RESEARCH ARTICLE

Operational Design Domain-Driven Coverage for the Safety Argumentation of Automated Vehicles

PATRICK WEISSENSTEINER [1], **GEORG STETTINGER** [2],
SIDDARTHA KHASTGIR [3], (Member, IEEE), AND
DANIEL WATZENIG [1,4], (Senior Member, IEEE)

[1] Virtual Vehicle Research GmbH, 8010 Graz, Austria
[2] Infineon Technologies AG, 85579 Neubiberg, Germany
[3] WMG, International Manufacturing Centre, University of Warwick, CV4 7AL Coventry, U.K.
[4] Institute of Automation and Control, Graz University of Technology, 8010 Graz, Austria

Corresponding author: Patrick Weissensteiner (patrick.weissensteiner@v2c2.at)

ABSTRACT Automated Vehicles aim to increase road safety as automated driving systems (ADS) take over the human driving task in the operational design domain (ODD), introducing severe challenges for safety validation. Pure driving over many kilometers to gather enough evidence for a safety argument is not feasible. Scenario-based testing is an approach to overcome this, but challenges like parameter discretization still prevail, hindering safety assurance. This work proposes contributions towards a traceable and efficient safety argumentation for ADS built upon ODD coverage. First, ODD coverage is thoroughly quantified across all scenario levels, assuming distribution functions' availability for the scenario parameters. Secondly, a sampling method for n-dimensional scenario parameter distributions is proposed. The provided algorithms adapt an initial k-means clustering using pre-defined boundary conditions requiring significantly fewer scenarios. Furthermore, a risk metric for urban intersections is presented for scenario evaluation. The risk metric consists of two parts, scene prediction of traffic participants (TPs) and risk assessment. The scene prediction uses a manoeuvre-based motion model with a data-driven approach towards trajectory prediction, increasing the validity. For the risk assessment, a probabilistic risk prediction for the TPs is performed for each scenario scene. The risk metric shows a reasonable tradeoff between sensitivity and specificity, outperforming time-to-collision. These contributions are exemplarily applied at an intersection using a simplified setup for generating TPs and ego vehicle trajectories. The results indicate that an increased safety argumentation is enabled using the proposed methods alongside a coverage process, facilitating further research.

INDEX TERMS Automated vehicles, autonomous vehicles, automated driving, validation, coverage, operational design domain, automotive safety, vehicle safety, safety argumentation.

I. INTRODUCTION

In 2016, more than 1.3 million traffic deaths happened worldwide. Compared to the growing global population, the death rate has, however, at least stagnated. Taking the increasing motorisation in large parts of the world into account suggests

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Ciani [ID].

that the rising safety measures in modern vehicles positively contribute to safety. [1].

Automated vehicles (AVs) are cyber-physical systems operating in open context [2]. AVs are equipped with automated driving systems (ADS) which are expected to increase the aforementioned safety benefits of vehicle automation further while enabling new mobility methods, e.g., robotaxis, in complex urban areas. ADS are SAE Level 3 to 5 [3]

as they take over the complete driving task without human supervision. The boundaries of operation for such systems are specified by the target operational design domain (ODD). The description of an ODD contains a set of parameters and their respective ranges. A respective taxonomy for such an ODD description is given in [4]. The open context brings various difficulties for the safety validation of these systems. In general, the definition of safety for ADS is a difficult concept which still needs to be clarified. However, certain initiatives exist (e.g. [5]). For example, in [6], specific interpretations of safety are defined, including the definition of safety as a process, which influences the possible validation strategies. Overviews of the different validation strategies are given in [7] and [8], respectively.

A. CHALLENGES OF SAFETY VALIDATION FOR ADS

Most of these strategies use the concept of scenarios, introduced in [9] and [10]. In there, different types of scenarios are defined. The logical scenario (LS) is a model of time sequences of consecutive scenes with parameters represented as ranges. The concrete scenario (CS) is an instance of a LS, with concrete values for each LS parameter. The continuous parameter (CP) is part of the LS and can take on every value between specific pre-defined ranges. In contrast, the discrete parameter can assume only specific values (e.g. binary values). However, it must be noted that the discrete parameters can also be used to split up existing LS into individual LS. These individual LS then contain only CP.

This leads to various validation strategies using scenario-based testing (SBT) as the basis. An overview is given in [11], [12], and [13]. The general issues regarding those strategies include:

- Defining the correct scenarios. These can come from certain databases (with data from traffic analysis, including accident data) and expert-based approaches.
- Providing suitable test methods. Even if the SBT approach offers ways to tackle the open context problem of ADS, there is still a tremendous amount of scenarios to be executed, which calls for suitable test methods, mostly virtual testing.
- Evaluating the executed scenarios and deciding on the correct metric to evaluate criticality.
- Provide a suitable safety argumentation which includes evidence of the achieved ODD coverage.

Potential solutions are formulated in various research initiatives, e.g. [14] and standardisation activities [15]. The question of how safe ADS-equipped vehicles need to be for operating in real traffic also concerns regulatory bodies on all levels. Recently, the United Nations Economic Commission for Europe (UNECE) proposed a regulatory framework for the safety assessment of ADS ([16], [17], and [18]). The framework consists of multiple pillars, including a scenario catalogue for conducting SBT, various test methods for scenario execution and assessment procedures. This framework aims to be repeatable, objective and evidence-based in assessing ADS safety. While the overall safety assessment

framework is clearly defined, specific details for an actual execution still need to be clarified, including methods for choosing CS' and metrics for scenario evaluation. This article aims to contribute to the resolution of these current issues.

There are also certain approaches in the industry towards the safety validation of ADS. General views on safety validation and certification are provided in [19] and [20]. A safety argumentation for assessing an automated lane-keeping system (ALKS) focused on virtual test methods is provided in [21]. However, the respective ADS has a narrow target ODD and severely bounded behaviour competencies, which makes the proposed methods only conditionally applicable for more complex use cases.

II. STRUCTURE OF THE ARTICLE

The remainder of the article is structured as follows: First, Sec. III will introduce the proposed methods and key results of this article, which all contribute towards an increased safety argumentation. Secondly, in Sec. IV, the methods are applied exemplarily. The generated results and their implications are discussed in Sec. V. The article concludes with a summary and an outlook in Sec. VI.

III. COVERAGE EVALUATION METHOD

As stated in [13], traditional distance-based statistical approaches are insufficient to cover the resulting test space. Various methods to explore the scenario space exist. In [22], a method for criticality identification to provide coverage estimates is proposed. It uses a criticality indicator for the exploration of the scenario space using a guided search. Another method is to estimate the probability of a random scenario being critical. The actual probability is unknown for a specific ADS, so estimation is necessary [13]. However, deriving confidence statements for such methods can take time and effort. They need to be derived for each ADS separately and also repeated for every change in the ADS. This introduces another source of potential error in the overall safety argumentation process.

In general, the required safety argumentation for AVs can be viewed as a combination of a microscopic and macroscopic assessment [11]. The microscopic assessment, as defined in [23], evaluates the individual scenarios using respective metrics. A metric for the microscopic assessment of scenarios at urban intersections is presented in Sec. III-D. The macroscopic assessment, as defined [24], compiles the individual microscopic assessments to statistical statements about the overall impact of AVs. A proposal for the macroscopic assessment is given in Sec. III-B. The combination of both assessments into an overall safety argumentation leads to the safety validation process, which is presented in Sec. III-A. Therefore, the goal of the safety validation process lies not in the search for critical scenarios or certain edge cases (e.g., [25], [26]). Such a search is essential during the development and internal validation of such systems, especially across different design iterations, as it can influence certain development decisions. Nevertheless, such edge
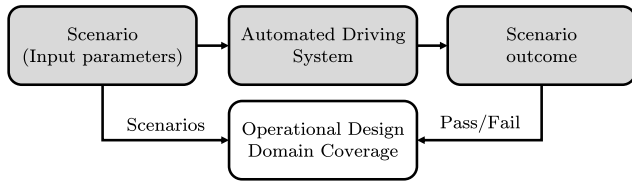
FIGURE 1. The coverage of the relevant ODD depends on the executed scenarios and if these are passed or failed.

cases inherently depend on the performance of the tested ADS regarding the experienced criticality, based on the pre-defined metrics.

The safety validation process essentially needs to be seen as a function of the defined target ODD of the ADS, as this defines the overall search space that needs to be covered. Additionally, it provides the necessary safety argumentation for the ADS and its respective target ODD. A proposal for such an overall process is shown in [27]. To determine the overall coverage, an exploration method independent of the respective ADS performance is needed. Hence, a deterministic and transparent way to calculate the overall coverage can only be achieved if it is based on the respective input parameters, meaning the executed scenarios. In other words, taking an open-loop approach towards coverage evaluation, as closing the loop and using the feedback of the experienced criticality in a guided search for the following scenario, would once again incorporate a specific ADS. This is displayed in Figure 1.

An overview of different methods for exploring the scenario space is given in [28]. As mentioned earlier, the focus is on sampling-based methods, as they provide the possibility to respect available parameter distributions, enabling the previously mentioned advantages. the advantage of which has also already been explained. Another advantage of sampling methods is that, since the drawn samples are mutually independent, the generated test cases can be executed in parallel, reducing overall testing time. Sampling methods, in general, tend to be inefficient if uniform distributions are assumed. Furthermore, the applied discretisation processes are not able to provide sufficient argumentations for the safety validation [29].

There are, after all, a few methods to increase efficiency for these assumptions (see [30] or [31]). In [31], different concepts for the coverage calculation, based on the past scenario in the overall scenario space, are presented.

However, this can be partly addressed using the respective parameter distributions of the CPs for each LS. These parameter distributions can e.g. include the likelihood of exposure to given scenarios. As opposed to using uniform distributions, realistic parameter distributions enable a more efficient design of the sampling process, e.g., by considering that higher probabilities of exposure in the real world are of high significance for the safety validation of ADS. A method for drawing efficient test case samples, while independent of ADS implementations, will be shown in Sec. III-C.

A. EXPLORING THE PROCESS OF COVERAGE EVALUATION
The aim of the coverage evaluation as part of the safety argumentation for ADS is to systematically ensure safe operation inside a pre-defined ODD. Therefore, a dedicated process needs to be in place to enable an efficient and traceable way to cover the target ODD of the ADS. Two crucial aspects of such a process are:
- Extendable towards all necessary target ODDs.
- The concrete ADS implementation is irrelevant.

In general, such a process starts by gathering all requirements for the process itself, including the ADS' target ODD, manoeuvers and the overall use case. These requirements are used for further execution of the process, presented in detail in [27].

The overall process consists of six steps, depicted in the left part of Figure 2. In *step 1*, the target ODD based on the top-level taxonomy from [4] is defined. Next, *step 2* defines the occurring disturbances, a concept defined in [14], which enhances the SBT approach with the underlying physical principles for each ADS subtask (perception, judgement and control in [14]). These disturbances also define the structure of the relevant LS, which is, together with the target ODD parameter of *step 1*, the input for *step 3*, the LS generation. In *step 3* and *step 4*, the CS' are created, whereas in *step 4+* they are executed using specific test methods (ranging from real-world proving grounds to virtual testing). Next, the executed test cases are evaluated using a pre-defined set of key performance indicators (KPIs) to determine if the test was passed or failed (*step 5*). This is then used as input for the determination of the achieved coverage (*step 6*).

While the presented process (left part of Figure 2) introduces an overall method together with a stepwise process regarding the coverage evaluation of ADS, certain aspects still need to be better defined. The aspects concerning the core part of the process (Figure 2) include a consistent definition of the coverage itself in the context of SBT, as introduced in Sec. I, and a respective set of metrics usable for pass/fail determination in an urban use case. Furthermore, the exact type of scenarios to include in the overall process, either gathered from scenario databases such as [32] or derived otherwise (e.g., using expert knowledge), is an open question. The first two aspects are being addressed in Sec. III-B and Sec. III-D, respectively. However, scenario creation is an equally important aspect that will be covered in future publications.

B. DEFINING COVERAGE IN THE EVALUATION PROCESS
As already examined at the beginning of Sec. III, most definitions of coverage in the domain of safety validation for ADS are either simplified pass/fail ratios of executed test cases or only cover the aspect of coverage partly (e.g., on parameter level) and not in the context of an overall SBT approach for safety argumentations. Therefore, the aim is to define coverage in a way that fits SBT, especially considering that for most LS descriptions, the associated CPs are defined as probability density functions (PDFs) using a specific metric.
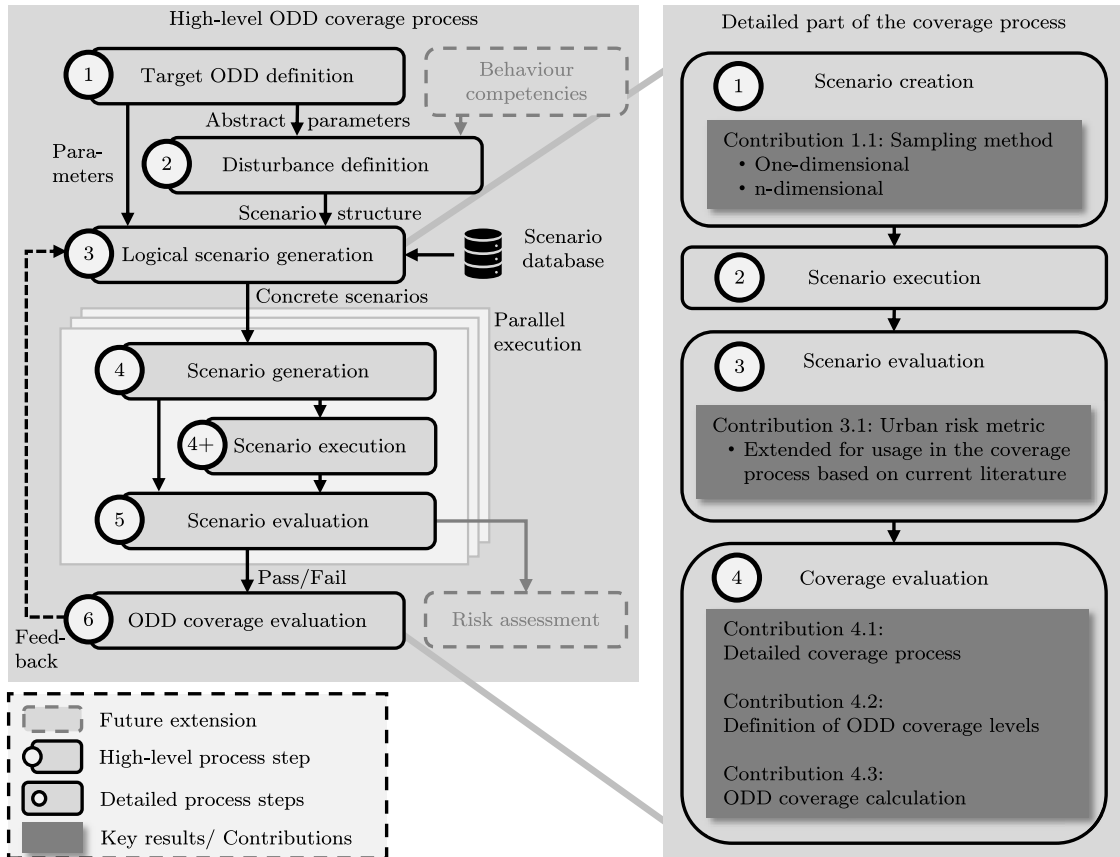
FIGURE 2. On the left, the high-level process for the determination of the ODD coverage, based on [27], can be seen. On the right, steps 3 to 6 of the high-level process are detailed. These four steps are the focus of this article, for which multiple contributions are provided.

This directly leads to the fact that each CS contributes differently to the overall coverage based on the associated metric in the CP PDF. This is in line with the basic idea of scenarios, where it is acknowledged that not every situation while driving is equally relevant. Furthermore, this means that some information is available regarding the scenarios (based on the utilised metric) which can be utilised for CS generation. For the remainder of this article, it is assumed that *the associated metric for the CP PDF is the probability of occurrence*. This means that values of the CP with higher densities are more likely to appear in the real world. However, other metric types can also be defined on the continuous parameter level and for CS and LS. Other possible metrics would be the associated risk of a CS (based on some pre-assessment) or specific metadata for a LS (e.g., amount of roads at a junction) or even a combination of those.

1) DEFINING THE DIFFERENT LEVELS OF COVERAGE

The previous section already mentioned that the different elements of a scenario (LS, CS, CP) could all have their metrics that influence the overall coverage. This fact is combined into a general overview, respecting the hierarchy between the different levels and includes statements regarding the test case domain and the number of test cases. It can be seen in Figure 3. Starting from the top, an ADS of SAE Level 5 would have to cover the entire input parameter space without any restrictions, leading to infinite test cases. The ODD is limited in the case of an ADS of SAE Level 3 or 4 (one level below in Figure 3). As the amount of test cases is still infinite, using the SBT approach, the ODD is split into a finite amount of LS, denoted as $S_L = \{S_{L,1}, S_{L,2}, \ldots, S_{L,n}\}$. However, for a respective LS and the associated scenario space, the test case domain is once again infinite. For each $S_{L,i}$, there is a number of CS defined, given as $S_{L,i} = \{S_{C,1}, S_{C,2}, \ldots, S_{C,n}\}$. Then again, for each $S_{C,i}$, concrete values are assigned, making it executable as a test case given a chosen test method. This is denoted as $S_{C,i} = \{x_1, x_2, \ldots, x_n\}$. At the lowest level, CPs define the actual values of each $S_{C,i}$. The CPs for a given $S_{L,i}$ are denoted as $S_{P,i} = \{S_{CP,1}, S_{CP,2}, \ldots, S_{CP,n}\}$. The elements of $S_{P,i}$ are the chosen test case values for each CP, e.g., $S_{CP,i} = \{x_1, x_2, \ldots, x_n\}$. Essentially, the generated CS for each LS can be seen as the cartesian product of each $S_{CP,i}$ associated with $S_{L,i}$, as the values and the number of resulting pairs are equal to the defined CS.

Having established the general relationship between the different levels of scenario descriptions, a thorough definition of coverage is possible. Starting with the overall coverage on
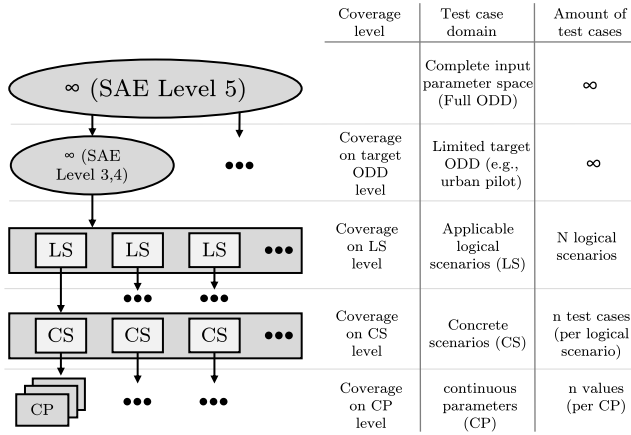
FIGURE 3. Defining the ODD coverage on different levels, from the target ODD towards the continuous parameters of the concrete scenarios.

FIGURE 4. Example PDF of a continuous parameter x as part of a concrete scenario. The area A_i equals the area under the curve between x_i, a and x_i, b and defines the coverage for that parameter range.

target ODD level, $c_{ODD} \in [0, 1]$ is given by

$$c_{ODD} = \sum_{i=1}^{|S_L|} c_{LS,i} m_{LS,i,coverage}, \quad (1)$$

where $c_{LS,i} \in [0, 1]$ is the coverage of one specific LS and $m_{LS,i,coverage}$ is the value of the specified metric to weigh the different LS across each other (e.g., according to the probability of occurrence). Over all LS, the sum of this metric is defined as

$$\sum_{i=1}^{|S_L|} m_{LS,i,coverage} := 1. \quad (2)$$

The individual coverage of a specific LS is

$$c_{LS,j} = \sum_{i=1}^{|S_{L,j}|} c_{CS,i}, \quad (3)$$

where $c_{CS,i} \in [0, 1]$ is defined as the coverage for a specific CS and is calculated using the following equation:

$$c_{CS,i} = \frac{\sum_{i=1}^{|S_{P,j}|} c_{CP,i}}{\left(\sum_{i=1}^{|S_{P,j}|} \frac{1}{|S_{CP,i}|}\right)|S_{L,j}|}, \quad (4)$$

where $c_{CP,i} \in [0, 1]$ is the coverage on CP level. To make sure that for a given LS $S_{L,j}$ the calculated coverage in Eq. 3 does not exceed the defined limit, the individual CS coverage calculated in Eq. 4 is corrected, using the amount of chosen CP values for each defined CP in the LS description and the total amount of CS for the specific LS. At last, $c_{CP,i}$ is given as

$$Pr(x_{i,a} \le X \le x_{i,b}) = \int_{x_{i,a}}^{x_{i,b}} f_X(x)dx = c_{CP,i}, \quad (5)$$

which equals A_i in Figure 4. This means that for a respective CP a specific value range will be represented by one concrete value, which becomes the test case value. This stems from the fact that specific values (for given value ranges) are needed to get executable scenarios and to cope with the infinite test
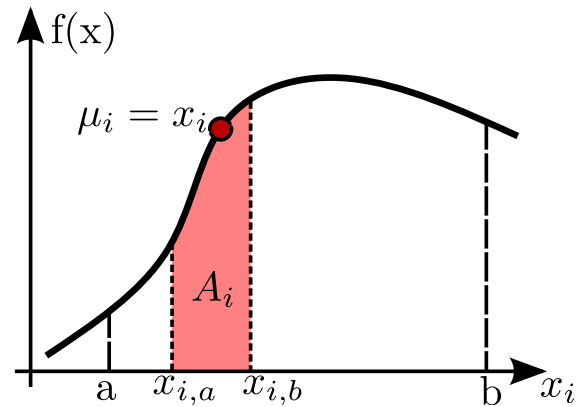
case domain of each LS. However, the probability that a value is occurring based on a given PDF is only defined for parameter ranges. Therefore, an approach combining specific value ranges with representative concrete values is needed. This representative value can e.g. be the mean of the value range, as this takes the shape of the associated PDF, for that specific range, into account.

It can be observed from Eq. 4 that each generated CS contributes differently to the overall coverage, depending on the covered area of each CP associated with the CS. Note that in Eq. 3 it is possible to include another metric (similar to $m_{LS,i,coverage}$ on LS level) to adjust the individual weighing of each CS. This would further influence how much it contributes to the overall coverage based on a metric defined on the CS level. This is omitted for now but will be explored in greater detail in future publications.

C. PROPOSED SAMPLING METHOD FOR EFFICIENT TEST CASE GENERATION

After defining a consistent way for ODD coverage calculation, the parameter discretisation process is examined further in this section. This process chooses exact values for each CP in a LS description to generate CS'. This technically belongs to step 1, depicted in Figure 2.

There are two extreme cases to consider when looking at sampling from a specific PDF for generating exact values of CP. On the one hand, choosing only one value and assigning the complete value range of a CP to this value, e.g. using the mean, would be possible. How advisable this is, depends on the shape of the PDF, as a narrow PDF is more suitable than a wide one. Therefore, the overall PDF's variance is essential (which is a property influenced by the shape of the PDF). For each area (e.g. A_i in Figure 4), the variance should be as slim as possible since, otherwise, the chosen representative value can differ a lot from actual values inside the respective value range. On the other hand, using Monte Carlo sampling with a high number of samples leads to minimal areas and narrow value ranges for each representative value, reducing

the respective variance and approaching zero in case of an infinite number of Monte Carlo samples. Even if both cases are unrealistic, they contribute towards displaying the necessity of generating representative samples for given CP PDFs in a traceable manner.

Exploiting the PDF of scenario parameters has already been explored in the literature. For example, in [33], the cumulative probability of a CP, based on synthetic data generated using traffic simulation, is split into equidistant areas, which all encompass the same amount of coverage on the CP level. In [34], existing parameter distributions are used for conditional sampling based on risk measures. As a concrete method for sampling, an expectation value-based kD-tree is used for dividing the parametric space, which can be n-dimensional [35]. The kD-tree performs a spatial subdivision based on a pre-defined number of divisions. Each resulting subspace has an equal amount of points sampled from the original PDF. It is in principle similar to the method in [33], however, extended towards higher dimensions, in case multiple CPs are dependent on each other and are defined over a joint PDF, as both methods aim for test cases of equal coverage on CP level.

If parameter distributions are used for the CPs of the defined scenarios (either one-dimensional, meaning one CP, or multi-dimensional, meaning multiple CPs together in a joint PDF), two main aspects need to be considered:

- Overall, the chosen test case values need to be as close as possible to the actual occurring values, given a pre-defined amount of values. Otherwise, Monte Carlo sampling would be the apparent default solution.
- The deviation must be limited for each test case value and must not exceed a certain predefined limit. This is necessary to ensure that a particular condition is met for each test case value, as the first aspect focuses on an overall condition across all test case values.

Therefore, in Sec. III-C2, specific metrics will be introduced to quantify different sampling methods across each other, including the proposed sampling method in Sec. III-C1, which covers both mentioned aspects.

1) METHOD FOR GENERATING SAMPLES FROM PARAMETER DISTRIBUTIONS OF CPs

Clustering is an unsupervised learning task already covered by literature in many aspects [36]. A well-known algorithm for clustering is called k-means [37], whose objective function is to minimise the total within-cluster distances (squared euclidian distances) to the closest centroids. This objective function directly addresses the first aspect stated at the end of Sec. III-C if the centroids from k-means are chosen as test case values. To apply clustering, samples must be drawn from the available parameter distributions of the CP. However, the second aspect (see the end of Sec. III-C) is still unaddressed. Therefore, the following proposed method will use k-means as the basis for the first clustering and further adapt this initial solution to address the second aspect. The steps of the

sampling method are all depicted in Figure 5 and will be explained stepwise in the following.

Step 1: It is assumed that the original data concerning some CP (in Figure 5 denoted as x_i and x_j) is available in point samples. Using kernel density estimation (KDE) on the data would lead to a respective estimation of the PDF. It could also be the case that parameter densities are already available, e.g. from extracted LS' from a scenario database.

Step 2: In any case, samples are drawn from the derived PDF in *step 1* and scaling is performed. Concretely, we denote $\mathcal{X} = \{x_1, x_2, \ldots, x_m\} \subseteq [0, 1]^d$ as the representation of the dataset on which the clustering is performed upon.

Step 3: Performing clustering on $\mathcal{X}$, we first define $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ as the respective partition of $\mathcal{X}$ into k clusters, with $C_i \cap C_j = \emptyset$, $i \neq j$. Additionally, the centroids are denoted as $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$, whereas $\theta_j \in [0, 1]^d$ is the centroid of C_j. It is expressed as $\theta_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$. Now, we define the within cluster distances W_{C_j} between samples $x \in C_i$ as

$$W(C_i) = \sum_{x \in C_i} \|x - \theta_i\|_2^2, \qquad (6)$$

with $\|.\|_2^2$ being the squared euclidian distance. This defines also the variance within each cluster. k-means is then defined as the minimisation of the objective function

$$J(C; \Theta) = \sum_{i=1}^{|\mathcal{C}|} W(C_i). \qquad (7)$$

The most common algorithm to tackle this problem is Lloyd's algorithm [38] which uses a greedy strategy to approximate $J(C; \Theta)$. As initialisation of the centroids for the first step, the k-means++ initialisation is used [39]. For the concrete implementation, used for generating the benchmark in Sec. III-C2, the Scikit-learn [40] package is utilised.

Step 4: After having completed the initial partition into k clusters, the clusters are now adapted in the following steps. First, the probability p_{C_i} that $x \in C_i$ is calculated using,

$$p_{C_i} = \frac{|C_i|}{\sum_{i=1}^{|\mathcal{C}|} |C_i|}. \qquad (8)$$

The two properties of each cluster, stated in Eq. 6 and 8, can be seen in a qualitative representation in Figure 5 for *step 4*. There, the x-axis, denoted as $P_{Occurence}$, represents the individual occurrence probabilities p_{C_i} for each cluster, whereas the y-axis, denoted as $\text{tr}(\Sigma)$, is the trace of the covariance matrix. This is used as a scalar-valued generalisation for higher dimensions interpreting the deviation in the variance definition as euclidian distance since in Eq. 6 the squared euclidian distance was also interpreted as a variance. If X is a column vector which elements are random variables X_i and Σ is the covariance matrix between each components of X, the scalar valued variance Var_s is defined as

$$\text{Var}_s(X) = \text{E}\big[\|X - \text{E}[X]\|_2^2\big]. \qquad (9)$$
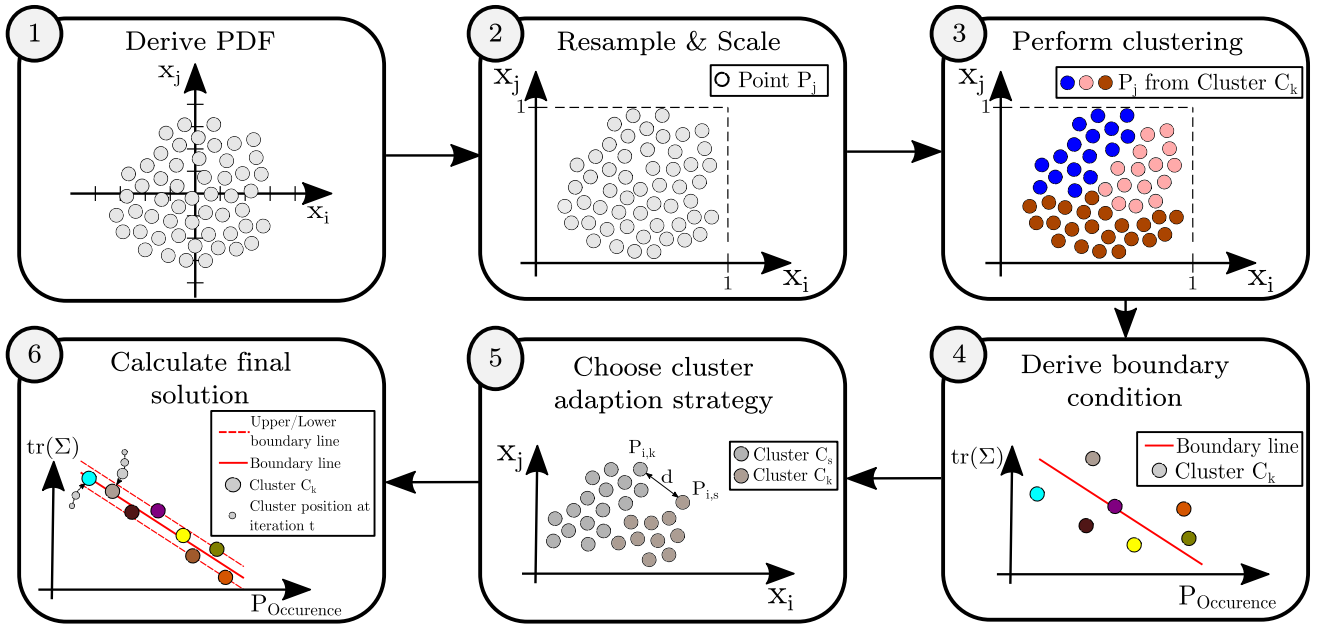
FIGURE 5. The six steps of the developed method for efficient sampling from PDFs of continuous parameters are part of logical scenario descriptions. The method applies to n-dimensional PDFs of *n* continuous parameters and is pictured for the two-dimensional case.

After some rewriting, which is omitted, this equals the sum of the individual variances of each X_i, which in turn equals $\text{tr}(\Sigma)$. Therefore, in the following, the use of $\text{tr}(\Sigma)$ is equal to $W(C_i)$. As the overall goal is to bound each within-cluster variance $W(C_i)$, based on some pre-defined condition to fulfil the second aspect presented in Sec. III-C, the following inequality is defined:

$$W(C_i) \leq g(p_{C_i}) + \epsilon \ \forall \ C_i \in \mathcal{C}, \tag{10}$$

with ϵ being a threshold value and g being described as follows:

$$g : p_{C_i} \in [0, 1] \mapsto k \, p_{C_i} + d \in \mathbb{R}^+, \tag{11}$$

where k and d are parameters to be defined. Eq. 10 determines the condition each cluster needs to obey. For a start, the condition itself, Eq. 11, is expressed as a boundary line and reflects the following assumption: For a cluster C_i, where its centroid θ_i represents the actual test case value, with higher occurrence probability, p_{C_i}, there should be a lower within-cluster variance W_{C_i}. This makes intuitive sense. For a cluster representing more of the sampled points than a cluster with lower occurrence probability, the imposed condition (in this case, the variance) should be more strict. How exactly this relationship ($\uparrow p_{C_i} \Rightarrow \downarrow W_{C_i}$) is characterised, needs to be explored more in future publications. As stated in Eq. 11, a simple boundary line with two parameters is chosen, as it reflects the basic intuitive assumption. The parameters are chosen using linear regression on the points defined by the properties of each cluster, as can be seen in *step 4* in Figure 5. This gives a realistic boundary, as some clusters will be above and beyond the line. For future publications,

assuming a predefined boundary condition will be explored further. Adjusting the sampling method presented in this article would enable the determination of the required amount of test cases, considering that each of these cases needs to obey the boundary condition.

Step 5: In this step, a respective adaption strategy needs to be specified so that all clusters obey the respective boundary, discussed in *step 4*. The general idea, as qualitatively depicted in Figure 5, is to efficiently search for pairs of points in the sample space that fulfil two criteria:

- They belong to different clusters and,
- the euclidian distance between them falls below a certain threshold.

These point pairs are potential candidates for exchange between their respective clusters. Because fundamentally, clusters above the boundary need to drop some points to reduce the occurrence probability. As the dropped points, by definition, are rather far off from their respective centroid, the within-cluster variance is also reduced. Eventually, if enough points are dropped, the boundary condition will be met. This already describes the basics of the developed algorithms. Concretely, two algorithms for cluster reduction were developed, Alg. 1 for the one-dimensional problem and Alg. 2 as a generalisation for the n-dimensional problem. In the following, both algorithms are explained.

The one-dimensional algorithm, shown in Alg. 1, deals with the situation of having clustered samples (re-scaled to [0,1]) from the parameter distribution of a CP, which is independent of all other CPs defined in a LS. This means that only the direction of the x-axis is left as the degree of freedom for dropping or absorbing points in clusters, which

is in maximum two other clusters ("left" and or "right" of the cluster in focus). Considering this fact enables an adapted algorithm design for the one-dimensional problem. Taking the overall point samples and the clusters and centroids from the k-means clustering as input, and the adapted clusters are returned. At first, the clusters are sorted based on the individual centroids, from small to large, and the cluster properties, W_{C_i} and p_{C_i}, are calculated. Then, until each cluster obeys the boundary condition, three options are available for each cluster. Option 1 is that the cluster is inside the ϵ-threshold, which means no further action is taken for this cluster. Option 2 means the cluster is above the boundary line $g(p_{C_i})$. Therefore, points are dropped towards the next cluster based on the sorting. The last option, a cluster being below $g(p_{C_i})$, means further points can be added. The euclidian distance threshold between points of different clusters determines how many points are dropped or added. If the threshold is high, the algorithm needs fewer iterations as more points are exchanged in each iteration and vice versa.

Algorithm 1 One-Dimensional Algorithm for Cluster Adaption

Require: $\mathcal{X} \subseteq [0, 1], \mathcal{C}, \Theta, \epsilon$, threshold,
    number of samples m, number of clusters k
Ensure: $\mathcal{C}$
  print sort($\mathcal{C}$) after Θ,
  Obtain $W(C_i)$ & p_{C_i} by Eq. 6 & 8 $\forall\ C_i \in \mathcal{C}$
  repeat
    for $i = 1, 2, \ldots, k$ **do**
      if $W(C_i) \leq g(p_{C_i}) + \epsilon$ **then**
        continue
      else
        for $j = 1, 2, \ldots m_i$ **do**
          $d_{j,l} = \|x_j - x_l\|_2\ \forall\ x_l \in C_{i+1}$
          if $d_{j,l}$ < threshold **then**
            if $W(C_i) > g(p_{C_i})$ **then**
              $C_i = \{C_i \setminus x_j\},\ C_{i+1} = C_{i+1} \cup \{x_j\}$
            else if $W(C_i) < g(p_{C_i})$ **then**
              $C_{i+1} = \{C_{i+1} \setminus x_l\},\ C_i = C_i \cup \{x_l\}$
            end if
          else
            continue
          end if
        end for
      end if
    end for
    update: $W(C_i)$ & p_{C_i} using Eq. 6 & 8 $\forall\ C_i \in \mathcal{C}$
  until $W(C_i) \leq g(p_{C_i}) + \epsilon\ \forall\ C_i \in \mathcal{C}$
  return $\mathcal{C}$

The generalisation of the one-dimensional algorithm towards n-dimensions is based on the same principles of exchanging points, which are part of the original dataset $\mathcal{X}$ and assigned to different clusters, to obtain a final set $\mathcal{C}$ that obeys Eq. 10. At first, the cluster properties W_{C_i} and p_{C_i} are calculated for each cluster. Additionally, we denote $\mathcal{S} \subseteq \mathcal{C}$

which is constructed as:

$$\mathcal{S} = \{C_i \in \mathcal{C} \mid W(C_i) > g(p_{C_i}) + \epsilon\}. \tag{12}$$

Now the algorithm repeats the following steps until $\mathcal{S}$ is empty: Iterating through $\mathcal{S}$, all possible other clusters (except the current cluster itself) are checked for how much they reduce the defined objective function

$$J(\mathcal{C}) = \sum_{i=1}^{|\mathcal{S}|} W(S_i) - g(p_{S_i}) - \epsilon. \tag{13}$$

Once again, points with less euclidian distance than a pre-defined threshold are considered for the exchange. After having iterated through every possible combination, the option which minimises the objective function in Eq. 13 the most, is chosen and a new iteration starts.

Algorithm 2 N-Dimensional Algorithm for Cluster Adaption

Require: $\mathcal{X} \subseteq [0, 1]^d, \mathcal{C}, \epsilon$, threshold
Ensure: $\mathcal{C}$
  print Obtain $W(C_i)$ & p_{C_i} by Eq. 6 & 8 $\forall\ C_i \in \mathcal{C}$; $\mathcal{S}$ from Eq. 12
  repeat
    for $S_i \in \mathcal{S}$ **do**
      for $C_j \in \{\mathcal{C} \setminus S_i\}$ **do**
        $D_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2,\ \boldsymbol{x} \in C_i,\ \boldsymbol{y} \in S_i$
        $(d_{i,j}) \in \mathbb{R}^{|C_i| \times |S_i|}$
        for $d_{i,j}$ < threshold **do**
          $C_i' = C_i \cup \{\boldsymbol{y}_j\},\ S_i' = \{S_i \setminus \boldsymbol{y}_j\}$
        end for
        $\mathcal{C}_{i,j}' = \{\mathcal{C} \setminus \{C_i, S_i\}\} \cup \{C_i', S_i'\}$
        $\mathcal{L} = \mathcal{L} \cup \{\mathcal{C}_{i,j}'\}$
      end for
    end for
    $\mathcal{C}' = \operatorname{argmin}_{\mathcal{C}' \in \mathcal{L}} J(\mathcal{C}')$
    update: $\mathcal{C} = \mathcal{C}'$, $\mathcal{S}$ from Eq. 12
  until $\mathcal{S} = \emptyset$
  return $\mathcal{C}$

Step 6: In this last step, the proposed algorithms, either Alg. 1 or Alg. 2, depending on the dimensionality of the problem, are applied to the original clustering of $\mathcal{X}$ using k-means, which leads, qualitatively, to the situation depicted for *step 6* in Figure 5, as all clusters, based on their properties, obey to the defined boundary condition. Next, the proposed sampling method (adapted k-means) will be benchmarked against other methods.

2) BENCHMARKING THE PROPOSED METHOD FOR CONTINUOUS PARAMETER SAMPLING

The proposed adapted k-means addresses both aspects defined in Sec. III-C that sampling methods should have. In order to compare this method to other ones and showcase in which aspects the method outperforms others, a short benchmarking is performed for the one-dimensional and two-dimensional cases. In total, five methods are included in the

benchmark. The first option is simply sampling from a given PDF, basically a Monte Carlo approach. As this method was mentioned in Sec. III-C as an extreme case, it was essential to include it here. Then, equidistant steps across the value range are used next for the one-dimensional case. For the two-dimensional case, Latin hypercube sampling (LHS), a much more efficient method based on [41], is used. Both methods discourage the available parameter distribution and assume an underlying uniform distribution. However, it should be showcased that if a respective PDF is available, it is best to utilise it. That is why these methods are included. The last two methods are the *k*-means and the proposed adapted *k*-means approach, respectively. Next, the two relevant metrics are introduced, followed by the definition of the used parameter distributions and the actual results of the benchmark.

The two metrics are:

- *Distance-based metric* (DBM): The idea of this metric is that if a large number of samples m is drawn from the parameter distribution, it should be measured how close the closest test case value is to each of the sample points. Concretely this is expressed as

$$\text{DBM} = \frac{1}{m} \sum_{i=1}^{m} \min_{\forall \theta \in \Theta} \|\boldsymbol{x}_i - \boldsymbol{\theta}\|_2. \quad (14)$$

The aim of a sampling method for parameter distributions of CP in a SBT approach is therefore to reduce DBM as much as possible as it reduces the uncertainty that is inevitably introduced since not every exact value that is possible can also be tested.

- *Standard deviation metric* (STDM): This metric enables measurement of the expected standard deviation for a chosen cluster, scaled by the actual occurrence probability of said cluster. This directly reflects the already mentioned fact that clusters with higher occurrence probability should have stricter boundaries on the within-cluster variance. It is defined as

$$\text{STDM} = \sqrt{W(C_i)}\, p_{C_i}. \quad (15)$$

Both metrics operate only on clusters and their properties, which means that they are independent of the dimensions of the parameter distributions.

To benchmark the methods, two examples, one and two-dimensional, have been constructed. For the one-dimensional case, the probability density is defined using a mixture of two Gaussian distributions,

$$f(x) = \sum_{i=1}^{n} w_i p_i(x), \quad (16)$$

with $n = 2$, $w_1 = 0.55$, $w_2 = 0.45$, $p_1 = \mathcal{N}(-2, 1)$ and $p_2 = \mathcal{N}(2, 0.04)$. For the two-dimensional case, two one-dimensional Gaussian distributions, $p_1(x) = \mathcal{N}(0.5, 0.25)$ and $p_2(x) = \mathcal{N}(0.2, 1)$, are defined. The two-dimensional samples are obtained by adding and subtracting $p_1(x)$ and $p_2(x)$ for the x and y axis, respectively. Executing all the methods on this given examples for various amounts of test cases

TABLE 1. For test case amounts between 10 and 20 (rows), for each given method (column) the test case reduction potential in per cent when using the adapted *k*-means, based on the DBM, is given. Positive values refer to reduction potential for the adapted *k*-means.

| Test cases	Test case reduction potential (%) compared to the adapted *k*-means:		
	K-means	Equidistant steps	Sampling from PDF
10	9.1	28.6	65.5
11	8.3	31.2	65.6
12	7.7	29.4	62.5
13	7.1	27.8	63.9
14	6.7	26.3	64.1
15	6.2	25.0	61.5
16	5.9	23.8	62.8
17	5.6	26.1	61.4
18	5.3	25.0	60.9
19	5.0	24.0	59.6
20	4.8	23.1	60.8

TABLE 2. For test case amounts between 5 and 20 (rows), for each given method (column) the test case reduction potential in per cent compared to the adapted *k*-means, based on the DBM, is given. Positive values refer to reduction potential for the adapted *k*-means.

| Test cases	Test case reduction potential (%) compared to the adapted *k*-means:		
	K-means	Sampling from PDF	LHS
5	-25	78.3	84.4
6	25	76.9	83.3
7	12.5	73.1	80.6
8	11.1	70.4	81.0
9	10	72.7	81.6
10	9.1	69.7	80.0
11	8.3	67.6	78.0
12	7.7	66.7	76.0
13	-8.3	65.8	74.0
14	6.7	66.7	72.0
15	11.8	64.3	70.0
16	-14.3	61.9	68.0
17	-6.2	62.2	66.0
18	14.3	64.0	64.0
19	-18.8	57.8	62.0
20	-5.3	59.2	60.0

leads to the results shown in Tab. 1 for the one-dimensional example, Tab. 2 shows the two-dimensional case. For both Tables, the values shown for each column represent the fractional of the necessary test cases for the adapted *k*-means over the other respective method to achieve the same value for the DBM. Both evaluations are also depicted for one and two dimensions in Figure 6 and Figure 7, respectively.

For the one-dimensional case, the mean of the reduction potential is approx. 6.5% over the *k*-means, 26.4% over the equidistant steps, and 62.6% over sampling from the PDF. For the two-dimensional case, these values are 2.4% over the *k*-means, 67.3% over sampling from the PDF, and 73.8% over using LHS.

The achieved results show various things: First, the performance of the adapted *k*-means in comparison to the *k*-means is not degraded by applying the cluster adaption algorithms, neither in one nor in two dimensions. Secondly, it can be observed that both clustering techniques outperform all other benchmarked methods severely. This indicates,
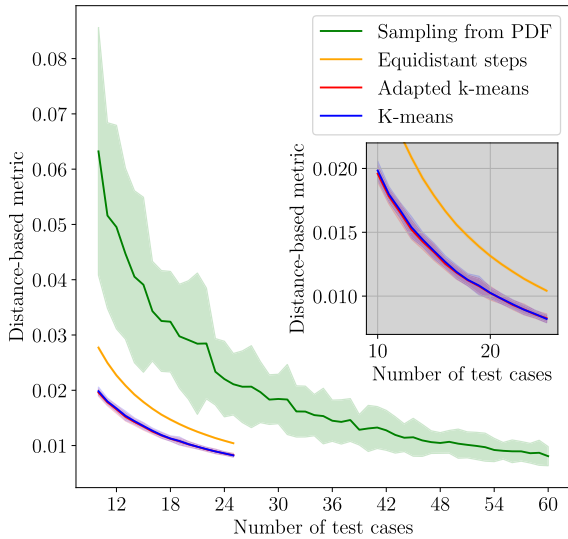
FIGURE 6. The number of test cases over the DBM for the different types of sampling methods for the one-dimensional case. The number of test cases refers to the number of samples drawn from the PDF of the continuous parameter.
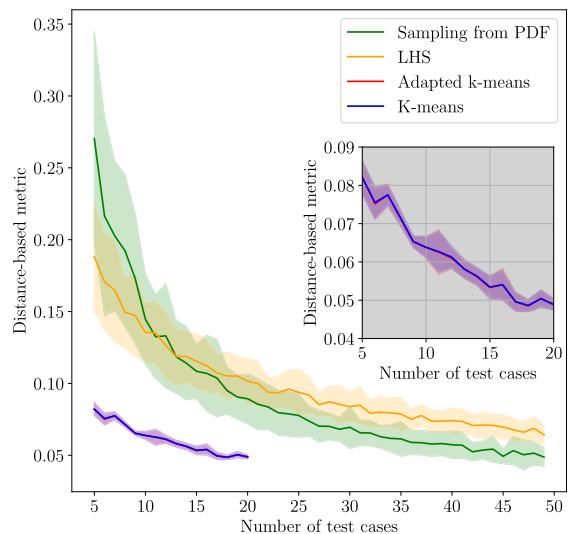
FIGURE 7. The number of test cases over the DBM for the different types of sampling methods for the two-dimensional case. The number of test cases refers to the number of samples drawn from the joint PDF of the continuous parameters.

based on the applied metrics, that utilising parameter distributions, when available, offers a large test case reduction potential on CP level. This effect is further enhanced on LS level since it is typically composed of many individual CP.

At last, for the one-dimensional case, the STDM is observed. Calculating this metric for each cluster in the given example is depicted in Figure 8. The upper boundary line is created by inserting Eq. 10 into the STDM (Eq. 15). This means, that for the adapted *k*-means the STDM is bounded based on the given boundary condition. In Figure 8 it can be observed that one specific cluster of the *k*-means approach is violating this boundary. All clusters of the adapted *k*-means are within the given boundary.
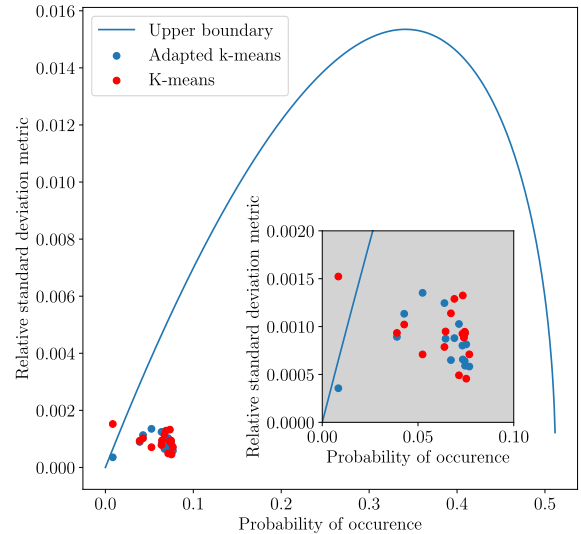
FIGURE 8. For the one-dimensional case, the probability of occurrence for each cluster is pictured over the STDM. It can be seen that in the case of the classical k-means approach, one cluster violates the chosen boundary condition by being above the upper boundary, depicted in blue.

D. METRICS FOR THE EVALUATION OF SCENARIOS

This section deals with the evaluation of executed scenarios. Following the main steps of the ODD coverage process, this corresponds to step 3 (see Figure 2). Evaluating scenarios means analysing if a certain amount of criticality did occur during the execution, whereas *criticality* is defined as *'the combined risk of the involved actors when the traffic situation is continued'* [42]. The most obvious way to determine that is by checking if an actual collision between entities of a scenario did occur. However, that would disregard certain near-misses and would provide no means of indicating the criticality of those scenarios. Therefore, many different criticality metrics exist, as they are not only relevant for evaluating scenarios during the safety validation phase using a SBT approach. There are applications for such metrics along the entire V-model [43]. The most well-known metric is the time to collision (TTC), which calculates the time until a predicted collision between traffic participants [44]. It originates from traffic conflict research and is, therefore, collision-focused. There exist many adapted and improved versions of TTC. In [45], different accident configurations are considered, taking the vehicles' width and length into account and providing a more accurate TTC. For a-posteriori traffic data analysis, the post encroachment time (PET) calculates the time gap between a traffic participant leaving and another entering a pre-defined area [46]. A comprehensive overview of criticality metrics is given in [47]. The authors of [47] also define a set of properties relevant to such metrics. These properties are (excerpt):

- *Reliability*: Describes how close repeated measurements are to one another.
- *Validity*: This property is concerned with how close the measurements are to the actual accident probability.

- *Sensitivity*: This is the true positive rate (TPR), which is defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad (17)$$

  whereas TP are the true positives and FN are the false negatives.
- *Specificity*: Similarly, specificity is the true negative rate (TNR), which describes the rate of correctly identified uncritical situations, expressed as

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \qquad (18)$$

  with TN and FP being the true negatives and the false positives, respectively.

In this article, the focus is on the test evaluation of certain scenarios, based on pre-defined pass/fail criteria to determine the overall coverage of the target ODD for a specific ADS. If the focus is on an urban use case, including complex intersections, TTC is considered to not be an ideal choice, since it suffers from reduced validity and sensitivity [47].

Various literature exists that aims to apply distinct risk metrics towards urban-based use cases. In [48], a probabilistic risk assessment algorithm considering occlusion, focusing on integration into planning algorithms, is proposed. In [49], a potential risk assessment for occluded areas in urban situations is proposed by threat modelling of the potential risk, however, it is designed for real-time purposes. In [50], a virtual risk assessment, determining the suitability of an automated shuttle deployment in suburban areas, is presented, taking into account the actual geo-location. Another risk metric for urban use cases is presented in [51]. The authors propose a threat metric in complex urban scenarios using the inD dataset [52]. Moreover, they perform a reachability analysis, a technique to compute reachable system states based on parametrised motion models. Also, respective lane association is considered. However, this association is static, not considering the typical driver's intention. This leads to a threat metric which is prone to reduced sensitivity.

1) STRUCTURE OF THE DEVELOPED METRIC FOR URBAN JUNCTIONS

A useful structure of such metric, which also applies towards simple ones like TTC, can be seen in Figure 9, which is based on [53]. It serves as an overview for the urban risk metric which is proposed to be used to evaluate executed scenarios during the ODD coverage process. The two main parts, the scene prediction and the risk assessment, will be explained in detail in Sec. III-D2 and Sec. III-D3, respectively.

2) SCENE PREDICTION

Many advanced metrics (including the proposed urban risk metric) calculate the risk for every scenario scene and then derive a final risk value for the whole scenario. On that basis, the pass/fail criteria are evaluated. To do that, for every scene in the scenario, a scene prediction needs to be carried out
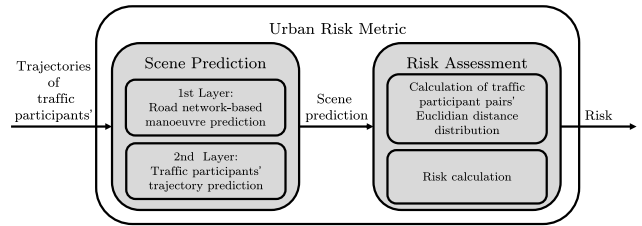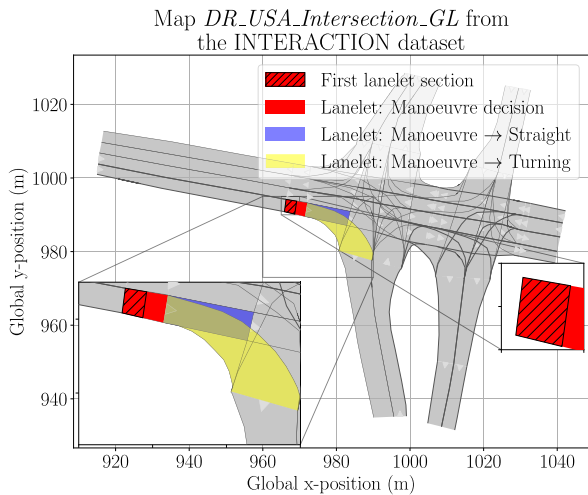

FIGURE 9. The overall structure of the urban risk metric, based on [53].

to calculate criticality based on a predicted future evolvement of the scenario. This prediction either, for a pre-defined prediction horizon, calculates a single evolution or multiple ones for each traffic participant in the scenario. For the single evolution, the worst case is mostly chosen, which leads to severely reduced specificity. Multiple evolvements can be carried out using a probabilistic framework or calculating reachable sets using reachability analysis.

In [54], motion models for prediction are categorized into three levels, with increasing abstraction. Physics-based models consider the motion of vehicles based on the laws of physics. Manoeuvre-based motion models consider the future motion of the vehicle also based on the potential driver's intention. Interaction-aware models are additionally taking into account the vehicles' manoeuvres (e.g., [55], [56]). Another option is to define potential functions for each object type and aggregate these functions to obtain a manoeuvre model [57].

If a manoeuvre-based motion model is used, the actual trajectory of the vehicle needs to be predicted. In [58], the authors perform trajectory prediction based on observations using image processing and map-matching techniques. It is shown that these generated predictions outperform the usual physics-based motion models in the observed intersections. Schreier et al. perform criticality assessment using a manoeuvre-based trajectory prediction utilising a Bayesian network where each manoeuvre is modelled separately [59]. Several other approaches towards trajectory prediction exist, using data-driven machine learning techniques (e.g., [60], [61], [62], [63]).

The scene prediction for the urban risk metric uses a manoeuvre-based motion model prediction with a data-driven approach towards concrete trajectory prediction. Using a data-driven approach increases the validity of the applied risk metric, although it is restricted to the locations with available data. However, if these locations represent complex intersections (e.g. included in the inD dataset [52] or the INTERACTION dataset [64]), these are vital aspects of the defined target ODD, which may justify the additional effort associated with gathering the necessary amount of traffic observations. Concretely, a specific map of the INTERACTION dataset is used to develop the scene prediction and provide an exemplary application in Sec. IV-A. The dataset provides naturalistic motions of different traffic participants for various complex intersections. One of these intersections is chosen to showcase the developed method and can be seen

FIGURE 10. One specific map of the INTERACTION dataset [64], provided in the format Lanelet2 [65].

TABLE 3. Performance comparison of different classifiers used for manoeuvre estimation for the manoeuvre decision depicted in Figure 10.

Type of classifier	Train scores	Test scores
Decision Tree Classifier	0.916	0.931
Random Forest Classifier	**0.944**	**0.951**
KNeighbors Classifier	0.933	0.945
SVC	0.908	0.915
Gaussian NB	0.871	0.883
Logistic Regression	0.885	0.895

in Figure 10. The junction is provided in the Lanelet2 format [65]. The chosen map provides an unsignaled intersection and approx. 260 minutes of birds-view video material with more than 10 thousand vehicles. The dataset can be evaluated extensively with the provided Lanelet2 map, which provides good starting points for further analysis.

Based on the overall structure of the urban risk metric (Figure 9), the scene prediction is divided into two separate layers. The first layer performs a road network-based manoeuvre prediction. In contrast, the second layer performs an actual trajectory prediction, providing a respective scene prediction for the second primary step, the risk assessment.

For the road network-based manoeuvre prediction, a classification problem is formulated, where the goal is to store pre-trained classifiers for each relevant lanelet section. A lanelet section is defined by four nodes (two nodes for each side). Doing that enables better performance of each individual classifier. All lanelets at respective road forks, and for which more than one option for future manoeuvres is possible, are considered. A respective ground truth (GT) is needed for training the classifiers. The GT is created by analysing the provided vehicle tracks in the dataset and matching it with a pre-defined list of lanelet IDs that belong to a global route (e.g. the route from the top left of Figure 10 going right (from top-view), crossing the red lanelet and taking the immediate right turn). Furthermore, certain manoeuvres are connected with that global route. Additionally, for these manoeuvres, it is defined at which respective lanelet the manoeuvre decision has to be made and which lanelets are the successors. For example, for the red lanelet in Figure 10, a decision if turn right or go straight needs to be made. The features for the classifier include the lateral position (in regards to the respective lane in the provided map), the vehicles' velocity and the global heading for each lanelet section. An overview of manoeuvre intention estimation at road intersections and its associated features is given in [54].

In our case, the required supervised learning algorithm acts on a dataset $\mathcal{D}$, drawn from an unknown distribution, which is defined as

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \subseteq \mathbb{R}^3 \times \mathcal{L}\} \quad (19)$$

where n is the size of the dataset, $\mathbb{R}^3$ is the feature space and $\mathcal{L}$ is the label space. Furthermore, y_i is the label and $\boldsymbol{x}_i$ is the feature vector of the i_{th} sample, which is expressed as

$$\boldsymbol{x}_i = [\text{velocity, global heading, lateral position}]^T. \quad (20)$$

The label space, for the example shown in Figure 10, is designed to be:

$$\mathcal{L} = [\text{Turning, Straight}] \quad (21)$$

The goal of the algorithm is to find $h : \mathbb{R}^3 \to \mathcal{L}$ so that for new samples $(\boldsymbol{x}_1, y_1)$ we get $h(\boldsymbol{x}) \approx y$.

Different classifiers were trained and evaluated for the situation depicted in Figure 10. The results of the test and train scores (80/20 split) can be seen in Tab. 3. The random forest classifier (RBF), which fits several decision tree classifiers on various sub-samples of the dataset [66], performs best for the test and the training dataset. In general, it can be observed that the manoeuvre intention is predictable with very high accuracy. Therefore, for future trajectories, for example, considering the decision for a right turn or going straight in Figure 10, it is not necessary to always chose the worst case, which would unnecessarily reduce specificity.

Suppose the probability for a particular predicted manoeuvre is below 0.65. In that case, the prediction is disregarded, all possible manoeuvres are considered future trajectories, and a worst-case estimation is performed in the risk assessment. Future advancements in the urban risk metric regard the investigation of calibrated classifiers. If such classifiers are being used, it would be possible to utilise the different prediction probabilities for each manoeuvre as a weighting factor for the predicted trajectories, which can be exploited for the risk assessment. This requires a classifier with a low log loss, which is necessary if the probability estimates of the classifier are utilised directly.

As the first layer of the scene prediction provided the manoeuvre estimation, the second layer predicts the actual future trajectories of the traffic participants. The output is thereby defined by the necessary input to the second major part of the urban risk metric, which is based on the probabilistic evolvement of object positions for the pre-defined
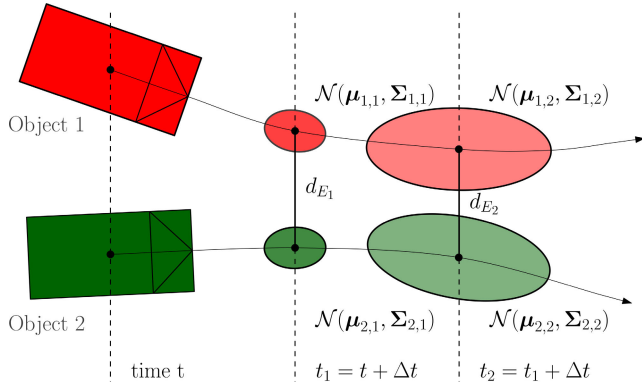
FIGURE 11. The potential evolvement of each object position for a specific time *t* in a scenario is modelled as bivariate Gaussian distribution for each following timestep.

FIGURE 12. The trajectory prediction of a traffic participant (Layer 2 of the scene prediction in the urban risk metric) on the map depicted in Figure 10.

prediction horizon. As Figure 11 shows, a bivariate Gaussian distribution is constructed for each object and each timestep Δt. This distribution is described in the respective local object coordinate system for each prediction timestep. It distinguishes between a longitudinal prediction along the axis in the driving direction and a lateral prediction perpendicular to this. It is crucial to notice that there is no restriction regarding the shape of the bivariate Gaussian distribution.

For the traffic participants' lateral prediction along the road, an Ornstein-Uhlenbeck process [67] is modelled based on [59]. The process has an exponentially decaying autocovariance function and is discretised as

$$y_{lat,i+1} = e^{-\alpha T} y_{lat,i} + (1 - e^{-\alpha T}) u + w_{y_{lat}} \forall i \in \{j : j + T_P\},$$
(22)

with α being a constant defining the rate of decay, T is the current time in the prediction horizon, and u is defined as the long-term mean of the process. Furthermore, $w_{y_{lat}}$ is the process noise scalar, with a variance stated as

$$Q_i = \sigma_{y_{lat}}^2 (1 - e^{-2\alpha T}),$$
(23)

where $\sigma_{y_{lat}}^2$ equals the limiting value of Q_i. The notation $\{j : j + T_P\}$ equals the sequence $\{j, j+1, \ldots, j+T_P$ with $T_P\}$ being the amount of prediction timesteps, calculated as
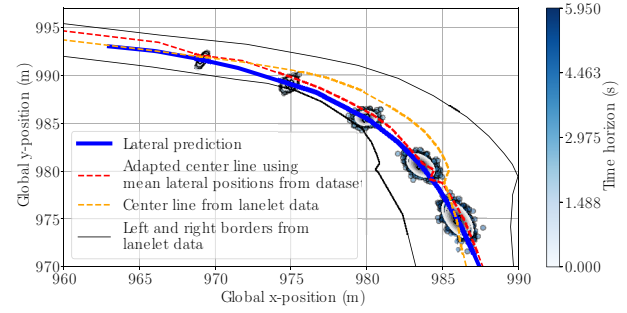
$$T_P = \frac{t_{pred}}{\Delta t},$$
(24)

with t_{pred} being the prediction horizon and Δt the prediction timestep. This leads to the following Gaussian distribution of the lateral prediction:

$$\mathcal{N}(e^{-\alpha T} y_{lat,i} + (1 - e^{-\alpha T}) u, Q_i).$$
(25)

The longidutinal prediction for the traffic participant at each timestep in the prediction horizon is defined as

$$x_{lon,i+1} = x_{lon,i} + v_{lon,i} \Delta t + \frac{1}{2} a_{lon,i} \Delta t^2 \forall i \in \{j : j + T_P\},$$
(26)

which equals a constant acceleration model, with $x_{lon,i}$ being the current position along the prediction horizon and $v_{lon,i}$ the current velocity. Analysing the INTERACTION dataset at the complex intersection shown in Figure 10 reveals that there is a correlation between the current velocity and the acceleration of traffic participants. This is utilised by conditionally sampling from the joint PDF of velocity and acceleration, derived for each section of relevant lanelets, to get point samples for accelerations based on the current $v_{lon,i}$. These samples are approximated by a Gaussian distribution by calculating the mean and variance of the acquired acceleration samples, which leads to

$$a_{lon,i} = \mathcal{N}(a_{mean_{lon,i}}, \sigma_{lon,i}^2).$$
(27)

Now, for both directions, longitudinal and lateral, a distribution for predicting the future motion of the respective traffic participant along the prediction horizon is in place. The lateral prediction is parametrised by setting $\alpha = 0.66$. This was also used in [59] for the conducted parameter study and lead to meaningful predictions. The long-term mean u is set to equal the mean of all relevant trajectories in the dataset for the specific global route. In Figure 12, this is equal to the dashed red line. The original centre line would be the dashed yellow line derived from the respective means of the lanelet boundaries. It can be observed that directly using an already adapted centre lane as the basis for defining the lateral prediction process is much more realistic. The limiting variance $\sigma_{y_{lat}}^2$ is defined so that the generated trajectories do not exceed the respective lanelets. These parameters have been verified to lead to feasible kinematic predictions using [68].

Observing Figure 12 shows that the lateral prediction converges towards the adapted centre line with ongoing prediction time. For each value, the respective point samples for this specific prediction timestep are plotted in the time horizon colour bar of Figure 12. As the prediction progresses, the variance of the longitudinal prediction grows more significant, as the longitudinal position is primarily influenced by potential braking manoeuvres caused by other traffic participants. While this is not accounted for in the motion model itself, to a certain degree, this is considered by the growing variance in the prediction, as these situations are part of the dataset

TABLE 4. Overall accuracy of the scene prediction for the specific example shown in Figure 10.

Sigma interval	Miss Rate on 3-second prediction
1σ	34 %
2σ	8.5 %
3σ	2.6 %

samples used for generating the longitudinal prediction. The variance of the lateral prediction also grows more prominent, as modelled with the Ornstein-Uhlenbeck process. However, it is outgrown by the longitudinal prediction variance. This makes sense, as the probability that the traffic participant stays inside its respective lane (and deviates to the calculated mean) is much more probable.

The generated probability distributions must reflect realistic traffic participant behaviour to serve as valuable scene predictions. To rate the overall scene prediction, a scene prediction for a prediction horizon of 3 seconds is performed for each object trajectory which is part of the displayed first section of the red lanelet in Figure 10. The generated probability distributions are then compared to the positions of all trajectories in the dataset following the same global route. The 1σ, 2σ and 3σ intervals are constructed based on the positions after 3 seconds. The resulting overall accuracy of the scene prediction based on the specific example shown in Figure 10 is displayed in Tab. 4. The accuracy for the 3σ interval is already quite good, considering it already includes both the manoeuvre prediction at the red lanelet and the following route prediction. The accuracy could be improved further if the trajectories in the dataset are filtered to exclude traffic participants performing a braking manoeuvre since this is not explicitly accounted for in the motion model as this would require an interaction-aware model.

3) RISK CALCULATION

Calculating a criticality measure for every scene in a given scenario, each for a pre-defined prediction horizon, requires a metric that can be calculated for arbitrary time intervals, also in the absence of evident object collisions. As the preceding scene prediction outputs respective position probability distributions, using a probability-based risk calculation is consistent. The probability distributions of objects in the scene reflect the growing uncertainty moving along the prediction horizon and can be accounted for directly in a probabilistic risk prediction. Eggert [69] use an exponential transform in combination with a survival function to estimate future event probabilities. The focus is on collision risk, which is accounted for by modelling a distance-dependent risk. Other types of risks could be modelled, e.g. the risk of a vehicle losing control in a curve. However, the focus is now on modelling the risk of traffic collisions and potential close calls.

The future collision event probability is derived by combining a survival probability, designed as a survival function that decays in proportion to the collision event

probabilities. Earlier events significantly influence the predictive risk calculations as potential risks that occur before others are considered more important. The distance-dependent risk directly influences the collision event probability. More detailed explanations are in [70] and [71], respectively.

Next, the concrete implementation of the risk calculation is explained. At first, the distance-dependent risk is defined as

$$r_{indi}(t_i) = \frac{1}{2}\left(\mathrm{erf}\left(\frac{\mathrm{threshold} - d_{adapted}}{\sqrt{2\sigma_i^2}}\right) + 1\right), \quad (28)$$

with $r_{indi} \in [0, 1]$ and the threshold calculated as

$$\mathrm{threshold} = \begin{cases} 0.25 + 0.002\, v_{rel}^2 & v_{rel} > 0 \\ 0.25 & v_{rel} \leq 0. \end{cases} \quad (29)$$

The relative velocity between traffic objects v_{rel} is calculated as a relative velocity vector in the direction of the distance vector between both points of the forward edge of the objects as described in [72]. The threshold is increased if v_{rel} is greater than zero. If $d_{adapted}$ falls beneath the threshold, the indicated risk approaches 1. In cases where the threshold is still higher than $d_{adapted}$, the risk is increased by a higher variance σ_i^2, which is used as a parameter to quantify the uncertainty in the distance estimate $d_{adapted}$. Higher uncertainty leads to higher risk estimates in cases above the threshold. A detailed explanation of how to analytically compute $d_{adapted}$ and σ_i^2 follows at the end of this section.

Next, r_{indi} is used to calculate the event probability

$$p_{event}(t_i) = \left(\frac{1 - e^{-r_{indi}(t_i)}}{1 - e^{-1}}\right), \quad (30)$$

with $p_{event} \in [0, 1]$. Furthermore, the survival probability is calculated as follows:

$$p_{survival}(t_i) = e^{-\int_0^{t_i} p_{event}(t_i)dt}, \quad (31)$$

with $p_{survival} \in [0, 1]$. The actual collision probability for timestep t_i in the prediction horizon is calculated as

$$p_{coll}(t_i) = p_{survival}(t_i)\, p_{event}(t_i), \quad (32)$$

with $p_{coll} \in [0, 1]$. The overall collision probability for a scenario is then determined using the maximum value over the prediction horizon across all individual scenes.

Observing Figure 11 already indicates that using the euclidian distance between the means of the probability distributions is not sufficient for $d_{adapted}$, as choosing σ_i^2 would still be an open question. The probability distribution of the distance between two bivariate Gaussian probability distributions can be solved analytically. First, the problem needs to be reformulated. This is achieved by transforming these two distributions into a single one, by applying a coordinate transform so that the mean of one distribution becomes the coordinate origin. The final covariance matrix Σ_d is obtained by applying uncertainty propagation as follows:

$$\Sigma_d = J\, \Sigma_{1,2}\, J^T, \quad J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (33)$$

with $\Sigma_{1,2}$ being the matrix of the combined individual covariance matrices defined as

$$\Sigma_{1,2} = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}. \tag{34}$$

Now the problem is reduced to finding the distribution of the distance vector from the coordinate origin to the individual points sampled from the combined bivariate Gaussian distribution. This problem, however, has been addressed in other domains before. In case the bivariate Gaussian is circularly-symmetric, the actual distance distribution is described by the Rice distribution [73]. However, since using a Rice distribution in our case would mean a restriction, as it was already shown in Sec. III-D2 that a circularly-symmetric assumption for the individual Gaussians would be unwanted simplification. However, the Beckmann distribution describes the distribution for the general case, with no assumptions [74]. From the Beckmann distribution, various others, including the Rice distribution can be derived.

Next, a simple example is constructed to showcase the different distributions and their effect on the distance estimate, also concerning the needed values, mean and variance of the distance distribution, for the distance-dependent risk in Eq. 28. The position of the first traffic object, in a global coordinate system, is described by a bivariate Gaussian distribution of a random vector $X_1 = (X, Y)^T \in \mathbb{R}^2$ with

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \tag{35}$$

with μ_1 and Σ_1 being defined as

$$\mu_1 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 0.35 & 0.3 \\ 0.3 & 1 \end{bmatrix}, \tag{36}$$

and similar for the second object with the random vector $X_2 = (X, Y)^T \in \mathbb{R}^2$ with

$$X_2 \sim \mathcal{N}(\mu_2, \Sigma_2), \tag{37}$$

with μ_2 and Σ_2 being expressed as

$$\mu_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 3 & -3.5 \\ -3.5 & 5 \end{bmatrix}. \tag{38}$$

Three different methods are investigated to construct the resulting distance vector distribution. The already mentioned Rice and Beckmann distributions together with the Monte Carlo approach. Starting with the latter, samples from both individual Gaussians are drawn and the euclidean distance between these points is calculated and stored. Performing this calculation for 10k samples each leads to the distribution depicted in Figure 13 as a dashed blue line. Next, the Rice and Beckmann distributions are calculated using R, a free software environment for statistical computing [75] with an additional package [76] for calculating the specific distributions. In the case of the Rice distribution it can be observed that, while the mean deviates not too much from the Monte Carlo approach, the variance is much greater. The Beckmann distribution is solved by numerical integration, which leads to small deviations compared to the Monte Carlo simulation

TABLE 5. Three different methods are compared for computing the distance vector distribution for bivariate Gaussian distributions.

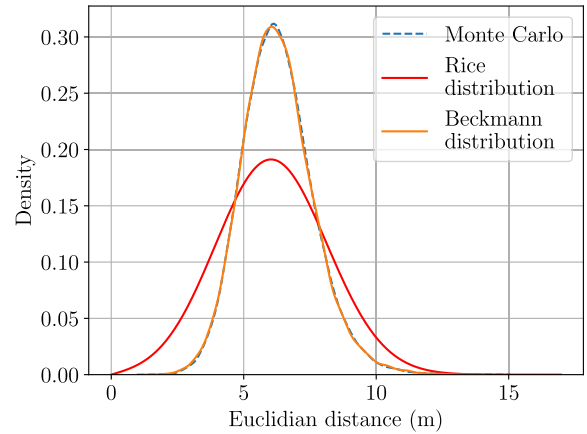| Type of method	Simulation time (s)	Mean (m)	Std (m)
Monte Carlo	4.4223	6.2957	1.3445
Rice distribution	0.0014	6.0907	2.0621
Beckmann distribution	0.0296	6.2833	1.3544

FIGURE 13. The PDFs of the euclidian distance between two traffic participants (d_E in Figure 11) assuming a Rice distribution (red line) and a Beckmann distribution (orange line) compared to the Monte Carlo evaluation (dashed blue line).

around the mean. However, overall, the computed Beckmann distribution is very close to the actual solution (derived using Monte Carlo simulation). The individual simulation time, the mean and the standard deviation for all methods can be seen in Tab.5.

To calculate the actual $d_{adapted}$ for Eq. 28, the shape of the traffic participants and uncertainty regarding the orientation of the object needs to be considered. Based on the objects' current orientation, $\pm 3°$ of angular deviation is considered, for which an enveloping ellipse is constructed taking the resulting object shapes into account. These ellipses act as a safety envelope, similar to the concepts presented in [77] and [78]. Next, the shortest distance between these two resulting ellipses is determined by sampling, using uniform distributed ellipse angles as shown in Figure 14. To account for the already discussed fact that the distance vector between those two objects, using bivariate Gaussians, actually follows a parameter distribution, the mean of the chosen distribution (e.g. the presented Beckmann distribution) $d_{mean,dist}$ is substracted from the simple euclidian distance $d_{euclidian}$ between the objects to account for the shifted mean and added to the shortest distance between the objects at each positional mean $d_{short,ellipses}$, concretely

$$d_{adapted} = d_{short,ellipses} + (d_{mean,dist} - d_{euclidian}). \tag{39}$$

Doing that, the final distance value includes the probabilistic approach regarding the objects' position as well as the object's global heading and also the actual dimensions of each respective object. The variance σ_i^2 for Eq. 28 is directly derived from the calculated distance vector distribution.
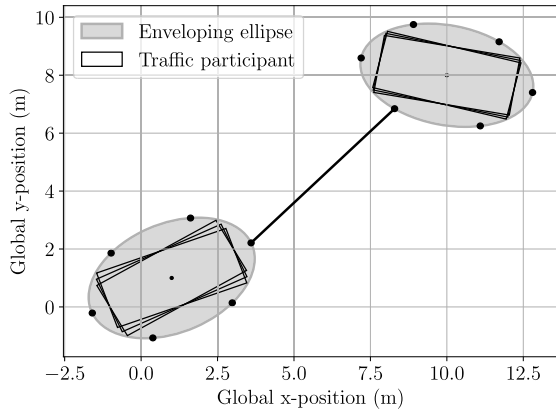
FIGURE 14. Calculating the shortest distance between two ellipses by sampling using uniformly distributed ellipse angles.
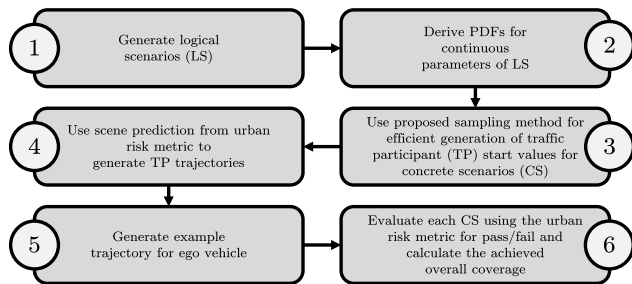

FIGURE 15. These six steps were taken in the execution of the application example.

IV. RESULTS

This section is structured into three main parts. First, the steps that have been taken to construct an application example of the coverage process are explained. Secondly, the produced results are presented in detail. Lastly, a detailed evaluation of the urban risk metric is included.

A. APPLICATION OF THE COVERAGE PROCESS

Two methods that advance the main parts of the coverage process (see Figure 2) have been presented. First, the creation of CS' by applying the proposed sampling method using an adapted *k*-means is tackled. Additionally, a respective metric for the evaluation of the created scenarios is presented. It is based on metrics already existing in the literature and has been further developed for use in specific urban areas and called the urban risk metric. To apply these two contributions to the coverage process, various steps have been taken, which can be seen in Figure 15 and are discussed next.

Step 1: In the first step, the LS need to be designed. For this application example, no LS from any scenario databases are used, but self-constructed ones. The LS is located at the exact complex intersection pictured in Figure 10. One traffic participant is approaching from the top left (from a bird's view) and is either turning right or going straight at the road fork and turning right one road fork later. This is equivalent to the manoeuvre decision in Figure 10. The other traffic object is meant to be representing the ego vehicle, which tries to turn
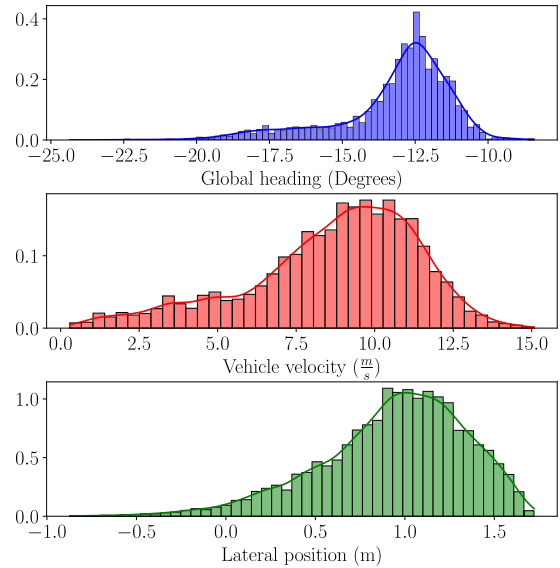

FIGURE 16. These are the generated parameter distributions for the application example. They are extracted for each chosen vehicle feature for the first lanelet section displayed in Figure 10.

**TABLE 6.** The parameters of the LS used in the application example.

Parameter	Range	Distribution
ego start position (x, y)	(992, 970)	fixed value
ego target position (x, y)	(1022, 982)	fixed value
ego start velocity $(\frac{m}{s})$	7	fixed value
TP start velocity $(\frac{m}{s})$	[0.5, 15]	from Figure 16 (middle)
TP start lateral pos. (m)	[-0.85, 1.75]	from Figure 16 (bottom)
TP start heading $(°)$	[-25, -8]	from Figure 16 (top)

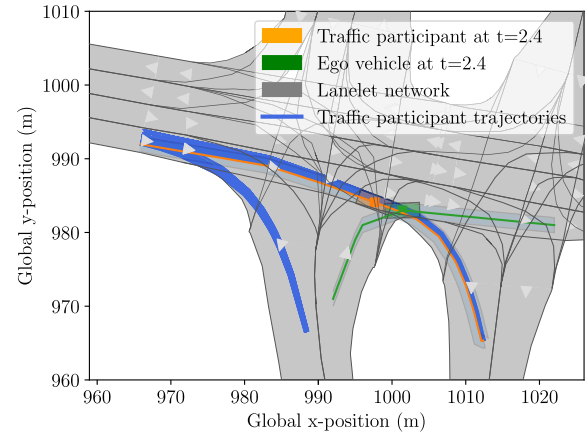right at the intersection. It is represented as the green vehicle in Figure 10.

Step 2: The CPs for this LS define the vehicle configuration of the traffic participant at the start of the scenario. Therefore the CPs are defined to be the vehicle's lateral position, the global heading and the velocity. The longitudinal starting position is set to be at the beginning of the lanelet section depicted in Figure 10. Based on the available data in the INTERACTION dataset, the values for these three CPs for all available vehicle trajectories in the dataset in this specific lanelet section are extracted. Using that, the respective PDFs, for each CP individually, assuming that these parameters are independent, are derived. They can be observed in Figure 16.

Step 3: With the individual PDFs for each CP of the LS in place, the proposed sampling method from Sec. III-C1 is used to efficiently choose values from the CP PDFs to generate the actual CS'. Concretely, the Alg.1 is applied to the parameter distributions in Figure 16. In the case of the vehicles' global velocity distribution, the generated test case values are shown in Figure 17 as individual black dots. Furthermore, the boundaries of the resulting coverage areas (value under the parameter distribution) for each test case value, are displayed. All parameters that define the LS and are used to construct the respective CS are shown in Tab. 6.
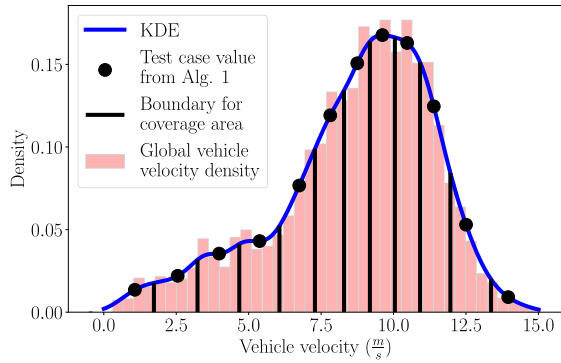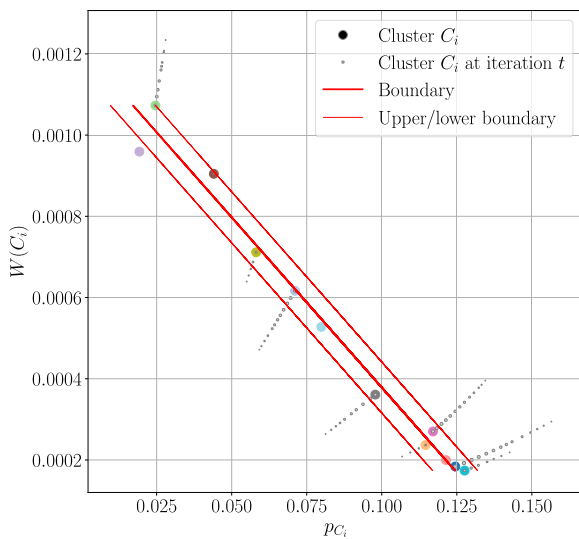
FIGURE 17. The generated test case values using Alg. 1 for the extracted parameter distribution of the global vehicle velocity in the first lanelet section displayed in Figure 10.

FIGURE 18. The Alg. 1 is applied to the parameter distribution for the global vehicle velocity (shown in the middle of Figure 16) to generate the required test case values. These values are shown in Figure 17. The iterative steps of the algorithm for each generated cluster using the proposed adapted *k*-means are shown here.

Step 4: Now, the scene prediction presented in Sec. III-D2 is used to generate the actual trajectories of the traffic participant for each CS. Another option would have been to choose the closest matching trajectory available in the dataset for the traffic participant. However, since the scene prediction has been shown to produce meaningful results, it was also applied for trajectory generation in this application example.

Step 5: For the ego vehicle, one trajectory was generated that resembles a typical turn manoeuver at the intersection. Here, the underlying assumption is that the behaviour of the ego vehicles ADS is always the same for each CS. Note that steps 4 and 5 tackle the scenario execution (step 4+ of the high-level ODD coverage process, see Figure 2) in a simplified manner. Since the aim of this application example is to show the proposed methods from Sec. III in an over-all example coverage process, this is sufficient but will be expanded in future publications.

FIGURE 19. The generated TP trajectories can be seen in blue. All TP's start at the beginning of the red lanelet depicted in Figure 10 and either turn right at the first or second possibility. The generated ego vehicle trajectory can be seen in green. For a specific pair, consisting of a TP trajectory and the trajectory of the ego vehicle, the time and object positions of the collision are depicted.

Step 6: As the last step, the CS' are evaluated. Since each CS consists of a pair of one TP trajectory and the ego vehicle trajectory, these two object trajectories are used as input to the urban risk metric. Based on the output of the risk metric, each CS is either categorised as passed or failed. This information is then used to compute the overall coverage value c_{ODD}.

V. DISCUSSION

All of the generated TP trajectories can be observed in Figure 19. For a specific pair, consisting of a TP trajectory and the trajectory of the ego vehicle, the time and object positions of the collision are depicted. Approx. 30% of the generated trajectories lead to an actual collision, with many near-misses.

For evaluating each CS, three different types of metrics are applied. First, the urban risk metric with the Rice distribution is used for the distance vector between the objects' positional distribution. Second, the urban risk metric with the Beckmann distribution and, as a third method, the well-known TTC calculation, using the improved calculation method of [45]. For each metric, the TPR and the TNR using Eq. 17 and Eq. 18 is calculated and shown in Tab. 7 for all metrics, whereas the positive case is referred to an actual collision, and vice versa. This can be done since this is an a-posteriori analysis of trajectory pairs. It can be observed that the urban risk metric (for both implemented cases) can observe nearly every collision. The risk assessment part of the urban risk metric is parametrised so that for every scenario where the calculated risk p_{coll} exceeds a value of 0.7 for at least one future scene evolvement, the scenario is flagged as failed. Based on how the actual risk is calculated, using an event-based distance-dependent risk, even in actual collision cases, the calculated risk is slightly below 0.7. This, however, can further be tuned based on the threshold parameter in Eq. 29. However, in any case, each CS will be additionally collision-checked with the actual trajectories.

**TABLE 7.** Three types of risk metrics are used to evaluate the generated CS' in the application example.

Type of risk metric	True positive rate	True negative rate
Urban risk metric (Rice)	0.96	0.011
Urban risk metric (Beckmann)	0.96	0.61
TTC calculation	1.0	0.0

It can be observed that the urban risk metric with the implemented Beckmann distribution provides the best tradeoff of sensitivity, with a high TPR, but also reasonable specificity with flagging near-miss cases as being critical. In the case of the urban risk metric with the Rice distribution, however, the TNR is severely reduced. This is a direct consequence of the greater variance that this distribution assumption has, which overstates the distance-depending risk in Eq. 28. Note that such a metric does not need to achieve a TNR of 1 since this would mean that no other than actual collision scenarios are flagged as critical. However, a simple collision check of both trajectories is enough to achieve the same. In an application with an actual ADS, it is much more essential to unveil potentially critical evolvements of an already executed scenario (since this is a-posteriori analysis) which can be further evaluated. The urban risk metric provides precisely that, as it generates realistic scene predictions, trained using actual traffic data, next to the reliability for the specific geolocation the metric is applied at. Using the urban risk metric with the Beckmann distribution for the overall ODD coverage evaluation leads to an achieved coverage value of 55.96%.

For the TTC calculation, the situation is different. As displayed in Tab. 7, the TTC metric categorised every scenario as critical. For the TTC metric to flag a scenario as critical, the TTC value needs to equal zero for at least one scene evolution. Consequently, this means that using the TTC metric for evaluating CS' in complex urban intersections turns out to be worthless, as no new information can be gathered by applying this metric. A fact discussed as well in [47].

Next, the validity of the urban risk metric (with the Beckmann distribution) and the TTC calculation is evaluated. For that, all executed CS' with collisions are analysed. For these scenarios, the actual time to collision is known in this a-posteriori analysis. Then, for each metric, the time offset is calculated from the actual time to collision in the first time step of the scenario. The distribution of the offset (in seconds) from the actual time to collision is shown in Figure 20. The urban risk metric has a smaller offset and less variance than the TTC calculation. This shows the urban risk metric's greater validity than the TTC calculation, as it already has quite a low error in predicting the collision time, even at the first scenario timestep, which has the largest prediction time until the actual collision.

Now, a specific CS with a collision is analysed in Figure 19, with the TP trajectory in orange and the ego vehicle displayed in blue. For this specific trajectory pair, the time (in seconds) which is left until a predicted collision is shown for the urban risk metric, the calculated TTC and the actual time until the collision is added as well (see Figure 21). It shows that for the
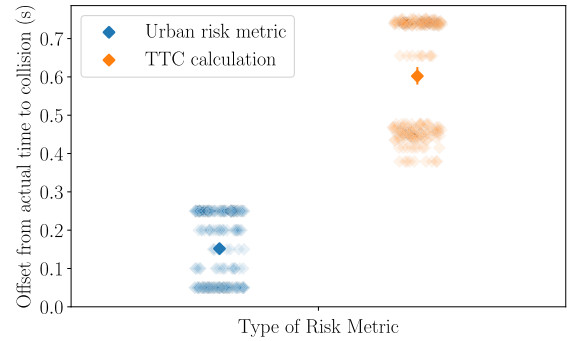
FIGURE 20. The distribution of the offset (in seconds) from the actual time to collision, for all generated scenarios with a collision, is shown for the TTC calculation and the urban risk metric. The latter has a smaller offset and also less variance.
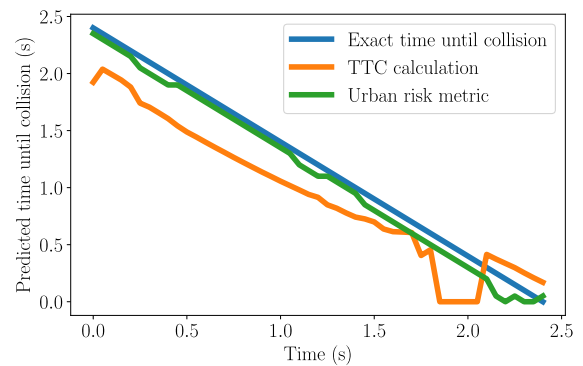
FIGURE 21. For a specific pair, consisting of a TP trajectory and the trajectory of the ego vehicle, also depicted as an example in Figure 19, the time (in seconds) which is left until a predicted collision is shown. The urban risk metric is more reliable in its predictions than the TTC calculation across the whole time horizon.

first timestep, the prediction of the TTC is worse, and also, until the actual collision, the urban risk metric proves to be more reliable.

These results show that using a manoeuvre-based motion model as part of the scene prediction used in the urban risk metric is suitable since the traffic participant has the right of way in this example. Otherwise, the motion model needs to be enhanced towards interaction awareness to provide reasonable estimates. Using the scene prediction from Sec. III-D2 for the generation of TP trajectories also, in turn, betters the performance of the urban risk metric, as the identical scene prediction is used to calculate potential future evolvements for each scene. However, the TTC calculation also partly benefits from this since if the generated TP trajectories would include interaction-aware breaking or accelerating, the performance of this metric would also be greatly deterred. In addition, the application example enabled an analysis of an urban risk metric based on the properties of validity, sensitivity and specificity. Additionally, analysing the derived PDFs from the INTERACTION dataset for the specific lanelet section (see Figure 10) shows a strong correlation between the global heading angle of the vehicle and its lateral position.

This can be utilised by applying Alg. 2 to the two-dimensional PDF of these two features to generate test cases.

VI. CONCLUSION AND OUTLOOK

This work proposes multiple extensions for a pre-existing high-level coverage process for the safety validation of ADS. The contributions are across a particular part of the coverage process, detailed in the right part of Figure 2, and enable more traceable and efficient safety argumentations for ADS'. Concretely, these contributions are (the first number of each contribution refers to the respective step of the right process in Figure 2):

- *Contribution 1.1*: A n-dimensional sampling method for the scenario discretisation process is presented, based on *k*-means clustering. It enables the definition of relevancy-dependent variance boundaries for the individual test case values.
- *Contribution 3.1*: A risk metric for the use in complex urban intersections is extended based on current literature. A significantly reduced specificity compared to time-to-collision is shown by using an exact mathematical formulation for the distribution of the distance vector between the positional distributions of traffic objects. Additionally, using actual traffic data from the respective geolocation increased the validity of the metric.
- *Contribution 4.1*: The coverage process is advanced by the individual contributions and the application example in Sec. IV.
- *Contribution 4.2*: Different levels of ODD coverage are introduced corresponding to the scenario ontology (LS, CS and CP).
- *Contribution 4.3*: The ODD coverage is thoroughly quantified across all scenario levels, assuming distribution functions' availability for the scenario parameters.

In summation, a transparent process to evaluate the achieved coverage for the target ODD, independent of the ADS implementation, is needed. Such a process is extended in multiple ways in this article, which in turn enables further extensions. Potential future work includes:

- The proposed coverage calculation can also be extended to include a metric to individually weigh each executed CS, e.g., taking into account the experienced criticality. Also, other types of metrics on the scenario level are possible, e.g., if the focus of the coverage evaluation needs to be steered towards assessing the perception layer.
- The coverage evaluation assigns every CS a certain value that contributes to the overall coverage. That information can be useful when deciding the specific test method for the respective CS, as CS, which contributes strongly to the overall coverage, need to be executed with a test method of high validity. Considering that most test methods have certain limitations (e.g., virtual testing regarding weather and the respective effects), this can be utilised when deciding the test method for the CS'. Furthermore, as the impact on the overall achievable

coverage can be stated explicitly for each CS, test method efforts can be guided more efficiently. This article does not cover this topic, but the proposed coverage calculation provides a good starting point for future research.

- Since the proposed sampling method for concrete CS generation is variance-bounded based on the individual coverage contribution, the number of necessary test cases for each CP can be calculated inversely given the respective boundary condition. That way, the boundaries can be adjusted for each CP individually, and the required amount of test cases (and their concrete values) can be automatically generated, enabling more automation in the coverage process.
- Currently, the proposed generation of CS using the adapted *k*-mean sampling is deterministic, as, for each generated cluster, the centroid is chosen as the actual test case value. However, every other rule of determining the actual values based on the generated clusters is possible. Suppose the generated CS should not be deterministic, e.g., because the whole process is repeated multiple times during an ADS development cycle. The actual test case value could again be a drawn sample from the respective cluster.
- The urban risk metric can also be extended in multiple ways. This concerns primarily scene predictions as this determines how valid the actual risk estimates are. Currently, the proposed scene prediction depends on the availability of enough traffic data to train the individual manoeuvre prediction classifiers. Also, for route prediction, this data is needed. However, the scene prediction could also be carried out with more generally applicable driver models which would lead to increased scalability.
- For this article, no concrete ADS was used in the application example, as the coverage calculation, the proposed sampling method and the urban risk metric can be exemplarily shown without a concrete ADS implementation. However, using a concrete ADS offers specific other usages. Concretely, the sensor setup requirements for a specific ADS with a pre-defined target ODD can be recursively determined using the urban risk metric in combination with a value on the necessary coverage. The urban risk metric can be applied to determine if the ADS detected particular other traffic participants that could lead to dangerous situations eventually.
- The proposed safety argumentation for AVs could also be extended towards other means of transportation where automated systems are deployed (for example, automated drones or ships). Possible adaptions to apply the safety argumentation proposed in this article in these domains can be explored in future research.

REFERENCES

[1] *Global Status Report on Road Safety 2018: Summary*, World Health Organization, Geneva, Switzerland, 2018. [Online]. Available: http://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf?ua=1

[2] A. Poddey, T. Brade, J. E. Stellet, and W. Branz, "On the validation of complex systems operating in open contexts," 2019, *arXiv:1902.10517*.

[3] *SAE J 3016 Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, SAE International, On-Road Automated Driving (ORAD) Committee, Warrendale, PA, USA, Apr. 2021.

[4] (Aug. 2020). *PAS 1883:2020*. [Online]. Available: https://www.bsigroup.com/en-GB/CAV/pas-1883/

[5] SaFAD. (2019). *Safety First for Automated Driving*. [Online]. Available: https://connectedautomateddriving.eu/mediaroom/framework-for-safe-automated-driving-systems/

[6] M. S. Blumenthal, L. Fraade-Blanar, R. Best, and J. L. Irwin, "Safe enough: Approaches to assessing acceptable safety for automated vehicles," RAND Corp., Santa Monica, CA, USA, Tech. Rep. RR-A569-1, Oct. 2020. [Online]. Available: https://www.rand.org/pubs/research_reports/RRA569-1.html

[7] F. Batsch, S. Kanarachos, M. Cheah, R. Ponticelli, and M. Blundell, "A taxonomy of validation strategies to ensure the safe operation of highly automated vehicles," *J. Intell. Transp. Syst.*, vol. 26, no. 1, pp. 1–20, Mar. 2020. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/15472450.2020.1738231

[8] A. Corso, R. Moss, M. Koren, R. Lee, and M. Kochenderfer, "A survey of algorithms for black-box safety validation of cyber-physical systems," *J. Artif. Intell. Res.*, vol. 72, Oct. 2021. [Online]. Available: https://jair.org/index.php/jair/article/view/12716

[9] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Canaria, Spain, Sep. 2015, pp. 982–988. [Online]. Available: http://ieeexplore.ieee.org/document/7313256/

[10] T. Menzel, G. Bagschik, L. Isensee, A. Schomburg, and M. Maurer, "From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment," 2019, *arXiv:1905.03989*.

[11] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9090897/

[12] T. Brade, B. Kramer, and C. Neurohr, "Paradigms in scenario-based testing for automated driving," in *Proc. Int. Symp. Elect., Electron. Inf. Eng.*, Feb. 2021, pp. 108–114. [Online]. Available: https://dl.acm.org/doi/10.1145/3459104.3459208

[13] C. Neurohr, L. Westhofen, T. Henning, T. de Graaff, E. Möhlmann, and E. Böde, "Fundamental considerations around scenario-based testing for automated driving," 2020, *arXiv:2005.04045*.

[14] Jama and Sakura. *Automated Driving Safety Evaluation Framework Ver. 1.0 -Guidelines for Safety Evaluation of Automated Driving Technology*. Accessed: Feb. 4, 2023. [Online]. Available: https://www.jama.or.jp/english/reports/docs/Automated_Driving_Safety_Evaluation_Framework_Ver2.0.pdf

[15] ASAM, "ASAM test specification study group report 2022," ASAM e.V., Hoehenkirchen, Germany, Tech. Rep., Version 1.0.0, Feb. 2022. [Online]. Available: https://report.asam.net/wp-content/uploads/2022/02/ASAM-Test-Specification-Study-Group-Report-2022_V1-0-0.pdf

[16] United Nations Economic Commission for Europe, "New assessment/test method for automated driving (NATM)," Econ. Social Council, Geneva, Switzerland, Submitted by the Working Party on Automated/Autonomous and Connected Vehicles Tech. Rep. ECE/TRANS/WP.29/2021/61, 2021. [Online]. Available: https://unece.org/sites/default/files/2021-04/ECE-TRANS-WP29-2021-61e.pdf

[17] B. Ciuffo, K. Mattas, and M. C. Galassi, "Safety assurance of automated driving systems. Raising the level of ambition," in *Proc. Eur. Commission Joint Res. Center*, Nov. 2020, pp. 1–10.

[18] R. Donà, B. Ciuffo, A. Tsakalidis, L. Di Cesare, C. Sollima, M. Sangiorgi, and M. C. Galassi, "Recent advancements in automated vehicle certification: How the experience from the nuclear sector contributed to making them a reality," *Energies*, vol. 15, no. 20, p. 7704, Oct. 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/20/7704

[19] *Building Safe Autonomous Systems—Handbook for the Verification and Validation (V&V) of Autonomous Systems*, Applied Intuition-Accelerating Safety, Mountain View, CA, USA, Jul. 2022.

[20] *Certification of Highly Automated Vehicles for Use on UK Roads—Creating An Industry-Wide Framework for Safety*, FIVE AI, Bristol, U.K., 2019.

[21] C. Miethaner and J.-E. Stavesand, "Virtual homologation of an ALKS according to UNECE R157," TÜV SÜD & dSPACE, Westendstr, Munich, Germany, White paper, 2022. Accessed: Feb. 4, 2023. [Online]. Available: https://www.tuvsud.com/en-gb/-/media/global/pdf-files/whitepaper-report-e-books/tuvsud_virtual-homologation-of-an-alks-according-to-unece-r157.pdf

[22] H. Hungar, "A concept of scenario space exploration with criticality coverage guarantees, extended abstract," in *Leveraging Applications of Formal Methods, Verification and Validation: Applications* (Lecture Notes in Computer Science), vol. 12478. Rhodes, Greece: Springer, Oct. 2020, ch. 19, pp. 293–306. [Online]. Available: http://link.springer.com/10.1007/978-3-030-61467-6_19

[23] P. Junietz, "Microscopic and macroscopic risk metrics for the safety validation of automated driving," Ph.D. thesis, Dept. Mech. Eng., Technische Universität Darmstadt, Darmstadt, 2019. [Online]. Available: https://tuprints.ulb.tu-darmstadt.de/9282/

[24] P. Junietz, U. Steininger, and H. Winner, "Macroscopic safety requirements for highly automated driving," *Transp. Res. Record, J. Transp. Res. Board*, vol. 2673, no. 3, pp. 1–10, Feb. 2019.

[25] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of test cases for automated driving systems using Bayesian optimization," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1961–1967. [Online]. Available: https://ieeexplore.ieee.org/document/8917103/

[26] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, "Systems approach to creating test scenarios for automated driving systems," *Rel. Eng. Syst. Saf.*, vol. 215, Nov. 2021, Art. no. 107610. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0951832021001551

[27] P. Weissensteiner, G. Stettinger, S. Genser, and D. Watzenig, "Operational design domain coverage for the safety validation of automated driving systems," in *Proc. Driving Simul.* Strasbourg, France: Driving Simulation Association, 2022.

[28] X. Zhang, J. Tao, K. Tan, M. Törngren, J. M. G. Sánchez, M. R. Ramli, X. Tao, M. Gyllenhammar, F. Wotawa, N. Mohan, M. Nica, and H. Felbinger, "Finding critical scenarios for automated driving systems: A systematic literature review," 2021, *arXiv:2110.08664*.

[29] T. Ken Mori, X. Liang, L. Elster, and S. Peters, "The inadequacy of discrete scenarios in assessing deep neural networks," *IEEE Access*, vol. 10, pp. 118236–118242, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9943537/

[30] (2010). *Design of Experiments (DoE)*, Technical Statistics. [Online]. Available: https://assets.bosch.com/media/global/bosch_group/purchasing_and_logistics/information_for_business_partners/downloads/quality_docs/general_regulations/bosch_publications/booklet-no11-design-of-experiments-doe_EN.pdf

[31] "Scenario-based verification and validation of self-driving vehicles: Relevant safety metrics," Siemens Digit. Industries Softw., Munich, Germany, White Paper 84312-D4 2/22 A, 2022.

[32] *Safety Pool—Powered by Deepen AI and WMG University of Warwick*. Accessed: Feb. 4, 2023. [Online]. Available: https://www.safetypool.ai/

[33] N. Weber, D. Frerichs, and U. Eberle, *A Simulation-Based, Statistical Approach for the Derivation of Concrete Scenarios for the Release of Highly Automated Driving Functions* (Automotive Meets Electronics), no. 11. German: Dortmund, Jan. 2020.

[34] Y. Akagi, R. Kato, S. Kitajima, J. Antona-Makoshi, and N. Uchida, "A risk-index based sampling method to generate scenarios for the evaluation of automated driving vehicle safety," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 667–672. [Online]. Available: https://ieeexplore.ieee.org/document/8917311/

[35] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2003, pp. 1–8.

[36] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1427769/

[37] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math., Stat., Prob.*, 1967, pp. 281–297.

[38] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[39] D. Arthur and S. Vassilvitskii, "K-Means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, Jan. 2007, pp. 1027–1035.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2012. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[41] M. D. McKay, W. J. Conover, and R. J. Beckman, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, May 1979.

[42] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas, "Criticality analysis for the verification and validation of automated vehicles," *IEEE Access*, vol. 9, pp. 18016–18041, 2021.

[43] *ISO 26262-1:2018–Road Vehicles–Functional Safety*, International Organization for Standardization, Geneva, Switzerland, 2018. [Online]. Available: https://cdn.standards.iteh.ai/samples/68383/4e26ddadc54a4198bed652afe29669fa/ISO-26262-1-2018.pdf

[44] J. C. Hayward, "Near-miss determination through use of a scale of danger," Highway Res. Rec., Pennsylvania Transp. Traffic Saf. Center, Pennsylvania, PA, USA, Tech. Rep., 1972, no. 384.

[45] F. Jiménez, J. E. Naranjo, and F. García, "An improved method to calculate the time-to-collision of two vehicles," *Int. J. Intell. Syst.*, vol. 11, no. 1, pp. 34–42, 2013. [Online]. Available: http://link.springer.com/10.1007/s13177-012-0054-4

[46] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," *Transp. Res. Rec.*, vol. 667, pp. pp. 67–74, 1978.

[47] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Arch. Comput. Methods Eng.*, vol. 30, no. 1, pp. 1–35, Aug. 2022, doi: 10.1007/s11831-022-09788-7.

[48] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Occlusion-aware risk assessment for autonomous driving in urban environments," 2018, *arXiv:1809.04629*.

[49] D. Wang, W. Fu, Q. Song, and J. Zhou, "Potential risk assessment for safe driving of autonomous vehicles under occluded vision," *Sci. Rep.*, vol. 12, no. 1, p. 4981, Mar. 2022. [Online]. Available: https://www.nature.com/articles/s41598-022-08810-z

[50] P. Weissensteiner, G. Stettinger, K. Tieber, and K. Rehrl, "Virtual risk assessment for the deployment of autonomous shuttles," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2675, no. 11, p. 10, Jul. 2021.

[51] P. Schneider, M. Butz, C. Heinzemann, J. Oehlerking, and M. Woehrle, "Towards threat metric evaluation in complex urban scenarios," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1192–1198. [Online]. Available: https://ieeexplore.ieee.org/document/9564790/

[52] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," 2019, *arXiv:1911.07602*.

[53] S. Wagner, K. Groh, T. Kühbeck, M. Dörfel, and A. Knol, "Using time-to-react based on naturalistic traffic object behavior for scenario-based risk assessment of automated driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1521–1528. [Online]. Available: https://ieeexplore.ieee.org/document/8500624/

[54] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Dec. 2014. [Online]. Available: https://robomechjournal.springeropen.com/articles/10.1186/s40648-014-0001-z

[55] K. Kreutz and J. Eggert, "Analysis of the generalized intelligent driver model (GIDM) for uncontrolled intersections," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3223–3230. [Online]. Available: https://ieeexplore.ieee.org/document/9564423/

[56] K. Mattas, G. Albano, R. Donà, M. C. Galassi, R. Suarez-Bertoa, S. Vass, and B. Ciuffo, "Driver models for the definition of safety requirements of automated vehicles in international regulations. Application to motorway driving conditions," *Accident Anal. Prevention*, vol. 174, Sep. 2022, Art. no. 106743. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0001457522001798

[57] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2008, pp. 3731–3736. [Online]. Available: http://ieeexplore.ieee.org/document/4543783/

[58] J. Quehl, H. Hu, S. Wirges, and M. Lauer, "An approach to vehicle trajectory prediction using automatically generated traffic maps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 544–549.

[59] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2751–2766, Oct. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7412746/

[60] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9690–9699. [Online]. Available: https://ieeexplore.ieee.org/document/9709630/

[61] D. Petrich, T. Dang, D. Kasper, G. Breuel, and C. Stiller, "Map-based long term motion prediction for vehicles in traffic environments," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2166–2172. [Online]. Available: http://ieeexplore.ieee.org/document/6728549/

[62] S. Biswas, M. Bianchini, R. N. Shaw, and A. Ghosh, "Prediction of traffic movement for autonomous vehicles," in *Machine Learning for Robotics Applications*, vol. 960. Singapore: Springer, 2021, pp. 153–168. [Online]. Available: https://link.springer.com/10.1007/978-981-16-0598-7_12

[63] V. P. Thazha, D. Filliat, and J. Ibañez-Guzmán, "Applying map-masks to trajectory prediction for interacting traffic-agents," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Dec. 2019, pp. 1–7.

[64] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION dataset: An INTERnational, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.

[65] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 1672–1679. [Online]. Available: http://www.mrt.kit.edu/z/publ/download/2018/Poggenhans2018Lanelet2.pdf

[66] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[67] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory, Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2001.

[68] C. Pek, V. Rusinov, S. Manzinger, M. C. Uste, and M. Althoff, "CommonRoad drivability checker: Simplifying the development and validation of motion planning algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1013–1020. [Online]. Available: https://ieeexplore.ieee.org/document/9304544/

[69] J. Eggert, "Risk estimation for driving support and behavior planning in intelligent vehicles," *Automatisierungstechnik*, vol. 66, no. 2, pp. 119–131, Feb. 2018. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/auto-2017-0132/html

[70] J. Eggert and T. Puphal, "Continuous risk measures for ADAS and AD," in *Proc. Int. Symp. Future Active Safety Technol. Towards Zero-Traffic-Accidents (FAST-Zero)*, 2017, pp. 1–8.

[71] J. Eggert, "Predictive risk estimation for intelligent ADAS functions," in *Proc. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 711–718. [Online]. Available: http://ieeexplore.ieee.org/document/6957773/

[72] E. K. Antonsson, "A general measure of collision hazard in traffic," 2022, *arXiv:2205.08640*.

[73] S. O. Rice, "Mathematical analysis of random noise," *Bell Syst. Tech. J.*, vol. 24, no. 1, pp. 46–156, 1945.

[74] P. Beckmann, "Statistical distribution of the amplitude and phase of a multiply scattered filed," *J. Res. Nat. Bureau Standards*, vol. 66D, pp. 231–240, May 1962.

[75] *R: A Language and Environment for Statistical Computing*, R Core Team, Vienna, Austria, 2022. [Online]. Available: https://www.R-project.org/

[76] D. Wollschlaeger. (2022). *ShotGroups: Analyze Shot Group Data*. R Package Version 0.8.2. [Online]. Available: https://cran.r-project.org/web/packages/shotGroups/index.html

[77] J. Bernhard, P. Hart, A. Sahu, C. Schöller, and M. Guzman Cancimance, "Risk-based safety envelopes for autonomous vehicles under perception uncertainty," 2021, *arXiv:2107.09918*.

[78] J. Bernhard and A. Knoll, "Risk-constrained interactive safety under behavior uncertainty for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 63–70. [Online]. Available: https://ieeexplore.ieee.org/document/9575956/

SIDDARTHA KHASTGIR (Member, IEEE) received the dual B.Tech. and M.Tech. degrees in mechanical engineering from the Indian Institute of Technology (IIT) Kharagpur, India, in 2011, and the Ph.D. degree from The University of Warwick, U.K., in 2019. He is currently the Head of Verification and Validation, Intelligent Vehicles with WMG, The University of Warwick. His research interests include trust in automation, system safety, and verification and validation of autonomous vehicles. He is a member of the Institution of Mechanical Engineers (IMechE), U.K. He has been named in the Forbes 30 under 30 Europe 2018—industry list and was awarded the prestigious UKRI Future Leaders Fellowship, in 2019. He is also a Chartered Engineer.

PATRICK WEISSENSTEINER received the B.Sc. and M.Sc. degrees in mechanical engineering and business economics from the Graz University of Technology, Graz, Austria, in 2015 and 2017, respectively.

He is currently working as a Senior Researcher with the Automotive Electronics and Software Department, Virtual Vehicle Research GmbH, Graz. His current research interest includes safety validation of automated vehicles, with a particular focus toward coverage methods for the respective operational design domain of such vehicles.

GEORG STETTINGER received the B.Sc. and M.Sc. degrees in electrical engineering from the Graz University of Technology (TUG), Graz, Austria, in 2009 and 2011, respectively, and the Ph.D. degree in information technology from the University of Klagenfurt (AAU), Klagenfurt, Austria, in 2015. From 2015 to 2018, he was a Senior Researcher with the Co-Simulation and Software Group, Virtual Vehicle Research GmbH, Graz. From 2018 to 2022, he led the Control Systems Team, Virtual Vehicle Research GmbH. He is currently a Senior Project Manager with the Research and Development Funding Department, Infineon Technologies AG, Munich, Germany. His current research interests include ODD-based testing and validation, particularly certification and homologation of automated vehicles. He is an active member of the Technical Committee of the Driving Simulation Association with a special focus on virtual verification and validation of ADAS/AD systems.

DANIEL WATZENIG (Senior Member, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in technical science from the Graz University of Technology, Graz, Austria, in 2002 and 2006, respectively, the Habilitation (Venia Docendi) degree in electrical measurement and signal processing, in 2009. Since 2006, he has been the Divisional Director and the Scientific Head of the Automotive Electronics and Software Department, Virtual Vehicle Research GmbH, Graz. In 2017, he was appointed as a Full Professor of autonomous driving with the Institute of Automation and Control, Graz University of Technology. He is the Founder and a Team Leader of the Autonomous Racing Graz Team, one of currently six teams of the global race series Roborace. He is the author or coauthor of more than 200 peer-reviewed papers, book chapters, patents, and articles. His research interests include sense and control of automated vehicles, signal processing, multi-sensor data fusion, uncertainty estimation and quantification, and robust optimization.

• • •