Simulations on Consumer Tests: A Systematic Evaluation Approach in an Industrial Case Study

Christian Berger
Department of Computer Science and Engineering, Chalmers, University of Gothenburg, Sweden

Delf Block
Volkswagen AG, Development Car Safety, Electronic and Testing, Wolfsburg, Germany

Sönke Heeren
Automotive Safety Technologies GmbH, Competence Cluster Wolfsburg, Ingolstadt, Germany

Christian Hons, Stefan Kühnel, and André Leschke
Volkswagen AG, Development Car Safety, Electronic and Testing, Wolfsburg, Germany

Dimitri Plotnikov and Bernhard Rumpe
Department of Software Engineering, RWTH Aachen, Germany

Abstract—Context: Consumer tests which assess safety features of modern vehicles have a tradition in Europe. Recently, such test protocols have been substantially extended to also cover active safety systems like Volkswagen's Front Assist.

Objective: Simulations for passive safety systems are already a widely adopted approach during vehicle development and internal assessments. As active safety systems are becoming an increasingly important element in a vehicle's safety concept and a differentiating feature, a systematic validation and assessment of such systems is necessary to successfully pass consumer tests and complimentarily identified, relevant traffic scenarios.

Method: With this work, we extend our previous conference publication about EuroNCAP CCRs tests by additionally investigating US NCAP scenarios for an AEB system. Therefore, we systematically modeled the allowed variations with a graph where the paths represent concrete test scenarios. These paths are used in a virtual test environment to assess the AEB system.

In our previous publication, we illustrated our method of test case generation and simulating consumer test scenarios by showing results of 27 specific test cases. In this work, we focused on integrating a test automatization routine as well as evaluating a set of test

Digital Object Identifier 10.1109/MITS.2015.2474956
Date of publication: 23 October 2015

1939-1390/15©2015IEEE

cases with a factor of 100 compared to our previous paper.

Results: We demonstrate the approach for both EuroNCAP's and US NCAP's CCRs scenarios with a total quantity of more than 2,700 test cases including re-runs to systematically evaluate an AEB algorithm. Our results unveiled varying action points in time for the same initial values for a given consumer test scenario while applying different allowed variations.

Conclusion: We foresee the importance of complementary virtual testing for real-world tests on proving grounds especially during the design phase. Our study shows that already small variations that yet accord with the test procedure specification influence the behavior of an active safety system and need to be investigated during development and vehicle testing.

> As a conclusion, we foresee the importance of complementary virtual testing for real-world tests on proving grounds especially during the design phase.

I. Introduction and Motivation

Advanced Driver Assistance Systems (ADAS) are playing an increasing role as safety features even in smaller vehicle classes like Golf, Polo, or up!. The development of active safety systems is a challenging task because they are designed to operate in potentially dangerous crash- and near-crash-scenarios. Furthermore, detecting such hazardous situations and reacting as quickly as possible and yet reliably is of central interest and will probably result in a better rating at consumer-test-organizations (CTOs).

A. Problem Domain and Motivation

European New Car Assessment Programme (EuroNCAP) and United States New Car Assessment Programme (US NCAP) started assessing active safety systems like Forward Collision Warning (FCW) and Autonomous Emergency Braking (AEB) (also known as Crash Imminent Braking (CIB)/Dynamic Brake Support (DBS) system) besides others. Additionally, consumers as well as Original Equipment Manufacturers (OEMs) are interested in getting the top rating for their safety features. Both, EuroNCAP and US NCAP specified their individual testing procedures and the allowed variation of several test parameters within certain tolerance ranges in detail [1], [2].

While real test runs are inevitable to evaluate the performance of such systems, a simulation approach will provide additional insights and allows further analysis of the system's behavior in border cases that are hardly achievable in reality due to riskiness of the test scenarios or controllability of the boundary conditions.

B. Research Goal and Research Questions

The research goal for this study is to systematically evaluate the EuroNCAP assessment procedure on the example of the Car-to-Car-Rear: stationary (CCRs) test in a simulation-based environment for analyzing the impact of parameter variations within the allowed tolerance ranges. In addition to that, US NCAP test scenarios for a CIB/DBS system are investigated as well to demonstrate the transferability of the approach and to complement the existing study. The following research questions are of specific interest:

RQ-1: To which extent would different lateral positions and heading of the vehicle within the allowed tolerance ranges by EuroNCAP CCRs tests influence an AEB algorithm and as a consequence the residual velocity in case of a collision between Vehicle-Under-Test (VUT) and the target vehicle?

RQ-2: How does the AEB algorithm used for RQ-1 perform in US NCAP settings?

C. Contributions of the Article

This article is an extended version of our previous work in which we presented an approach to model and simulate EuroNCAP test scenarios and the allowed tolerance ranges for several test parameters [3]. In addition to our previous work, we have used updated parameter settings regarding the AEB algorithm due to modification by the supplier. Complementary, we have also applied the approach to the US NCAP test procedure both, (a) to show the transferability of the method and (b) to investigate the robustness of our active safety system. Our experiments unveiled the effect of lateral deviation on the trigger points of an AEB algorithm in both types of consumer tests. Furthermore, we extended our model based infrastructure through the generation of concrete scenarios from scenario models [4], [5], [6] in order to enable the automation of the test execution [7].

Beside our numerical results, we also integrated a test automatization routine which supports the evaluation of a 100 times larger set of test cases compared to our previous study. Thus, even a larger set of the 2,700 individual test cases could be handled by our simulation environment.

D. Structure of the Article

Sec. II outlines a selection of related work. Sec. III describes test procedures and allowed variation parameters for EuroNCAP's and US NCAP's test protocol regarding AEB/FCW systems. Our experimental study is described in Sec. III before we summarize and conclude the article in Sec. V.

In our first publication regarding the simulation of consumer test scenarios for AEB systems, we outlined to what extent a virtual testing approach may support their development.

II. Related Work

In our first publication regarding the simulation of consumer test scenarios for AEB systems, we outlined to what extent a virtual testing approach may support their development. We illustrated that the Equivalence Class Partitioning (ECP) test method known from software testing is insufficient to evaluate such an active safety system, if solely applied [8]. Furthermore, we showed in [3], [6] how the allowed tolerance ranges of EuroNCAP's test protocol can be modeled and simulation runs can be generated to analyze an AEB algorithm under these boundary conditions.

Belbachir et al. present a method for evaluating ADAS including an assessment architecture containing environmental and vehicle components in a simulation. The objective of this simulation-driven approach is to validate such systems by explicitly considering different self-designed evaluation criteria like pedestrian detection error or driver safety estimation [9].

Schuldt et al. outlined a modular testing toolbox for the purpose of evaluating ADAS in an virtual environment. The objective of that work is to reduce the overall numbers of test cases that are necessary to sufficiently validate different types of vehicle functions within a simulation environment. The approach focuses on the intelligent combination of an ADAS's components with certain evaluation criteria from a general point of view [10], [11].

The work of Rauskolb et al. describes the realization of an autonomous driving vehicle for the 2007 DARPA Urban Challenge. One part focuses especially on the use of a simulation environment for acceptance testing in accordance to the requirements given by that competition. The hardware-independent approach aims on modifying object data within the restricted operating environment of the vehicle [12], [13].

Another hardware-independent simulation approach of ADAS is illustrated by Martinus et al. who developed a Virtual Application Platform (VAP) for Software-in-the-Loop (SiL)-tests to support the frontloading during the software development process of ADAS concentrating on functional tests. The virtual platform is based on the AUTOSAR-Standard to deploy software releases without the need for real hardware. Moreover, they also combined the VAP with virtual test driving including a vehicle dynamics model and the according environment simulation without a special focus on the test scenarios [14].

The simulation approach by Nentwig et al. focuses using the original hardware of the supplier realizing a Hardware-in-the-Loop (HiL)-testbed based on the software tools Virtual Test Drive (VTD) and Automotive Data and Time Triggered Framework (ADTF). The simulation environment whose capabilities are described in [15] addresses the functional testing and system testing of video-based systems [16], [17].

Schick et al. worked on a similar research simulation framework for video-based ADAS. They use a different toolset provided by IPG Automotive GmbH in contrast to the aforementioned toolchain to access time-dependent data from virtual camera and radar sensors to validate sensor data fusion algorithms [18]. In [19], a use-case for evaluating a chassis control system is illustrated using the vehicle dynamics simulation of IPG.

Chucholowski et al. worked on a real-time numerical simulation environment to model the vehicle dynamics of a passenger car for the ISO slalom test [20]. Tideman et al. present the toolset "PreScan" by TNO on the basis of manually creating test scenarios and the evaluation of a Lane-Keeping Assist (LKA) from a functional point of view [21], [22].

To the best knowledge of the authors, the design of a structured experiment and its results from systematically applying a simulation-based approach to evaluate an active safety system according to new car assessment programs in a real industrial setting have not been published so far from other research/industry collaborations.

III. Simulating Consumer Tests

At first, we are shortly presenting the purpose and structure of the consumer test protocols for evaluating active safety systems for modern cars. Furthermore, a brief overview of the simulation environment is given.

A. EuroNCAP's AEB Test Protocol

EuroNCAP is a non-profit organization composed of several stakeholders including seven European Governments, motoring, and consumer organizations. It assesses independently new cars with respect to their passive and active safety performances. For that purpose, a test catalogue was compiled evaluating the different functions in realistic scenarios:

- Adult Occupant Protection (frontal, side, and pole impact tests)
- Child Occupant Protection (frontal and side impacts with child restraint systems)
- Pedestrian Protection (front-end structure tests)
- Safety Assist (test of other safety technologies like ADAS)

For a detailed description of the different tests and score calculation, we refer to [24].

Since 2014, there are three different test scenarios representing typical types of crashes occurring in city and inter-urban areas and being addressed by AEB/FCW systems as depicted by Fig. 1:

- Car-To-Car-Rear: stationary (CCRs)
- Car-To-Car-Rear: moving (CCRm)
- Car-To-Car-Rear: braking (CCRb)

The description of the CCRs scenario is provided in section IV. CCRm and CCRb are characterized by a moving target vehicle, driving at a speed of either 20 *km/h* or 50 *km/h* respectively, while the VUT's velocity ranges from 30 to 80 *km/h* by 5 *km/h* steps dependent on the assessed function. In 2016, additional test scenarios including different types of pedestrians will complement this catalogue.

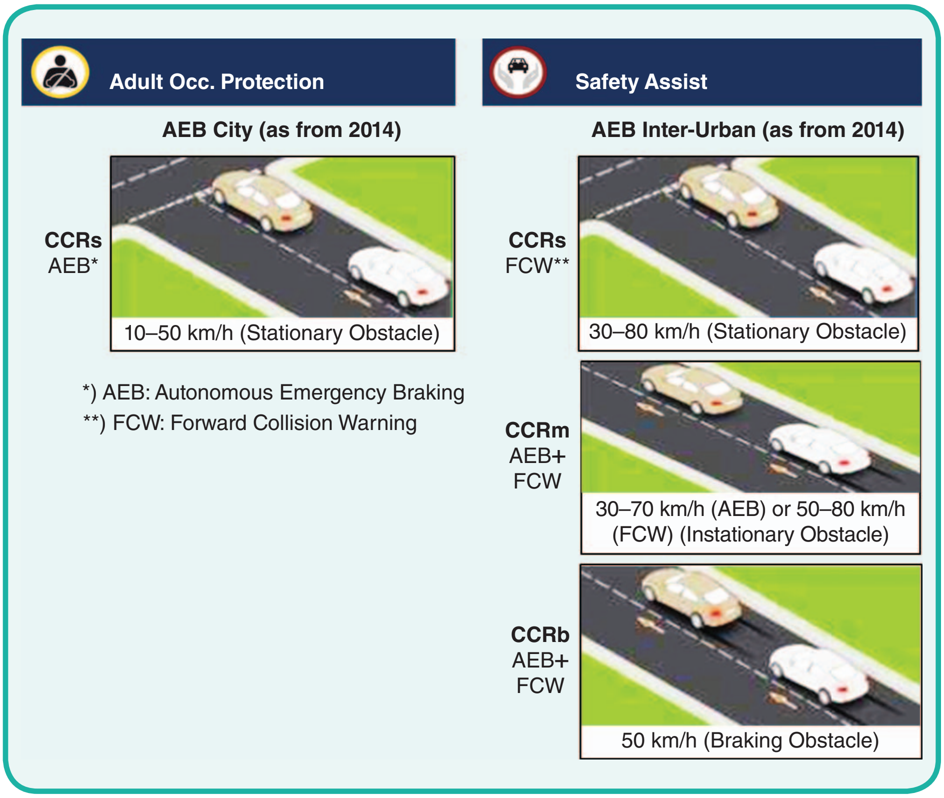

FIG 1 Summary of EuroNCAP's test scenarios and the assessed functions (based on [23]).

For conducting a successful test on a real proving ground, several test parameters regarding the CCRs scenario have to be within the following ranges [1]:

- Speed of VUT (test speed + 1.0 *km/h*)
- Lateral deviation from test path (0 ± 0.1 *m*)
- Yaw velocity (0 ± 1.0°/s)
- Steering wheel velocity (0 ± 15.0°/s)

These parameters are relevant between 4.0 *s* before the VUT probably hits the target vehicle depending on its test speed and the actual activation of the active safety system. Otherwise, the test is considered incorrect according to EuroNCAP's test protocol.

B. US NCAP's Crash Imminent Braking Systems Performance Evaluation

In the United States, there are corresponding test procedures to assess CIB systems (also known as AEB systems). They are described in a different protocol [2] that is currently a working draft and differs with respect to the total amount of test trials and the underlying tolerance ranges for a valid test performance due to being conducted by a human driver instead of a driving robot. The differences are:

- The corresponding CCRs equivalent is performed only with two nominal test velocities of 25 *mph* (40.2 *km/h*) and 45 *mph* (72.4 *km/h*), respectively.
- The allowed tolerance ranges for the VUT may vary ±1 *mph* (1.6 *km/h*) regarding its test speed.
- The maximum lateral deviation from the ideal path is up to ±1.0 *ft* (0.6 *m*), while the maximum yaw velocity can vary in the range of ±2.0°/*s*.

Because of the finalization process regarding this test protocol, we assume the same validity period as described in EuroNCAP's with a start at 4.0 *s* Time-To-Collision (TTC) and the actual activation of the system itself.

Both test procedures are conducted very similar. The main difference between them is that the US-NCAP parameters allow a steering of the VUT by a human driver while the vehicle must be driven by a driving robot at EuroNCAP.

C. Simulation-Based Evaluation for Active Safety Systems

Our simulation approach bases on a systematic enumeration of allowed paths in a directed graph $\mathcal{G}$ as depicted in Fig. 2. The root node for this graph represents T_{end} and all preceding children nodes extend the graph towards the beginning of T_0. The extending depth levels symbolize a time step while all children per depth level represent allowed variations; in our case, we allow lateral deviations per time step resulting in an oscillating manner. Any concrete path p from the depth level at T_0 towards T_{end} encodes a possible test scenario representing a driving trajectory within the allowed tolerance range. Having this inverse orientation in $\mathcal{G}$ allows to model possible concrete scenarios that result in the final situation at T_{end}.

> The ability to reproduce test runs is one of the major advantages of a simulation environment allowing a more systematic investigation of active safety systems than in reality.

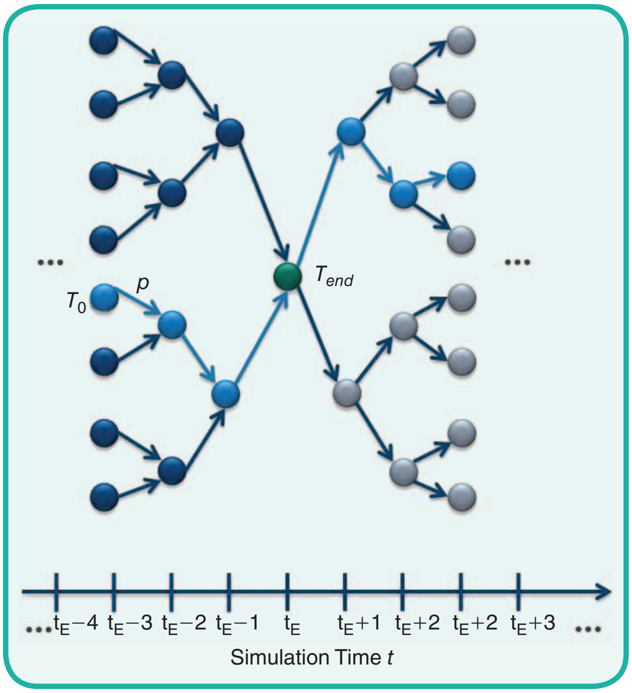

FIG 2 Graph representing possible test cases for the allowed variations: A concrete test case is depicted in red as path p from T_0 to T_{end}.

There are different reasons why an automotive Original Equipment Manufacturer (OEM) is interested in simulating active safety systems and attaining additional insights with respect to real test runs on proving grounds. Firstly, the effort is quite high to prepare VUTs for such test runs with driving robots and adjusting their parameters for example. Furthermore, the ability to reproduce test runs is one of the major advantages of a simulation environment allowing a more systematic investigation of active safety systems than in reality. Nevertheless, simulated test runs will not replace real runs on proving grounds like it is still common practice in developing passive safety features.

In [8] we outlined a method for designing a simulation environment more systematically. One key aspect in that work was to analyze the scope of application for that environment. Two questions are fundamental for an engineer:

1) *Which engineering problem should be addressed by the simulation?*
2) *Which result is finally aimed for?*

The first question focuses on the concrete development task, for example "tolerance analysis regarding specific consumer test scenarios". The other question concentrates on defining the insight that should be attained by the simulation process because the model abstraction depends foremost on the desired result, in this case "Which parameter has a greater influence on the distribution of the remaining speed than others?" for example. Thus, the use of the outlined simulation environment serves primarily at design-time to, for example, also guide the clarification of requirements by systematically experimenting with different potential sensor configurations.

In the following, a potential technical solution to answer such questions is described that we use for our experiments.

D. Simulation Runtime Environment

The simulation environment that is used at the industrial partner consists of several software components whose elements are briefly described as they constitute the basis for our experiment in the industrial setting.

1) Virtual Test Drive: VTD is a software tool developed by VIRES Simulationstechnologie GmbH for openSuSE Linux. It simulates automotive vehicles in a 3D virtual environment including surrounding objects like other traffic participants, pedestrians, vegetation, buildings, traffic signs, and terrain. It has a modular layout so that different vehicle dynamics and environmental sensor modules can be deployed for example. The individual modules communicate via the Runtime Data Bus (RDB), which provides detailed information about all objects; the Simulation Control Protocol (SCP) is used to control the simulation flow [25], [26].

2) Automotive Data and Time Triggered Framework: The ADTF is a software tool developed by Audi Electronics Venture GmbH (AEV) either running on Windows or Linux platforms. It is used for the development and testing of driver assistance systems and safety functions in the automotive industry especially designed for recording and re-playing large amounts of vehicle data [27], [28].

3) *Complementing Core Modules:* The simulation runtime environment for controlled simulation of EuroNCAP test cases consists of two virtual machines running in an Oracle VirtualBox [29]. The first virtual machine is a Windows instance hosting the ADTF configuration including the AEB algorithm. The second virtual machine is a Linux openSuSE instance hosting the VTD environment.

The developed simulation architecture is comprised of a virtual driver module and a vehicle dynamics module. The former is implemented as an ADTF filter that provides stimuli data like braking and acceleration rates for the AEB algorithm running in the ADTF configuration; the latter positions the VUT on precomputed coordinates in the VTD scene during a simulation run.

4) Simulation Automatization: Since several thousand test runs for the introduced experiments must be simulated and evaluated, we extend our previous approach [6] with an automatization approach that executes test runs and archives corresponding traces exports. To fulfill this requirement we integrated a Subversion (SVN) server, that triggers a simulation as soon as there is a new scenario identified. After the particular simulation run is finished, the trace data of the run is stored alongside with simulation data on the SVN server. Furthermore, the SVN server enables versioning of the results. Thus, the function engineer is able to research influence of a parameter change on the function's result by navigating to previous results.

5) Usage of the Simulation Environment: This section shortly describes a hands-on on the developed simulator. In order to evaluate an experiment either the test case generator is adjusted for the particular experiment or it is extended if the experiment is not supported at the moment. In the case of the extension the developer can lean on the already implemented scenarios.

The test case generator produces a set of test cases as depicted in Fig. 3. Each test case is composed of a scene description written in the ScenarioDSL [6], a path file for the VUT, and a property file containing information about the scenario type, speed, and positions of participants. ScenarioDSL which is written with the MontiCore language workbench [30], [31], [32] provides an abstraction over particular simulation technology. The scene is described in an technology agnostic manner and only the code generator produces the platform dependent Extensible Markup Language (XML) representation for the VTD. However, the VTD could be exchanged through another simulation technology in future without altering already existing test scenarios by exchanging the code generator.

Also the granularity of the speed and yaw increments can be adjusted there. The possibility to increase granularity can be useful to analyze a dedicated region of the test protocol where an anomaly occurred in a finer resolution.

Each generated test case is contained in an own folder. After all desired test cases are generated, they can be placed on the SVN Server where they are processed automatically and simulation results are stored alongside with the test case data in the same folder. After all test cases are simulated, the results can be analyzed. The test case property file is also used to support the analysis and to enable a visualization of the results.

Thus, the use of the outlined simulation environment serves primarily at design-time to also guide the clarification of requirements by systematically experimenting with different potential sensor configurations, for example. Moreover, an algorithm can be intensively tested and verified by a huge set of varied parameters. In combination with a distributed hardware infrastructure and distributed computing architecture, a large number of test cases can be handled in less time than it would be possible by HiL simulation environments, for instance.

IV. Systematic Evaluation of Test Case Variations for Stationary Consumer Tests – An Industrial Case Study

In the following, we are describing our experimental study on systematically evaluating tolerance ranges for consumer tests on the example of the CCRs test scenario in an industrial case study. We are reporting according to the guidelines from Jedlitschka et al. [33] and Runeson and Höst [34].

A. Experimental Setup

For our experimental setting, we are focusing on the EuroNCAP's CCRs scenario and its equivalent at US NCAP that are characterized by a target vehicle as a static obstacle being placed in front of the VUT at a certain distance. The VUT, which is equipped with an emergency braking system, has to drive at constant test speed towards the target until the system performs the emergency braking maneuver. Because of different parameter settings we are conducting two experiments that also address our two research questions as described in the following.

Exp-1: Oscillating the VUT alongside the vehicle's x-axis within the allowed EuroNCAP tolerance ranges

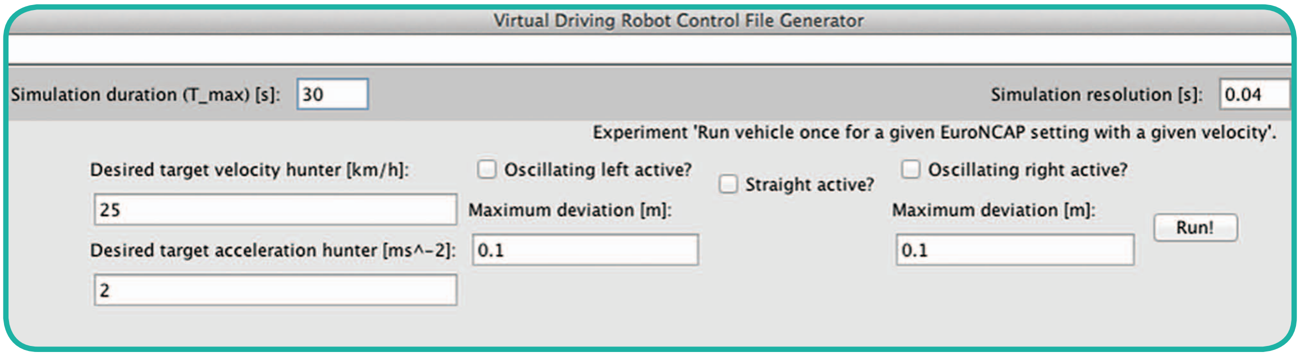

FIG 3 Testcase Generation Tool: Options to adjust CCRs Experiment.

All simulations for this paper were executed and evaluated automatically in the infrastructure available at the industrial partner.

with the goal to analyze the AEB algorithm's behavior at a test case's boundaries.

Exp-2: Oscillating the VUT alongside its x-axis again but now according to US NCAP test protocol and tolerance ranges to analyze the AEB algorithm's behavior at these boundaries.

For each test case there are specific points in time which are indicating the beginning of the test, its ending, and the actual triggering of the AEB algorithm. The test formally starts when the TTC equals 4.0 *s* (T_0) and it ends when either the velocity of the VUT is lower than the target's one (i.e. the VUT stops in time) or the hits the target vehicle. (T_{AEB}) marks the point of time when the safety function is activated as shown in Fig. 5. As we referred in Sec. III, we assumed for both test protocols that only variations between T_0 and T_{AEB} are allowed for the relevant parameters of the VUT.

1) [Exp-1:]: As we referred in [3], we increased the VUT's velocity by 5.0 *km/h* steps from the interval [10.0, 50.0] *km/h* that we call the nominal test cases. In addition, we vary the velocity for each nominal test case in 0.1 *km/h* steps to address the tolerance range of +1.0 *km/h*. We also allow a yaw rate ψ of 1.0°/s in accordance with the test protocol. This leads to an overall quantity of 297 test cases for three types of trajectories: A left-handed and a right-handed one as well as an ideal one.

2) [Exp-2:]: The VUT's velocity will be steadily increased by 0.1 *km/h* steps from the interval [38.6, 41.8] *km/h* and [70.8, 74.0] *km/h* in each test case. This also includes a lateral deviation of [.05; .10; .20; .40; .60] *m* each from the perfect straight line between both vehicles as well as changes in the VUT's heading angle ψ of 2.0°/s resulting in 715 individual test cases.

3) Assumptions for Estimating the Residual Velocity: After the function is activated, we are considering that after AEB algorithm fired the braking system needs a delay time of 0.3 *s* to fully establish the desired deceleration rate to a limited 3.5 m/s^2 due to illustration purposes.

According to the given test speed we calculated the TTC, which depends on the deceleration rate a and the distance D_x between both vehicles by using Eq. 1:

$$\mathrm{TTC} = -\frac{v_{start}}{a} - \sqrt{\frac{v_{start}^2}{a^2} + 2 * \frac{D_x}{a}} \qquad (1)$$

Each TTC describes the time the VUT needs for travelling D_x with a given speed and a constant deceleration. The residual velocity v_{res} is estimated by Eq. 2:

$$v_{res} = v_{start} - a * \mathrm{TTC} \qquad (2)$$

In the following, the experimental procedure is described to realize the aforementioned experiments.

B. Experimental Procedure

A possible path p from our scenario model graph is used in the simulation environment to operate the involved participants, e.g. VUT or EuroNCAP Vehicle Target (EVT). The simulation is controlled via a time discrete clock that sends a tick every 40 ms. Every tick triggers a processing of the next node from p that determines the properties of the VUT and EVT, for instance the position, the speed, and the acceleration. During the simulation all relevant parameters are plotted in the form of comma-separated values files (CSV) including relevant parameters as status of the AEB algorithm or current braking level.

After an initial initialization, the EuroNCAP test simulation is divided into two phases. The first phase serves for a positioning of the VUT during the simulation run, before an emergency braking action is initiated through the AEB algorithm, cf. `Phase 1` in Fig. 4. The first phase begins at the start of the simulation and ends with a warning level output of the AEB algorithm. Afterwards, the second phase is activated whose goal is the computation of the residual velocity and the respective EuroNCAP score, cf. `Phase 2` in Fig. 4. In the following, the functional principle of the vehicle dynamics and the virtual driver is described.

The virtual driver processes a given path p from the scenario model graph stored in a textual representation and sends commands like accelerate, braking rate, and steering angle as an RDB message to the other modules, e.g. vehicle dynamics and the AEB algorithm. The vehicle dynamics, based on the same path p, positions the VUT to the prescribed place including (x, y, z) and the vehicle heading angle ψ in the simulated scene.

In order to fulfill a simulation run, the VTD scene, virtual driver, and vehicle dynamics are initialized according to the scene description and path p. Afterwards, the simulation is invoked via the SCP and runs until the warning level of the AEB algorithm rises. Based on the distance D_x to the target, the remaining speed v_{res} is computed that serves as the input for the EuroNCAP scoring.

All simulations for this paper were executed and evaluated automatically in the infrastructure available at the industrial partner, cf. Fig. 4. An average time for a simulation test run amounts approximately 1 minute excluding subsequent data evaluation. For this paper, *2,732* test cases were simulated including additional proof runs of the same experiment type. The overall time for the simulation took approximately *45.5 hours* of computation required for

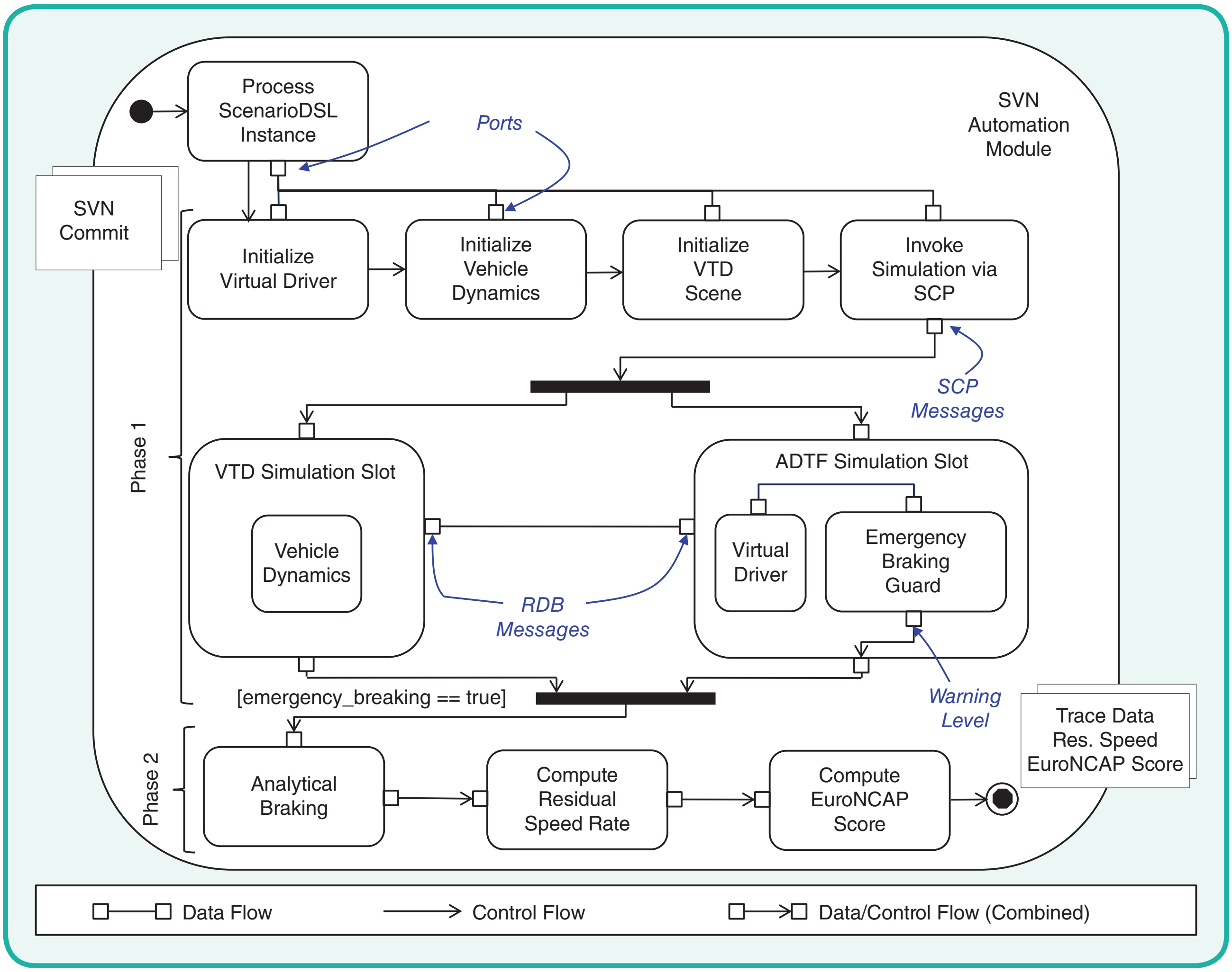

FIG 4 Workflow of an EuroNCAP Test Simulation.

a full experiment handling three path variations for each experiment.

C. Results

The first subsection shows the results from [*Exp-1*] that focused on EuroNCAP's CCRs scenario, the second subsection presents the results from [*Exp-2*].

1) [Exp-1:]: The lower chart in the Fig. 6 on the right-hand side represents all points in time T_{AEB} of the underlying black box AEB algorithm according to each test velocity and each trajectory indicated by its yaw rate. The time scale is converted into an TTC scale, which indicates the time until a collision would occur depending on the VUT's velocity at that very moment.

The upper chart in Fig. 6 on the same side illustrates a varying lateral deviation y_{dev} with respect to the ideal trajectory according to the different test velocities at the point in time T_{AEB}.

We recognized that a possible situation in which the VUT would hit the target vehicle following a trajectory with a lateral deviation while avoiding a contact on an ideal path, lies beyond the official EuroNCAP test specification. To identify this particular case, we simulated additional test cases with velocities from the interval [31.1, 34.9] km/h increased by 0.1 km/h steps.

Fig. 5 illustrates this test case "CCRs AEB 32.6 km/h" with three different trajectories of the VUT towards the target. This particular velocity reveals the transition where a collision would not occur in case of an ideal trajectory, but would happen on one of the oscillating trajectories. The different points of time T_0, T_{AEB}, and T_{end} describe the official test beginning, trigger point, and the test ending as aforementioned. After the AEB algorithm has sent its trigger signal to the braking system, the residual velocity v_{res} can be estimated.

2) [Exp-2:]: Due to the an allowed y-deviation of 0.6 m, we focus on the results from those simulation runs because it is very likely that the biggest difference between the ideal and oscillated trajectories can be noticed within these test cases. Fig. 6 shows the results of the different trigger points

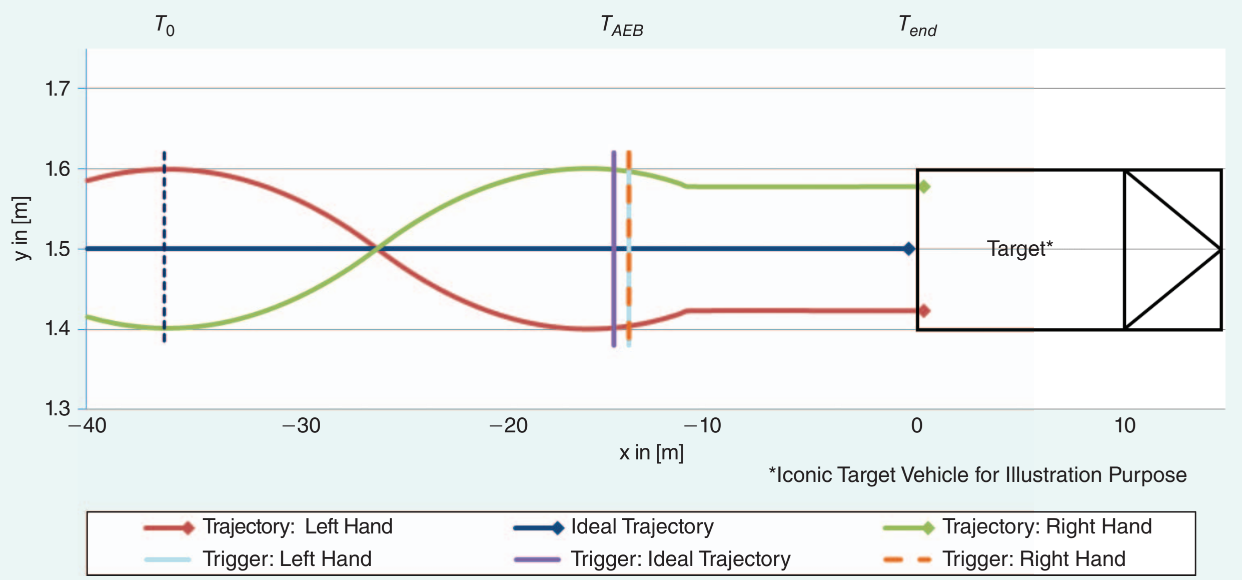

FIG 5 Test case with a VUT's velocity of 32.6 *km/h*, in which a collision occurs following the right-handed and left-handed trajectory respectively compared to the test case with an ideal trajectory.

of the AEB-algorithm for the corresponding test scenarios that are part of US NCAP's CIB/DBS assessment. In Fig. 6, the upper chart on the left-hand side illustrates the lateral position of the VUT y_{dev} compared to the ideal trajectory depending on the different test velocities and the underlying trajectory, while the lower chart illustrates the TTC of the VUT being left after the algorithm triggers an emergency braking maneuver.

D. Analysis and Discussion

Now, the results of the two experiments are analyzed and discussed. Because of the mass of data we received by the simulation runs, we limit our analysis and discussion on an extract of the data to demonstrate what kind of analysis and insights the described approach provides to the vehicle development and testing.

1) [Exp-1:]: In Fig. 6 the right-handed charts reveal that the AEB algorithm regularly behaves as expected with respect to its trigger points for the different trajectories. Particularly, this means that the ideal trajectory leads to an earlier sent signal for activating the brakes by the AEB algorithm than compared to the other two trajectories with an y-deviation. But there are some test cases in which the AEB algorithm triggers at the same point in time and independently from the travelled trajectory. In order to identify a possible anomaly in the AEB algorithm the upper right-handed chart must be considered showing y_{dev} at the moment of the triggering. In case of the $25.x$ km/h trials, we identify the biggest gap between the different trigger points, while the y-deviation delta is only about 0.02 m (cf. A and B). On the other hand, the $30.x$ km/h and $35.x$ km/h test reveal that the AEB algorithm sends its signal at the same time, while the lateral deviation of the VUT is close to the allowed maximum of 0.1 m (cf. C and D). This behavior is different to the actual expectation of having a later triggering with an increasing lateral deviation. Thus, we would declare this an anomaly, which should be further investigated by real test runs on the one hand and with involvement of the underlying supplier on the other.

2) [Exp-2:]: The left-handed charts in Fig. 6 show that the TTC values resulting from the right- and left-handed trajectories are lower than compared to the ideal trajectory, which is expected as well. But a comparison of the TTC values between the lower velocity scenario (38.6 – 41.8 km/h) and the higher velocity scenario (70.9 – 74.0 km/h) reveals that the time gap is as twice as high with a higher speed, although the lateral deviation is quite similar: Approx. 0.54 m in case of 38.6 – 41:8 km/h test cases compared to 0.48 m for the 70.9 – 74.0 km/h test cases (cf. E, F and G, H in Fig. 6). Because this could be an expected behavior, we would not declare this as an anomaly, but it should be further discussed if such a difference resulting in a delta velocity of almost 8 km/h is tolerable for a real assessment run later on.

E. Threats to Validity

We report about threats to validity for our work following the guidelines from Runeson and Höst [34].

Related to construct validity, we set up a clear design for our experiments to investigate the identified research questions based on official test protocols released from third parties (EuroNCAP and US NCAP). Therefore, the

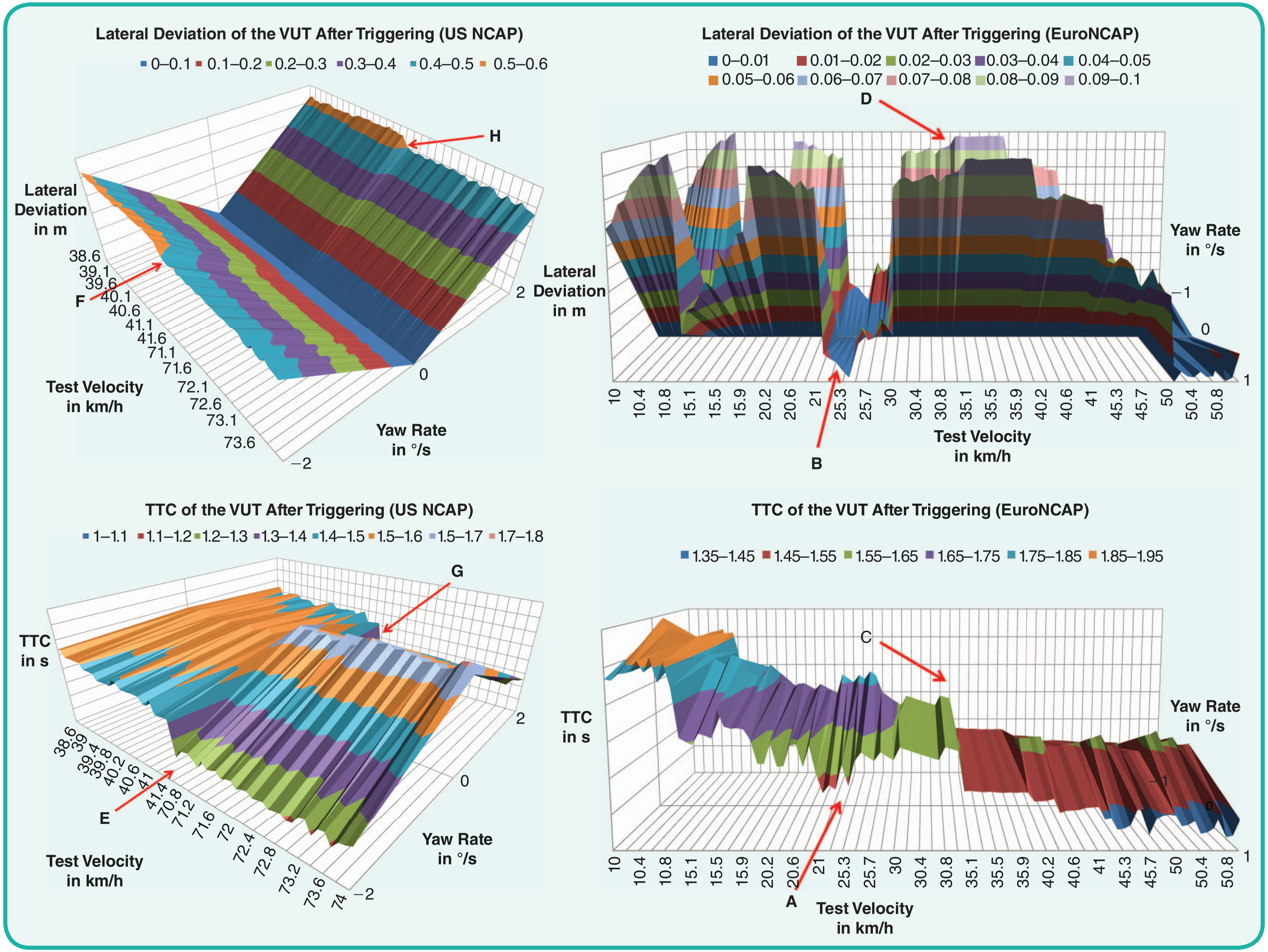

FIG 6 The different results from the simulation runs of EuroNCAP CCRs scenario and its equivalent at US NCAP. The upper left chart illustrates the lateral deviation of the VUT in case of the AEB algorithm's activation. The chart below illustrates the TTC values, respectively. The same allocation can be found on the right hand side with regard to the EuroNCAP simulation runs.

> The virtual test environment that we used to complement real world testing on proving ground allows to study anomalies in a systematic and repeatable manner.

design of our experiment is considered as relevant and appropriate to address the topics of interest. We conducted the experiments with tools that were defined and provided by our industrial partners; in the data analysis phase we observed in very rare cases latencies in the communication between the simulation components. We further analyzed this effect and its impact: For 1,188 simulations runs with different velocity settings and trajectories of the VUT (four runs in total for the CCRs experiment and each run is composed by 297 test cases) we quantified the effect less than 3.7%.

Considering internal validity, the externally supplied test protocols from the consumer test organizations with the allowed tolerance ranges prevent us from favoring specific experimental setup over another. Regarding external validity, the design of experiment can be transferred to other settings as their boundaries where derived from external test protocols; therefore, the method can be considered as rather generalizable. Though, as the experiments where conducted in a virtual environment, specific observations need to be confirmed on a proving ground as well.

V. Conclusions

This article extends our previous work by describing a systematic evaluating of the influence of tolerance ranges from both, EuroNCAP and US NCAP consumer tests on the example of an AEB system. We conducted two experiments to investigate how the lateral deviation from the perfect trajectory affects the point in time when the AEB system triggers an emergency braking maneuver.

Our first experiment addressing our first research question showed that small variations within the allowed tolerance range of $y = \pm 0.1\ m$ in the official EuroNCAP test cases unveil unexpected behavior, for example the same trigger time point despite different lateral deviations; these effects need to be validated on real proving grounds as the test conditions are within the allowed range. We could illustrate how the tolerance ranges of the EuroNCAP CCRs scenario may effect an AEB algorithm and thus, the residual velocity later on if a collision would occur.

In the second experiment which focused on the second research question, we could unveil an anomaly in the US NCAP test cases in comparison between the lower and the higher velocity range despite a similar lateral deviation from the ideal trajectory.

The virtual test environment that we used to complement real world testing on proving ground allows to study anomalies in a systematic and repeatable manner. Therefore, a model-based representation of all relevant test scenarios allows developers and tester to identify unwanted behavior in complex systems and to evaluate countermeasures. Our study also showed that appropriate means are required in the automotive domain to handle the growing amount of "big data" originating from such important experiments as the number of vehicles to be tested is increasing as well as the number of relevant consumer tests around the globe.

Future work needs to include dynamic driving scenarios (CCRm and CCRb) that we are currently working with. Furthermore, the very rare communication latencies need to be address in cooperation with the tool suppliers to further scale up the simulations especially for cloud-based environments.

VI. Acronyms

ADAS Advanced Driver Assistance Systems
ADTF Automotive Data and Time Triggered Framework
AEV Audi Electronics Venture GmbH
AEB Autonomous Emergency Braking
CIB Crash Imminent Braking
DBS Dynamic Brake Support
CCRs Car-to-Car-Rear: stationary
CTOs consumer-test-organizations
ECP Equivalence Class Partitioning
FCW Forward Collision Warning
EuroNCAP European New Car Assessment Programme
US NCAP United States New Car Assessment Programme
HiL Hardware-in-the-Loop
LKA Lane-Keeping Assist
OEM Original Equipment Manufacturer
OEMs Original Equipment Manufacturers
RDB Runtime Data Bus
SCP Simulation Control Protocol
SiL Software-in-the-Loop
TTC Time-To-Collision
VAP Virtual Application Platform
VTD Virtual Test Drive
VUT Vehicle-Under-Test
EVT EuroNCAP Vehicle Target
XML Extensible Markup Language

About the Authors

Christian Berger is associate professor in the Department of Computer Science and Engineering at Chalmers | University of Gothenburg, Sweden. He received his Ph.D. from RWTH Aachen University, Germany in 2010 for his work on challenges for the software engineering for self-driving cars. He

received the “Borchers” award in 2011 for his dissertation. He coordinated the interdisciplinary project for the development of the self-driving car “Caroline”, which participated in the 2007 DARPA Urban Challenge Final in the US. His research expertise is on simulative approaches, formal methods, and model-based software engineering. He published more than 50 peer-reviewed articles in workshops, conferences, journals, and books.

Delf Block is engineer for car safety engineering at the safety development department, Integrated Safety subdivision of Volkswagen AG in Wolfsburg, Germany. In 1997, he received his German Diplom in mechanical engineering for automotive engineering from the Technical University of Brunswick, Germany. He started working in several fields of the car safety development at Volkswagen AG including passive safety, car body structure design and vehicle crash testing. Since 2005, he focused on the combined active and passive safety and worked on precrash functions like the Proactive Occupant Protection System. Currently, he is responsible for the AEB/FCW functions on vehicles and pedestrians in consumer tests. His main interests are active safety, algorithms, sensor development and vehicle testing in consumer tests.

Sönke Heeren is a function developer at the Automotive Safetey Technologies GmbH in Wolfsburg, Germany. His main focus at work is the development of a simulation environment for active safety functions. In 2012, he received his B.Sc. in electrical engineering at the Ostfalia - University of Applied Sciences, Wolfenbüttel, Germany. He is currently working on his Master Thesis in the field of measuring and mapping electromagnetic fields with the help of an automobile platform.

Christian Hons received his German Diplom in mechanical engineering from the Technical University of Brunswick in 1995. From 1995 to 2000, he worked there as a research assistant in the field of force measuring systems. In 2000 he joined Volkswagen AG, Development Car Safety starting as a project manager in airbag algorithms and occupant sensing systems. He became the head of the Development of Airbag and Occupant Sensing subdivision in 2004. Since 2009, he is the head of the Integrated Safety subdivision of Volkswagen AG in Wolfsburg, Germany. His main interests focus on active safety systems like the Proactive Occupant Protection System, AEB/FCW systems and other types of ADAS in consumer tests.

Stefan Kühnel received his German Diplom in business informatics from the Technical University of Brunswick, Germany in 2010. Currently, he is a Ph.D. candidate at Volkswagen AG, Development Car Safety, Electronic and Testing in Wolfsburg, Germany and academically guided by the Department of Software Engineering at RWTH Aachen University. His research interests include automotive software engineering and testing as well as the development of simulative approaches for ADAS in consumer tests.

André Leschke is the head of the Development Car Safety, Electronic and Testing division in Wolfsburg, Germany. In 1997, he received his German Diplom in electrical engineering from the Technical University of Brunswick, Germany in 1997. After joining Volkswagen Bordnetze GmbH in 1997, he was responsible for the Volkswagen T5 on-board electronics project. In 2002, he joined Volkswagen AG, Development Car Safety and filled several leading positions, i. a. head of the Development of Frontal Seats subdivision as well as the Integrated Safety subdivision. His main interests focus on airbag electronics, crash sensors and algorithms, integrated safety systems, Car-to-X communications and crash testing.

Dimitri Plotnikov received his B.Sc. and M.Sc. degrees in computer science from RWTH Aachen University in 2010 and 2012. Currently, he is a research assistant and Ph.D. candidate at Software Engineering at RWTH Aachen University. His research interests cover software engineering, design and development of domain specific languages and corresponding code generators.

Bernhard Rumpe is chair of the Department for Software Engineering at the RWTH Aachen University, Germany. His main interests are software development methods and techniques that benefit from both rigorous and practical approaches. This includes the impact of new technologies such as model-engineering based on UML-like notations and domain specific languages and evolutionary, test-based methods, software architecture as well as the methodical and technical implications of their use in industry. He has furthermore contributed to the communities of formal methods and UML. Since 2009 he started combining modelling techniques and Cloud Computing. He is author and editor of 19 books and Editor-in-Chief of the Springer International Journal on Software and Systems Modeling. See http://www.se-rwth.de/topics/ for more.

References

[1] European New Car Assessment Programme, "Test protocol—AEB systems," Test Protocol 1.0, Tech Rep., July 2013.

[2] National Highway Traffic Safety Administration, "Crash imminent braking system performance evaluation (draft)," NHTSA, Test Protocol, Tech. Rep., Aug. 2014.

[3] C. Berger, D. Block, S. Heeren, C. Hons, S. Kühnel, A. Leschke, D. Plotnikov, and B. Rumpe. (2014). Simulations on consumer tests: A systematic evaluation approach in an industrial case study. in *Proc. IEEE 17th Int. Conf. Intelligent Transportation Systems*, Piscataway, NJ: IEEE Press, pp. 1474–1480. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2014.6957640

[4] H. Grnniger, H. Krahn, B. Rumpe, and M. Schindler. (2006). Integration von modellen in einen codebasierten softwareentwicklungsprozess. in *Proc. Modellierung* (Series Lecture Notes Informatics, vol. 82), H. Mayer and R. Breu, Eds. Innsbruck, Tirol, Österreich: Gesellschaft für Informatik, März, pp. 67–81. [Online]. Available: http://serwth. de/staff/rumpe/publications20042008/GKRS Modellierung06.pdf

[5] B. Rumpe. (2012, June). *Agile Modellierung mit UML: Codegenerierung, Testfälle, Refactoring*. 2nd ed. Berlin, Germany: Springer-Verlag, [Online]. Available: http://www.se-rwth.de/mbse/

[6] C. Berger, D. Block, S. Heeren, C. Hons, S. Kühnel, A. Leschke, D. Plotnikov, and B. Rumpe. (2014). Simulations on consumer tests: Systematic evaluation of tolerance ranges by model-based generation of simulation scenarios. in *30. VDI-VW-Gemeinschaftstagung Fahrerassistenz und Integrierte Sicherheit*, (Series VDI-Berichte, vol. 2223). Düsseldorf, Germany: VDI-Verlag, pp. 403–418. [Online]. Available: http://d-nb.info/1063049393

[7] C. Basarke, C. Berger, and B. Rumpe. (2007, Oct.). Software & systems engineering process and tools for the development of autonomous driving intelligence. *J. Aerosp. Comput. Inform. Commun.* [Online]. *4(12)*, pp. 1158–1174. Available: http://www.se-rwth.de/rumpe/publications20042008/Software-and-Systems-Engineering-Process-and-Tools-for-the-Development-of-Autonomous-Driving-Intelligence.pdf

[8] D. Block, S. Heeren, S. Kühnel, A. Leschke, B. Rumpe, and V. Serebro. (2014). Simulations on consumer tests: A perspective for driver assistance systems. presented at the Int. Workshop Engineering Simulations for Cyber-Physical Systems, New York, NY: ACM, pp. 38–38. [Online]. Available: http://doi.acm.org/10.1145/2559627.2559633

[9] A. Belbachir, J.-C. Smal, J.-M. Blosseville, and D. Gruyer. (Transport Research Arena 2012). Simulationdriven validation of advanced driving-assistance systems. *Proc. Soc. Behav. Sci.* [Online]. *48*, pp. 1205–1214. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042812028327

[10] F. Schuldt, B. Lichte, M. Maurer, and S. Scholz. (2014). Systematische auswertung von testfällen für fahrfuntionen im modularen virtuellen testbaukasten. presented at 9. Workshop Fahrerassistenzsysteme, (vol. 9), pp. 169–178. [Online]. Available: http://www.unidas. de/documents/FAS2014/FAS2014 Tagungsband.pdf

[11] F. Schuldt, F. Saust, B. Lichte, and M. Maurer, "Effiziente systematische testgenerierung für fahrerassistenzsysteme in virtuellen umgebungen," in *Automatisierungssysteme, Assistenzsysteme und Eingebettete Systeme Für Transportmittel*, 2013.

[12] F. W. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, J.-M. Wille, P. Hecker, T. Nothdurft, M. Doering, K. Homeier, J. Morgenroth, L. Wolf, C. Basarke, C. Berger, T. Gülke, F. Klose, and B. Rumpe. (2009). Caroline: An autonomously driving vehicle for urban environments. in *The DARPA Urban Challenge*, (Series, Springer Tracts in Advanced Robotics, vol. 56), M. Buehler, K. Iagnemma, and S. Singh, Eds. Berlin, Heidelberg, Germany: Springer-Verlag, pp. 441–508.

[13] C. Berger, "Automating acceptance tests for sensor- and actuator-based systems on the example of autonomous vehicles," in *Aachener Informatik-Berichte, Software Engineering Band*, vol. 6. Aachen, Germany: Shaker Verlag, 2010.

[14] M. Martinus, M. Deicke, and M. Folie. (2013). Virtual test driving hardware-independent integration of series software. *ATZelektronik Worldwide*. [Online]. *8(5)*, pp. 16–21. Available: http://dx.doi.org/10.1365/s38314-013-0193-7

[15] K. V. Neumann-Cosel, "Virtual test drive," Ph.D. Dissertation, Technische Univ. München, Munich, Germany, 2014.

[16] K. V. Neumann-Cosel, M. Nentwig, D. Lehmann, J. Speth, and A. Knoll, "Preadjustment of a vision-based lane tracker," in *Proc. Driving Simulation Conf. Europe*, Monaco, 2009.

[17] M. Nentwig, R. Schieber, and M. Miegler. (2011). Hardware-in-the-loop testing of advanced driver assistance systems. *ATZelektronik worldwide eMagazine*, [Online]. *6(4)*, pp. 10–15. Available: http://dx.doi.org/10.1365/s38314-011-0034-5

[18] B. Schick and S. Schmidt, "Evaluation of video-based driver assistance systems with sensor data fusion by using virtual test driving," in *Proc. FISITA World Automotive Congr.* (Series Lecture Notes in Electrical Engineering, vol. 196). Berlin, Heidelberg, Germany: Springer-Verlag, 2012, pp. 1363–1375.

[19] H. Holzmann, K.-M. Hahn, and M. K. B. Klein-Ridder, "Virtual test driving at general motors europe," *ATZelektronik worldwide*, vol. 3, no. 1, pp. 48–51, Feb. 2008.

[20] C. Chucholowski, M. Vögel, O. von Stryk, and T.-M. Wolter, "Real time simulation and online control for virtual test drives of cars," in *High Performance Scientific and Engineering Computing*, Berlin, Heidelberg, Germany: Springer-Verlag, 1999.

[21] M. Tideman, "Scenario-based simulation environment for assistance systems," *ATZautotechnology*, vol. 10, no. 1, pp. 28–32, 2010.

[22] M. Tideman, R. Bours, H. Li, T. Schulze, and T. Nakano, "Integrated simulation toolset for ada system development," in *Proc. FISITA World Automotive Congr.* (Series Lecture Notes in Electrical Engineering, vol. 200). Berlin, Heidelberg, Germany: Springer-Verlag, 2013, pp. 25–36. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33838-0 3

[23] W. Hulshof, I. Knight, A. Edwards, M. Avery, and C. Grover, "Autonomous emergency braking test results," in *Proc. 23rd Int. Technical Conf. Enhanced Safety of Vehicles*, 2013.

[24] EuroNCAP. (2014, June). EuroNCAP—The official site of the european new car assessment programme. [Online]. Available: http://www.euroncap.com/

[25] Audi Electronics Venture GmbH. (2014, Mar. 6). Virtual test drive—simulation und absicherung vorausschauender funktionen in virtuellem umfeld. [Online]. Available: http://www.audielectronics-venture.de/aev/brand/de/projekte/virtual test drive.html

[26] Vires Simulationstechnologie GmbH. (2014, Mar. 6). Virtual test drive. [Online]. Available: http://vires.com/products.html

[27] Audi Electronics Venture GmbH. (2014, Mar. 6). ADTF software-umgebung für applikationen und tools. [Online]. Available: http://www.audi-electronicsventure.de/aev/brand/de/leistungen/Entwicklungstools/adtf.html

[28] Elektrobit Corporation. (2014, Mar. 6). EB assist ADTF—Driver assistance applications and safety system development. [Online]. Available: http://automotive.elektrobit.com/driverassistance/eb-assist-adtf

[29] Oracle. (2014, Oct. 6). VirtualBox. [Online]. Available: https://www.virtualbox.org/

[30] H. Krahn, B. Rumpe, and S. Völkel. (2007, Oct.). Integrated definition of abstract and concrete syntax for textual languages. in *Proc. Model Driven Engineering Languages Systems*. (Series Lecture Notes in Computer Science, vol. 4735), G. Engels, B. Opdyke, D. Schmidt, and F. Weil, Eds. Nashville, TN: Springer-Verlag, pp. 286–300. [Online]. Available: http://www.serwth.de/rumpe/publications20042008/Integrated-Definition-of-Abstractand-Concrete-Syntax-for-Textual-Languages.pdf

[31] H. Krahn, B. Rumpe, and S. Völkel. (2008, July). Monticore: Modular development of textual domain specific languages. presented at 46th Int. Conf. Objects, Models, Components, Patterns (TOOLS-Europe), Zurich, Switzerland, (Series Lecture Notes in Business Information Processing, vol. 11), R. Paige and B. Meyer, Eds. Berlin, Heidelberg, Germany: Springer-Verlag, pp. 297–315. [Online]. Available: http://www.serwth.de/ rumpe/publications20042008/MontiCore-Modular-Developmentof-Textual-Domain-Specific-Languages.pdf

[32] S. Völkel. (2011). Kompositionale entwicklung domänenspezifischer sprachen. in *Software Engineering*. (Series Aachener Informatik-Berichte, vol. 9). Aachen, Deutschland: Shaker Verlag. [Online]. Available: http://www.se-rwth.de/phdtheses/Diss-Voelkel-Kompositionale-Entwicklung-domaenenspezifischer-Sprachen.pdf

[33] (2008). *Guide to Advanced Empirical Software Engineering*. F. Shull, J. Singer, and D. I. K. Sjø berg, Eds., London, U.K.: Springer London. [Online]. Available: http://www.springerlink.com/index/10.1007/978-1-84800-044-5

[34] P. Runeson and M. Höst. (2008, Dec.). Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.*, [Online]. *14(2)*, pp. 131–164. Available: http://link.springer.com/10.1007/s10664-008-9102-8

ITS