IEEE Open Journal of
Intelligent Transportation
Systems

Received 28 February 2021; revised 13 April 2021; accepted 3 May 2021. Date of publication 6 May 2021; date of current version 17 May 2021.

Digital Object Identifier 10.1109/OJITS.2021.3078121

Online Verification Enabling Approval of Driving Functions—Implementation for a Planner of an Autonomous Race Vehicle

TIM STAHL (Member, IEEE), AND FRANK DIERMEYER

Chair of Automotive Technology, Technical University of Munich, 80333 Munich, Germany

CORRESPONDING AUTHOR: T. STAHL (e-mail: tim.stahl@tum.de)

ABSTRACT Safety guarantees and regulatory approval for autonomous vehicles remain an ongoing challenge. In particular, software that is frequently adapted or contains complex, non-transparent components, such as artificial intelligence, is exceeding the limits of safety standards. This paper presents a detailed implementation of an online verification module – the Supervisor – that copes with these challenges. The presented implementation focuses on autonomous race vehicles without loss of generality. Following an identified holistic list of safety-relevant requirements for a trajectory, metrics are developed to monitor whether the trajectory can safely be executed. To evaluate safety with respect to dynamic objects in a semi-structured and highly dynamic racing environment, rule-based reachable sets are presented. As a result, the pure reachable set is further constrained by applicable regulations. Real-time capability and effectiveness are demonstrated in fault-injected scenario-based tests and on real-world run data. The implemented Supervisor will be publicly available on GitHub.

INDEX TERMS Autonomous vehicles, formal verification, runtime environment, software safety, vehicle safety.

I. INTRODUCTION

EVERY year, road traffic accidents claim approximately 1.35 million lives around the world, according to the World Health Organization (WHO) [1]. Passive safety mechanisms and the market introduction of advanced driver assistance systems (ADAS) have led to a reduction in traffic deaths [2]. Highly automated driving aims to reduce the traffic death rate even further, if not guarantee absolute safety.

Since the beginning of this century, the field of autonomous driving has experienced an increase in interest. Accordingly, a large number of methods and approaches has been developed and evaluated [3]. The field of trajectory planning alone includes numerous search-based, optimization-based and even artificial intelligence (AI)-based approaches [4], [5], [6]. However, more complex software (SW) always goes hand in hand with increasing difficulties in safeguarding and approval. Existing safeguarding procedures and standards cannot yet map such complex systems, especially when online learning methods are involved. As a consequence, new approval and safeguarding strategies have to be developed [7], [8]. One promising concept, making it possible to cope with complex and AI-based algorithms, is online verification (OV) [8], [9].

In this paper, we present a holistic OV framework for trajectories of autonomous vehicles (AVs) (Fig. 1). The implementation focuses on safety guarantees in a semi-structured environment – i.e., a race track without lane markings – and high dynamic capabilities of the agents in the scene. The OV framework is independent of the underlying trajectory planner and can therefore be used to safeguard various planning approaches, ranging from classic to AI-based algorithms. The contributions of this work can be summarized as follows:

- Structured and holistic (targeting all aspects) OV framework for trajectory planning modules.
- Approach capable of handling a semi-structured environment (e.g., wide track with no lane allocation) by introduction of rule-based reachable sets.

The review of this article was arranged by Associate Editor A. H. Al-Kaff.

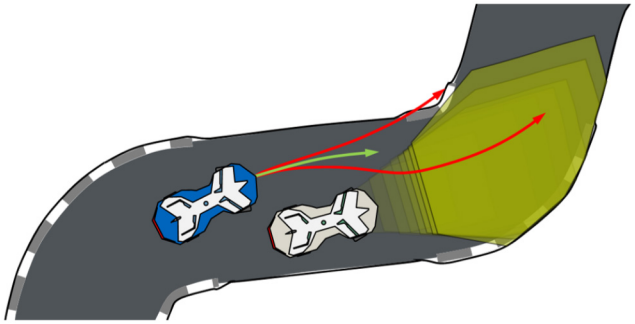


FIGURE 1. Illustrative sketch. A gray vehicle with its reachable set and a blue vehicle with three trajectory candidates. Unsafe trajectories (red) are rejected by the proposed OV framework.

- Method does not build upon moderate dynamics and is validated with accelerations up to $12\,\mathrm{m\,s^{-2}}$.
- Structured validation including fault-injected simulative and real-world autonomous race vehicle runs.

We provide an OV framework that can be used to tackle approval issues when facing complex, frequently changing or AI-based algorithms. Furthermore, the method can be used to pinpoint safety issues during the development of a trajectory planner. Compared to related work, the presented framework aims – as introduced in [9] – for OV of a holistic list of properties essential for a safe execution. Besides apparent aspects of a safe trajectory, such as no collisions with track boundaries or other traffic participants, the dynamic properties or safe end states must also be examined. Other than in related work, the list of monitored entities is elaborated in a structured way, reducing the risk of neglecting individual aspects.

Without loss of generality, we show an illustrative implementation for autonomous race vehicles. These are particularly suitable because they can be evaluated in a closed environment without putting humans at risk, and at the same time be tested at dynamic limits of handling and high speeds. Existing OV approaches often rely heavily on a structured environment. For example, public roads are divided into lanes and allow assumptions to be made about other vehicles, such as that they will not leave their lanes or will change to the adjacent lane only (and not beyond). In this work, we present an approach that copes with a semi-structured environment, i.e., a race-track. Here, the agents can choose their path freely and do not have to adhere to lanes. Furthermore, vehicles with high acceleration capabilities result in large reachable sets within a short period of time. By integrating applicable regulations in reachable sets for each agent, we obtain rule-based reachable sets. In doing so, we do allow for interactive situations in semi-structured environments, even at high acceleration capabilities. The presented framework is evaluated in simulative scenarios as well as on real-world race vehicle runs. Adapting the rules allows the approach to be adapted to road traffic – especially for highways due to their similar structure.

The remainder of this paper is structured as follows. Section II covers related work in the field. Preliminaries on reachable sets and the underlying OV concept are outlined in Section III. The developed OV method with all its subcomponents is outlined in Section IV, details on the assessment metrics are given in Section V. Section VI presents evaluation results from simulation and real-world runs. A discussion and concluding remarks are provided in Section VII.

II. RELATED WORK

In this section, first of all, relevant standards for safeguarding and release of commercial AVs are outlined and their limitations highlighted (Section II-A). Subsequently, relevant work in the field of online monitoring is presented and the research gap is pointed out (Section II-B).

A. STANDARDS

Since the introduction of electrical and/or electronic (E/E) systems in commercial vehicles, safeguarding and approval have become more challenging because of the vulnerable nature of these systems. In order to guarantee a minimum level of safety, a homologation – a set of legal requirements to be verified in tests – must be passed before new vehicles are introduced to the market. However, since these tests can only provide spot checks, a comprehensive safety assessment must be carried out by the original equipment manufacturer (OEM) within the scope of product liability. This task is also the key concern of this work, in comparison to the more political undertaking of homologation. In the automotive industry, standards have been established to guarantee safe development and validation. Safety is often interpreted as three separate fields, each of which is covered by a dedicated standard[1]: Safety of the Intended Functionality (SOTIF) (ISO/WD PAS 21448 [10]), Functional Safety (FuSa) (ISO 26262 [11]) and (Cyber)Security (ISO/SAE AWI 21434 [12]).

In this paper, the focus is on the software aspect within FuSa and SOTIF. The basics of each of the standards are summarized in the next two paragraphs, followed by limitations regarding the latest development in the domain of AVs.

The ISO 26262 (FuSa) is designed in line with the V-model development principle [13], and therefore supports the whole development phase including market release and operation in the light of FuSa. A core principle of the standard is the principle of defining the risk of a system. The automotive safety integrity level (ASIL) – ranging from A to D – serves as risk label for a system component, deduced from system risk and identified failure modes. Each level comes with specified safety requirements to be fulfilled during development.

The ISO 21448 (SOTIF) strives to prove that a target function is sufficiently safe. This process is not only defined for

1. The listed standards are most common in European countries, other countries may focus more on other standards, e.g., the UL 4600.

the final product, but also targets the specification, development, verification and validation phase. In order prove this, the key idea of the standard is an interplay of the four following fields: known safe scenarios (to be maximized during development), known uncertain scenarios (to be mastered), unknown safe scenarios (no further measures), and unknown uncertain scenarios (to be uncovered and moved to any of the other classes).

With autonomous driving functions becoming more complex and harder to explain by human experts [14] – especially when building upon AI-based algorithms – the current standards reach their limits [15], [16], [17], [18]. Two exemplary issues when the ISO 26262 (FuSa) is applied to projects including such algorithms are:

- *Function Specification not Possible for Training-Based Algorithms:* The behavior of a learning-based method cannot be specified by the algorithm alone, since it is substantially shaped by the provided training data. This data inevitably hosts gaps and it is not possible to provide a proof of dependable failure prevention [8], [15].
- *Modular and Transparent Properties are Lost:* The ISO 26262 requests a modular, manageable and transparent SW structure. AI-based functions soon become impossible for human experts to explain and leave the frame of the standard [8], [19], [20].

Currently, the ISO/WD PAS 21448 (SOTIF) is rather vague in contrast to ISO 26262, not least due to its relatively young age and ongoing development, and does not provide detailed instructions. However, it can already be considered for AVs (with possibly additional required measures). Thus, the standard is hard to implement due to a lack of details, but does not present any explicit problems in its application.

In conclusion, the ISO 26262 reaches it limits when facing complex or non-transparent SW stacks. As a consequence, new safeguarding techniques have to be developed. Related work proposes several approaches to tackle safeguarding or approval of AV SW. While offline methods (e.g., formal offline approval) cannot cope with continuing learning during runtime, online monitoring methods are considered a promising approach [8], [21]. Parallel online monitoring (also known as doer/checker principle) goes well with the principles of ASIL-decomposition in the current version of the ISO 26262 [11], [22]. The following section reviews existing work in the field of online monitoring.

B. ONLINE MONITORING

Within the domain of online monitoring, we identified two clusters of approaches. One cluster focuses on online risk assessment, primarily with stochastic methods, while the other performs OV with formal methods. The following paragraphs present exemplary publications in each field and close with findings as well as gaps in research.

Online risk assessment methods commonly aim to calculate a collision probability. Various approaches have been evolved. One approach is to determine the most probable maneuver and set it into relation with the planned motion of the ego-vehicle [23], [24], [25]. Another group of authors uses Monte Carlo simulation [26], [27] or Markov Chains [28], in order to evaluate not only a single most probable motion. However, all approaches in this cluster have in common that they do rely on stochastic techniques that hinder approval, in line with safety standards introduced in Section II-A. Due to this fact, the cluster of OV – elaborated in the following – seems more promising.

In the cluster of OV, there are a number of approaches verifying processes with modal logics online. Xu *et al.* [29] use spatial logic in order to safeguard motion primitives at a roundabout. Also, traffic rules have been transcribed in temporal logics and evaluated online. In this regard, Esterle *et al.* [30] applied Linear Temporal Logic (LTL) and Maierhofer *et al.* [31] used Metric Temporal Logic (MTL). This group of methods focuses mainly on the verification of state transitions and traffic rules, but lacks a holistic view. Consequently, safety approval does not fall within the scope of these approaches.

Pek *et al.* [32] developed a drivability checker that verifies trajectories within moderate calculation times. However, the main target is the verification of trajectories with perfect knowledge about the motion of other vehicles.

Shalev-Shwartz *et al.* [33] use a formal description of worst-case maneuvers in lateral and longitudinal directions to detect unsafe behavior. In order to do so, the ego-vehicle must at least keep a distance from other vehicles great enough to cope with sudden brake/steer maneuvers while incorporating the ego-vehicle's acceleration potential and reaction time (calculating a "proper response"). While this comprehensive approach covers several situations in regular traffic, the calculated response is not guaranteed to be feasible for every road geometry. Furthermore, this approach focuses on dynamic collision mitigation and misses the holistic view required for approval.

A similar underlying idea is used by reachable sets, which calculate sets of states that a vehicle can reach in a certain time interval. Several authors [34], [35], [36], [37], [38], [39] use this approach to formally state whether planned actions of the ego-vehicle are collision-free. In doing so, the points in time of the planned ego movement are checked for overlaps with a corresponding time interval in the reachable set of other objects. While most of the approaches focus only on the aspect of dynamic collisions, missing the holistic view required for safety approval, the concept of reachable sets seems promising for a formal safety approval. Basic principles of the concept are revised in Section III. Individual approaches, and here Pek *et al.* [37] should be emphasized, cover further aspects such as dynamic and kinematic restrictions of the vehicle. Nevertheless, all of the approaches are missing a structured elaboration of a holistic list of features needed for safety. Furthermore, the basic reachable sets shown in the approach strongly limit the ego-vehicle's operating range – especially when targeting a

racing application with a semi-structured environment and high acceleration capabilities.

The OV approach presented in this paper builds on formal methods – including some of those presented in this section. In contrast to related work, we combine both, a detailed implementation of the approach and the consideration of the suitability for approval in the lights of applicable standards. Existing approaches are extended to cope with semi-structured environments and a high acceleration range. In this regard, applicable regulations are represented in the reachable sets. The presented approach is evaluated on critical simulative scenarios and real-world runs of a full-scale race AV.

III. PRELIMINARIES

Following the notation of Pek *et al.* [32], we define the state space $\mathcal{X} \subset \mathbb{R}^n$ as the set of possible states ξ and $\mathcal{U} \subset \mathbb{R}^m$ as the set of admissible control inputs u for the ego-vehicle. The motion of a vehicle is described by the differential equation

$$\dot{\xi}(t) = f(\xi(t), u(t)), \tag{1}$$

with $u([t_0, t_h])$ describing an input trajectory covering the time interval $[t_0, t_h]$, $t_0 < t_h$ and t_0 being the initial time. The solution of (1) at time $t \in [t_0, t_h]$ with initial state $\xi(t_0) = \xi_0$ and input trajectory $u([t_0, t_h])$ is denoted by $\chi(t, \xi(t_0), u([t_0, t_h]))$. In line with Pek *et al.* [32], we assume that state trajectories $\xi([t_0, t_h])$ of the ego-vehicle are provided to our OV module.

The race track is defined by a left $\mathcal{B}_\mathrm{l}$ and right boundary $\mathcal{B}_\mathrm{r}$, each represented by a polyline. Every point p on the track can be either described by a Cartesian coordinate $\langle x, y\rangle$ or within the Frenet-frame $\langle s, n\rangle$ – a lane-based coordinate system in which s describes the arc length along the reference line of the track and n describes the lateral offset at that point of the reference line along the normal vector. The track coordinate for a given state ξ is expressed with the notation $\cdot(\xi)$, e.g., $n(\xi)$ for the lateral coordinate. Furthermore, we denote the lateral offset of the left and right boundary to the reference line at a coordinate s with $n_{\mathcal{B}_\mathrm{l}}(s)$ and $n_{\mathcal{B}_\mathrm{r}}(s)$, respectively.

The set of states $\mathcal{X}$ occupied by a vehicle at a certain state ξ is indicated by the operator $\mathcal{E}(\xi) : \mathcal{X} \to \mathrm{P}(\mathbb{R}^2)$, where $\mathrm{P}(\mathbb{R}^2)$ denotes the power set of $\mathbb{R}^2$. The occupation operator for a set $\mathcal{X}$ is defined as $\mathcal{E}(\mathcal{X}) := \{\mathcal{E}(\xi) | \xi \in \mathcal{X}\}$.

Following Althoff *et al.* [40], the set of states a vehicle can reach is called reachable set $\mathcal{R} \subseteq \mathcal{X}$. Starting from an initial set of states $\mathcal{R}_0$, the reachable set $\mathcal{R}^e$ is defined as the set of states that can be reached by executing any possible action,

$$u(t) = \begin{bmatrix} u_1(t) & u_2(t) \end{bmatrix}^T, \forall t : u(t) \in \mathcal{U},$$

where for an AV, u_1 describes normalized steering and u_2 normalized acceleration:

$$\mathcal{R}^e(t_f, \xi_0, \mathcal{U}) := \big\{ \chi(t_f, \xi_0, u(\cdot)) | \xi_0 \in \mathcal{R}_0, \forall \tau \in [0, t_f] : u(\tau) \in \mathcal{U} \big\}.$$

Since it is not possible to determine the exact reachable set $\mathcal{R}^e$ [41], a spatial over-approximation is commonly used:

$$\mathcal{R}(t_f, \xi_0, \mathcal{U}) \supseteq \mathcal{R}^e(t_f, \xi_0, \mathcal{U}).$$

All points occupied by the reachable states can be accessed with the previously defined occupation operator $\mathcal{E}(\mathcal{R}^e(t_f, \xi_0, \mathcal{U}))$. Althoff and Magdici [42] provide a computation-efficient method for the over-approximated occupation of the reachable set $\mathcal{E}(\mathcal{R}(t_f, \xi_0, \mathcal{U}))$, which is, for convenience, referenced as $\mathcal{R}^{\mathcal{E}}(t_f, \xi_0, \mathcal{U})$ in the course of this work.

In order to establish relations between two sets $\mathcal{X}_1$ and $\mathcal{X}_2$, the following set operations are used: $\mathcal{X}_1 \cup \mathcal{X}_2$ describes the union of the sets, $\mathcal{X}_1 \cap \mathcal{X}_2$ denotes the intersection, and $\mathcal{X}_1 \setminus \mathcal{X}_2 := \{\xi \in \mathcal{X}_1 : \xi \notin \mathcal{X}_2\}$ the set difference.

Temporal relations are formulated in past time linear temporal logic (ptLTL) introduced by Havelund and Roşu [43]. Besides standard propositional operators like $\neg$ (negation), $\vee$ (disjunction), and $\wedge$ (conjunction), further operators are defined[2]: $\odot$ (previously), and $\mathcal{S}_s$ (strong since). The semantics of ptLTL specific operators are given in the following. If $t = s_1 s_2 \ldots s_m$ is a finite sequence of abstract states s_i then t_i denotes the trace $s_1 s_2 \ldots s_i$ for each $1 \leq i \leq m$. Given F, being a boolean value, variable or an arbitrary number of atomic propositions, then the semantics of the introduced operators are:

$$t \models \odot F \iff t' \models F, \text{ with } t' = \begin{cases} t_{m-1}, & \text{if } m > 1 \\ t, & \text{if } m = 1 \end{cases},$$

$$t \models F_1 \mathcal{S}_s F_2 \iff \big(\exists j \in [1\,..\,m]\ t_j \models F_2\big) \wedge (\forall i \in [j\,..\,m]\ t_i \models F_1).$$

IV. METHOD

Given a trajectory planner, which cannot be approved by conventional methods due to complexity, frequent updates or online learning, the goal is to develop new methods that enable approval. In previous work [9], we present a detailed concept targeting OV of SW components used in AV. In the following subsections, we briefly revisit the key development steps (Fig. 2) for a trajectory planner OV module – called the Supervisor – and give details of selected steps in the course of this section. Each of the following subsections covers one of the four (1-4) stages within the Supervisor (S) framework. Further details on all the individual steps and additional information can be found in the referenced publication.

A. REQUIREMENTS (S-1)

In the first stage, requirements for a safe trajectory, as well as requirements posed on the OV module itself, have to be defined, based on the superordinate safety goal. Since it is not possible to avoid all collisions [33], the goal is to avoid any collision caused by the ego-vehicle in a environment

2. List limited to the operators used in this paper, further operations in the ptLTL can be found in the referenced source.

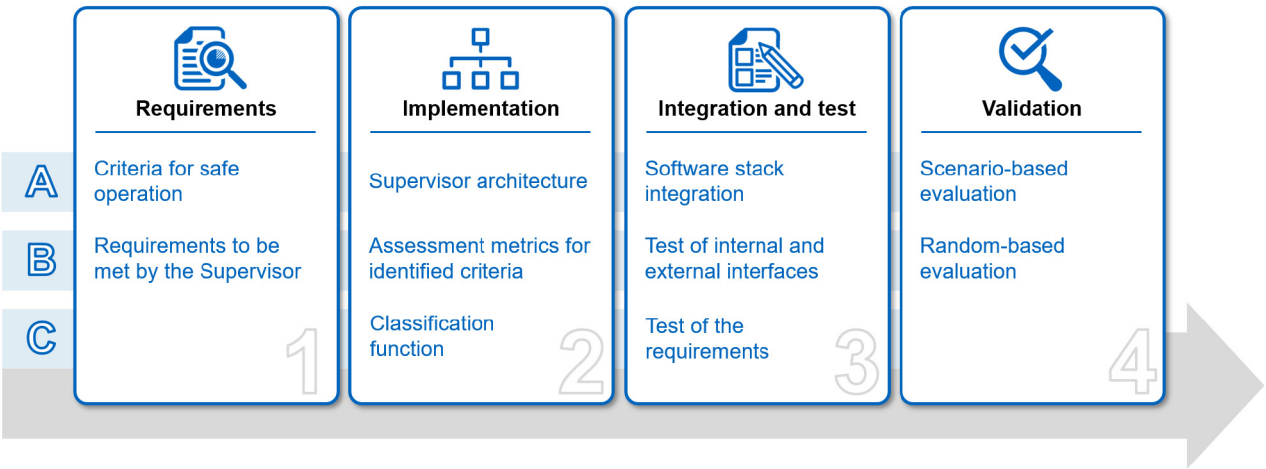


FIGURE 2. Generic method for the development of an OV module. The method is structured into four stages (1-4 – indexed from left to right), each comprising up to three steps (A, B, C – indexed from top to bottom) [9].

TABLE 1. Identified list of criteria that must hold for a safe trajectory – keys holding the identifiers for Supervisor (S), first stage (1), step (A) and an incrementing number (1 - 6) [9].

Key	Criteria
S-1-A-1	Objects in the scene must be detected and perceived properly
S-1-A-2	Physical interactions with static objects or other traffic participants initiated by the AV must not occur at any time
S-1-A-3	Trajectory must match the actual pose in the real world
S-1-A-4	Trajectory must respect the friction limits at all times
S-1-A-5	Kinematic and dynamic vehicle properties must be respected
S-1-A-6	Applicable rules of conduct (e.g. race rules) must be obeyed

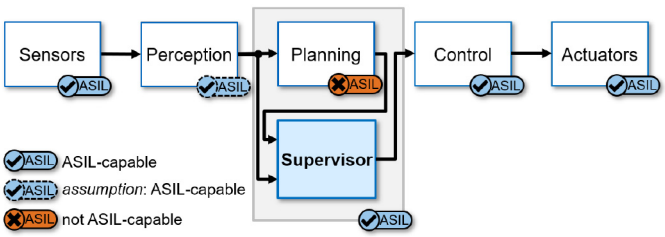


FIGURE 3. Supervisor architecture for a trajectory OV. The planning module is assumed to be not approvable by standard methods (not ASIL-capable), but with insertion of the Supervisor, the overall system can be approved.

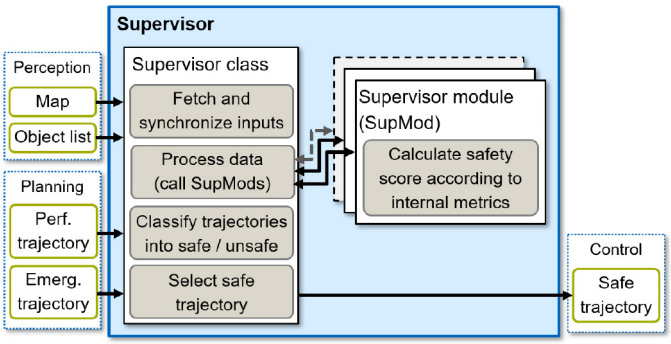


FIGURE 4. Framework of the proposed OV module. The Supervisor takes a map, object-list, performance trajectory and emergency trajectory as input. The input data is processed in Supervisor modules (SupMod) and then classified as either safe or unsafe. A safe trajectory is forwarded to the controller.

governed by rules. To this end, a structured approach should be pursued to identify a holistic list of criteria for a safe trajectory. In the aforementioned paper [9], we introduced one possible approach based on interfaces between SW modules. Resulting key criteria derived in the referenced work are listed in Table 1.

B. IMPLEMENTATION (S-2)

The second stage addresses the implementation. First, the architecture of the OV is defined, then assessment metrics for the previously specified criteria are developed and merged with a classification function. The architecture of the OV follows previous findings [9]. Accordingly, the OV runs in series to the planning module (Fig. 3) and receives environmental data (including environment map $\mathcal{M}$ and object list $\mathcal{O}(t)$), a planned performance trajectory $\xi_{\text{perf}}([t_0, t_h])$, and an emergency trajectory $\xi_{\text{em}}([t_0, t_h])$ as input. The safety rating of these trajectories is the output, which in turn regulates the routing of only safe trajectories to the controller. The interfaces as well as the key internal components and steps are visualized in Fig. 4. A core element of the Supervisor is a set of Supervisor modules (SupMods) Ψ, that each evaluate a certain aspect of the safety criteria (Table 1). Each of the trajectories (performance and emergency) is passed to a dedicated set of SupMods $\Psi_{\text{perf}} \subset \Psi$ and $\Psi_{\text{em}} \subset \Psi$. Each SupMod $\psi_i \in \Psi$ is provided with inputs and returns a boolean safety rating $s_i = \psi_i(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h]))$. Details on the safety assessment metrics in the SupMods are given in Section V. For each of the trajectories (performance and emergency – in the following formulated for the performance trajectory) the boolean safety ratings $s_{\text{perf},i}$ of the N_{perf} SupMods are fused with a conjunction over all ratings $s_{\text{perf}} = s_{\text{perf},1} \wedge s_{\text{perf},2} \wedge \cdots \wedge s_{N_{\text{perf}}}$. Based on the overall rating of both trajectories, the latest safe trajectory is forwarded to the controller (Fig. 5). It is noted that in order to switch to an emergency trajectory from a past time step, performance and emergency trajectory must

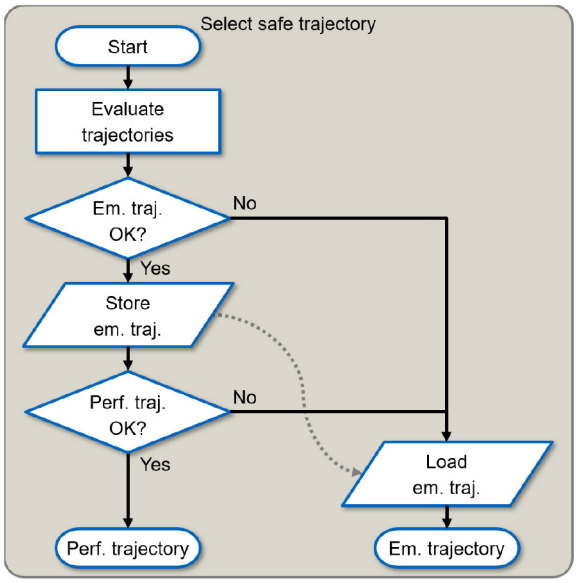


FIGURE 5. Decision metric for the selection of a safe trajectory.

be congruent for a certain time period (safety factor times average update rate).

C. INTEGRATION AND TEST (S-3)

The third stage of the development addresses the integration and verification tests against the requirements. This procedure is state of the art, and structured recommendations are given in ISO 26262-4:2018, 7 [11]. This paper only briefly addresses aspects of this procedure in Section VI.

D. VALIDATION (S-4)

The fourth stage tackles the validation of the established Supervisor module. The goal of this process is to reveal possible design and implementation flaws affecting the safety goal in the overall framework. To this end, a scenario-based and a field-test validation is recommended. Section VI provides further details on this.

V. SAFETY ASSESSMENT METRICS

In this section, the required safety metrics are derived and elaborated in the following subsections. In order to attest safe behavior, the criteria for a safe trajectory (Table 1) have to be checked with suitable metrics. In the following, we allocate measures taken for each of the criteria listed:

- *S-1-A-1 (Perception of Objects):* No further actions taken, safeguarding targets planning module, correct perception data is assumed (assured by V2X communication).
- *S-1-A-2 (Physical Interactions With Objects):* This requirement can be divided into two sub-aspects. On the one hand the trajectories have to be checked against static objects (on a race track especially track limitations, Section V-A), on the other hand against dynamic objects (implemented here with rule-based reachability, Section V-B). In addition, the emergency trajectory must be certified as valid for an infinite time horizon (e.g., if no new valid trajectory is found), therefore the trajectory is checked to ensure that it results in a safe end state (Section V-C).
- *S-1-A-3 (Localization Must be Correct):* No further actions taken, as correct perception data is assumed (*S-1-A-1*).
- *S-1-A-4 (Friction Limits Must be Respected):* The actual friction limit is assumed to be provided via the perception module, the acceleration limits given the friction potential are checked with a tire model, detailed in Section V-D.
- *S-1-A-5 (Kinematic and dynamic properties)*: Tracked with straightforward checks along the trajectory, presented in Section V-E.
- *S-1-A-6 (Rules of Conduct):* Tackled in Section V-F and partially included in Section V-B, but not the main scope of this paper. Detailed approaches targeting this aspect can be found in related work [31].

All referenced metrics are implemented as a SupMod within the Supervisor framework, each detailed in the following.

A. TRACK BOUNDARY COLLISION CHECKS

The ego-vehicle must stay within the track boundaries $\mathcal{B}_\mathrm{l}$ and $\mathcal{B}_\mathrm{r}$ at any time. With the assumption of each trajectory $\xi([t_0, t_h])$ hosting at least one point on the track, trajectories not colliding with the boundaries are formally stated as

$$\forall t \in [t_0, t_h], \forall \mathcal{B}_i \in \{\mathcal{B}_\mathrm{l}, \mathcal{B}_\mathrm{r}\} : \mathcal{E}(\xi(t)) \cap \mathcal{B}_i = \emptyset. \tag{2}$$

In order to properly assess the stated property, one has to examine every pose of the trajectory for stated intersections. Maierhofer *et al.* [31] pursued similar approaches within their road-compliance checks. However, since their procedure is computationally demanding, we pursue a simplified approach building on the trajectory alone and rejecting orientation information of the individual poses along the trajectory.

In this regard, the curve formed by the sequence of positions hosted by a trajectory $\xi([t_0, t_h])$ is inflated by a specified distance d_infl. The inflation $\mathcal{I}(\mathcal{X}) : \mathcal{X} \rightarrow \mathrm{P}(\mathbb{R}^2)$ hosts all points in the $\mathbb{R}^2$ domain that fall within the Euclidean distance d_infl to any point in the given set $\mathcal{X}$. An efficient implementation of this operation is given with the 'buffer' method of the python library 'shapely'. In order to assure absolute safety, the distance parameter d_infl chosen must be larger or equal to half of the vehicle's footprint diagonal.

The overall Boolean safety rating $\psi_\mathrm{stat}(\cdot)$ w.r.t. to the static environment (i.e., the track boundaries) calculated by this SupMod is formalized as follows:

$$\psi_\mathrm{stat}(\mathcal{M}, \xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall \mathcal{B}_i \in \{\mathcal{B}_\mathrm{l}, \mathcal{B}_\mathrm{r}\} : \mathcal{I}(\xi([t_0, t_h])) \cap \mathcal{B}_i = \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In order to achieve a gap-free detection of this property, the discrete trajectory states are represented by continuous lines in the 'shapely' library and then checked for intersections with the boundaries using the library's methods.

The underlying approach can be replaced by any other approach assuring safety. It should be noted that within the Supervisor framework only the planned trajectory is approved. Control errors are not considered in this step. However, a safety margin can be added to the distance parameter – i.e., guarantee safety within a broader corridor – and the controller choses a strategy to stay within this given tube (e.g., learning tube model predictive control).

B. DYNAMIC COLLISION CHECKS

The ego-vehicle must not collide with any object o associated with a dedicated state $\xi_o(t)$ in the object-list $\mathcal{O}(t)$ at any time. Within our application, the entries in the object list are restricted to other race vehicles, but the method holds for any entity including static obstacles. Trajectories not colliding with objects on the track are formally stated as

$$\forall t \in [t_0, t_h], \forall o \in \mathcal{O}(t) : \mathcal{E}(\xi(t)) \cap \mathcal{E}(\xi_o(t)) = \emptyset. \quad (4)$$

Since the future ($t > t_0$) behavior and, therefore, occupation $\mathcal{E}(\xi_o(t))$ of other objects in the scene is unknown, an OV concept must assume any viable behavior of other entities. A regular prediction approach is unsuitable in this case, since other objects can move differently at any time. As deduced in Section II-B, we rely on reachable sets for the reasons given.

Reachable sets – especially in the domain of race vehicles, which holds large acceleration potentials and high speeds – grow to a large occupation area in a short time. Without further measures or structural limitations such as lanes, vehicles cannot meet or even overtake each other without these sets intersecting the planned trajectories. Therefore, we propose reducing the sets to only rule-conform reachable sets. This extension limits the possible states of other vehicles and at the same time supports the overall safety goal – prevention of accidents caused by the ego-vehicle. In order to do this, the standard occupation set of reachable sets $\mathcal{R}^{\mathcal{E}}$ is further reduced based on the set of applicable rules $\mathcal{K}$,[3] i.e., race regulations in this case. Each rule $k \in \mathcal{K}$ is composed of a ptLTL triggering condition c_k and an associated mathematical formulation of a reduction set $\mathcal{Q}_k \in \mathrm{P}(\mathbb{R}^2)$. The occupation of rule-conform reachable sets $\mathcal{R}^{\mathcal{E}}_{\mathcal{K}}$ is formally described as follows:

$$\mathcal{R}^{\mathcal{E}}_{\mathcal{K}}(t_h, \xi_0, \mathcal{U}) = \mathcal{R}^{\mathcal{E}}(t_h, \xi_0, \mathcal{U}) \setminus \bigcup_{k \in \mathcal{K}_c} \mathcal{Q}_k,$$
$$\text{with } \mathcal{K}_c = \{k \in \mathcal{K} | c_k\}. \quad (5)$$

3. Applicable rules in this sense mean rules that constrain the motion of other vehicles in certain situations. Other rules, such as rules only affecting the ego-vehicle, are addressed in Section V-F.

It should be noted that, for the sake of simplicity, the formula is limited to a single time interval $[t_0, t_h]$, where the actual implementation builds on this formulation for each individual subinterval, enclosed in $[t_0, t_h]$. Furthermore, the rules are evaluated for the considered situation and projected forward in time for computational reasons. A detailed analysis would require an evaluation along the evolution of the reachable sets.

The set of triggering conditions c_k evaluates the interaction of the ego-vehicle with surrounding vehicles and the environment. ptLTL can be used to evaluate past and present relationships. In order to do so, all entities holding the time t as an argument $\cdot(t)$ can be understood as a discretized series of states, starting at the launch of the vehicle or scenario up to the current time instance. For the sake of convenience, the mathematical formulation is mostly defined in the lane based coordinate system. In this paper, we showcase the implementation of rules and regulations within the Roborace Season Alpha, as well as current Formula 1 regulations [45], [46]. Table 2 holds the rule description, the mathematical description of the trigger condition c_k and the mathematical description of the reduction set $\mathcal{Q}_k$, each in a separate column. This approach could be applied to regular road traffic when formalizing the applicable traffic rules – especially for highways by their similar nature. Imprecise and soft rules can be challenging to formalize, thus past court decisions are included in related work [31].

The overall Boolean safety rating $\psi_{\mathrm{dyn}}(\cdot)$ w.r.t. to the dynamic environment (i.e., the movable objects on the track) calculated by this SupMod is formalized as follows:

$$\psi_{\mathrm{dyn}}(\mathcal{M}, \mathcal{O}(t_0), \xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_{\mathrm{h}}], \forall o \in \mathcal{O}(t_0): \\ & \mathcal{E}(\xi(t)) \cap \mathcal{R}^{\mathcal{E}}_{\mathcal{K},o}(t, \xi_0, \mathcal{U}) = \emptyset \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

In this setting $\mathcal{R}^{\mathcal{E}}_{\mathcal{K},o}(t, \xi_0, \mathcal{U})$ indicates the occupation of the rule-based reachable set for the object o, for a given time stamp t. Within a temporally discretized reachable set, consisting of consecutive time intervals within the range $[t_0, t_{\mathrm{h}}]$, t should be interpreted to use the reachable set with the matching interval. It is important to note that the temporal resolution of the reachable set poses a trade off between calculation time and false positive alarms – the coarser the temporal resolution, the more likely false alarms will occur. Static objects on the track can be considered by providing a static reachable set hosting a single time step.

C. SAFE END STATE

Since the emergency trajectory hosts the backup strategy until a new valid set of trajectories is found (Fig. 5), the trajectory must be safe for an infinite time horizon. In order to ensure this, the emergency trajectory must host a safe end state $\xi(t_h)$ in addition to all other criteria. For the context of our application, a stationary vehicle on the track is defined as sufficiently safe. More stringent requirements can also be

TABLE 2. List of three exemplary reduction sets $\mathcal{Q}_k$ and their respective trigger condition c_k. For every object o in the scene, the active reduction sets are subtracted from the unconstrained reachable set in order to obtain a rule-based reachable set.

Description	Mathematical formulation
Roborace – overtaking regulation Season Alpha If a chase vehicle approaches a lead vehicle and its headway falls below a threshold t_{trig} within a specified trigger zone $\mathcal{Z}_{\text{T}} \in \text{P}(\mathbb{R}^2)$, an overtake is granted within a dedicated overtake zone $\mathcal{Z}_{\text{O}} \in \text{P}(\mathbb{R}^2)$.[a] Within the overtake zones paired with a valid activation in a preceding trigger zone, the lead vehicle must stay on a specified side $\mathcal{Z}_{\text{O,l}} \subset \mathcal{Z}_{\text{O}}$ or $\mathcal{Z}_{\text{O,r}} \subset \mathcal{Z}_{\text{O}}$ of the track.	c_k: $c_k = (\neg F_{\text{T}}(t) \vee F_{\text{C}}(t))\, \mathcal{S}_s\, (F_{\text{T}}(t) \wedge F_{\text{C}}(t))$, where $F_{\text{T}}(t) = \mathcal{E}(\xi(t)) \cap \mathcal{Z}_{\text{T}}$ and $F_{\text{C}}(t) = \left(\frac{s(\xi_{\text{o}}(t)) - s(\xi(t))}{v(t)} < t_{\text{trig}} \right)$ Note: Simplified representation.[b] $\mathcal{Q}_k$: $\mathcal{Q}_k = \begin{cases} \mathcal{Z}_{\text{O,l}}, & \text{if overtake specified left} \\ \mathcal{Z}_{\text{O,r}}, & \text{otherwise} \end{cases}$
Formula 1 – racing alongside another car The following regulations are considered: • "Manoeuvres liable to hinder other drivers, such as deliberate crowding of a car beyond the edge of the track or any other abnormal change of direction, are strictly prohibited" [45]-2b. • "It is not permitted to drive any car unnecessarily slowly, erratically or in a manner deemed potentially dangerous to other drivers at any time" [45]-2d. • "Any driver [...] may use the full width of the track [...], provided no significant portion of the car attempting to pass is alongside his. For the avoidance of doubt, if any part of the front wing of the car attempting to pass is alongside the rear wheel of the car in front this will be deemed to be a 'significant portion'" [46]-27.7. Hence, as soon as the footprint of two vehicles overlap along the track coordinate s by more than d_{overlap}, the other vehicle must not cut into the ego driving tube. In this case, both vehicles must always leave a space of one vehicle width w_{veh} to the edge.	c_k: $c_k = \odot\left((s(\xi_{\text{o}}(t)) - s(\xi(t))) < \left(1 - d_{\text{overlap}}\right) l_{\text{veh}}\right)$ $\mathcal{Q}_k$: $\mathcal{Q}_k = \left\{ \langle s, n \rangle \in \text{P}(\mathbb{R}^2) \,|\, \forall s \in [s(\xi(t_0)), s(\xi(t_{\text{h}}))] : n \in [l(s), u(s)] \right\}$, with $l(s) = \begin{cases} n_{\mathcal{B}_{\text{l}}}(s), & \text{if } n(\xi(t_0)) < n(\xi_o(t_0)) \\ \max\left(n_{\mathcal{B}_{\text{l}}}(s) + c(s),\ n_{\mathcal{B}_{\text{l}}}(s) + w_{\text{veh}}\right), & \text{otherwise} \end{cases}$, and $u(s) = \begin{cases} \min\left(n_{\mathcal{B}_{\text{l}}}(s) + c(s),\ n_{\mathcal{B}_{\text{r}}}(s) - w_{\text{veh}}\right), & \text{if } n(\xi(t_0)) < n(\xi_o(t_0)) \\ n_{\mathcal{B}_{\text{r}}}(s), & \text{otherwise} \end{cases}$, where $c(s) = \left(|n_{\mathcal{B}_{\text{l}}}(s)| + |n_{\mathcal{B}_{\text{r}}}(s)|\right) \frac{|n(\xi(t_0))| + |n(\xi_o(t_0))|}{2}$ Note: Simplified representation.[c]
Formula 1 – defending off-line The Formula 1 sporting regulations state that "Any driver moving back towards the racing line, having earlier defended his position off-line, should leave at least one car width between his own car and the edge of the track on the approach to the corner" [45]-2b. Hence, as soon as an object vehicle is closer than d_{corner} to a corner, while defending (distance between the two vehicles along the s-coordinate smaller than d_{defend}) off-line (within the track half preceding the curve's inside $\mathcal{Z}_{\text{I}}$), the defender must leave one vehicle width w_{veh} to the outer edge. A corner is defined as a track segment where the absolute value of the curvature of the centerline exceeds a threshold curvature κ_{thr}. Thereby, s_{corner} denotes the s-coordinate of the corner first exceeding this threshold, starting from $s(\xi_{\text{o}}(t))$.	c_k: $c_k = ((s_{\text{corner}} - s(\xi_{\text{o}}(t))) \leq d_{\text{corner}})\, \mathcal{S}_s\, (F_{\text{O}}(t) \wedge F_{\text{D}}(t))$, where $F_{\text{O}}(t) = \mathcal{E}(\xi(t)) \cap \mathcal{Z}_{\text{I}}$ and $F_{\text{D}}(t) = ((s(\xi_{\text{o}}(t)) - s(\xi(t))) \leq d_{\text{defend}})$ Note: Simplified representation.[b] $\mathcal{Q}_k$: $\mathcal{Q}_k = \left\{ \langle s, n \rangle \in \text{P}(\mathbb{R}^2) \,|\, \forall s \in [s(\xi(t_0)), s(\xi(t_{\text{h}}))] : n \in [l(s), u(s)] \right\}$, with $l(s) = \begin{cases} n_{\mathcal{B}_{\text{r}}}(s) - w_{\text{veh}}, & \text{if left turn} \\ n_{\mathcal{B}_{\text{l}}}(s) & \text{otherwise} \end{cases}$, and $u(s) = \begin{cases} n_{\mathcal{B}_{\text{r}}}(s) & \text{if left turn} \\ n_{\mathcal{B}_{\text{l}}}(s) + w_{\text{veh}}, & \text{otherwise} \end{cases}$ Note: Simplified representation.[c]

[a] This principle is similar to the DRS regulation in the Formula 1 series (Article 3.18 of the F1 Technical Regulations).
[b] The difference of s-coordinates (e.g. $s_o(t) - s(t)$) is only valid for unclosed tracks. For round courses, crossing of the start-finish-line must be tracked.
[c] A track wider than $2w_{\text{veh}}$ at any position is assumed. If this is not the case, the leading vehicle must be prioritized for the shared space.

imposed here, as they are developed in various works [47]. A trajectory $\xi([t_0, t_h])$ holding a sufficiently safe end state is formally stated as

$$\forall t \geq t_{\text{h}} : v_{\text{x}}(\xi(t)) = 0 \iff v_{\text{x}}(\xi(t_{\text{h}})) = 0, \tag{7}$$

where $v_{\text{x}}(\xi(t))$ denotes the associated longitudinal velocity of state $\xi(t)$.

The Boolean safety rating $\psi_{\text{ses}}(\cdot)$ w.r.t. to a safe end state generated by this SupMod is formalized as follows:

$$\psi_{\text{ses}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } v_{\text{x}}(\xi(t_{\text{h}})) = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

D. ACCELERATION LIMITS

The requested acceleration of a trajectory must stay within the road-tire friction potential at all times. The actual friction limit is assumed to be provided by the perception module.[4] A trajectory $\xi([t_0, t_h])$ respecting the acceleration limits is formally stated as

$$\forall t \in [t_0, t_{\text{h}}] : F_{\text{a}}(\xi(t)) \leq F_{\text{a,max}}(\xi(t)), \tag{9}$$

where $F_{\text{a}}(\xi(t))$ denotes the requested force and $F_{\text{a,max}}(\xi(t))$ the maximum possible force at state $\xi(t)$.

The force requested by a trajectory $\xi([t_0, t_h])$ at state $\xi(t)$ depends on the longitudinal acceleration $a_{\text{x}}(\xi(t))$ and lateral acceleration $a_{\text{y}}(\xi(t))$. These quantities can be determined based on the velocity $v(\xi(t))$, the longitudinal acceleration $a_{\text{x,traj}}(\xi(t))$ and the curvature $\kappa(\xi(t))$ associated with state $\xi(t)$. All listed variables can be determined via the change

4. A convenient way to guarantee safety is to underestimate the actual friction potential.

of successive states. It should be noted that the longitudinal component must be offset against the drag, since only the portion carried by the tire is of interest. Given a vehicle with mass m, the combined force F_a to be transmitted by the tire is given by applying a simple tire model,[5] the circle of forces:

$$F_\mathrm{a}(\xi(t)) = m\sqrt{a_\mathrm{x}(\xi(t))^2 + a_\mathrm{y}(\xi(t))^2} \tag{10}$$

$$= m\sqrt{\begin{matrix}\left(a_{\mathrm{x,traj}}(\xi(t)) + v_\mathrm{x}(\xi(t))^2 c_{\mathrm{d,ext}}\right)^2 \\ +(v_\mathrm{x}(\xi(t))^2\kappa(\xi(t)))^2\end{matrix}}, \tag{11}$$

with the extended drag coefficient $c_{\mathrm{d,ext}} = \frac{c_\mathrm{d}\rho A}{2m}$ being a vehicle-specific value considering reference area A, mass density of the air ρ, and the pure drag coefficient c_d.

The maximum allowed force $F_{\mathrm{a,max}}(\xi(t))$ is determined based on the friction coefficient $\mu(\xi(t))$ and the normal force F_N,

$$F_{\mathrm{a,max}}(\xi(t)) = \mu(\xi(t))F_\mathrm{N}(\xi(t)). \tag{12}$$

For the sake of brevity, the friction coefficient is assumed to be constant across the track (no state dependence). As the aero-induced downforce is not included in the normal force, a constant maximum allowed force is deduced.

The overall Boolean safety rating $\psi_{\mathrm{acc}}(\cdot)$ w.r.t. to the acceleration limits calculated by this SupMod is formalized as follows:

$$\psi_{\mathrm{acc}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_\mathrm{h}]: \\ & F_\mathrm{a}(\xi(t)) \leq F_{\mathrm{a,max}} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Although simplified measures have been taken with the basic tire model and the aero-induced downforce, it should be noted that all these measures have been chosen to keep the overall behavior on the conservative (safe) side.

E. KINEMATIC AND DYNAMIC FEASIBILITY

In order to assure drivability and thereby validity of the other safety checks (e.g., collision with other vehicles), the trajectory $\xi([t_0, t_h])$ must be inherently valid and respect a set of hardware-related limitations $\mathcal{L}$, which is formally stated as

$$\forall t \in [t_0, t_h], \forall l \in \mathcal{L} : l(\xi(t)), \tag{14}$$

where $l(\xi(t))$ holds for a limitation l that is respected in state $\xi(t)$ of the trajectory.

The limitations considered in this paper are the turn radius r_{turn} and motor acceleration limits $a_{\mathrm{lim}}(v)$. The curvature κ of the trajectory $\xi([t_0, t_h])$ is necessarily the result of the sequence of states $\xi(t)$ and must not violate the turn radius r_{turn} of the vehicle at any time, $\forall t \in [t_0, t_h] : \kappa(\xi(t)) \leq \frac{1}{r_{\mathrm{turn}}}$. While a vehicle's brakes are usually designed in such a way that tire friction is the limiting factor during negative acceleration, the engine is the limiting factor during positive acceleration and must be considered explicitly. The requested acceleration must not exceed the velocity-dependent acceleration limits, $\forall t \in [t_0, t_h] : a(\xi(t)) \leq a_{\mathrm{lim}}(v(\xi(t)))$.

The Boolean safety rating $\psi_{\mathrm{kd}}(\cdot)$ w.r.t. to stated limits generated by this SupMod is formalized as follows:

$$\psi_{\mathrm{kd}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : \kappa(\xi(t)) \leq \frac{1}{r_{\mathrm{turn}}} \\ & \quad \wedge\ a(\xi(t)) \leq a_{\mathrm{lim}}(v(\xi(t))) \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

F. RULES

In order to fulfill the stated safety goal of not causing a collision, the ego-vehicle must adhere to a set of applicable rules $\mathcal{Z}$, formally stated as:

$$\forall z \in \mathcal{Z} : z(\xi([t_0, t_h])), \tag{16}$$

where $z(\xi(t))$ holds for a rule z that is respected in state $\xi(t)$ of the trajectory. In order to evaluate whether rules hold, the rules have to be formalized with a temporal logic. Since this is itself a comprehensive field of research, this paper limits itself to a single exemplary rule in the sense of its illustrative character. Further progress in the area of rule formalization can be found in related work [31].

The rule considered here is simple in nature but no less important. In races, the race control defines a maximum speed v_{max} that must be adhered to. The corresponding formalized rule can be described as follows:

$$z_{\mathrm{vmax}}(\cdot) = \begin{cases} 1, & \text{if } \forall t \in [t_0, t_h] : v(\xi(t)) \leq v_{\mathrm{max}} \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The Boolean safety rating $\psi_{\mathrm{rule}}(\cdot)$ w.r.t. to a set of rules generated by this SupMod is formalized as follows:

$$\psi_{\mathrm{rule}}(\xi([t_0, t_h])) = \begin{cases} 1, & \text{if } \forall z \in \mathcal{Z} : z(\xi([t_0, t_h])) \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

VI. EVALUATION

The developed Supervisor has to be validated systematically to guarantee a sufficiently safe behavior. ISO 26262-4:2018, 8.4.3.4 [11] proposes a selection of measures for validation. In this course, field tests are employed, which represent a typical exposure. However, since critical situations are underrepresented in these cases, additional scenario-based tests are used, which deliberately test the limits of the Supervisor with fault injection, for instance. First (Section VI-A), fault-injected scenarios, targeting domains of individual SupMods are elaborated in hand with the general scenario-based testing strategy [48]. Second (Section VI-B), an exemplary real-world field test is presented. Finally (Section VI-C), we provide insights in implementation details. It should be noted that while the measures shown may be sufficient for a proof of concept in the context of this work, the testing effort would need to be significantly increased for actual approval. Related work deals with the required test coverage for AV function validation [49].

5. Of course, this can be replaced by any more sophisticated tire model.

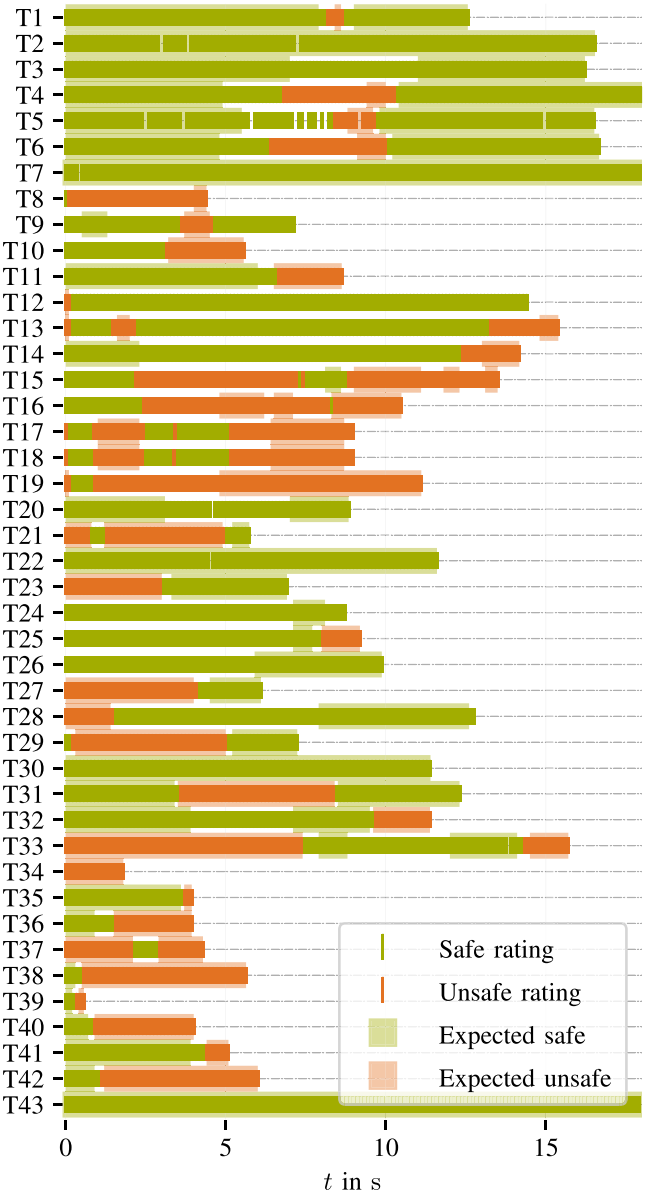


FIGURE 6. Passive Supervisor ratings for the 42 scenarios implemented (T1-T42) paired with their expected outcome. Regions where no expected outcome is stated can take any Supervisor rating, as only regions with an unambiguous outcome are labeled with expected ratings. This view is cropped to a duration of 18.0 s for better readability.

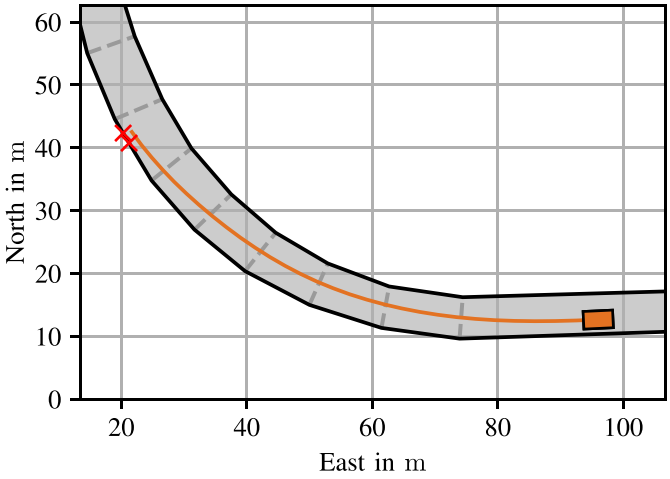


FIGURE 7. The planned trajectory of the ego-vehicle (orange line) at time-stamp 1.0 s in scenario T33 colliding with the track boundaries (black lines) is flagged as unsafe by the Supervisor. The intersection points are indicated with red crosses.

A. SCENARIO-BASED EVALUATION – NUMERICAL EXAMPLES

For the scenario-based evaluation, we showcase 42 scenarios with up to three vehicles, generated from scratch as well as based on real race tracks [50]. The scenarios are each of different durations and host various types of injected faults (e.g., collision with other vehicles). For each scenario, we define time intervals in which the rating of the Supervisor is expected to be safe, as well as regions where the rating is expected to be unsafe. Since it is not possible to state the proper rating for every point in time (e.g., only critical situation in some constellations), intervals hosting no expected outcome (any rating by the Supervisor is allowed) were included during labeling. The labeling followed fixed rules, e.g., 'unsafe' whenever a collision is unavoidable and 'safe' when no issues within a static environment are present and appropriate distance to other vehicles is kept. The ground truth regarding other vehicles was objectively generated via the Difference of Space distance and Stopping distance (DSS) [51]. Values above 10 m must be evaluated as safe, values below 0 m as unsafe. Regions with no expected rating should – in tendency – receive more safe ratings for a high availability. However, by nature these regions must also hold unsafe ratings when approaching expected unsafe regions, in order to prevent possible collisions. The implemented scenarios with their expected ratings and the actual ratings by the passive (i.e., rating, but not intervening) Supervisor are depicted in Fig. 6.

The passive Supervisor ratings for all scenarios fall within the expected ratings. In the following, snapshots of selected scenarios are elaborated in more detail in order to demonstrate the underlying evaluation metric of the implemented SupMods. Furthermore, the Supervisor is able to avoid a collision in all of the scenarios, when used in the active pipeline (i.e., Supervisor rates and intervenes). An example demonstrating this behavior is given at the end of this subsection.

In scenario T33 (Fig. 7) the ego-trajectory is planned too close to the boundaries, meaning that the vehicle footprint would collide. The Supervisor successfully detects this incident. The calculated intersection points are highlighted with red crosses.

Collisions with dynamic objects are primarily checked by rule-based reachable sets (Section V-B). In scenario T16 (Fig. 8) none of the rule-based reductions is active, since the object vehicle (blue) is in front of the ego-vehicle (orange) and on the racing line. As a consequence, the occupation of the ego-vehicle's emergency trajectory is rated against the

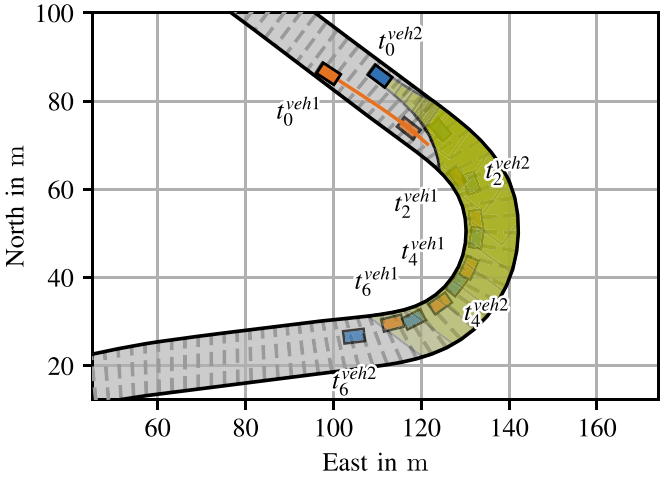


FIGURE 8. Planned emergency trajectory of the ego-vehicle (orange line) and reachable set of the object vehicle (green) at 3.2 s into scenario T16. Future vehicle poses are drawn at 1.0 s increments, while every second instance *i* is denoted by t_i^{veh1} and t_i^{veh2} for the ego and object vehicle, respectively.

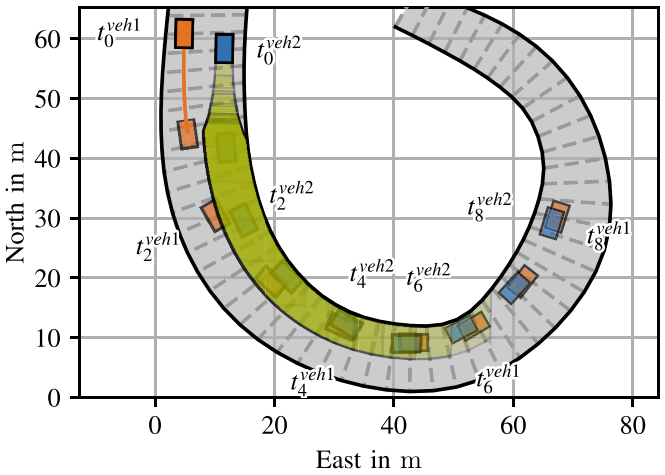


FIGURE 9. Planned emergency trajectory of the ego-vehicle (orange) and rule-based reachable set of the object vehicle (green) at 0.5 s into scenario T19. Future vehicle poses are drawn at 1.0 s increments, while every second instance *i* is denoted by t_i^{veh1} and t_i^{veh2} for the ego and object vehicle, respectively.

pure reachable set (green) of the object vehicle. For every planning-step, the reachable set is calculated in slices of 0.2 s interval. The occupation of every state in the trajectory is then checked for intersection with the corresponding reachable set interval. In this case, the ego-vehicle is moving too fast on the inside of the curve to properly handle the situation, in the case of the object vehicle deciding to immediately shear to the right. Thus, the resulting rating for the given time stamp is 'unsafe'.

In scenario T19 (Fig. 9) the 'racing alongside' rule is active, since the footprint of the two vehicles overlaps in a longitudinal direction. As a consequence, the object vehicle (blue) must not push the ego-vehicle towards the track boundaries. Correspondingly, the reachable set of the object vehicle is reduced by a corresponding spatial representation. The emergency trajectory of the ego-vehicle does not intersect the rule-based reachable set at any state. The resulting rating for the given planning step is 'safe'.

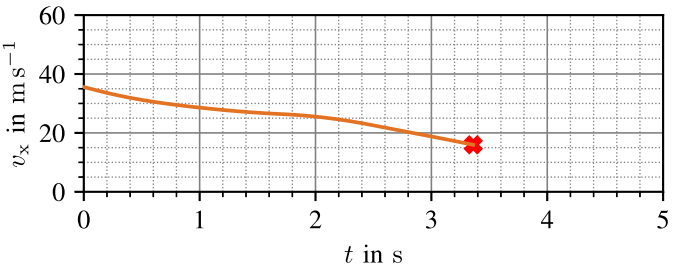


FIGURE 10. The planned emergency trajectory of the ego-vehicle (orange line) at 1.0 s into scenario T33 does not host a safe end state. The end state violating the condition is marked with a red cross.

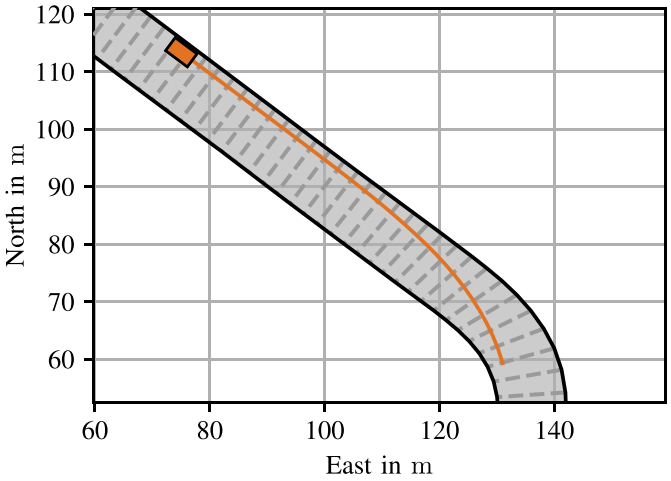
Not only does the scenario T33 discussed previously (Fig. 7) cause a collision of the performance trajectory with the track limits, but also the emergency trajectory does not host a safe end state (Fig. 10). The emergency trajectory provided causes the vehicle to brake down, but not to a standstill. Reasons for this phenomenon in a real situation could be a vehicle's high entry speed at the corner paired with too short a planning horizon, when only a moderate potential for longitudinal acceleration is left. The Supervisor rates the trajectory as 'unsafe'.

In scenario T28 (Fig. 11) the planned performance trajectory exceeds the allowed acceleration limit (blue line in Fig. 11(b)). The trajectory shown decelerates from about 35 m s^{-1} to about 25 m s^{-1} on the straight segment, but then does not apply any deceleration, since the lateral acceleration resulting from the curvature of the path requests the full available friction potential. The Supervisor results in an 'unsafe' rating.

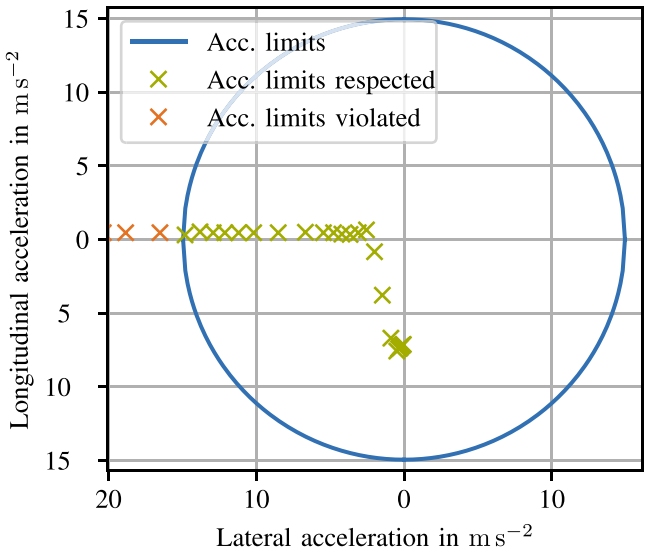
All scenarios were also simulated with an active Supervisor. In this case, as soon as one of the trajectories is evaluated as unsafe, the last valid emergency trajectory is executed (Fig. 5) until a new trajectory is considered safe again. One example of an active intervention is shown in Fig. 12. Therefore, scenario T16, which was presented before with a passive Supervisor (Fig. 8), is simulated with an active Supervisor. Here, it is clear that at the same plotted time step, the ego-vehicle has been slowed down by the Supervisor to such an extent that no collision can occur at the visualized time. Furthermore, there is no collision in the corner itself, which can be seen by comparing the future vehicle positions.

B. FIELD-TEST EVALUATION – REAL-WORLD EXAMPLE

In addition to scenario-based tests with deliberate fault injection, long runs in a real environment are essential to verify a low false alarm rate. In this course, we used the data of a trajectory planner [44] in autonomous races to evaluate the Supervisor. As an example, we elaborate a race with 10 laps and two cars on a racetrack in Modena, Italy (Fig. 13). The two vehicles were operated autonomously following Roborace regulations (Table 2) with speeds up to 160 km h^{-1}. Several overtaking maneuvers took place in compliance with regulations during the race, and no unsafe



(a) Planned performance trajectory.



(b) Acceleration profile along the planned trajectory.

FIGURE 11. Planned performance trajectory of the ego-vehicle at 0.4 s into scenario T28.

situations occurred. The Supervisor did not show any false positives ('unsafe' rating) during the entire race.

C. IMPLEMENTATION DETAILS

While the safety-compliant final version is to be implemented in a C-based language, the prototype Supervisor presented in this paper is implemented in Python 3.7 for development convenience. The average calculation time for one iteration of the Supervisor (safety evaluation of the performance and emergency trajectory) is 50.93 ms. However, it should be noted that the calculation time depends greatly on the proximity to other vehicles and active regulations. Thus, a distribution of the calculation time for the real-world run presented previously is shown in Fig. 14. The code was executed on an Intel Xeon E3-1245 3.3 GHz and key parameters influencing the performance are listed in Table 3. The computation time can be drastically reduced if shorter or more coarsely discretized trajectories are provided.

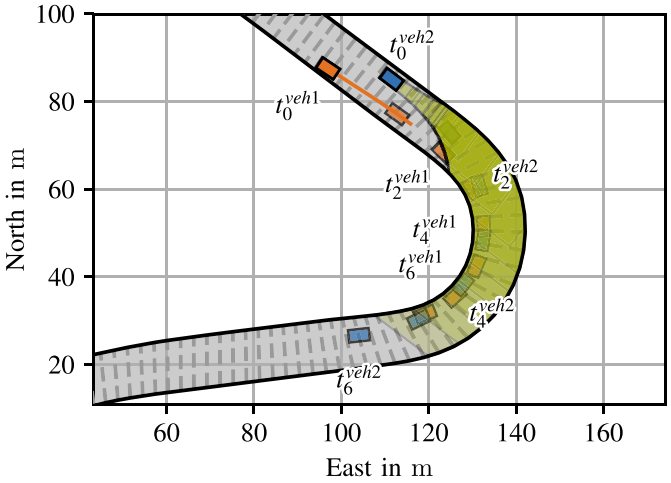
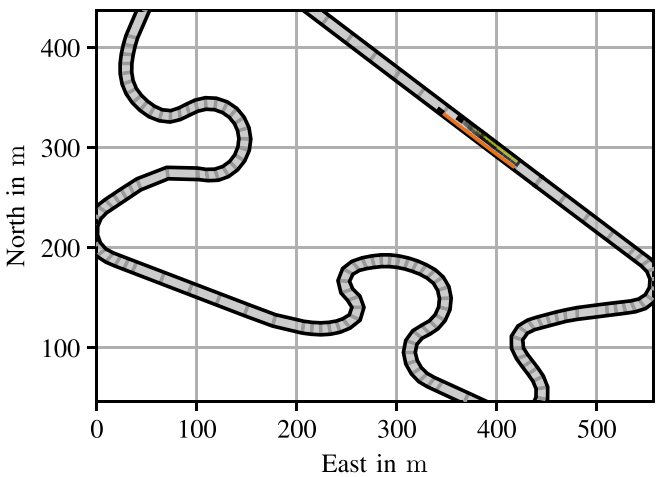


FIGURE 12. Planned emergency trajectory of the ego-vehicle (orange line) and reachable set of the object vehicle (green) at the 3.2 s time stamp in scenario T16 with an active Supervisor, i.e., whenever a trajectory is rated unsafe, the last valid emergency trajectory is executed. Future vehicle poses are drawn in 1.0 s increments, while every second instance i is denoted by t_i^{veh1} and t_i^{veh2} for the ego and object vehicle respectively.

(a) Roborace Devbot 2 on the racetrack.



(b) Track layout with two vehicles on the track. Planned emergency trajectory of the ego-vehicle in orange, reachable set of the other vehicle in green.

FIGURE 13. Real-world runs with the Roborace Devbot 2 on the racetrack in Modena, Italy.

In addition, it should be noted that the approach is compliant with applicable standards. In particular, the critical points presented in Section II-A, such as functional specification, modularity and transparency, can be achieved without any problems.

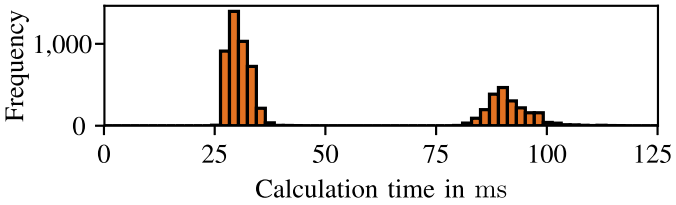


FIGURE 14. Distribution of required calculation time per iteration (rating of both, performance and emergency trajectory).

TABLE 3. Key parameters influencing performance.

Description	Value
Planning horizon (performance)	200 m - 600 m (velocity dependent)
Planning horizon (emergency)	200 m - 600 m, crop at $v = 0\,\mathrm{m\,s^{-1}}$
Trajectory step size	2 m, i.e. 100 - 300 states
Reachable set horizon	3 s
Reachable set step size	0.2 s

VII. CONCLUSION

We have introduced an OV - the Supervisor - that makes it possible to safeguard a planning module of arbitrary complexity (including online learning AI) during runtime. Based on an identified list of features for a safe trajectory, we developed safety metrics. One of the core components is the safety evaluation in the environment of other road users. Here, the rule-based reachable sets presented have proven to be effective. Compared to previous work, it is thus also possible to operate in semi-structured environments such as a race setting. The implemented prototype works in real time and was convincing both in error-injected scenario-based tests and on real-world run data. The Supervisor prototype meets the requirements of existing safety-related standards introduced in Section II-A. Future work includes the integration of further regulations, tests in a larger set of scenarios, and the realization of a C-based implementation. While the implementation for a racing environment was demonstrated, the generic method is transferable to road vehicles if the applicable regulations are adapted. The implemented Supervisor will be publicly available on GitHub (github.com/TUMFTM/TrajectorySupervisor).

ACKNOWLEDGMENT AND CONTRIBUTIONS

Research was supported by TÜV Süd. As the first author, Tim Stahl initiated the idea of this paper and is responsible for the concept and implementation presented. Frank Diermeyer contributed to the conception of the research project and revised the paper critically for important intellectual content. He gave final approval of the version to be published and agrees to all aspects of the work. As guarantor, he accepts responsibility for the overall integrity of the paper.

REFERENCES

[1] World Health Organization. *Global Status Report on Road Safety 2018*. Accessed: Jan. 2021. [Online]. Available: https://www.who.int/publications-detail-redirect/9789241565684

[2] "Advanced driver assistance systems," Eur. Commiss., Brussels, Belgium, Rep., 2016. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/ersosynthesis2016-adas15_en.pdf

[3] H. Cao, M. Li, S. Zhao, and X. Song, *Decision Making, Planning, and Control Strategies for Intelligent Vehicles* (Synthesis Lectures on Advances in Automotive Technology). San Rafael, CA, USA: Morgan & Claypool, Jul. 2020.

[4] B. Paden, M. Čap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[5] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," Jan. 2018. [Online]. Available: arXiv:1812.03079.

[6] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[7] H. Winner, "Introducing autonomous driving: An overview of safety challenges and market introduction strategies," *Automatisierungstechnik*, vol. 66, no. 2, pp. 100–106, Jan. 2018.

[8] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE Int. J. Transp. Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[9] T. Stahl, M. Eicher, J. Betz, and F. Diermeyer, "Online verification concept for autonomous vehicles—Illustrative study for a trajectory planning module," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2020, pp. 1–7.

[10] *Road Vehicles—Safety of the Intended Functionality*, ISO/PAS Standard 21448:2019, Jan. 2019.

[11] "Road vehicles—Functional safety," Int. Org. Stand., Geneva, Switzerland, Rep. ISO 26262, 2018.

[12] *Road Vehicles—Cybersecurity Engineering*, ISO/SAE Standard 21434:2019, Jan. 2019.

[13] IT-Beauftragter der Bundesregierung. *V-Modell XT*. Accessed: Aug. 19, 2020. [Online]. Available: https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html

[14] D. Gunning, *Explainable Artificial Intelligence (XAI)*. Defense Adv. Res. Projects Agency (DARPA), Arlington, TX, USA, Jan. 2017.

[15] R. Salay, R. Queiroz, and K. Czarnecki, "An analysis of ISO 26262: Using machine learning safely in automotive software," Jan. 2017. [Online]. Available: arXiv:1709.02435.

[16] H. Monkhouse, I. Habli, J. McDermid, S. Khastgir, and G. Dhadyalla, "Why functional safety experts worry about automotive systems having increasing autonomy," in *Proc. Int. Workshop Driver Driverless Cars Competition Coexistence*, Aug. 2017, pp. 1–6.

[17] A. Koenig, K. Witzlsperger, F. Leutwiler, and S. Hohmann, "Overview of HAD validation and passive HAD as a concept for validating highly automated cars," *Automatisierungstechnik*, vol. 66, no. 2, pp. 132–145, Jan. 2018.

[18] *Safety First for Automated Driving*, Aptiv, Audi, Apollo, BMW, Continental, Daimler, FCA, Here, Infineon, Intel, and Volkswagen, Dublin, Ireland, Ingolstadt, Germany, Munich, Germany, Hanover, Germany, Stuttgart, Germany, Auburn Hills, MI, USA, Amsterdam, The Netherlands, Neubiberg, Germany, Santa Clara, CA, USA, and Wolfsburg, Germany, 2019.

[19] S. Wachter, B. Mittelstadt, and L. Floridi, "Transparent, explainable, and accountable AI for robotics," *Sci. Robot.*, vol. 2, no. 6, p. 6080, Jan. 2017.

[20] M. Henzel, W. Hermann, and B. Lattke, "Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen," in *Proc. Workshop Fahrerassistenzsysteme*, Jan. 2017, pp. 136–148.

[21] P. Feth, D. Schneider, and R. Adler, "A conceptual safety supervisor definition and evaluation framework for autonomous systems," in *Computer Safety, Reliability, and Security* (Lecture Notes in Computer Science), S. Tonetta, E. Schoitsch, and F. Bitsch, Eds. Cham, Switzerland: Springer, 2017, pp. 135–148.

[22] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," SAE Int., Warrendale, PA, USA, Tech. Paper 2018-01-1071, Apr. 2018.

[23] 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.
[24] S. Annell, A. Gratner, and L. Svensson, "Probabilistic collision estimation system for autonomous vehicles," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Jan. 2016, pp. 473–478.
[25] A. Houenou, P. Bonnifait, and V. Cherfaoui, "Risk assessment for collision avoidance systems," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Jan. 2014, pp. 386–391.
[26] J. Ward, G. Agamennoni, S. Worrall, and E. Nebot, "Vehicle collision probability calculation for general traffic scenarios under uncertainty," in *Proc. IEEE Intell. Veh. Symp.*, Jan. 2014, pp. 986–992.
[27] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 137–147, Mar. 2008.
[28] B. Kim, K. Park, and K. Yi, "Probabilistic threat assessment with environment description and rule-based multi-traffic prediction for integrated risk management system," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 8–22, Jan. 2017.
[29] B. Xu, Q. Li, T. Guo, Y. Ao, and D. Du, "A quantitative safety verification approach for the decision-making process of autonomous driving," in *Proc. Int. Symp. Theor. Aspects. Softw. Eng. (TASE)*, 2019, pp. 128–135.
[30] K. Esterle, V. Aravantinos, and A. Knoll, "From specifications to behavior: Maneuver verification in a semantic state space," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2019, pp. 2140–2147.
[31] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff, "Formalization of interstate traffic rules in temporal logic," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 752–759.
[32] C. Pek, V. Rusinov, S. Manzinger, M. C. Üste, and M. Althoff, "CommonRoad drivability checker: Simplifying the development and validation of motion planning algorithms," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1–8.
[33] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," Jan. 2017. [Online]. Available: arXiv:1708.06374.
[34] F. Gruber and M. Althoff, "Anytime safety verification of autonomous vehicles," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Jan. 2018, pp. 1708–1714.
[35] B. Schurmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Koster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Jan. 2017, pp. 1–8.
[36] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, Aug. 2014.
[37] C. Pek, M. Koschi, and M. Althoff, "An online verification framework for motion planning of self-driving vehicles with safety guarantees," in *Proc. AAET*, Jan. 2019, pp. 260–274.
[38] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Jan. 2018, pp. 1447–1454.
[39] A. Rizaldi, F. Immler, B. Schürmann, and M. Althoff, "A formally verified motion planner for autonomous vehicles," in *Automated Technology for Verification and Analysis* (Lecture Notes in Computer Science), S. K. Lahiri and C. Wang, Eds. Cham, Switzerland: Springer, 2018, pp. 75–90.
[40] M. Althoff, D. Heß, and F. Gambert, "Road occupancy prediction of traffic participants," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 99–105.
[41] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic reachability computation for families of linear vector fields," *J. Symbolic Comput.*, vol. 32, no. 3, pp. 231–253, Sep. 2001.
[42] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Trans. Intell. Veh.*, vol. 1, no. 2, pp. 187–202, Jun. 2016.
[43] K. Havelund and G. Roşu, "Synthesizing monitors for safety properties," in *Tools and Algorithms for the Construction and Analysis of Systems* (Lecture Notes in Computer Science), J.-P. Katoen and P. Stevens, Eds. Heidelberg, Germany: Springer, 2002, pp. 342–356.
[44] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oct. 2019, pp. 3149–3154.
[45] "Appendix L to the international sporting code," Federation Internationale L'Automobile, Paris, France, Rep., 2020. [Online]. Available: https://www.fia.com/sites/default/files/appendix_l_2020_publie_le_16_decembre_2020.pdf
[46] "Formula 1—Sporting regulations," Federation Internationale L'Automobile, Paris, France, Rep., 2016. [Online]. Available: https://www.fia.com/file/40714/download
[47] R. Homma, T. Wakasugi, and K. Kodaka, "Influence of difference of minimum risk maneuver of an automated vehicle on the following vehicle," *Trans. Soc. Automot. Eng. Japan*, vol. 51, no. 1, pp. 149–154, Jan. 2020.
[48] 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.
[49] C. Amersbach and H. Winner, "Defining required and feasible test coverage for scenario-based validation of highly automated vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oct. 2019, pp. 425–430.
[50] T. Stahl and J. Betz, "An open-source scenario architect for autonomous vehicles," in *Proc. 15th Int. Conf. Ecol. Veh. Renew. Energies (EVER)*, 2020, pp. 1–8.
[51] S. M. S. Mahmud, L. Ferreira, M. S. Hoque, and A. Tavassoli, "Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs," *IATSS Res.*, vol. 41, no. 4, pp. 153–163, Dec. 2017.

TIM STAHL (Member, IEEE) received the B.Eng. degree in mechatronics from the Technical University of Applied Sciences Rosenheim, Germany, in 2014, and the M.Sc. degree in electrical engineering and information technology from the Technical University of Munich, Germany, in 2016, where he is currently pursuing the Ph.D. degree with the Chair of Automotive Technology. His research interests are focused on safety assessment and trajectory planning for autonomous race vehicles.

FRANK DIERMEYER received the Diploma and Ph.D. degrees in mechanical engineering from the Technical University of Munich, Munich, Germany, in 2001 and 2008, respectively, where he has been a Senior Engineer with the Chair of Automotive Technology Since 2008, and he has been the Leader of the Automated Driving Research Group. His research interests include teleoperated driving, humanmachine interaction, and safety validation.