - t_{\text{ref},j} \tag{9}$$

with $t_{\text{warn},j}$ the first sample for which $w_j > 0$ (*i.e.*, a warning) and $t_{\text{ref},j}$ the first sample for which $w_{\text{ref},j} > 0$.

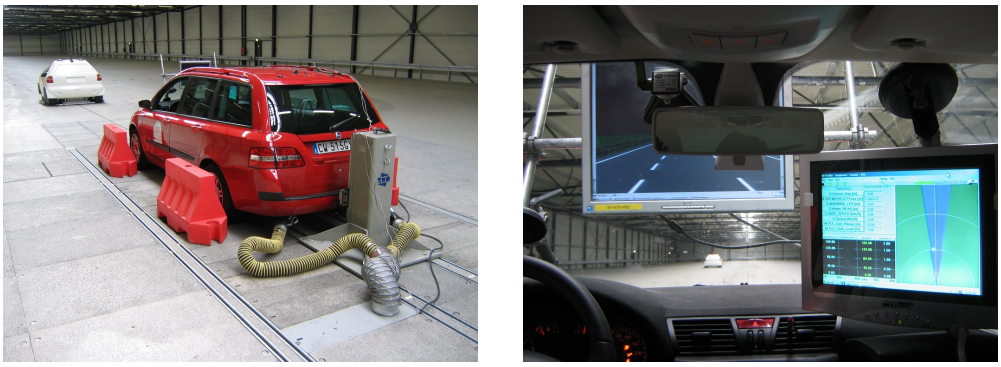
The impact of the system on traffic safety and driver comfort is validated by comparing a scenario where a driver is assisted by the SASPENCE system to the same scenario where the SASPENCE system is not operational. The safety is measured in terms of the minimum time-to-collision (TTC) during the scenario and the comfort in terms of the RMS value of the longitudinal acceleration.

With respect to the dependability of the system, we require a maximum false alarm rate p_{FP} in the order of $1 \cdot 10^{-4}$, and a maximum missed alarm rate p_{FN} in the order of $1 \cdot 10^{-3}$. These values are chosen in accordance with the state-of-the-art values discussed in Section 2. The objective of this case study is to validate the comfort, performance, and dependability of the warning functions of the SASPENCE system against the system requirements. The system must conform to the requirements for a representative set of traffic scenarios, operating conditions, and driver characteristics.

The value of these performance measures for a particular traffic scenario j obviously depends on the perturbations imposed by that scenario. Based on the system specifications [35], a parameter set is defined, composed of traffic scenarios, operating conditions, sensor characteristics, and driver characteristics. For the traffic scenario parameters we use the microscopic traffic model developed in [26]. The measurement noise of the environment sensors is taken from the system specifications [33]. The driver is modelled for conventional car-following behaviour after the Gipps model [40], using three different driver types: conservative, intermediate, and aggressive drivers.

4.3 Functional validation with VeHIL tests

Obviously, in practice simulations have their limitations with regard to the credibility of the results. Driving simulator tests have therefore been carried out in [41], but these do not take into account the actual hardware of the vehicle and the SASPENCE system. On the other hand, driving tests with the demonstrator vehicles have also been performed [42], but they are limited in their ability to test safety-critical scenarios. To provide a preliminary functional validation of the SASPENCE system in an early stage of its development, the most critical scenarios that were identified with the simulator study are therefore selected to be reproduced in the VeHIL laboratory.



(a) (b)

Fig. 12. VeHIL laboratory setup: (a) the SASPENCE vehicle is set up on the chassis dyno (beneath the floor), and approaches another road user, represented by the moving base; (b) the real-time traffic scene is projected on a display in front of the camera system (located behind the rear-view mirror), while a preliminary HMI display shows the output of the SASPENCE system.

As shown in Fig. 12(a), the Fiat demonstrator vehicle is mounted on the chassis dyno to emulate the tire-road interaction and the moving base is used to emulate the preceding vehicle. The visual input to the camera system is emulated by projecting a real-time animation of the traffic scene in front of the camera, as shown in Fig. 12(b). Similarly, DGPS satellite navigation and vehicle-to-vehicle communication are emulated by a real-time ethernet link from the real-time simulation environment.

Since the SASPENCE system is a driver warning system, a closed-loop configuration requires that a driver reacts to warnings and takes appropriate action. The prototype vehicle is therefore instrumented with a driving robot, consisting of two actuators to control the brake and throttle pedal positions. The driving robot is linked to the driver model, as illustrated in Fig. 13. The driver model receives real-time information on the host absolute state ${}^{G'}\mathbf{x}_2$ and the relative target motion ${}^{L_2}\mathbf{x}_1$ from the simulation environment. The driver model also receives the warning level w from the SASPENCE system, such that it can calculate a desired speed v_{ref}, which is sent to the actuator controller of the driving robot. Hence, the experiment is a *closed-loop* hardware-in-the-loop simulation. The flow of information between these components is illustrated in the schematic diagram in Fig. 13.

Using the results of the simulation study, a test schedule has been developed for a representative set of traffic scenarios. The test schedule is biased towards scenarios that are considered more critical, *i.e.*, with a lower value for the minimum TTC, as obtained from the simulation study. Correspondingly, scenario parameters are selected, such as the velocities $v_1(0)$ and $v_2(0)$, target acceleration a_1, initial distance $x_{\mathrm{r}}(0)$, and initial lateral offset $y_{\mathrm{r}}(0)$. Fig. 14 shows an overview of these parameters, where the SASPENCE vehicle, driving at

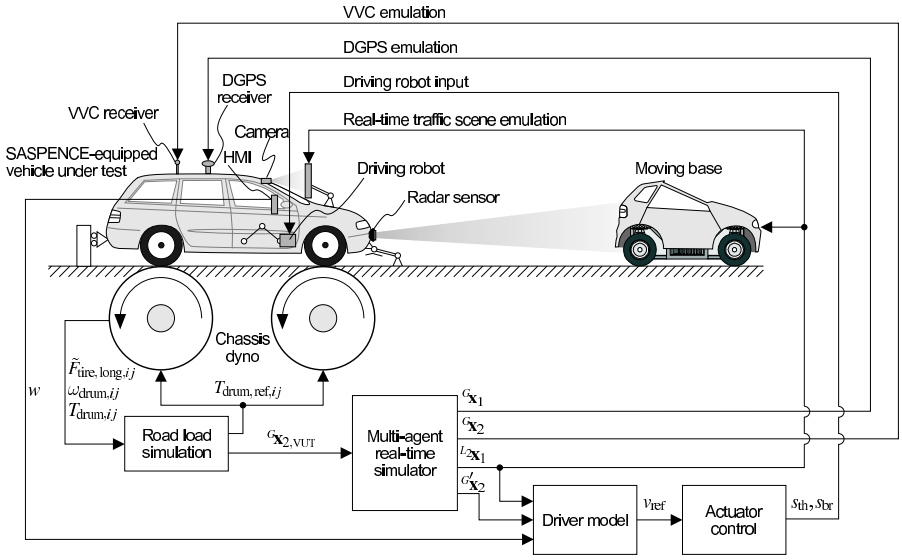


Fig. 13. Schematic representation of the closed-loop VeHIL setup for the SASPENCE vehicle. The inputs from the driver, camera, DGPS, and vehicle-to-vehicle communication (VVC) are emulated from the real-time simulation environment, whereas the relative motion for the radar sensor is emulated by the moving base.

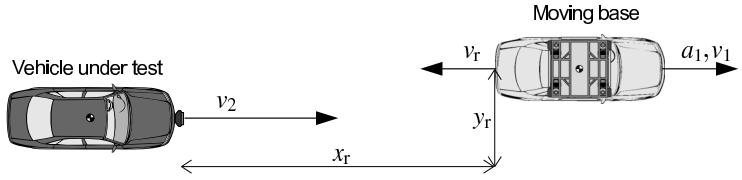


Fig. 14. Overview of the scenario parameters in an approach scenario.

a velocity v_2, approaches a slower target vehicle, driving at v_1. The resulting relative motion for the moving base is $v_{\mathrm{MB}} = v_{\mathrm{r}} = v_1 - v_2$. Around 150 tests were carried out, for each of which the performance criteria were calculated. For comparison of the VeHIL experimental results with the PreScan simulation results, the approach scenarios (Scenario 5) were carried out both *with* and *without* the SASPENCE system. On a test track it would be very difficult to safely and reproducibly carry out such a test with human drivers, but in VEHIL the scenario can be accurately reproduced.

4.4 Experimental results

Fig. 15 illustrates these VeHIL test results for a typical approach scenario with the SASPENCE-equipped vehicle approaching a slower target vehicle. The initial velocities for this test are $v_{\mathrm{VUT}}(0) = v_2 = 33.3\,\mathrm{m/s}$ and $v_1(0) = 22.2\,\mathrm{m/s}$, and the initial distance $x_{\mathrm{r}}(0) = 0$. It can be seen that at $t = 5.1\,\mathrm{s}$, the time-to-collision drops below 6 s, which causes the reference algorithm to give a

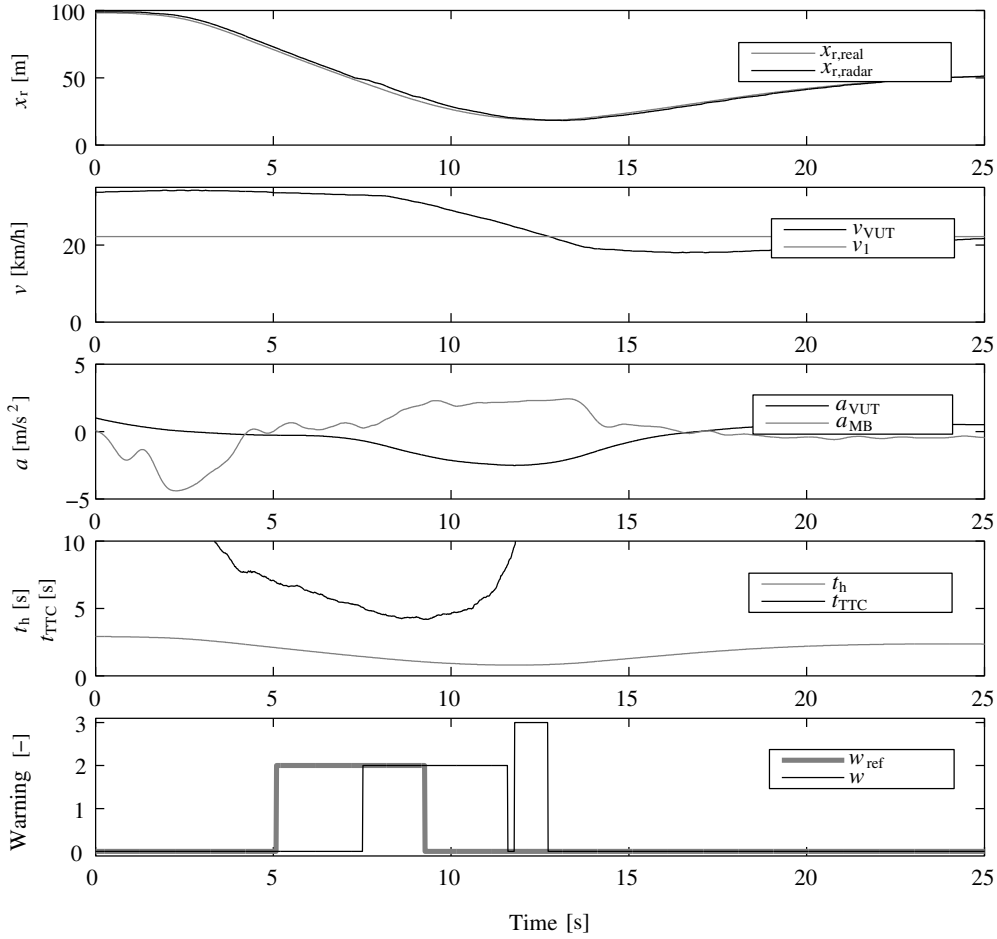


Fig. 15. VeHIL test results for an approach scenario.

warning signal w_{ref}. However, only at $t = 7.5\,\text{s}$ the SASPENCE system gives a warning, after which the driving robot decelerates the vehicle. As the vehicle decelerates, the TTC rises again from $t = 9.3\,\text{s}$ onwards, indicating that a collision is being averted. Correspondingly, the reference signal w_{ref} disappears. However, the SASPENCE warning signal w remains present, and the warning level is even increased at $t = 11.8\,\text{s}$, even though the distance x_r is increasing again.

Of course, the choice of the reference model is quite arbitrary, and a more conservative or a more sensitive algorithm might be selected. Nevertheless, the difference between the reference warning w_{ref} and the SASPENCE warning w is used to illustrate the dependability validation. Fig. 16 shows the distribution of the timeliness of the warning. On average, the SASPENCE warning is given at approximately the same time instance as the reference warning, which indicates that the reference warning w_{ref} has acceptable behaviour. However, the time difference ranges between 5 s too soon and 5 s too late. This variety in warning timeliness means that the presence or absence of a warning might be interpreted by the driver as a false or missed alarm.

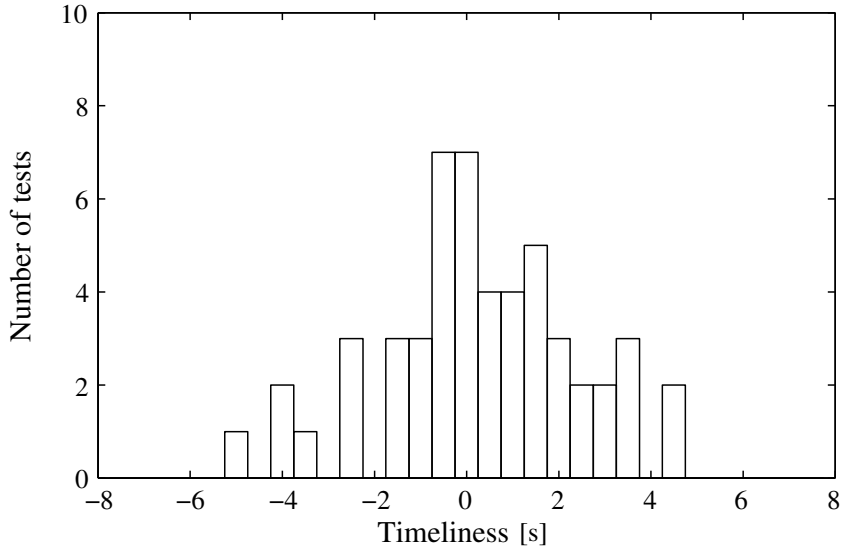


Fig. 16. Distribution of the timeliness of the warning (negative means too soon, positive too late).

If we look again at Fig. 15, the lack of a SASPENCE warning in the time interval $t = [5.1, 7.5]$ indicates a missed alarm. Similarly, the time interval [9.3, 12.7] indicates a false alarm. By combining the test results for all scenarios, it is possible to obtain a representative overview of the dependability of the SASPENCE system. The results show that the estimated missed alarm rate is $\hat{p}_{\text{FN}} = 0.021$ and the estimated false alarm rate $\hat{p}_{\text{FP}} = 0.011$, *i.e.*, the reliability measures are all in the order of 10^{-2}.

These results show that in practice the dependability and timeliness of the SASPENCE warning function must be improved. This could also be observed during the VeHIL tests, where occasionally the SASPENCE system did not consider safety-critical scenarios threatening. Vice versa, warnings were sometimes given, even when the preceding vehicle did not pose any threat (it was far away or driving away from the vehicle under test). Because of the inherent trade-off between false and missed alarms for a given detection accuracy, it is not possible to simply lower or raise the obstacle detection thresholds. Instead, the path prediction and reference maneuver algorithms should be further fine-tuned to reduce the above-mentioned probabilistic values.

Despite the fact that the dependability of the system must be improved, the effectiveness of SASPENCE in terms of traffic safety and driver comfort can still be validated for the scenarios with a correct alarm. For this purpose the experimental results are compared for scenarios *with* and *without* the SASPENCE system, as well as for different driver types (conservative, medium or aggressive), considering *equal* initial conditions for all six experiments.

In Table 4, the minimum TTC that occurs during an approach scenario is displayed for different driver types. From the table it can be concluded that

Table 4
VeHIL test results for the minimum TTC [s] at different initial relative velocities $v_r(0)$ [m/s] for different drivers.

		Initial relative velocity $v_r(0)$ [m/s]				
Driver type	SASPENCE	−2.8	−5.6	−8.3	−11.1	−13.9
Conservative	off	*	7.35	7.22	2.89	4.76
	on	20.22	13.79	9.41	6.85	4.84
Intermediate	off	11.83	7.15	5.33	3.87	1.86
	on	16.00	9.77	7.21	5.18	3.87
Aggressive	off	12.17	2.01	3.40	4.18	*
	on	*	*	*	1.84	2.98

*Data not available.

Table 5
VeHIL test results for the RMS of the longitudinal acceleration [m/s^2] at different initial relative velocities $v_r(0)$ [m/s] for different drivers.

		Initial relative velocity $v_r(0)$ [m/s]				
Driver type	SASPENCE	−2.8	−5.6	−8.3	−11.1	−13.9
Conservative	off	*	0.64	0.87	0.71	0.88
	on	0.17	0.15	0.26	0.70	0.99
Intermediate	off	0.48	0.10	0.80	0.90	1.00
	on	0.11	0.17	0.26	0.69	1.05
Aggressive	off	0.24	0.66	0.64	0.97	*
	on	*	*	*	0.62	1.12

*Data not available.

the SASPENCE system has a positive effect on the safety of conservative and medium drivers, since the minimum TTC increases for them. Not enough consistent results for the aggressive driver were available to validate the benefit of the system for these drivers.

The effect that the SASPENCE system has on comfort is expressed in the RMS value of the longitudinal acceleration. VeHIL test results are given in Table 5. This table shows that the RMS value of the acceleration generally decreases when using the SASPENCE system, which means that the SASPENCE system also increases driver comfort.

Obviously, in the end outdoor test drives are still necessary to evaluate the system's performance on the road and to provide a subjective assessment of the SASPENCE system. However, these tests can now be focussed on specific problem areas, since the system has already been thoroughly tested for a large number of scenarios in PreScan and VeHIL. These test drives can be used to evaluate the performance and dependability over a longer period of time. This will serve as validation of the expected probabilistic values from the simulation study and the VeHIL test results. In addition, test drives will be used to assign a subjective rating to the system and to test the HMI. It is a topic of ongoing research to carry out these test drives and to compare the test results with those presented in this chapter [42].

5 Conclusions

This paper has presented the new VeHIL concept for testing ADASs, where a real intelligent vehicle is operated in a HIL environment. VeHIL experiments are performed in an accurate, reproducible, and controllable way to create a representative test environment. Furthermore, tests are performed more efficiently than with outdoor test drives, and test scenarios can be varied very easily, due to the connection to the underlying simulation environment.

It was demonstrated that VeHIL has an added value in the development process of an ADAS, using a case study of a driver information and warning system for safe speed and safe distance (SASPENCE). Results of the VeHIL experiments show that the warning and intervention strategies need to be fine-tuned to further improve the dependability of the system. Based on these findings, the scenario assessment modules of the SASPENCE system can be modified.

Subsequent test drives can then be performed with a much higher confidence in the system, since the SASPENCE system has already been thoroughly tested in VeHIL. VeHIL is therefore *not* meant to replace MIL simulations and test drives, but to form an efficient link between them. Consequently, the number of iteration loops in the development process is reduced, saving on time and costs.

References

[1] International Road Traffic Database (IRTAD).

[2] S. Shladover, Review of the state of development of advanced vehicle control systems, International Journal of Vehicle System Dynamics 24 (6–7) (1995) 551–595.

[3] R. Bishop, Intelligent Vehicle Technology and Trends, Artech House, Norwood, MA, USA, 2005. ISBN: 1-58053-911-4.

[4] P. Zador, S. Krawchuk, R. Voas, Automotive collision avoidance system (ACAS) program, Final Report DOT HS 809 080, DOT/NHTSA, Washington, DC, USA (Aug. 2000).

[5] H. Jagtman, V. Marchau, T. Heijer, Current knowledge on safety impacts of Collision Avoidance Systems (CAS), in: P. Herder, W. Thissen (Eds.), Proceedings of the 5th International Conference on Technology, Policy and Innovation, Delft, 2001, paper number: 1152.

[6] J. Golias, G. Yannis, C. Antoniou, Classification of driver-assistance systems according to their impact on road safety and traffic efficiency, Transport Reviews 22 (2) (2002) 179–196.

[7] H. Winner, S. Witte, W. Uhler, B. Lichtenberg, Adaptive cruise control system aspects and development trends, SAE Technical Paper Series 961010.

[8] M. Flament, Integrated passive and active safety solutions, in: Proceedings of the 5th European MADYMO User Meeting, Cambridge, UK, 2005.

[9] NHTSA, Automotive collision avoidance system field operational test (ACAS FOT), Final Program Report DOT HS 809 866, DOT/NHTSA, Washington, DC, USA (May 2005).

[10] R. Kiefer, D. LeBlanc, M. Palmer, J. Salinger, R. Deering, M. Shulman, Development and validation of functional definitions and evaluation procedures for collision warning/avoidance systems, Final Report DOT HS 808 964, DOT/NHTSA, Washington, DC, USA (Aug. 1999).

[11] P. Seiler, B. Song, J. Hedrick, Development of a collision avoidance system, SAE Technical Paper Series 98PC-417.

[12] A. Doi, T. Butsuen, T. Niibe, T. Yakagi, Y. Yamamoto, H. Seni, Development of a rear-end collision avoidance system with automatic braking control, JSAE Review 15 (4) (1994) 335–340.

[13] J. Pimentel (Ed.), Safety-Critical Automotive Systems, No. PT-103, SAE International, Warrendale, PA, USA, 2006. ISBN: 0-7680-1243-0.

[14] R. Isermann, Mechatronic Systems, Springer-Verlag, London, 2003. ISBN: 1-85233-693-5.

[15] J. Schwarz, editor, Code of Practice for the design and evaluation of ADAS, RESPONSE 3 – PReVENT subproject, Brussels, Belgium (Oct. 31, 2006).

[16] International Organization for Standardization (ISO), Transport information and control systems — forward vehicle collision warning systems — performance requirements and test procedures, ISO Standard 15623, First edition 2002-10-01, Geneva, Switzerland (2002).

[17] R. Abou-Jaoude, ACC radar sensor technology, test requirements, and test solutions, IEEE Transactions on Intelligent Transportation Systems 4 (3) (2003) 115–122.

[18] C. Hôte, Abstract interpretation techniques for software testing, Business Briefing: Global Automotive Manufacturing & Technology (2002) 1–7.

[19] Z. Papp, K. Labibes, A. Thean, M. van Elk, Multi-agent based HIL simulator with high fidelity virtual sensors, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Columbus, OH, USA, 2003, pp. 213–219.

[20] J. Li, F. Yu, J.-W. Zhang, J.-Z. Feng, H.-P. Zhao, The rapid development of a vehicle electronic control system and its application to an antilock braking system based on hardware-in-the-loop simulation, Proceedings of the Institution of Mechanical Engineers – Part D: Journal of Automobile Engineering 216 (2002) 95–105.

[21] R. Isermann, J. Schaffnit, S. Sinsel, Hardware-in-the-loop simulation for the design and testing of engine-control systems, Control Engineering Practice 7 (5) (1999) 643–653.

[22] R. Isermann, Diagnosis methods for electronic controlled vehicles, International Journal of Vehicle System Dynamics 36 (2–3) (2001) 77–117.

[23] K. Athanasas, C. Bonnet, H. Fritz, C. Scheidler, G. Volk, VALSE – validation of safety-related driver assistance systems, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Columbus, OH, USA, 2003, pp. 610–615.

[24] K. Yi, M. Woo, S. Kim, S. Lee, Study on a road-adaptive CW/CA algorithm for automobiles using HiL simulations, JSME International Journal Series C – Mechanical Systems Machine Elements and Manufacturing 42 (1) (1999) 163–170.

[25] G. Reymond, A. Heidet, M. Canry, A. Kemeny, Validation of Renault's dynamic simulator for adaptive cruise control experiments, in: Proceedings of the Driving Simulation Conference (DSC2000), Paris, France, 2000, pp. 181–192.

[26] O. Gietelink, Design and validation of advanced driver assistance systems, T2007/11, TRAIL Thesis Series, Delft University of Technology, Delft, The Netherlands (Nov. 2007).

[27] Z. Papp, M. Dorrepaal, D. Verburg, Distributed hardware-in-the-loop simulator for autonomous continuous dynamical systems with spatially constrained interactions, in: Proceedings of the 11th IEEE/ACM International Workshop on Parallel and Distributed Real-Time Systems, Nice, France, 2003.

[28] J. Craig, Introduction to Robotics, Mechanics and Control, 3rd Edition, Addison-Wesley, Boston, MA, USA, 2004. ISBN: 0-20110-326-5.

[29] J. Ploeg, A. van der Knaap, D. Verburg, ATS/AGV, design, implementation and evaluation of a high performance AGV, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Vol. 1, Versailles, France, 2002, pp. 127–134.

[30] J. Ploeg, J. Vissers, H. Nijmeijer, Control design for an overactuated wheeled mobile robot, in: Proceedings of the 4th IFAC Symposium on Mechatronic Systems (MECHATRONICS 2006), Heidelberg, Germany, 2006.

[31] Integrated Project PReVENT – Preventive and Active Safety Applications.

[32] PREVENT-SASPENCE Consortium, SASPENCE project description – appendix of PReVENT Technical Annex, European Commission, IP PReVENT, Subproject SASPENCE, Brussels, Belgium (2003).

[33] F. Tango, A. Saroldi, System specifications, Deliverable D20.34, European Commission, IP PReVENT, Subproject SASPENCE, Brussels, Belgium (Jun. 9, 2005).

[34] F. Tango, A. Saroldi, Towards a new approach in supporting drivers function: specifications of the SASPENCE system, in: Proceedings of the 5th European Congress and Exhibition on Intelligent Transport Systems and Services (ITS), Hannover, Germany, 2005, pp. 614–620.

[35] M. Alonso, P. Garayo, L. Herrán, Functional requirements, Deliverable D20.33, European Commission, IP PReVENT, Subproject SASPENCE, Brussels, Belgium (Nov. 30, 2004).

[36] H. Weigel, H. Cramer, G. Wanielik, A. Polychronopoulos, A. Saroldi, Accurate road geometry estimation for a safe speed application, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Tokyo, Japan, 2006, pp. 516–521.

[37] N. Floudas, M. Tsogas, A. Amditis, H. Weigel, Positioning and path prediction for scenario assessment of safe speed system, in: Proceedings of the 14th World Congress on Intelligent Transport Systems and Services (ITS), Beijing, P.R. China, 2007.

[38] F. Biral, M. Da Lio, E. Bertolazzi, Combining safety margins and user preferences into a driving criterion for optimal control-based computation of reference maneuvers for an ADAS of the next generation, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2005, pp. 36–41.

[39] R. Kiefer, M. Cassar, C. Flannagan, D. LeBlanc, M. Palmer, R. Deering, M. Shulman, Forward Collision Warning Requirements Project Task 1, Final Report DOT HS 809 574, DOT/NHTSA, Washington, DC, USA (Jan. 2003).

[40] P. Gipps, Behavioural car-following model for computer simulation, Transportation Research B 15B (2) (1981) 105–111.

[41] L. Bruel, J.-P. Colinot, E. Adell, A. Várhelyi, M. Dalla Fontana, M. D'Alessandro, HMI tests in simulator, Deliverable D20.54, European Commission, IP PReVENT, Subproject SASPENCE, Brussels, Belgium (Jul. 4, 2006).

[42] O. Gietelink, M. Lammers, S. Jansen, SASPENCE validation results, Deliverable D20.70, European Commission, IP PReVENT, Subproject SASPENCE, Brussels, Belgium, to be published (2007).