Simulation-Based Methods for Validation of Automated Driving: A Model-Based Analysis and an Overview about Methods for Implementation

Stefan Jesenski[1], Jan Erik Stellet[1], Wolfgang Branz[1] and J. Marius Zöllner[2]

***Abstract*— The release of highly automated driving functions requires a thorough validation of safety. In this paper, the recently introduced 3-circles model is recapitulated. Based on the 3-circles model, the possible areas of application of simulations to the validation procedure are analyzed and possible challenges and limitations of simulation-based methods are given and discussed. An overview and classification of current simulation concepts is presented. Afterwards, recent research on the implementation details of simulations is discussed.**

I. Introduction

Recently, a lot of research has been performed on the development of highly automated driving (HAD) functions. Hence, a high number of functions of SAE-level 3 and higher [1] are expected to hit the markets in the next years. To release HAD functions automotive manufacturers must thoroughly validate them to make sure, that they are sufficiently safe. However, due to system complexity, emergent behavior of system components, the high complexity of the context of HAD functions and the low exposure to critical situations such as accidents, traditional statistical validation approaches as endurance runs become infeasible. Therefore, manufacturers must develop new methods for validation. Simulations promise to be beneficial since they allow testing a large variation of contextual scenarios reproducibly in economical time spans. Since the process of validation is a complex task, it is necessary to know which of its sub-tasks can be accomplished by simulations and which can not. Besides, it is important to know the demands simulations must fulfill to address the sub-tasks.

For these reasons, this paper discusses the applicability of simulations to sub-tasks of validation. Additionally, an overview about recent research regarding the implementation of simulation-based methods to fulfill sub-task demands is given.

Our **contributions** are as follows:

- We analyze the fields of application of simulations for validation based on the 3-circles model from [2] and show parts of the validation procedure which might be solved by simulations. We give possible challenges and limitations.
- We give a specification of simulation concepts extracted from recent research.
- We summarize and discuss recent research done on the implementation details of simulations for validation.

The paper is **structured** into the following sections: Sec. II reviews the 3-circles model. Sec. III explains the types of evidence which can be gained from simulations and Sec. IV discusses parts of the validation procedure represented in the 3-circles model which might be solved by simulations. Sec. V illustrates a specification of possible simulation concepts and Sec. VI summarizes recent research on the implementation of simulation-based methods. Sec. VII concludes the paper and presents open questions.

II. Validation triangle and 3-circles model

As shown in [2], the aim of validation is to show that a system *realization* fulfills its *purpose* in its *context*. The unity of realization, purpose and context forms the *validation triangle*. These constituents of the validation triangle are interdependent and can not be explicitly described in a complete manner for a HAD function. Reasons are the open context, the complexity of the implicit purpose and the emergent behavior of the realization. These problems make it very challenging to derive a valid specification and to implement a valid realization for the automated system.

In [2] the terms of required behavior (RB), specified behavior (SB) and implemented behavior (IB) were introduced. RB describes the infinitely complex behavior required in reality, SB is the explicitly formalized part of it (described in a specification) and IB includes the implemented results. The Venn diagram in Fig. 1 shows the three behaviors. Each of the three behaviors contains a validation triangle.

The triangle in RB shows that the required realization of the system must fulfill the aimed purpose on the ∞-complex context of reality. The aimed purpose as well as the ∞-complex context and the required realization can not be defined explicitly and formally complete since they are based on implicit assumptions and the real-world operational design domain (ODD) is an unstructured open context. A valid system is not allowed to leave the ODD.

SB contains a formally complete expressed version of the triangle ("expected to be relevant" context, intended purpose and specified realization) and IB contains a triangle which comprises the entities describing the implementation (effective context, effective purpose and implemented realization).

The validation process has to show that the sets of RB, SB and IB show a sufficient overlap and cut-set 3 is maximized (see Fig. 1). For a more elaborated discussion see [2].

[1]S. Jesenski, J. Stellet and W. Branz are with Robert Bosch GmbH, Corporate Research, 71272 Renningen, Germany.

[2]J. Marius Zöllner is with Research Center for Information Technology (FZI), 76131 Karlsruhe, Germany.

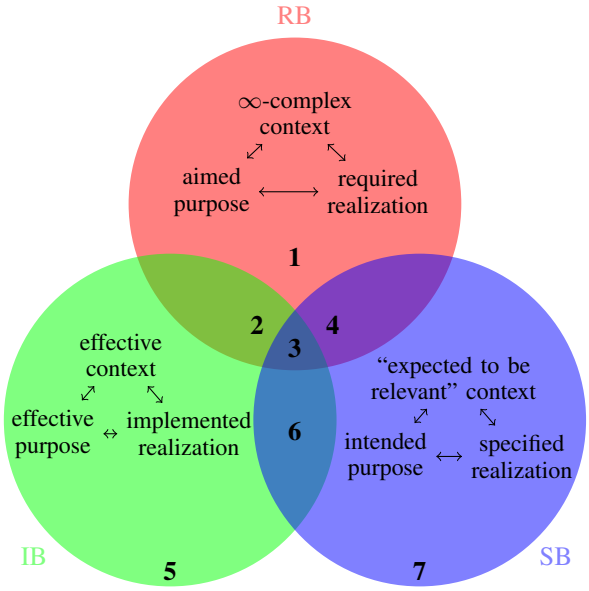


Fig. 1: 3-circles model describing the important entities necessary for validation of HAD functions. The figure is based on a figure from [2] but includes additional information on the relation between validation triangles and behavior sets.

III. Types of Evidence Generated by Simulation

Simulations are useful to generate *evidence* that a particular validation triangle is *consistent*. Consistency means that the realization is able to fulfill its purpose on the given context in the triangle. This evidence can be gained by sampling scenarios (*test cases*) from a *test space* (representation of context) and using them to evaluate a *test object* (representation of realization) by using a *test metric* (checks the fulfillment of the purpose for the context). A more detailed view on the procedure for simulations can be found in Sec. V. We now discuss two types of evidence:

A. System knowledge

The aim is to simulate as many diverse test cases as possible to get confidence in the system's realization. As a result, one can improve the overall understanding of the system's behavior. The application of *microscopic metrics* (see Sec. VI-E) allows infering this system knowledge.

B. Statistical evidence

The aim is to derive statistical evidence about the appearance of errors arising in the triangle of a HAD system. Statistical evidence should allow assessing the influence of HAD systems on real traffic. That means, these statements could for example allow showing that an automated car is statistically as save as a human driver. *Macroscopic metrics* (see Sec. VI-E) are needed to compute statistical evidence.

IV. Usage of Simulation-Based Methods for the Accomplishment of Validation Tasks in the 3-Circles Model

In the following, it shall be assessed how simulations for validation can be classified into the 3-circles model. Therefore, we discuss which parts of the 3-circles model are addressed by recent research. Analyzing the model, there are at least three obvious types of validation tasks, which can contribute to the proof of validity of a system:

- verification (SB $\Longrightarrow$ IB)
- validation of the specification (RB=SB)
- validation of the implementation (RB=IB)

Tab. I shows a summary of the following discussion regarding the simulation-based fulfillment of validation tasks.

A. Verification (SB $\Longrightarrow$ IB)

The process of verification is a well-known problem in the development of complex systems. It aims to check whether the developed and implemented system fulfills the demands of its explicit specification. As a simulation-based example, Bühler et al. [3] demonstrate a hardware-in-the-loop simulation of an automated parking system. It is tested if the system can solve the problem of parking in a variety of well-specified rectangular parking spaces.

Formally, such a simulation tests the consistency of the triangle of implemented realization, "expected to be relevant" context and intended purpose. The test object is a model r^* of the implemented realization. It should be expressed by very low-level models (the modeling needs assumptions c^* and p^* about the effective context and effective purpose) of the ego vehicle or by using as much of the final ego-vehicle hardware as possible. Especially hardware/software/vehicle-in-the-loop simulations seem to be appropriate. For the test space, a model obtained from the "expected to be relevant" context is needed. The test metric of the simulation can be derived by mainly considering the "expected to be relevant" context and the intended purpose.

There are some challenges inherent to the simulations:

- Low-level modeling of the implemented realization is very demanding and complicated. Full reality can not be modeled due to the system's complexity and its emergent behavior. If simulation models are replaced by hardware the results become more realistic. However, the execution speed will slow down. At worst, the execution speed can be limited to real-time (e.g. for VeHiL simulations [4]). Consequently, depending on the amount of used hardware, r^* will include a shift from reality and create a behavior set $\mathrm{IB}_{\mathrm{bias}}$; see Fig. 2a.
- The execution of a simulation can be very time-consuming depending on the complexity and the computational costs of the low-level models.
- Usually the test space can not be sampled exhaustively. An intelligent sampling method is required.

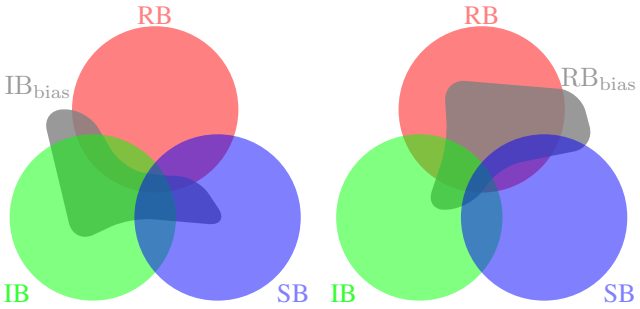
A valid simulation generates evidence on the cut-set 4 and the area 7 (see Fig. 1).

B. Validation of the specification (RB=SB)

The validation of the specification tries to show that the explicitly expressed specification of a system meets the (implicit) demands of the stakeholders. In other words, it must be shown that a specified system fulfills the demands of ∞-complex reality. In the examples [5]–[8] models

TABLE I: Summary of validation tasks and description of the respective simulation properties.

Task	Test object	Test space	Test metric derived from	Prerequisites for the validity of the simulations
verification (SB $\Longrightarrow$ IB, Sec. IV-A)	model r^* of the implem. realization	model of the "expected to be relevant" context	"expected to be relevant" context and intended purpose	$r^* \approx$ implemented realization
validation of specification (RB=SB, Sec. IV-B)	model of the specified realization	model c' of the ∞-complex context	model c' of the ∞-complex context and model p' of aimed purpose	$c' \approx \infty -$ complex context, $p' \approx$ aimed purpose
validation of implementation (RB=IB, Sec. IV-C)	model r^* of the implem. realization	model c' of the ∞-complex context	model c' of the ∞-complex context and model p' of aimed purpose	$c' \approx \infty -$ complex context, $p' \approx$ aimed purpose, $r^* \approx$ implemented realization
consistency of specification (Sec. IV-E)	model of the specified realization	model of the "expected to be relevant" context	"expected to be relevant" context and intended purpose	—



(a) (b)

Fig. 2: **(a)**: The model r^* (and c^*, p^*) uses assumptions and thereby does not exactly represent IB, but a new set $\mathrm{IB}_{\mathrm{bias}}$. Therefore, a lot of new cut sets appear. To handle this cut-sets it is important to check if the difference of $\mathrm{IB}_{\mathrm{bias}}$ and IB is small enough. **(b)**: Same argumentation also holds for $\mathrm{RB}_{\mathrm{bias}}$ caused by c', p' (and r'). For reasons of clarity and comprehensibility, the sets are drawn into separate figures.

for the specified realization, which are given by simple mathematical/control theoretical expressions (what we grasp as specifications since they are high level statements), are checked against test cases (context model).

In a more formal description, these simulations are generating evidence that the validation triangle of specified realization, ∞-complex context and aimed purpose is consistent. The test object is defined by a model of specified realization, the test space is given by a model c' of the ∞-complex context and the test metric is derived by c' and a model p' of the aimed purpose, compare Tab. I. Since it is unlikely, that the ∞-complex context and the related purpose can be defined explicitly and complete, assumptions might be necessary for obtaining c' and p'. Due to the interdependency in the triangles also a model r' of the required realization is (implicitly) assumed during the modeling process. [9]–[14] give examples for this modeling process (especially for c').

The validation of the specification contains challenges:

- It might be hard to reach a satisfying level of accuracy and completeness for c' and p': It is not expected for c' to describe the ∞-complex context, but a biased context c_{bias}. The same applies to p' ($\Longrightarrow p_{\mathrm{bias}}$). The inexact modeling causes a new set $\mathrm{RB}_{\mathrm{bias}}$ in the 3-circles model, see Fig. 2b.
- Lots of data might be necessary to create c' and p' which implies large measurement efforts.
- The execution of the simulation probably will be very time-consuming due to the complexity of c' and p'.
- It might be impossible to sample the test space exhaustively. An efficient sampling method is essential.
- The definition of a test metric to evaluate the results of the simulation is demanding since it must be derived from RB and therefore must handle the open context challenge and use as few assumption as possible.

The simulations might create confidence that the cut-sets 2, 6 and the areas 1, 7 are small. Besides, it might be beneficial to use additional methods to show that the "expected to be relevant" context and the intended purpose are well specified, that means that they are "close enough" to the implicit ∞-complex context and aimed purpose.

C. Validation of implementation (RB=IB)

For classical validation, the implemented system must accomplish the (implicit) needs of its stakeholders. Therefore, the system implementation is compared to the demands of reality as given in RB. By way of example, O'Kelly et al. [15] have implemented a black box framework which should in principle be able to test an entire automated function against a context learned from real world data. Schmidt [17] proposes a framework for HiL-testing of camera-based functions. [18]–[20] propose concepts for the validation of the implementation by incorporating a HAD function in a car. This implemented HAD function passively analyzes the scenarios the car experiences. The open-loop of the analysis of the scenarios is closed using offline simulations.

These types of simulations aim to show the consistency of the triangle of implemented realization, ∞-complex context and aimed purpose. The test object, test space and test metric have to be chosen accordingly; see Tab. I.

For the simulation a combination of challenges from Sec. IV-B and Sec. IV-A arises:

- challenges arising from models c' and p'; see Sec. IV-B.
- challenges corresponding to r^*; see Sec. IV-A.

A valid simulation should generate evidence on the cut-sets 4, 6 and the areas 1, 5.

D. Example: automated emergency braking (AEB)

The validation tasks are illustrated by an AEB which is a part of an automated vehicle. In its RB the system shall prevent collisions with other traffic participants on highways.

For an **example of SB $\Longrightarrow$ IB**, we specify the AEB to prevent collisions with a predecessor vehicle (intended purpose) on straight highways containing exactly one predecessor vehicle besides the ego vehicle (test space). It can be assumed that the properties of these specified scenarios are limited, e.g. assume $-5\,\frac{\mathrm{m}}{\mathrm{s}^2}$ to $5\,\frac{\mathrm{m}}{\mathrm{s}^2}$ to be the range of the acceleration of the predecessor. For the simulation one could draw synthetic scenarios from this specified test space. A possible test metric is the time to collision (TTC) between the ego vehicle and the predecessor vehicle. A VeHiL setup might give the test object. The simulations give us evidence about the errors of r^* within the specified test space. Exemplarily, Berger et al. [21] executed similar simulations to test AEB functions in the well specified context of NCAP tests.

The previous example could not give us any hints about the accurateness of the specification itself. An **example of RB = SB** can show how to get these hints. For that, we must add a specified realization consistent with the specified test space and intended purpose: e.g. decelerate the vehicle maximally for TTC $< 1\,\mathrm{s}$ and keep its velocity for TTC $\geq 1\,\mathrm{s}$. The test space (c') on which we test, might be modeled by directly sampling from a sufficiently large dataset or by fitting a model (e.g. by the use of machine learning) to the data. By sampling from the test space, one can detect possible errors of the specified realization which do not occur in the specified test space but on its outside, e.g. at curved highway sections. This hints at an inaccurate specification.

For an **example of RB = IB** the simulations can be executed by using the same test space and metric as for $RB = SB$. For the low-level test object we might again use a VeHiL system. This gives hints if the implemented system works for all parts of reality described by c' and p'.

E. Discussion and further approaches

Fig. 3 summarizes the discussed tasks in two validation task groups. Additional to the already discussed tasks, a task to show the *consistency of the specification* is included. The reason is, that specifications of complex systems can become quite complicated. Therefore simulations showing the consistency of the specification could be helpful. Evidence to this consistency could be generated by showing that the elements of the specification triangle ("expected to be relevant" context, intended purpose and specified realization) are in a sensible interdependent relationship.

The blue task group and the green task in Fig. 3 should include a certain degree of redundancy since both should be able to generate evidence of the equivalence of the implemented and the required behavior. However, by the application of the blue group more insights into the system and its behavior could become possible, e.g. the possible existence of specification-related cut-set 2 and area 7 would be neglected when only using RB=IB. Especially cut-set 2 could become dangerous, since its behavior was implemented correctly by chance and not by design. Hence, the behavior in cut-set 2 was not understood explicitly and could become invalid unnoticedly.

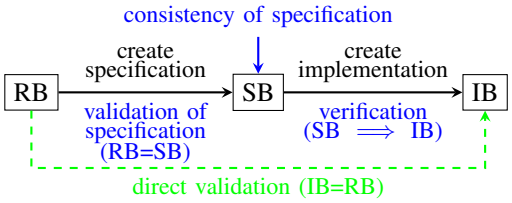


Fig. 3: Two validation task groups exist: The blue marked group can be applied during the development process, whereas the green, dashed group can be applied afterwards.

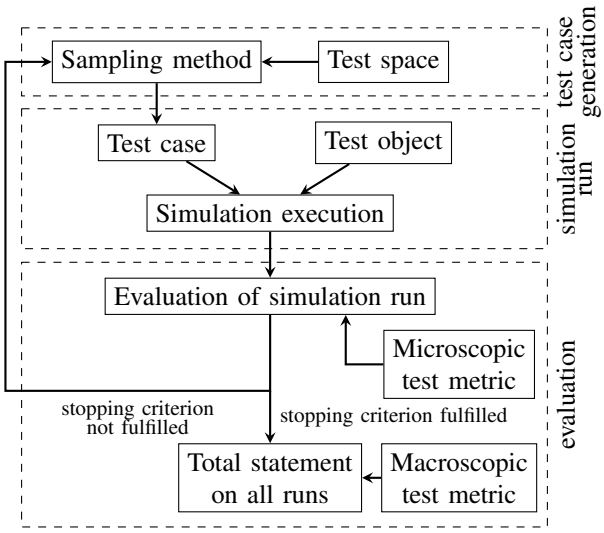


Fig. 4: Simulation-based procedure for validation. Simulation runs are conducted until a stopping criterion is fulfilled. This stopping criterion could be a statement about the completeness of the tested space or a statement about accuracy of a statistical result.

V. Classification of Simulation Concepts

In this section we want to highlight the basic principles of the simulations which can be used to show the consistency of validation triangles as described above. The basic procedure is illustrated in Fig. 4. A sampling method is used to create test cases by sampling the parameter values of the test space. Test cases are then used to simulate the response of the test object during the simulation execution. The response is evaluated by using a microscopic test metric. If a stopping criterion is met, a total final statement on all simulation runs is calculated and the simulation is aborted. The total statement is calculated using a macroscopic test metric. Otherwise, a new test case is sampled and another simulation run is conducted. Due to the emergent behavior of complex HAD systems, it might be beneficial to solve the validation task at system-level. Thereby, the following study considers the properties of system-level simulations. More information to test metrics can be found in Sec. VI-E.

A. Properties of simulation run

Dependent on the test object and the validation aims, several properties of a simulation run can be adjusted.

1) Granularity: One of the most important properties of a simulation is its granularity. One can distinguish between macroscopic, microscopic and sub-microscopic simulations. For the validation of HAD functions, sub-microscopic simulations are preferable since they consider vehicle sub-structures (e.g. single hardware components). In contrast, macroscopic simulations only consider thermodynamical traffic flow variables and microscopic simulations consider single vehicles but not the vehicle sub-structures.

2) Closed loop vs. open loop: Another important decision is whether a simulation is executed in a closed or an open loop. Within open loop simulations, the decisions and behavior of the test object do not affect its surrounding. An example is the (augmented) replay of measured data [22]. Since a HAD function is very complex and heavily influences its surrounding vehicles, it seems to be preferable to execute a validation simulation in a closed loop manner.

3) Inclusion of reality in simulations: As already explained in the former section, it can be quite demanding to model the complex system of the test object. Thereby, it can be beneficial to use parts of a real vehicle. This is denoted as x-in-the-loop procedure. Within x-in-the-loop methods, parts of simulation models are replaced by reality. An example is the Hardware-in the-loop (HiL) technique, which for example replaces (parts of) the HAD function-model by real software running on a real control unit. Of course, this replacement can be done on different scales. A Vehicle-Hardware-in-the-loop (VeHil) approach would go further and replace the whole test object including its sensors and near surrounding vehicles by reality [4]. Basically, the more models are replaced the better the simulation results become. However, simulation speed will (strongly) decrease with an increasing amount of reality.

B. Properties of test case generation

Besides the simulation run itself also the properties of the sampling method and the test space must be chosen carefully. A high-level categorization of recent work can roughly be done by using two axes: the type of model used to describe the test space and the basic principle of sampling.

1) Test space model: For system-level simulations, the test space mainly includes the surrounding of a HAD vehicle. Recent research on modeling the test space of system-level simulations can be distinguished into the categories of **generic** modeling and **maneuver-based** modeling.

A **maneuver-based model** describes a particular type of traffic maneuver, e.g. models for car following scenarios [5], [23], cut-in scenarios [6], [7], [24], lane departure events [25] or parking scenarios [3]. As an advantage, such a model does not have to be generic which makes it easier to design. Additionally, such models need less parameters and they usually have an expressive meaning. However it is disadvantageous, that an extra model is needed for each maneuver (e.g. the ∞-complex context would not be represented by one model c', but by a set of models $c' = \bigcup c'_i$). Since the open context reality contains a lot of different scenarios, the number of needed models is high. Therefore, a scenario catalog, which includes the relevant scenarios, is required [23] for validation. It is very time-consuming or even impossible to generate a complete scenario catalog of an open context system. Basic approaches to scenario catalogs can be found in [7], [8]. There are data-driven methods for scenario catalogs as in [26], [27] and formal approaches, e.g. based on ontologies [28], [29].

A **generic approach** towards test space modeling is not limited to the description of a specific maneuver. That means one does not need a scenario catalog and should be able to obtain scenarios of different maneuver type by just sampling from the model's parameter space. The disadvantages of generic modeling include parameter explosion, less expressive parameters and the difficulty to be accurate and complete on the ∞-complex context (maybe impossible). [9], [10], [12], [15], [16] give examples for generic modeling techniques.

2) Sampling strategies: A sampling strategy is needed to make sure that the results of the simulation have a sensible meaning and are calculated efficiently. For open context systems it is usually impossible to sample exhaustively. This fact strongly increases the demands on the sampling scheme. Additionally, it is important to develop a method to handle continuous parameters (e.g. discretize them) and to find valid parameter ranges. In literature, there are two approaches to sample: **coverage-based** and **statistical** sampling.

Coverage-based sampling is based on the principle to sample as many different parameter sets as possible. This is useful for getting insights into the HAD system and exploring the interactions with its surrounding. It is possible to sample evenly distributed over the whole test space or to use optimization methods to just sample interesting parameter configurations. Recent work on these principles is given in Sec. VI-C.2. It is often hard to define a stopping criterion for coverage-based sampling since exhaustive sampling is impossible and a metric, which determines when "enough" or the "relevant" parameter values have been checked, is difficult to find [30]. When using suitable coverage-based sampling methods it is possible to get system knowledge and statistical evidence about the test object [7].

Statistical sampling tries to find statistical evidence about the test object. Such evidence could be the probability of an accident in real traffic. Statistical sampling methods mostly sample in highly probable areas of the parameter space. These areas usually are understood quite good. That means, the possibility to get new insights into the system is reduced. Techniques to mitigate this challenge are given in Sec. VI-C.2. A stopping criterion can be defined based on the variance of the statistical results of the simulation [31].

VI. Implementation

This section gives an overview about the parts, elements and methods needed for the efficient implementation of the

validation concepts which were given in Sec. V.

A. Basis for model creation

In order to create models of the test object or the test space, it is necessary to have a source to deduce the models from. This source can be data-driven or driven by (human) knowledge. For the search of statistical evidence a data-driven test space model approach is essential. The allowed level of abstractness (objects lists, raw data, ...) of the dataset is dependent on the scope of the simulations. For the validation of an open context system it is a huge challenge to obtain a dataset which is as complete as possible.

B. On the representation of models

The mathematical models are fundamentally separated into black-box and white-box models. Some evaluation methods may need a white-box model. For mathematical purposes and flexibility, the test object and the test space are divided into several sub-models.

1) Modeling of test object: The test-object model is divided into several sub-models. On the one hand, there are models for the sensor hardware in the automated vehicle. They can be quite complicated depending on their accuracy. In our previous work [32] we distinguish physical low-level models and phenomenological high-level models. Of course, one could also use sensor hardware in a X-in-the-loop procedure. Additionally, models for electronic control units and models for the vehicle dynamics such as steering and braking are needed. Especially for vehicle dynamics there exists a wide variety of models, starting with simple single-track models which are improved by more accurate and more complicated multi-track ones [33]. It is also important to model the functional chain of the HAD function which shall be tested. As an alternative, one could use its software code/hardware components (see Sec. V-A.3).

2) Modeling of test space: The open context surrounding of the test object is complex and contains a lot of different entities. A categorization of surrounding entities into 4 hierarchic classes is given by Schuldt [30].

Since the validation of HAD functions benefits from a closed loop simulation, the surrounding model often separates into static scene models [9] and behavior models [13] which return the dynamics of dynamic environment objects. Some attempts also use hybrid models unifying scene and behavior models. However, such hybrid approaches often are not as generic and flexible as the separated approach, e.g. [5] only models a particular scenario.

C. Methods for efficient and effective test case generation

It is of utmost importance for simulations that the test case generation (see Fig. 4) is done in an effective and efficient manner.

1) Methods for test space generation: A major step in implementing the test case generation is to find a good way to parameterize the test space. Firstly, that includes the selection of features/parameters [34]. Secondly, parameter ranges and distributions to sample from must be defined. Thirdly, a good discretization for continuous parameters or another way to handle the continuum must be implemented. In the simplest case equidistant bins are used [9]. [35] presents a more elaborate discretization scheme for Bayesian networks. A high-level discussion of the generation of the test space and the inherent demands is given by Schuldt [30].

2) Methods for efficient sampling: Additionally, a method for efficient and effective sampling is needed. The main challenge is that for open context systems an exhaustive sampling becomes impossible. Hence, a strategy to select the "relevant" test cases is needed. Various aspects of the sampling methods given in Sec. V-B.2 are discussed below:

For the **statistical sampling** there is the challenge to find a good statistical model describing the scenario distribution. In general, one can distinguish statistical models relying on a particular parametric family of probability distributions and statistical models working with distribution-free methods.

Advantages of distribution-based statistical models are, that they often show few, but expressive parameters and that the models' output is comprehensible. The disadvantage is that the structure of such models is fixed and thereby there are problems using them to represent an open context. Distribution-based behavior models often are split up into models for lane-following [13], [14], [36], lane-change [37], [38] and gap acceptance. In [39], these models and some of their main parameters are discussed.

Distribution-free statistical models are more flexible in their structure and therefore might better handle the open context of a HAD function. A comparison of the performance of distribution-based and distribution-free models for vehicle speed prediction is given in [40]. Distribution-free models often contain a larger number of parameters with a non-expressive meaning. Examples for distribution-free models are graphical models as Bayesian networks [9], [11], [16], [41], factor graphs [10], tree diagrams [11] and neuronal networks [42]. Models based on generative adversarial techniques [12] are used to solve the problem of cascading errors which appear when supervised methods are used to learn sequential decision processes like behavior models.

Methods to increase the efficiency of the statistical sampling approach are importance sampling [5], [6], [15], [23], [31] and a Markow Chain Monte Carlo method in conjunction with the subset simulation method [43]. Additionally, surrogate functions based on Kriging models have been proposed to further enhance importance sampling [44].

The **coverage-based sampling** approach can be realized by a large bunch of sampling strategies. An overview about suitable combinatorial methods and coverage criteria, especially for the field of software testing, can be found in [45]. Schuldt [30] discusses the application of combinatorial methods in association with equivalence classes and boundary value analysis. A combinatorial t-wise sampling strategy is applied in [46]. Of course the coverage-based approach can also be realized by only sampling "interesting" parameter values. There are optimization-based [47]–[49], learning-based [50] and search-based [51], [52] methods to find error-prone regions of the parameter space, methods to

TABLE II: Specification of safety metrics based on [56].

	Metrics for accident severity	Metrics for criticality
microscopic	• physical metrics (collision speed, ...) • physiological metrics (injury severity [5], ...) • economic metrics	• physical metrics (PET [57], WTTC [58], TTX (TTC [59], TTB [60], ...), headway, DCE [61], ...)
	Metrics on accident cases	**Metrics on all cases**
macroscopic	• injury rate [5] • fatality rate • ...	• accident rate [54] • prevention rate • conflict rate [5] • ...

find feature interaction failures [53], methods to find the safety boundaries of error-prone regions [54] and methods to vary recorded scenes [22]. Additionally. there are approaches to derive test cases from test models [55].

D. Model validation

As already explained in Sec. IV-B it is important to validate that $c' \approx \infty$-complex context and $p' \approx$ aimed purpose. This consists of the problems of validating accurateness and completeness, which requires the definition of metrics.

For the validation of accurateness, methods based on comparisons of "emergent" behavior [9], [11], [12], comparisons of parameter distributions, e.g. by Kullback-Leibler divergence [12], [42], cross-validated likelihoods [11], root mean squared error (RMSE) [40], [41] and mean absolute error (MAE) [40] are used. These metrics have in common that they compare the learned models with datasets.

For the validation of completeness metrics directly evaluating the completeness off datasets are needed.

E. Evaluation

Simulations are evaluated by the use of test metrics. Helmer [56] has introduced a classification for metrics which distinguishes between microscopic and macroscopic metrics as can be seen in Tab. II. In addition to safety, metrics evaluating other types of effects of automated vehicles are required, e.g. traffic quality metrics as discussed in [60].

VII. Conclusion

We recapitulated the 3-circles model and analyzed parts which possibly can be handled by simulations. We discussed the types of evidence which can be generated by simulations and saw that simulations addressing validation tasks (e.g. RB=SB, RB=IB and SB $\Longrightarrow$ IB) possess diverse difficulties. We also pointed out, that there are some prerequisites for the simulations in the validation process which must be fulfilled for the simulations to be valid (compare Tab. I).

Afterwards, we introduced a categorization of simulation concepts depending on the properties of the simulation run and test case generation. The main axis of the properties of the simulation run is the inclusion of reality, whereas the test case generation must be specified along the two axes of test space model and sampling strategy. We finished the paper with an overview about recent research on methods needed for an efficient implementation of test case generation.

Despite extensive research, there is still a bunch of unsolved questions: Metrics evaluating the completeness of a dataset are needed. Additionally, there needs to be done more work on metrics evaluating the accurateness of a fit of the models (c' and p') to the dataset. Further work needs to be done on metrics which evaluate the results of simulations. A related problem is the definition of stopping criteria for sampling methods. Further research on the sampling methods themselves must be conducted. Another open problem is the modeling of complex system behavior (r^*) since there is a trade-off between inclusion of reality and simulation speed.

References

[1] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," in *SAE Standard J3016 JUN2018*, 2018.

[2] J. E. Stellet, T. Brade, A. Poddey, S. Jesenski and W. Branz, "Formalisation and algorithmic approach to the automated driving validation problem," in *IEEE Intelligent Vehicles Symposium*, Paris, 2019.

[3] O. Bühler and J. Wegener, "Automatic Testing of an Autonomous Parking System Using Evolutionary Computation," *SAE Technical Paper*, 2004.

[4] 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," in *Mechatronics*, vol. 19 no. 7, 2009, pp. 1091-1104.

[5] D. Zhao, X. Huang, H. Peng, H. Lam and D. J. LeBlanc, "Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, 2018, pp. 733-744.

[6] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa and C. S. Pan, "Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, 2017, pp. 595-607.

[7] J. Zhou and L. del Re, "Reduced Complexity Safety Testing for ADAS & ADF," in *IFAC-PapersOnLine*, Vol. 50, no. 1, 2017, pp. 5985-5990.

[8] R. van Tongeren, O. Gietelink, B. De Schutter and M. Verhaegen, "Traffic Modelling Validation of Advanced Driver Assistance Systems," in *IEEE Intelligent Vehicles Symposium*, Istanbul, 2007, pp. 1246-1251.

[9] T. A. Wheeler, M. J. Kochenderfer and P. Robbel, "Initial Scene Configurations for Highway Traffic Propagation," in *IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, 2015, pp. 279-284.

[10] T. A. Wheeler and M. J. Kochenderfer, "Factor graph scene distributions for automotive safety analysis," in *IEEE 19th International Conference on Intelligent Transportation Systems*, Rio de Janeiro, 2016, pp. 1035-1040.

[11] T. A. Wheeler, P. Robbel and M. J. Kochenderfer, "Analysis of microscopic behavior models for probabilistic modeling of driver behavior," in *IEEE 19th International Conference on Intelligent Transportation Systems*, Rio de Janeiro, 2016, pp. 1604-1609.

[12] A. Kuefler, J. Morton, T. Wheeler and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium*, Los Angeles, 2017, pp. 204-211.

[13] M. Treiber, A. Hennecke and D. Helbing, "Congested traffic states in empirical observations and microscopic simuations," in *Phys. Rev. E*, vol. 62, no. 2, 2000, pp. 1805-1824.

[14] P. G. Gipps, "A behavioral car-following model for computer simulation," in *Transportation Research Part B: Methodological*, vol. 15, no. 2, 1981, pp. 105-111.

[15] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, Russ and J. C. Duchi, "Scalable End-to-End Autonomous Vehicle Testing via Rare-event Simulation" in *Advances in Neural Information Processing Systems 31*, 2018, pp. 9827-9838.

[16] S. Jesenski, J. Stellet, F. Schiegg and J. Zöllner, "Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions," in *IEEE Intelligent Vehicles Symposium*, Paris, 2019.

[17] F. Schmidt, "Funktionale Absicherung kamerabasierter Aktiver Fahrerassistenzsysteme durch Hardware-in-the-Loop-Tests", PhD thesis, Universität Kaiserslautern, 2012.

[18] E. Rocklage, “Teaching self-driving cars to dream: A deeply integrated, innovative approach for solving the autonomous vehicle validation problem,” in *IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1-7.

[19] A. Koenig, K. Witzlsperger, F. Leutwiler and S. Hohmann, ”Overview of HAD validation and passive HAD as a concept for validating highly automated cars,“ in *at - Automatisierungstechnik*, vol. 66, no. 2, 2018, pp. 132-145.

[20] W. Wachenfeld and H. Winner, “Virtual assessment of automation in field operation a new runtime validation method,” in *Workshop Fahrassistenzsysteme*, Walting, 2015, pp. 161-170.

[21] C. Berger et al., “Simulations on Consumer Tests: A Systematic Evaluation Approach in an Industrial Case Study,” in *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 4, 2015, pp. 24-36.

[22] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm and J. M. Zöllner, “Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems,” in *18th International Conference on Information Fusion*, Washington DC, 2015, pp. 1422-1428.

[23] E. de Gelder and J. Paardekooper, “Assessment of Automated Driving Systems using real-life scenarios,” in *IEEE Intelligent Vehicles Symposium*, Los Angeles, 2017, pp. 589-594.

[24] J. Zhou and L. del Re, “Identification of critical cases of ADAS safety by FOT based parameterization of a catalogue,” in *11th Asian Control Conference*, Gold Coast, 2017, pp. 453-458.

[25] W. Wang and D. Zhao, “Evaluation of Lane Departure Correction Systems Using a Regenerative Stochastic Driver Model,” in *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, Sept. 2017, pp. 221-232.

[26] S. Li, W. Wang, Z. Mo and D. Zhao, “Cluster Naturalistic Driving Encounters Using Deep Unsupervised Learning,” in *IEEE Intelligent Vehicles Symposium*, Changshu, 2018, pp. 1354-1359.

[27] D. Zhao, Y. Guo and Y. J. Jia, “TrafficNet: An open naturalistic driving scenario library,” in *IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1-8.

[28] F. Klück, Y. Li, M. Nica, J. Tao and F. Wotawa, “Using Ontologies for Test Suites Generation for Automated and Autonomous Driving Functions,” in *IEEE International Symposium on Software Reliability Engineering Workshops*, Memphis, 2018, pp. 118-123.

[29] G. Bagschik, T. Menzel and M. Maurer, “Ontology based Scene Creation for the Development of Automated Vehicles,” in *IEEE Intelligent Vehicles Symposium*, Changshu, 2018, pp. 1813-1820.

[30] F. Schuldt, “Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen”, PhD thesis, Technische Universität Braunschweig, 2017.

[31] O. Gietelink, B. De Schutter, and M. Verhaegen, “Adaptive importance sampling for probabilistic validation of advanced driver assistance systems,” in *Proceedings of the 2006 American Control Conference*, Minneapolis, 2006, pp. 4002-4007.

[32] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels and J. M. Zöllner, “Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions,” in *IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, 2015, pp. 1455-1462.

[33] M. Mitschke and H. Wallentowitz, “Dynamik der Kraftfahrzeuge”, 5th edition, Springer Berlin Heidelberg, 2015.

[34] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” in *J. Mach. Learn. Res.*, vol.3, March 2003, pp. 1157-1182.

[35] Y. Chen, T. A. Wheeler and M. J. Kochenderfer, “Learning Discrete Bayesian Networks from Continuous Data,” in *Journal of Artificial Intelligence Research*, vol. 18, June 2017, pp. 103-132.

[36] R. Wiedemann, “Simulation des Straßenverkehrsflusses,” in *Schriftenreihe des IfV*, vol. 8, 1974.

[37] A. Kesting, M. Treiber, D. Helbing, “General Lane-Changing Model MOBIL for Car-Following Models,” in *Transportation Research Record*, vol. 1999, no. 1, Jan. 2007, pp. 86-94.

[38] J. Erdmann, “Lane-changing model in SUMO,” in *Proceedings of the SUMO2014 Modeling mobility with Open Data*, Berlin, 2014.

[39] P. Bonsall, R. Liu and W. Young, “Modelling safety-related driving behaviour – impact of parameter values” in *Transportation Research Part A: Policy and Practice*, vol. 39, no. 5, 2005, pp. 425-444.

[40] S. Lefèvre, C. Sun, R. Bajcsy and C. Laugier, “Comparison of parametric and non-parametric approaches for vehicle speed prediction,” in *American Control Conference*, Portland, 2014, pp. 3494-3499.

[41] T. Gindele, S. Brechtel and R. Dillmann, “Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning,” in *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, Spring 2015, pp. 69-79.

[42] J. Morton, T. A. Wheeler and M. J. Kochenderfer, “Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, May 2017, pp. 1289-1298.

[43] S. Zhang, H. Peng, D. Zhao and H. E. Tseng, “Accelerated Evaluation of Autonomous Vehicles in the Lane Change Scenario Based on Subset Simulation Technique,” in *21st International Conference on Intelligent Transportation Systems*, Maui, 2018, pp. 3935-3940.

[44] Z. Huang, H. Lam and D. Zhao, ”Towards affordable on-track testing for autonomous vehicle – A Kriging-based statistical approach,” in *IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1-6.

[45] M. Grindal, J. Offutt and S. F. Andler, “Combination testing strategies: a survey,” in *Softw. Test. Verif. Reliab.*, vol. 15, 2005, pp. 167-199.

[46] E. Rocklage, H. Kraft, A. Karatas and J. Seewig, “Automated scenario generation for regression testing of autonomous vehicles,” in *IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 476-483.

[47] C. E. Tuncali, T. P. Pavlic and G. Fainekos, “Utilizing S-TaLiRo as an automatic test generation framework for autonomous vehicles,” in *IEEE 19th International Conference on Intelligent Transportation Systems*, Rio de Janeiro, 2016, pp. 1470-1475.

[48] H. Beglerovic, M. Stolz and M. Horn, “Testing of autonomous vehicles using surrogate models and stochastic optimization,” in *IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, 2017, pp. 1-6.

[49] M. Koren, S. Alsaif, R. Lee and M. J. Kochenderfer, “Adaptive Stress Testing for Autonomous Vehicles,” in *IEEE Intelligent Vehicles Symposium*, Changshu, 2018, pp. 1-7.

[50] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin and J. Schroeder, “Accident Scenario Generation with Recurrent Neural Networks,” in *21st International Conference on Intelligent Transportation Systems* , Maui, 2018, pp. 3340-3345.

[51] R. B. Abdessalem, S. Nejati, L. C. Briand and T. Stifter, “Testing advanced driver assistance systems using multi-objective search and neural networks,” in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, New York, 2016, pp. 63-74.

[52] R. B. Abdessalem, S. Nejati, L. C. Briand and T. Stifter, “Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms,” in *IEEE/ACM 40th International Conference on Software Engineering*, Gothenburg, 2018, pp. 1016-1026.

[53] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand and T. Stifter, “Testing autonomous cars for feature interaction failures using many-objective search,” in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, New York, 2018.

[54] J. Zhou and L. del Re, “Safety Verification Of ADAS By Collision-free Boundary Searching Of A Parameterized Catalog,” in *Annual American Control Conference*, Milwaukee, 2018, pp. 4790-4795.

[55] T. Hempen, S. Biank, W. Huber and C. Diedrich, “Model Based Generation of Driving Scenarios,” in *Intelligent Transport Systems – From Research and Development to the Market Uptake*, 2018, pp. 153–163.

[56] T. Helmer, “Development of a Methodology for the Evaluation of Active Safety using the Example of Preventive Pedestrian Protection,” PhD thesis, Technische Universitat Berlin, 2014.

[57] P. Nitsche, R. H. Welsh, A. Genser and P. D. Thomas, “A novel, modular validation framework for collision avoidance of automated vehicles at road junctions,” in *21st International Conference on Intelligent Transportation Systems*, Maui, 2018, pp. 90-97.

[58] W. Wachenfeld, P. Junietz, R. Wenzel and H. Winner, “The worst-time-to-collision metric for situation identification,” in *IEEE Intelligent Vehicles Symposium* , Gothenburg, 2016, pp. 729-734.

[59] J. C. Hayward, “Near-miss determination through use of a scale of danger, ” in *Highway Research Board*, no. 384, 1972, pp. 24-34.

[60] S. Hallerbach, Y. Xia, U. Eberle and F. Koester, “Simulation-based identification of critical scenarios for cooperative and automated vehicles,” SAE Technical Paper, 2018.

[61] J. Eggert, “Predictive risk estimation for intelligent ADAS functions,” in *17th International IEEE Conference on Intelligent Transportation Systems*, Qingdao, 2014, pp. 711-718.