Fail-safe Motion Planning for Online Verification of Autonomous Vehicles Using Convex Optimization

Christian Pek[1], *Member, IEEE,* and Matthias Althoff[2], *Member, IEEE*

***Abstract*—Safe motion planning for autonomous vehicles is a challenging task, since the exact future motion of other traffic participant is usually unknown. We present a verification technique ensuring that autonomous vehicles do not cause collisions by using fail-safe trajectories. Fail-safe trajectories are executed if the intended motion of the autonomous vehicle causes a safety-critical situation. Our verification technique is real-time capable and operates under the premise that intended trajectories are only executed if they have been verified as safe. The benefits of our proposed approach are demonstrated in different scenarios on an actual vehicle. Moreover, we present the first in-depth analysis of our verification technique used in dense urban traffic. Our results indicate that fail-safe motion planning has the potential to drastically reduce accidents while not resulting in overly conservative behaviors of the autonomous vehicle.**

***Index Terms*—Formal verification, motion planning, autonomous vehicles, fail-safe operation, safe states, set-based computation.**

I. Introduction

SAFE motion planning remains an open issue in terms of realizing autonomous vehicles. Although existing motion-planning techniques are able to generate collision-free trajectories in many situations, they still can cause accidents in critical situations. Online verification approaches are particularly well suited for ensuring that autonomous vehicles do not cause accidents [1]–[3]. In contrast to testing-based methods, online verification approaches perform the safety analysis online in each situation and thus never miss verifying a scenario that results in a safety-critical situation. For this reason, these approaches are able to ensure that planned motions are provably correct with respect to a given specification and require less effort for certification.

Nevertheless, existing online verification approaches are not yet ready for autonomous vehicles. The reasons for this deficiency are manifold: 1) Online verification approaches are still not computationally efficient enough to be used in motion planning frameworks with fast replanning rates of 20 Hz and higher. 2) These approaches cannot ensure that the vehicle will not cause collisions for arbitrarily planned motions and in arbitrary traffic situations over an infinite time horizon. 3) In case a situation is unsafe, many approaches lack a mechanism that provides alternative motion plans to avoid collisions. These shortcomings still endanger passengers and other traffic participants and must be resolved to exploit the full potential of online verification.

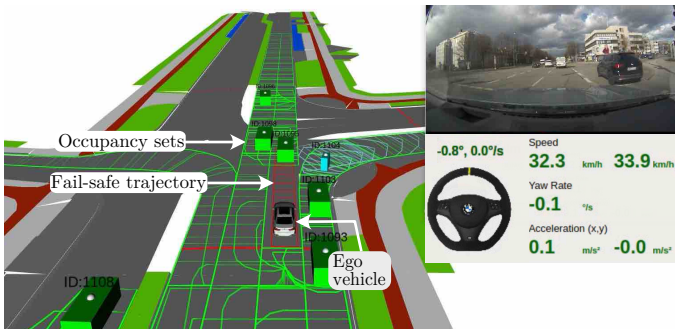


Fig. 1. Snapshot from our driving experiments with a BMW 7-series test-vehicle. Our verification technique is real-time capable and can be used to verify the safety of arbitrarily planned trajectories online.

A. Literature overview

In the following paragraphs, we extensively review state-of-the-art motion planning and verification techniques for autonomous vehicles.

a) Trajectory Planning: Many exciting motion-planning approaches have been introduced in the last decade. Extensive overviews of these techniques are given in [4]–[6]. Although machine learning techniques have become very popular recently, e.g., [7]–[12], we do not consider them for generating fail-safe trajectories, since candidate fail-safe planners need to be strictly real-time capable and auditable.

Discrete planning approaches are popular planning techniques for autonomous vehicles. These approaches discretize the search space (state or input space) to obtain feasible trajectories. For instance, planning with motion primitives is one discrete planning technique. Motion primitives are precomputed trajectory pieces which are concatenated online through search-based algorithms [13]–[18]. Since the underlying motion primitives are precomputed offline, the primitive computation can incorporate complex vehicle models, such as multi-body models [19].

Sampling-based trajectory planners sample states in the search space to obtain feasible trajectories. For instance, *rapidly-exploring random trees* (RRTs) [20]–[23], randomly sample and connect states toward a goal region to generate trajectories. Through the random sampling strategy, RRTs are perfectly suited for traversing high-dimensional search

[1]Christian Pek is with the Division of Robotics, Perception and Learning, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden. This work was conducted when Christian was a PhD student at the Technical University of Munich. e-mail: pek2@kth.se
[2]Matthias Althoff is with the Department of Computer Science, Technical University of Munich, D-85748 Garching, Germany.
e-mail: althoff@in.tum.de

 https://doi.org/10.1109/TRO.2020.3036624

spaces. Nevertheless, RRTs and their variants might not obtain (optimal) motions in time due to the randomized sampling [4].

Graph-search approaches, such as *state lattices*, are yet another form of discrete planning techniques and work on fixed graph structures [24]–[29]. They obtain sets of trajectories whose goal states are vertices in a fixed predefined grid, resulting in a lattice structure. State lattices have been combined with optimal control techniques in [30] to compute jerk-optimal trajectories. In general, state lattices generate drivable trajectories, but lack optimality and completeness due to the fixed grid.

Although discrete planning approaches are often easy to implement and solve motion problems effectively, they have major disadvantages. Due to the discretization strategy, they may fail to obtain solutions in safety-critical scenarios with small and convoluted solution spaces. For the same reason, they may also fail to determine trajectories ending in small safe terminal sets (cf. completeness in [31, pp. 79-80]). However, both requirements are crucial for meeting the high demands of fail-safe trajectory planning.

To overcome the limitations of discretization, continuous model predictive control approaches generate collision-free trajectories by minimizing a cost function with respect to state and input constraints (and possibly a set of disturbances). The underlying optimization problems are defined, e.g., as mixed-integer programs in [32]–[37] and as sequential quadratic programs in [38]–[42]. In general, the resulting optimization problems are non-convex and are thus harder to solve and usually not real-time capable; one of the reasons for this is that solvers can get stuck in local minima [43].

The generally non-convex motion planning problem can be approximated as a convex problem, e.g., by linearizing the non-linear, non-holonomic vehicle dynamics [44], [45] and separating the motion into a longitudinal and a lateral component [46]. The resulting convex optimization problems can be efficiently solved with global convergence as succesfully shown in [47]–[51]. Convex optimization techniques provide promising results for real-time planning in complex traffic situations. However, approaches which separate the longitudinal and lateral motion may obtain infeasible trajectories in complex scenarios in which both components are heavily linked [52]. We address this feasibility problem by focusing on simple evasive maneuvers and providing safe fallback solutions (see experiments in Sec. VI).

b) Safety Verification: Different verification techniques, such as barrier certificates, correct-by-construction controller synthesis, or model checking, have been introduced over the years. In this paper, we focus on popular formal verification techniques within the domain of autonomous vehicles.

In theorem proving, desired system properties are formulated using logical formulas. The verification is then performed by checking the satisfiability of the logical formulas. For instance, theorem proving has been applied to highway entry systems [53], to lane change controllers [54], and to adaptive cruise control systems [55], [56]. Although theorem proving is quite powerful and effective, it usually requires manual intervention to generate desired system behaviors and logical formulas must often be adapted to new scenarios.

Autonomous vehicles are also safe if they never enter inevitable collision states (ICS) [57]. ICS are states in which all possible trajectories of the autonomous vehicle eventually collide with an obstacle [58]–[64]. Note that ICS reason over infinite time horizons. Determining ICS in arbitrary traffic scenarios is computationally expensive, and most works lower the computational effort by only considering a single trajectory prediction of traffic participants [61].

Complementary to ICS, controlled invariant sets (CIS) [65]–[67] guarantee persistent feasibility. By definition, there exist at least one collision-free trajectory for every state within a CIS with respect to the future behavior of other traffic participants; thus, the vehicle remains safe. Unfortunately, obtaining CIS in dynamic environments is challenging due to the unknown future motion of obstacles. However, in our previous work [68], we have shown that invariably safe sets guarantee persistent feasibility and that an under-approximation of these sets can be obtained in real-time.

Set-based reachability analysis has also been succesfully applied to check whether trajectories are collision-free while accounting for any feasible future motion of dynamic obstacles [1], [69]–[71]. Loosely speaking, the reachable set of a dynamical system corresponds to the set of states the system is able to reach over time considering an initial set of states and all possible system inputs. Future collisions of the autonomous vehicle can be identified by checking for intersections of its reachable set with the ones associated with the obstacles. However, reachability analysis comes with the disadvantage that unsafe regions may grow rapidly over time, since any feasible future motion of obstacles is considered. As a result, planned motions may often be rejected as being potentially unsafe, leaving the autonomous vehicle without a safe trajectory. We resolve this issue in our verficiation technique by combining reachable sets with fail-safe trajectory planning.

Recent efforts in vehicle safety are made towards providing a formal safety model such Responsibility-Sensible Safety (RSS) [72]. Inspired by traffic rules (e.g., vehicles are not allowed to cause rear-end collisions), RSS identifies safety-critical situations and appropriate responses by the autonomous vehicle. In particular, RSS involves the computation of (longitudinal and lateral) safe distances to other traffic participants and braking maneuvers to avoid collisions if safe distances will be violated [73]. Nevertheless, RSS in its current form is not able to provide strong safety guarantees, since it assumes that other traffic participants act according to common sense rules which may not prove true in reality [72, Sec. III]. Moreover, RSS does not provide a formal specification, which makes it hard to verify trajectories of the autonomous vehicle during operation. In contrast, our verification technique provides a formal (and parameterizable) model that can be certified by authorities.

B. Contributions

Following up on our previous work [68], [74], we present a verification technique that ensures the safety of planned motions of an autonomous vehicle, denoted as *ego vehicle*

in the following sections, and eliminates the shortcomings of existing verification approaches. We present the following innovations compared to our previous work [68], [74]:

1) Fail-safe trajectories are verified over an infinite-time horizon to ensure that the ego vehicle remains safe at all times. This property is achieved by linearizing invariably safe sets and integrating them as additional terminal constraints.
2) Fail-safe trajectories now start at the latest possible point in time along planned motions and improve passenger comfort through the use of slack variables.
3) The benefits of fail-safe trajectories and the utilized linear vehicle have been extensively tested on an actual vehicle in safety-critical situations, e.g., when generating random steering and acceleration inputs.
4) The intervention rate of our verification technique is evaluated in recorded urban traffic.

C. Outline of the paper

This paper is structured as follows: In Sec. II, we introduce required mathematical models and definitions. Sec. III presents the general procedure of our verification technique. In Sec. IV and V, we describe the computation of invariably safe sets and the generation of fail-safe trajectories using convex optimization, respectively. The benefits of our approach are demonstrated in safety-critical situations on an actual test vehicle in Sec. VI. We finish with conclusions in Sec. VII.

II. Preliminaries

Let us introduce the configuration space $\mathcal{X} \subset \mathbb{R}^n$ as the possible set of states x, the input set $\mathcal{U} \subset \mathbb{R}^m$ as the set of admissible control inputs u, and the set $\mathcal{Z} \subset \mathbb{R}^q$ as possible disturbances z acting on the ego vehicle, whose motions are governed by the differential equation

$$\dot{x}(t) = f\big(x(t), u(t), z(t)\big). \tag{1}$$

We use the notation $x^{(i)}, i \in \mathbb{N}_0$, to describe the i-th component of the state variable x. We adhere to the notations $x\big([t_0, t_h]\big), x(t) \in \mathcal{X}$, and $u\big([t_0, t_h]\big), u(t) \in \mathcal{U}$, to denote state and input trajectories for the time interval $[t_0, t_h]$, respectively. In addition, $\chi\big(t_h, x(t_0), u([t_0, t_h])\big)$ denotes the solution of (1) at the point in time t_h with respect to the initial state $x(t_0) = x_0$ and the input trajectory $u\big([t_0, t_h]\big)$. By an abuse of notation, we use $u\big([t_1, t_2]\big) = \Phi(\phi_{\mathrm{ref}}), t_1 \leq t_2$, to emphasize that an input trajectory is generated by a feedback control law Φ for a given reference ϕ_{ref}, e.g., tracking a desired velocity.

We consider a lane-based environment for our online verification technique, which is modeled as a subset of the Euclidean space $\mathbb{R}^2$ and provided by the environment model of the vehicle. For motion planning, we use a curvilinear coordinate system that is aligned to a given reference path $\Gamma := (p_0, p_1, \ldots, p_{k_\Gamma}), \forall i \in [0, \ldots, k_\Gamma] : p_i \in \mathbb{R}^2, k_\Gamma \in \mathbb{N}$. The reference path is usually given by a high-level route planner and may, for instance, correspond to the centerline of the current lane. Using the curvilinear coordinate system, positions $p \in \mathbb{R}^2$ will be described in terms of the arc length s along Γ and the orthogonal deviation d (cf. Fig. 5). The operator $\Upsilon(p) = (s, d)^T$ transforms positions $p \in \mathbb{R}^2$ to the curvilinear coordinate system. We use the following operations between two sets $\mathcal{X}_1$ and $\mathcal{X}_2$: $\mathcal{X}_1 \cup \mathcal{X}_2$ denotes the union of sets, $\mathcal{X}_1 \cap \mathcal{X}_2$ is the intersection of sets, and $\mathcal{X}_1 \setminus \mathcal{X}_2 := \{x_1 \,|\, x_1 \in \mathcal{X}_1 \wedge x_1 \notin \mathcal{X}_2\}$ denotes the set difference. Moreover, the Minkowski sum of two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ is defined as $\mathcal{X}_1 \oplus \mathcal{X}_2 := \{x_1 + x_2 \,|\, x_1 \in \mathcal{X}_1 \wedge x_2 \in \mathcal{X}_2\}$.

To obtain the occupied space of the autonomous vehicle (i.e., its footprint in $\mathbb{R}^2$), we introduce the operator occ:

Definition 1 (Occupancy of States)
The operator $\mathrm{occ}(x)$ relates the state vector x to the set of points in the environment occupied by the system as $\mathrm{occ}(x) : \mathcal{X} \to \mathrm{P}\big(\mathbb{R}^2\big)$, where $\mathrm{P}(\mathbb{R}^2)$ is the power set of $\mathbb{R}^2$. Given a set $\mathcal{X}'$, we define $\mathrm{occ}(\mathcal{X}') := \{\mathrm{occ}(x) \,|\, x \in \mathcal{X}'\}$.

The set $\mathcal{B}$ describes all safety-relevant obstacles within the environment. The information about obstacles (state, type, and measurement uncertainties) is provided by the environment model, which is usually obtained using on-board sensors of the vehicle [75]. We assume that obstacles are detected as soon as they enter the ego vehicle's field of view. To account for the uncertain future motion of obstacles, we make use of the set-based prediction in [71], [76] which computes all feasible legal future motions of obstacles over time using reachability analysis and provides us with occupancy sets:

Definition 2 (Occupancy Set $\mathcal{O}$)
The occupancy set $\mathcal{O}_b(t) \subseteq \mathbb{R}^2$ describes the set of possibly occupied points in the environment by an obstacle $b \in \mathcal{B}$ at a point in time t. For a time interval $[t_1, t_2], t_1 \leq t_2$, we define $\mathcal{O}_b([t_1, t_2]) = \bigcup_{t_1 \leq t \leq t_2} \mathcal{O}_b(t)$.

Using the introduced operator in Def. 1 and the predicted occupancies $\mathcal{O}_b(t)$ of obstacles $b \in \mathcal{B}$, we can compute the set of collision-free states at a point in time t:

Definition 3 (Collision-free States $\mathcal{F}$)
The set $\mathcal{F}(t) \subseteq \mathcal{X}$ is the set of collision-free states at time t considering the occupancy $\mathcal{O}_\mathcal{B}(t) = \bigcup_{b \in \mathcal{B}} \mathcal{O}_b(t)$ of obstacles $\mathcal{B}$, i.e., $\mathcal{F}(t) := \{x \in \mathcal{X} \,|\, \mathrm{occ}(x) \cap \mathcal{O}_\mathcal{B}(t) = \emptyset\}$.

For fail-safe trajectories, we are particularly interested in finding (collision-free) states that allow the ego vehicle to remain collision-free for an infinite time horizon. We define such safe states through recursion [68]: we denote a state as safe if we can determine a collision-free trajectory to another safe state. This recursive definition allows us to derive subsets (cf. Fig. 2) of the set of collision-free states $\mathcal{F}(t)$ (cf. Def. 3). By definition, these subsets only contain states that guarantee a safe transition to another safe state for an infinite time horizon. As a result, these subsets do not include ICS and are thus invariably safe. We formally define the set of invariably safe states as:

Definition 4 (Invariably Safe Set $\mathcal{S}$)
The invariably safe set $\mathcal{S}(t)$ for a point in time t contains all states that allow the ego vehicle to remain safe for an infinite time horizon and is defined as:

$$\mathcal{S}(t) := \big\{ x \in \mathcal{F}(t) \,\big|\, \forall t' > t : \chi\big(t', x, \Phi(\phi_{\mathrm{ref}})\big) \in \mathcal{F}(t') \big\},$$

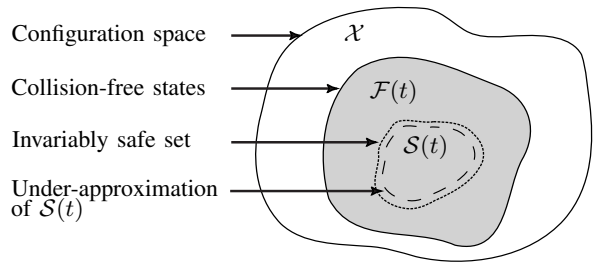


Fig. 2. Relation of the configuration space $\mathcal{X}$, collision-free states $\mathcal{F}(t)$, and invariably safe sets $\mathcal{S}(t)$.

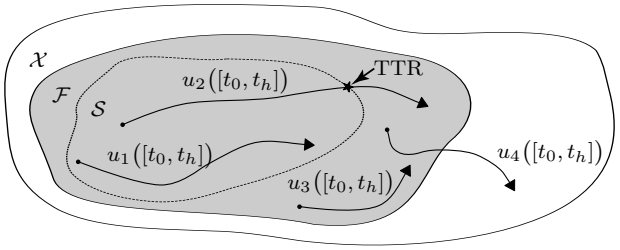


Fig. 3. Safety properties of trajectories. Only trajectory u_1 is an invariably safe trajectory which remains safe for $t > t_h$, since it is enclosed in the invariably safe set $\mathcal{S}(t)$. Trajectory u_2 leaves $\mathcal{S}(t)$ at the time-to-react TTR. Trajectory u_3 is just a collision-free trajectory, since it is only enclosed in the set of collision-free states $\mathcal{F}(t)$. Trajectory u_4, on the other hand, is not collision-free.

where ϕ_{ref} is an arbitrary desired reference, e.g., tracking a desired velocity while remaining in $\mathcal{F}(t)$.

With invariably safe sets, we are able to check whether a given trajectory $u\big([t_0,t_h]\big), t_0 < t_h$, remains safe for times $t' > t_h$:

Definition 5 (Invariably Safe Input Trajectory)
The input trajectory $u\big([t_0,t_h]\big), t_0 < t_h$, is called an invariably safe input trajectory if $u\big([t_0,t_h]\big)$ is collision-free and $\chi\big(t_h, x(t_0), u([t_0,t_h])\big) \in \mathcal{S}(t_h)$ (cf. Def. 4).

Another use of invariably safe sets is to obtain the *time-to-react* (TTR) [77, Sec. II], which is the last state along a trajectory for which a collision-free evasive trajectory still exists:

Definition 6 (Time-To-React)
Assuming that $x(t_0) \in \mathcal{S}(t_0)$, the time-to-react (TTR) is the maximum time the ego vehicle can continue the input trajectory $u\big([t_0,t_h]\big)$ for which the existence of a collision-free trajectory is guaranteed, i.e., $t_{\mathrm{TTR}} := \sup\big\{t \in [t_0,t_h] \,|\, \chi\big(t, x(t_0), u([t_0,t])\big) \in \mathcal{S}(t)\big\}$.

Fig. 3 illustrates the different safety properties of trajectories and the TTR.

III. Overview of the Online Verification Approach

Before we introduce the general procedure of our verification approach, we introduce different types of trajectories to guide the reader through this section.

1) *Intended motion* $\mathfrak{I}$: Trajectory with a (typically) long planning horizon that the ego vehicle should follow. $\mathfrak{I}$ is planned by an arbitrary motion planner and optimized for passenger comfort by considering the most likely motion of other traffic participants.
2) *Invariably safe part* $\mathfrak{I}_{\mathrm{safe}}$: Part of an intended motion $\mathfrak{I}$ that is invariably safe, computed using the TTR.
3) *Fail-safe trajectory* $\mathfrak{F}$: An invariably safe trajectory which serves as an emergency trajectory that keeps the ego vehicle within a safe state at all times, e.g., standstill in dedicated areas.
4) *Provably safe trajectory* $\mathfrak{I}_{\mathrm{ver}}$: Combination of $\mathfrak{I}_{\mathrm{safe}}$ and $\mathfrak{F}$ that is provably safe with respect to all feasible legal motions of other traffic participants.

Although motion planning algorithms generate collision-free motions $\mathfrak{I}$, these motions might not be safe when executed in actual traffic. The reason for this is that planners are designed to generate comfortable and anticipatory motions: motions $\mathfrak{I}$ are only collision-free against the most likely motion of other traffic participants (cf. intended motion and most likely trajectory in Fig. 4a). However, if traffic pariticipants deviate from the most likely trajectory, motions $\mathfrak{I}$ may no longer be safe.

Our verification approach is based on the policy that the ego vehicle is only allowed to execute verified trajectories $\mathfrak{I}_{\mathrm{ver}}$. Following the widely accepted Vienna Convention on Road Traffic [72], [78], we verify if a motion plan is collision-free against all possible legal motions of other traffic participants. Therefore, we use the set-based prediction in [71], [76] to compute all feasible legal[1] future motions of obstacles in the environment (cf. blue regions in Fig. 4).

Since we consider all possible future evolutions of a scenario with the formal prediction, motions $\mathfrak{I}$ might be rejected as potentially unsafe (cf. intersection of intended motion and blue region in Fig. 4a). However, many motions $\mathfrak{I}$ that are initially unsafe for the entire considered time horizon might be safe for a short period of time. Our verification technique determines the safe part $\mathfrak{I}_{\mathrm{safe}}$ of $\mathfrak{I}$ using the TTR (cf. black circle in Fig. 4). Since $\mathfrak{I}_{\mathrm{safe}}$ does not ensure that the ego vehicle will remain within a safe state at all times, e.g., standstill in dedicated areas, we append a fail-safe trajectory $\mathfrak{F}$ (cf. red paths in Fig. 4). We use the obtained TTR as the optimal point along $\mathfrak{I}$ to plan fail-safe trajectories. This choice reflects the fact that system designers usually want safety systems to intervene at the latest possible point in time.

An intended motion $\mathfrak{I}$ is verified if both $\mathfrak{I}_{\mathrm{safe}}$ and $\mathfrak{F}$ have been correctly computed. In this case, we allow the ego vehicle to execute the provably safe motion $\mathfrak{I}_{\mathrm{ver}}$, which ensures safety even if other traffic participants deviate from their most likely trajectories. Let us now consider how our verification technique ensures that the ego vehicle only executes verified motions $\mathfrak{I}_{\mathrm{ver}}$. Without loss of generality, we assume that the motion planner of the ego vehicle generates motions $\mathfrak{I}_c$ in consecutive planning cycles $c \in \mathbb{N}_+$, where $\mathfrak{I}_c$ will be

[1]The set-based prediction in [71], [76] also adjusts to individual misbehavior of traffic participants. However, an advantage of our approach is that if a collision occurs, we can verifiably argue that another traffic participant has violated traffic rules.

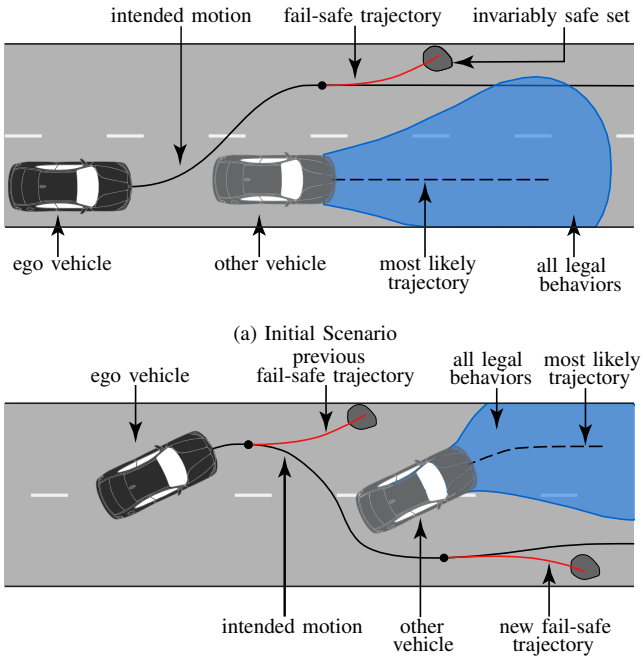


Fig. 4. Fail-safe trajectories are collision-free with respect to any feasible behavior of obstacles and end in invariably safe sets to guarantee safety for an infinite time horizon (a). While the ego vehicle proceeds along its intended motion, new fail-safe trajectories are computed to ensure safety at any time (b). In case no new valid fail-safe trajectory is found, the previously computed fail-safe trajectory must be executed.

executed in cycle $c+1$. For the sake of brevity, we sketch the inductive proof of our verification technique:

Base case ($c=0$): Initially, the ego vehicle is at a standstill and the motion planner generates motion $\mathfrak{I}_0$. If we can verify the safety of $\mathfrak{I}_0$, then the ego vehicle is allowed to execute $\mathfrak{I}_{\text{ver},0}$ in cycle $c=1$ (cf. Fig. 4a). Otherwise, the ego vehicle executes fail-safe trajectory $\mathfrak{F}_0$, which corresponds to remaining at a standstill.

Inductive step ($c=k$): Assuming that the ego vehicle is executing a verified trajectory $\mathfrak{I}_{\text{ver},k}$ (i.e., both $\mathfrak{I}_{\text{safe},k}$ and $\mathfrak{F}_k$ are correctly computed) for an arbitrary planning cycle $c=k, k\in\mathbb{N}_+$, we show that it remains safe in cycle $c=k+1$. Similar to the base case, we distinguish between two cases: 1) If we can verify $\mathfrak{I}_{k+1}$, then the ego vehicle is allowed to execute $\mathfrak{I}_{\text{ver},k+1}$ (cf. situation in Fig. 4b). 2) If $\mathfrak{I}_{k+1}$ cannot be verified, then the ego vehicle will still remain safe, since it can continue to execute the previous provably safe trajectory $\mathfrak{I}_{\text{ver},k}$ including $\mathfrak{F}_k$ (cf. assumption of inductive step), which ensures that the ego vehicle remains within a safe state at all times (cf. previous fail-safe trajectory in Fig. 4b).

It should be noted that even though the ego vehicle executes a fail-safe trajectory, it can return to its comfort driving mode by verifying a new intended motion if the safety-critical situation resolves. Moreover, we can integrate model inaccuracies of the ego vehicle (that result in tracking errors) in our approach by enlarging the dimensions of obstacles (see Def. 2) using a conservative safety bound or checking whether the reachable set of the ego vehicle along its trajectory intersects with unsafe sets, similar to [1]. In addition, one can ensure the drivability of provably safe trajectories using optimal control approaches [13]. However, neither is the focus of this work.

IV. Computation of Invariably Safe Sets

In our previous work [68], we presented how to obtain a tight under-approximation of $\mathcal{S}(t)$ in real-time. Let us briefly recall the results from [68]. To compute an under-appoximation of $\mathcal{S}(t)$, we utilize control laws Φ that maintain:

1) Formal safe distances according to [79]. If the ego vehicle respects the safe distance to its preceding vehicle, it is guaranteed that the ego vehicle can avoid a collision by braking, even if the preceding vehicle performs emergency braking.
2) Evasive distances according to [80]. If the ego vehicle respects the evasive distance to its preceding vehicle, it is guaranteed that the ego vehicle can avoid a collision by swerving to an adjacent lane, even if the preceding vehicle performs emergency braking.

It should be noted that if collisions occur due to misbehaviors on the part of other traffic participants, e.g., crashing into a tailback or cutting the ego vehicle off, the ego vehicle would not be accountable, since the other traffic participants violated traffic rules [78].

Safe and evasive distances are computed in the curvilinear coordinate system. Without loss of generality, we model the state of the ego vehicle as $x=(s,d,v)^T$ in the following paragraphs, where s and d are longitudinal and lateral positions, respectively, and v is the velocity. We enlarge the predicted occupancy sets $\mathcal{O}_b(t)$ (cf. Def. 2) for each safety-relevant obstacle $b\in\mathcal{B}$ with a circle R_{lon}, which denotes the smallest circumscribing circle covering the dimensions of the ego vehicle. The enlarged occupancy is computed as $\mathcal{O}_{b,\text{enl}} := \mathcal{O}_b(t)\oplus R_{\text{lon}}$ and allows us to over-approximate collision constraints. Next, we transform $\mathcal{O}_{b,\text{enl}}$ into the curvilinear coordinate system, resulting in occupancies $\mathcal{O}_{b,\text{cls}}(t) := \{\Upsilon(p)\,|\,p\in\mathcal{O}_{b,\text{enl}}(t)\}$.

According to [81], [82], the computation of the minimum required safe distance between the ego vehicle with velocity v_{ego} and absolute deceleration $-|a_{\text{s,max}}|$ and a preceding obstacle $b\in\mathcal{B}$ with velocity v_b and maximum absolute deceleration $-|a_{\text{s,max,b}}|$ depends on the following condition:

$$\begin{aligned}\big(|a_{\text{s,max,b}}|<|a_{\text{s,max}}|\big)\wedge\big(v_b^*<v_{\text{ego}}\big)\wedge\\ \big(v_{\text{ego}}/|a_{\text{s,max}}|<v_b^*/|a_{\text{s,max,b}}|\big),\end{aligned}\tag{2}$$

where δ_{brake} denotes the reaction time of the ego vehicle to perform braking and v_b^* denotes the remaining velocity of obstacle b after an emergency brake maneuver of obstacle b with duration δ_{brake}, defined as [82]:

$$v_b^* := \begin{cases} v_b - |a_{\text{s,max,b}}\delta_{\text{brake}}| & \delta_{\text{brake}} \le {v_b}/{|a_{\text{max,b}}|},\\ 0 & \text{otherwise.}\end{cases}\tag{3}$$

If condition (2) evaluates to true, the ego vehicle has to maintain the safe distance $\Delta_{\text{safe},1}$ to obstacle b, otherwise safe distance $\Delta_{\text{safe},2}$ [82]:

$$\begin{aligned}\Delta_{\text{safe},1}(v_{\text{ego}}, b) &:= \frac{(v_b - |a_{\text{s,max,b}}|\delta_{\text{brake}} - v_{\text{ego}})^2}{-2(|a_{\text{s,max,b}}| - |a_{\text{s,max}}|)} - v_b\delta_{\text{brake}} \\ &\quad + 1/2|a_{\text{s,max,b}}|\delta_{\text{brake}}^2 + v_{\text{ego}}\delta_{\text{brake}}, \\ \Delta_{\text{safe},2}(v_{\text{ego}}, b) &:= \frac{v_b^2}{-2|a_{\text{s,max,b}}|} - \frac{v_{\text{ego}}^2}{-2|a_{\text{s,max}}|} + v_{\text{ego}}\delta_{\text{brake}}.\end{aligned} \tag{4}$$

Note that the initial state of the ego vehicle and the prediction of other traffic participants are given.

We are now able to define the set of states in which the ego vehicle respects the safe distance to a preceding obstacle. Without loss generality, we assume that other traffic participants are able to decelerate at least as much as the ego vehicle, i.e., $|a_{\text{s,max,b}}| \geq |a_{\text{s,max}}|$. With this assumption, (2) evaluates to false and the ego vehicle has to respect safe distance $\Delta_{\text{safe},2}$ in longitudinal direction.

Proposition 1 (Safe Distance Set)
The set of states respecting a (longitudinal) safe distance to a preceding obstacle $b \in \mathcal{B}$ is defined as $\mathcal{S}^1(t) = \{(s,d,v)^T \in \mathcal{X} \,|\, \forall (s_b, d_b)^T \in \mathcal{O}_{b,\text{cls}}(t) : s \leq s_b - \Delta_{\text{safe},2}(v,b)\}$.

Proof. The safety that safe distances provide is shown in [81], [82]. ■

To compute evasive distances, we first introduce d_{eva} as the lateral distance necessary to fully enter an adjacent lane from a given lateral position d (whole shape of the ego vehicle). Based on the maximum lateral acceleration $a_{\text{d,max}}$, the required time t_{eva} to perform the evasive maneuver is computed as:

$$t_{\text{eva}} := \sqrt{2d_{\text{eva}}/a_{\text{d,max}}} + \delta_{\text{steer}}, \tag{5}$$

where δ_{steer} denotes the reaction time of the steering system. Using the dynamics of a double integrator system, we compute the traveled distance Δs_b of obstacle b during emergency braking with a deceleration of $-|a_{\text{s,max,b}}|$:

$$\begin{aligned}\Delta s_b &:= v_b t_b - \frac{1}{2}|a_{\text{s,max,b}}|t_b^2, \\ t_b &:= \min(t_{\text{eva}}, v_b/|a_{\text{s,max,b}}|).\end{aligned} \tag{6}$$

The evasive distance Δ_{eva} to the preceding obstacle b is obtained by [80, Eq. 12-13]:

$$\Delta_{\text{eva}}(v_{\text{ego}}, b) := v_{\text{ego}} t_{\text{eva}} - \Delta s_b. \tag{7}$$

Similar to safe distances, we are now able to define the set of states in which the ego vehicle respects the evasive distance to a preceding obstacle.

Proposition 2 (Evasive Distance Set)
The set of states respecting the evasive distance to a preceding obstacle $b \in \mathcal{B}$ is defined as $\mathcal{S}^2(t) = \{(s,d,v)^T \in \mathcal{X} \,|\, \forall (s_b, d_b)^T \in \mathcal{O}_{b,\text{cls}}(t) : s \leq s_b - \Delta^t_{\text{eva}}(v,b)\}$

Proof. The safety that evasive distances provide is shown in [80]. ■

Note that the set $\mathcal{S}^2(t)$ remains safe even if a preceding vehicle performs a lane change maneuver, since the utilized set-based prediction considers all feasible legal behaviors of other traffic participants (see scenario in Sec. VI-B).

Both sets, $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$, allow us to efficiently compute an under-approximation of $\mathcal{S}(t)$:

Proposition 3 (Under-Approximation of $\mathcal{S}$)
Given the set of states respecting safe distances $\mathcal{S}^1(t)$ and the set of states respecting evasive distances $\mathcal{S}^2(t)$, it holds that $\mathcal{S}^1(t) \cup \mathcal{S}^2(t) \subset \mathcal{S}(t)$.

Proof. The soundness has been shown in [68, Prop. 1]. ■

The algorithm to compute the under-approximation is given in [68, Alg. 1]. It should be noted that the set can also be computed for curved roads by considering the road curvature as shown in [68]; we omitted this for the sake of brevity.

V. Generation of Fail-safe Trajectories

To determine fail-safe trajectories with low computational effort, we make use of a convex approximation of the motion planning problem [83] by separating motions into longitudinal (cf. Sec. V-A) and lateral components (cf. Sec. V-B). In addition, we use linear vehicle models as well as linear state and input constraints to formulate convex linear-quadratic programs for each motion component [83, Sec. 4.4].

In Sec. V-A and V-B, we first introduce the optimization problems for the longitudinal and lateral component, respectively. The presented cost functions J in these sections are examples and can be modified to include other terms, e.g., separate costs for the final state of a trajectory or penalizing large inputs (cf. cost functions in [19]). Subsequently in Sec. V-C and V-D, we extract collision constraints from the predicted occupancy sets of obstacles (cf. Def. 2) and the computed invariably safe sets to generate fail-safe trajectories.

A. Longitudinal motion

We describe the state of the longitudinal motion as $x_{\text{lon}} = (s, v, a, j)^T$, where s is the longitudinal position, v is the velocity, a is the acceleration, and j is the jerk of the center point of the rear axle along a given reference path Γ (cf. Fig. 5). We choose the center of the rear axle as a reference point, since this model allows us to disregard the slip angle of the vehicle [49]. Moreover, we use the input $u_{\text{lon}}(t) = \dddot{a}(t)$ to generate smooth longitudinal trajectories, and describe the disturbance-free longitudinal motion of the vehicle by the linear time-invariant system:

$$\frac{d^4}{dt^4} s(t) = u_{\text{lon}}(t). \tag{8}$$

Since obstacles in the environment may restrict the feasible positions along the reference path Γ, we add the following time-variant collision constraint:

$$s_{\min}(t) \leq x^{(0)}_{\text{lon}}(t) \leq s_{\max}(t). \tag{9}$$

Furthermore, we apply the following time-invariant state constraints to ensure that trajectories are feasible:

$$\begin{aligned} v_{\min} &\leq x^{(1)}_{\text{lon}}(t) \leq v_{\max}, \\ a_{\min} &\leq x^{(2)}_{\text{lon}}(t) \leq a_{\max}, \\ j_{\min} &\leq x^{(3)}_{\text{lon}}(t) \leq j_{\max}, \end{aligned} \tag{10}$$

where $v_{\min}/v_{\max}$ represent the minimum and maximum velocity, $a_{\min}/a_{\max}$ the minimum and maximum acceleration, and $j_{\min}/j_{\max}$ the minimum and maximum jerk.

Acceleration profiles with partly constant acceleration phases enhance driving comfort for passengers by reducing maximum accelerations [84]. We model these constant acceleration phases by integrating slack variables [83, pp. 131-132] and a two-stage cost increase into our longitudinal optimization problem. Slack variables are used in optimization to loosen constraints. For the sake of clarity, we demonstrate the approach for the case of braking; however, the approach works analogously for positive accelerations. We introduce two additional deceleration limits, $a_{\mathrm{lim},1}$ and $a_{\mathrm{lim},2}$, with $a_{\min} < a_{\mathrm{lim},2} < a_{\mathrm{lim},1} < 0$. Furthermore, we define slack variables $\varsigma_{\mathrm{lon},1} \geq 0$ and $\varsigma_{\mathrm{lon},2} \geq 0$ and add the following time-invariant constraints to the longitudinal motion problem:

$$\begin{aligned} x_{\mathrm{lon}}^{(2)}(t) &\geq a_{\mathrm{lim},1} - \varsigma_{\mathrm{lon},1}, \\ x_{\mathrm{lon}}^{(2)}(t) &\geq a_{\mathrm{lim},2} - \varsigma_{\mathrm{lon},2}. \end{aligned} \tag{11}$$

Although the constraints (11) are soft, the constraints (10) still limit the maximum feasible deceleration. Thus, the introduced slack variables only affect the acceleration profile, but not the feasibility of the trajectory.

Note that the slack variables $\varsigma_{\mathrm{lon},1}$ and $\varsigma_{\mathrm{lon},2}$ become a part of the optimization vector (they can be appended to u_{lon}) and are determined during the optimization. By inducing linear costs for $\varsigma_{\mathrm{lon},1}$ and quadratic costs for $\varsigma_{\mathrm{lon},2}$, we can model constant acceleration phases, since the solver aims to minimize costs. Fig. 6 illustrates the resulting acceleration profiles. For instance, profiles with accelerations $a \leq a_{\mathrm{lim},1}$ are smoothed due to the linear costs. In the second stage, profiles with accelerations $a \leq a_{\mathrm{lim},2}$ are optimized as partly constant, since the quadratically increasing costs are minimized.

The quadratic cost function J_{lon} is chosen to favor comfortable longitudinal trajectories by punishing high accelerations and jerk with weights $w_a, w_j \in \mathbb{R}_+$, and the use of slack

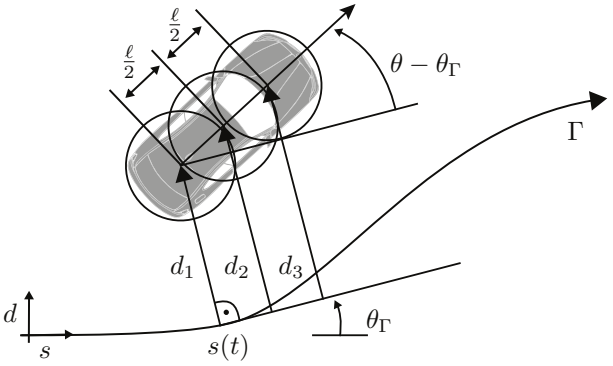


Fig. 5. Kinematic model with respect to a curvilinear coordinate system aligned to a reference Γ with orientation θ_Γ. The vehicle's pose is described by the longitudinal position s, the lateral deviation d, and the orientation θ.

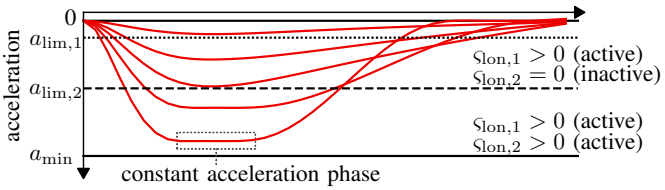


Fig. 6. Obtained acceleration profiles if slack variables are used to enhance comfort. Planned accelerations are penalized with costs in a two-stage approach: accelerations $a \geq a_{\mathrm{lim},1}$ induce linear costs and $a \geq a_{\mathrm{lim},2}$ induce quadratic costs, resulting in tub-shaped profiles.

variables with weights[2] $w_{\varsigma_1}, w_{\varsigma,2} \in \mathbb{R}_+$:

$$\begin{aligned} J_{\mathrm{lon}}\big(x(t), u(t)\big) = w_a x_{\mathrm{lon}}^{(2)}(t)^2 + w_j x_{\mathrm{lon}}^{(3)}(t)^2 \\ + w_{\varsigma_1}\varsigma_{\mathrm{lon},1} + w_{\varsigma_2}\varsigma_{\mathrm{lon},2}^2. \end{aligned} \tag{12}$$

B. Lateral motion

The lateral motion is described by the state vector $x_{\mathrm{lat}} = (d, \theta, \kappa, \dot{\kappa})^T$, where d is the lateral distance of the center of the rear axle to the reference path Γ, θ is the orientation, κ is the curvature, and $\dot{\kappa}$ is the change of curvature of the ego vehicle. Since the vehicle is supposed to move along the predefined reference path Γ, we can assume that the orientation difference $\Delta = \theta - \theta_\Gamma$ between the orientation of the vehicle θ and the orientation θ_Γ of the reference path is negligibly small. This assumption allows us to approximate the trigonometric functions as $\sin(\Delta) \approx \Delta$ and $\cos(\Delta) \approx 1$.

The computed longitudinal motion profile (cf. Sec. V-A) is used to determine the longitudinal positions $s(t)$ and the velocity $v(t)$ of the ego vehicle along reference path Γ. For collision avoidance, we limit the minimal and maximal lateral deviation of the vehicle from the reference path Γ at the given positions $s(t)$ (obtained in the previous optimization of the longitudinal trajectory). To compute the lateral deviation, we need to keep track of the orientation θ_Γ of the reference path. To not introduce a new state variable, we use the disturbance term $z_{\mathrm{lat}}(t) = \theta_\Gamma(s(t))$ instead [49]. The lateral motion of the vehicle with respect to the input $u_{\mathrm{lat}}(t) = \ddot{\kappa}(t)$ is given by the time-invariant linear system:

$$\dot{x}_{\mathrm{lat}} = \begin{pmatrix} 0 & v(t) & 0 & 0 \\ 0 & 0 & v(t) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} x_{\mathrm{lat}}(t) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_{\mathrm{lat}}(t) + \begin{pmatrix} -v(t) \\ 0 \\ 0 \\ 0 \end{pmatrix} z_{\mathrm{lat}}(t). \tag{13}$$

Please note that (13) qualifies as a linear system, since $v(t)$ is not a state variable for the lateral dynamics, but rather a time-variant parameter obtained from the longitudinal trajectory.

We approximate the shape of the ego vehicle using three circles with equal radius r (cf. Fig. 5) to model collision avoidance [85]. Without loss of generality, the centers of the first and third circles coincide with the rear and front axles, respectively. We denote the distance between the center points of the first and third circle as ℓ. The center of the second circle is positioned equidistantly between the other circles. By

[2] w_{ς_1} and $w_{\varsigma,2}$ must be carefully chosen to not distort the original solution of the unaltered optimization problem as described in [49].

considering the orientation of the ego vehicle and the reference path, the lateral distance d_i of the center of circle $i \in \{1,2,3\}$ to Γ is computed as:

$$d_i = d + \frac{i-1}{2}\ell \sin(\theta - \theta_\Gamma) \approx d + \frac{i-1}{2}\ell(\theta - \theta_\Gamma). \quad (14)$$

Using linear equation (14), we are able to constrain the lateral deviation of the ego vehicle in the lateral optimization problem with respect to the vehicle's orientation. We define the constrained values of the lateral motion as $x_{\mathrm{lat,constr}} = (d_1, d_2, d_3, \kappa, \dot{\kappa})^T$, computed as:

$$x_{\mathrm{lat,constr}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{2}\ell & 0 & 0 \\ 1 & \ell & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x_{\mathrm{lat}}(t) + \begin{pmatrix} 0 \\ -\frac{1}{2}\ell \\ -\ell \\ 0 \\ 0 \end{pmatrix} z_{\mathrm{lat}}(t). \quad (15)$$

Collision avoidance is incorporated by computing the minimum and maximum lateral displacements, $d_{i,\min}(t)$ and $d_{i,\max}(t)$, for each circle $i \in \{1,2,3\}$ along the reference path Γ. Together with physical constraints imposed by the steering system, we apply the following time-variant constraints to obtain drivable trajectories:

$$\underbrace{\begin{pmatrix} d_{1,\min}(t) \\ d_{2,\min}(t) \\ d_{3,\min}(t) \\ \kappa_{\mathrm{lim,min}}(t) \\ \dot{\kappa}_{\min}(t) \end{pmatrix}}_{x_{\min}(t)} \leq x_{\mathrm{lat,constr}}(t) \leq \underbrace{\begin{pmatrix} d_{1,\max}(t) \\ d_{2,\max}(t) \\ d_{3,\max}(t) \\ \kappa_{\mathrm{lim,max}}(t) \\ \dot{\kappa}_{\max}(t) \end{pmatrix}}_{x_{\max}(t)}, \quad (16)$$

where $d_{i,\min}/d_{i,\max}$ are the minimum and maximum allowed lateral deviation of circle $i \in \{1,2,3\}$, $\kappa_{\mathrm{lim,min}}/\kappa_{\mathrm{lim,max}}$ the minimum and maximum allowed curvature, and $\dot{\kappa}_{\min}/\dot{\kappa}_{\max}$ the minimum and maximum allowed change of the curvature, obtained from the technical specification of the steering system. To limit the maximum feasible curvature depending on the velocity of the ego vehicle, we use the friction circle [86] and set the lateral acceleration to $a_{\mathrm{lat}} = v(t)^2\kappa$. Solving for the curvature results in $\kappa = \sqrt{a_{\max}^2 - a(t)^2}/v(t)^2$, where $v(t)$ and $a(t)$ are obtained from the preplanned longitudinal motion. We consider steering actuator limitations by constraining the possible curvature range to $\kappa_{\min}$ (or $\kappa_{\max}$) by setting:

$$\begin{aligned} \kappa_{\mathrm{lim,min}}(t) &= \max\left(\frac{-\sqrt{a_{\max}^2 - a(t)^2}}{v(t)^2}, \kappa_{\min}\right), \\ \kappa_{\mathrm{lim,max}}(t) &= \min\left(\frac{\sqrt{a_{\max}^2 - a(t)^2}}{v(t)^2}, \kappa_{\max}\right). \end{aligned} \quad (17)$$

The quadratic cost function J_{lat} with weights $w_d, w_\theta, w_\kappa, w_{\dot{\kappa}} \in \mathbb{R}_+$ minimizes the lateral distance and orientation deviation to the reference path Γ and punishes high curvature rates to achieve comfortable lateral motions:

$$\begin{aligned} J_{\mathrm{lat}}\big(x(t), u(t)\big) = {} & w_d x_{\mathrm{lat}}^{(0)}(t)^2 + w_\theta\big(x_{\mathrm{lat}}^{(1)}(t) - \theta_\Gamma(t)\big)^2 \\ & + w_\kappa x_{\mathrm{lat}}^{(2)}(t)^2 + w_{\dot{\kappa}} x_{\mathrm{lat}}^{(3)}(t)^2. \end{aligned} \quad (18)$$

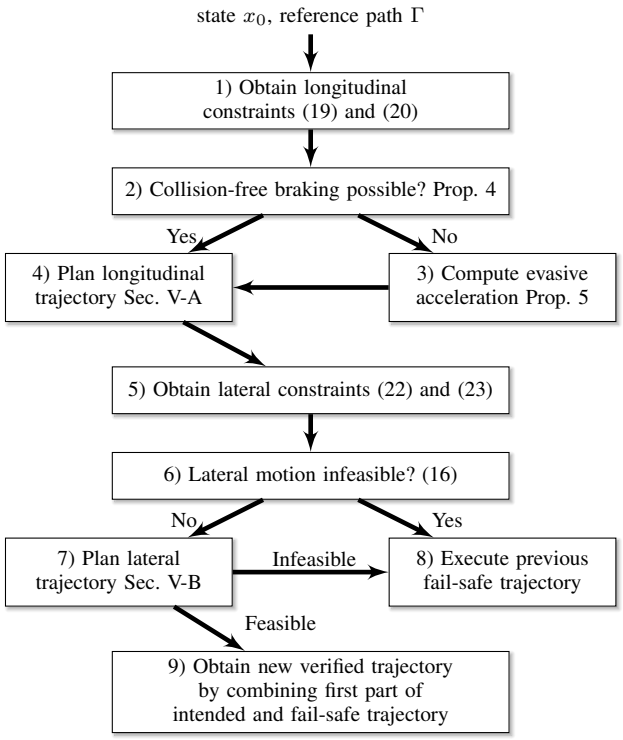


Fig. 7. Procedure for computing a fail-safe trajectory with a given initial state x_0 for the emergency trajectory and reference path Γ.

C. Fail-safe trajectory generation

Fig. 7 summarizes the necessary steps to generate fail-safe trajectories using the planners in Sec. V-A and Sec. V-B. We compute the initial state x_0 of the fail-safe trajectory along the given intended trajectory using the time-to-react (cf. Sec. II). We transform x_0 to curvilinear coordinates in our planner.

In Step 1 of Fig. 7, the collision constraints for the longitudinal motion of the fail-safe trajectory are extracted. We use $\mathcal{B}_{\mathrm{fol}} \subseteq \mathcal{B}$ and $\mathcal{B}_{\mathrm{pre}} \subseteq \mathcal{B}$ to denote the sets of following and preceding obstacles within the lane of the ego vehicle, respectively. Since we enlarge the occupancies with a circle (cf. Sec. IV) whose center coincides with the center of the ego vehicle, we need to transform the reference point of the longitudinal planning problem. Let us introduce Δ_{cor} as the correction term to transform the reference point of the ego vehicle on the rear axle to the center of its shape. Based on the longitudinal position of the ego vehicle s_0, the maximum longitudinal position constraints $s(t) \leq s_{\max}(t)$ (cf. (9)) are obtained as (Fig. 8 visualizes the constraint extraction):

$$\begin{aligned} s_{\max}(t) = \inf\big\{ s - \Delta_{\mathrm{cor}} \,|\, \forall b \in \mathcal{B}_{\mathrm{pre}} : s - \Delta_{\mathrm{cor}} > s_0 \\ \wedge (s,d)^T \in \mathcal{O}_{b,\mathrm{cls}}(t) \big\}. \end{aligned} \quad (19)$$

The minimum longitudinal position constraints $s(t) \geq s_{\min}(t)$ are obtained similarly as:

$$\begin{aligned} s_{\min}(t) = \sup\big\{ s - \Delta_{\mathrm{cor}} \,|\, \forall b \in \mathcal{B}_{\mathrm{fol}} : s - \Delta_{\mathrm{cor}} < s_0 \\ \wedge (s,d)^T \in \mathcal{O}_{b,\mathrm{cls}}(t) \big\}. \end{aligned} \quad (20)$$

It should be noted that (20) is only used if the ego vehicle changes lanes as described in [52]. For the current lane of the

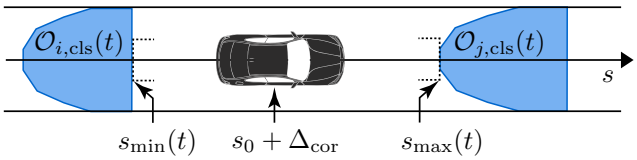


Fig. 8. Illustration of computing the longitudinal collision constraints, $s_{\min}$ and $s_{\max}$, in a lane for given occupancy sets $\mathcal{O}_{i,\mathrm{cls}}, i \in \mathcal{B}$, and $\mathcal{O}_{j,\mathrm{cls}}, j \in \mathcal{B}$.

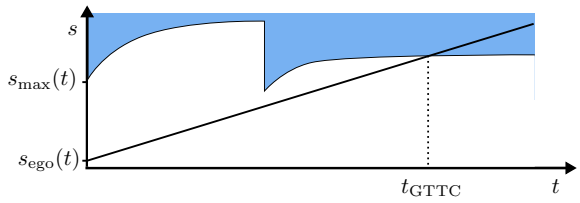


Fig. 9. Illustration of calculating the GTTC using the predicted position of the ego vehicle $s_{\mathrm{ego}}(t) = s_0 + v_0 t$ and the maximum longitudinal position constraints $s_{\max}(t)$.

ego vehicle, $s_{\min}(t)$ is omitted, since following vehicles need to keep a safe distance to the ego vehicle [78].

For fail-safe planning, we apply the following heuristics to decide whether to plan a braking or evasive maneuver. In Step 2, we check if a braking maneuver alone is sufficient for collision avoidance with any occupancy set, since braking maneuvers are often considered to be the most preferred evasive maneuvers for passengers in emergency situations [87]. Since the occupancy sets include information about the positions of the obstacles during emergency braking over time, we can use (19) for this check.

Proposition 4 (Collision Avoidance Through Braking)
A collision with obstacles can be avoided for the initial position s_0, velocity v_0, and reaction time δ_{brake} of the ego vehicle using emergency braking with $-|a_{\max}|$ if

$$\forall t \in [t_0, t_h] : s_0 + v_0(\tau) - \frac{1}{2}|a_{\max}| \max(\tau - \delta_{\mathrm{brake}}, 0)^2 \leq s_{\max}(t), \tau := \min(t, {}^{v_0}/_{|a_{\max}|} + \delta_{\mathrm{brake}}).$$

Proof. Using the longitudinal dynamics and applying maximum feasible deceleration $a_{\max}$, we compute the future positions of the ego vehicle over time by: $s_0 + v_0(\tau) - \frac{1}{2}|a_{\max}| \max(\tau - \delta_{\mathrm{brake}}, 0)^2$, where $\tau := \min(t, {}^{v_0}/_{|a_{\max}|} + \delta_{\mathrm{brake}})$ avoids driving backwards. If the ego vehicle does not occupy positions $s > s_{\max}(t)$, it can avoid a collision by braking. ■

If the ego vehicle is able to avoid potential collisions with a braking maneuver alone, we compute the longitudinal motion using the longitudinal planner described in Sec. V-A. It should be noted that this approach also works with crossing traffic.

Otherwise, the ego vehicle may avoid collisions by swerving to an adjacent lane using an evasive maneuver. For these situations, we must ensure that the required maximum lateral acceleration a_{eva} for evading is feasible throughout the planned maneuver despite the decoupled longitudinal and lateral dynamics of the vehicle. In the worst case, the evasive maneuver no longer allows braking, since $|a_{\max}| = |a_{\mathrm{eva}}|$. Therefore, let us first introduce the *guaranteed time-to-collision* as the time until the ego vehicle intersects with occupancy sets, encoded in the maximum allowed position $s_{\max}(t)$, when driving with constant velocity (cf. Fig. 9).

Definition 7 (Guaranteed Time-To-Collision)
Assuming a collision is possible, the guaranteed time-to-collision (GTTC) with respect to the initial longitudinal position s_0 and velocity v_0 of the vehicle and the maximum allowed position $s_{\max}(t), t \in [0, t_h]$ is defined as

$$t_{\mathrm{GTTC}} := \operatorname*{argmin}_{t \in [0, t_h]} \left| (s_0 + v_0 t) - s_{\max}(t) \right|.$$

We further introduce the duration of the evasive maneuver as t_{GTTC}, assuming no deceleration of the ego vehicle (i.e., the worst case $|a_{\max}| = |a_{\mathrm{eva}}|$), and the lateral distance to fully reach an adjacent lane as $d_{\mathrm{eva}} > 0$. Finally, we are able to compute the required lateral acceleration during the evasive maneuver.

Proposition 5 (Evasive Acceleration)
The required lateral acceleration a_{eva} of an evasive maneuver with initial lateral velocity $v_{\mathrm{lat}} \geq 0$ over the lateral distance d_{eva} with duration t_{GTTC} and reaction time for steering $\delta_{\mathrm{steer}} < t_{\mathrm{GTTC}}$ is obtained as:

$$a_{\mathrm{eva}} = \frac{2\big(d_{\mathrm{eva}} - |v_{\mathrm{lat}}| t_{\mathrm{GTTC}}\big)}{(t_{\mathrm{GTTC}} - \delta_{\mathrm{steer}})^2}.$$

Proof. The soundness has been shown in [80, III-A]. ■

Based on the maximum possible acceleration $|a_{\max}|$, the maximum allowed longitudinal acceleration is:

$$a_{\mathrm{lon}} = \sqrt{a_{\max}^2 - a_{\mathrm{eva}}^2}, \tag{21}$$

which is ensured in the longitudinal optimization problem by adding this limit as a constraint to the longitudinal optimization problem.

In Step 5 of Fig. 7, the constraints on the lateral motion are computed. Therefore, we first predict the poses of the ego vehicle along Γ considering the planned longitudinal motion and the orientation $\theta(s(t)) = \theta_\Gamma(s(t))$. As described in Sec. V-B, we approximate the shape of the vehicle with three circles. For each of these circles, we compute the minimum and maximum lateral deviation from Γ under the constraint that no collisions with occupancies occur. Let $\mathrm{circ}_i(d, t)$ denote the occupancy of circle $i \in \{1, 2, 3\}$, which is shifted by d along the normal direction (note the sign of d) considering the ego pose at time t along Γ. The maximum lateral offset constraints are:

$$d_{i,\max}(t) = \sup\{d \geq 0 \,|\, \mathrm{circ}_i(d, t) \cap \mathcal{O}_{\mathcal{B}}(t) = \emptyset\}. \tag{22}$$

The minimum lateral offset constraints $d_{i,\min}(t)$ are obtained similarly for negative values of d:

$$d_{i,\min}(t) = \inf\{d \leq 0 \,|\, \mathrm{circ}_i(d, t) \cap \mathcal{O}_{\mathcal{B}}(t) = \emptyset\}. \tag{23}$$

Fig. 10 illustrates the computation of the lateral constraints for each circle for two consecutive time steps t_1 and t_2. Note

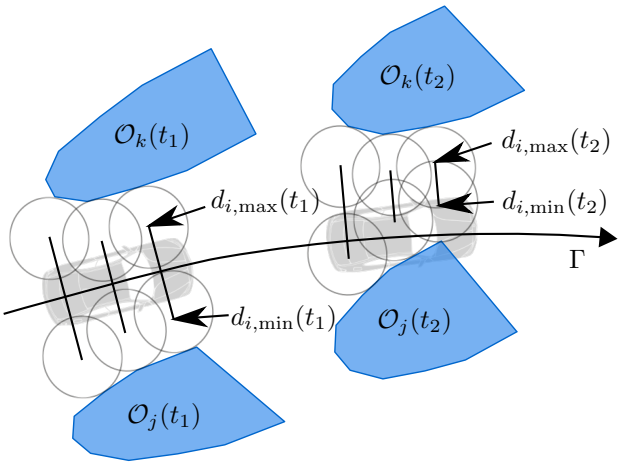


Fig. 10. Illustration of computing the lateral collision constraints, $d_{i,\min}$ and $d_{i,\max}$, of each circle i for example time steps, t_1 and t_2. The pose along the reference path Γ is predicted according to the planned longitudinal motion. The constraints are obtained by shifting each circle i along predefined passing directions so that they do not collide with occupancy sets of obstacles, e.g., $\mathcal{O}_k$ and $\mathcal{O}_j$.

that if a circle initially intersects with an occupancy set for $d = 0$, the circle must be shifted to determine whether the ego vehicle should pass left or right. For instance, the circles for the minimum position constraints at time step t_2 in Fig. 10 are shifted in positive d-direction to pass occupancy $\mathcal{O}_j(t_2)$ on the left. The passing side can be decided with reachability analysis, for example [64], and is not the focus of this work.

In Step 6, we perform a pre-solve check of the lateral optimization problem by evaluating whether the condition $\exists t \in [0, t_h] : d_{\min}(t) > d_{\max}(t)$ holds (cf. lateral position constraints in Sec. V-B). If the condition proves true, there is no longer a feasible solution, since the position constraint (16) in the lateral planner has been violated. In this case, we directly switch to the previously computed fail-safe trajectory, which is still valid (cf. Fig. 4). However, if the evasive maneuver option is feasible, we plan the lateral motion of the ego vehicle as described in Sec. V-B and obtain the new valid fail-safe trajectory.

D. Invariably safe set constraints

Our fail-safe trajectory planner only accepts linear constraints. For this reason, we present how the under-approximation of $\mathcal{S}(t)$ can be transformed to sets of linear constraints for the convex optimization problems (cf. Sec. V-A and V-B).

The integration of evasive distance constraints is simpler than safe distances, since (7) is already in linear form. Evasive distances are added to the longitudinal optimization problem with the constraint:

$$x_{\text{lon}}^{(0)}(t) + \Delta_{\text{eva}}(x_{\text{lon}}^{(1)}(t), b) \leq s_{\max,b}(t), \tag{24}$$

where $s_{\max,b}(t)$ is the maximum position constraint with respect to the obstacle b. For the preceding obstacle $b \in \mathcal{B}$ in the lane (or target lane) of the ego vehicle, we add (24) to the longitudinal optimization problem, resulting in one additional terminal constraint in the longitudinal optimization problem.

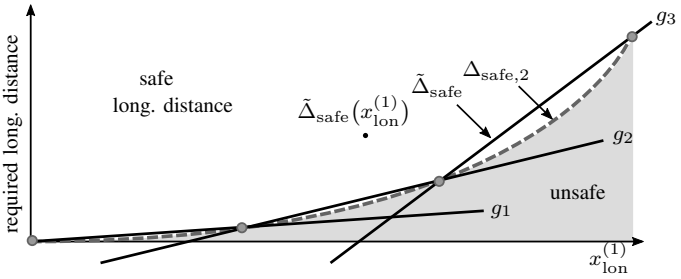


Fig. 11. Piecewise linear approximation of safe distances. The ego vehicle must maintain the longitudinal safe distance $\Delta_{\text{safe},2}$ to preceding obstacles to remain safe (white area). This convex safe distance $\Delta_{\text{safe},2}$ can be approximated by a linear piecewise function $\tilde{\Delta}_{\text{safe}}$, composed of h linear functions $g_i, i \in \{1, \dots, h\}$. A safe point $\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)})$ which respects the linear safe distance also fulfills $\forall i \in \{1, \dots, h\} : \tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) \geq g_i(x_{\text{lon}}^{(1)})$.

Safe distances cannot be directly included in linear-quadratic programs, since they are quadratic in the velocity of the ego vehicle. To circumvent this problem, we exploit the convexity of the safe distance functions and use a piecewise linear approximation of the safe distance instead. The resulting approximation is over-approximative and therefore still ensures safety.

We use h linear functions $g_1, g_2, \dots, g_h : \mathbb{R} \to \mathbb{R}$ to approximate the safe distance $\Delta_{\text{safe}} \in \{\Delta_{\text{safe},1}, \Delta_{\text{safe},2}\}$. To obtain the linear functions g_i, we divide the velocity range $[v_{\min}, v_{\max}], 0 \leq v_{\min} < v_{\max}$, of the ego vehicle into h equally large intervals $[v_i, v_{i+1}], i \in \{0, \dots, h-1\}$.

For the sake of brevity, we demonstrate the linearization with $\Delta_{\text{safe},2}$ in the following paragraphs. The linearization of $\Delta_{\text{safe},1}$ is done similarly. For each interval, we approximate the safe distance $\Delta_{\text{safe},2}$ using linear functions g_i, resulting in the linear safe distance formulation:

$$\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) = \begin{cases} g_1(x_{\text{lon}}^{(1)}), & v_0 \leq x_{\text{lon}}^{(1)} < v_1, \\ g_2(x_{\text{lon}}^{(1)}), & v_1 \leq x_{\text{lon}}^{(1)} < v_2, \\ \vdots & \\ g_p(x_{\text{lon}}^{(1)}), & x_{\text{lon}}^{(1)} \geq v_{h-1}. \end{cases} \tag{25}$$

Fig. 11 illustrates the piecewise linear approximation of the safe distance. The ego vehicle is not allowed to enter the shaded region in order to guarantee safety.

To integrate the h linear functions into the optimization problem, we make use of the fact that each convex, piecewise linear function can be represented as a maximum function [88]. Thus, the safe distance can be reformulated as:

$$\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) = \max\left(g_1(x_{\text{lon}}^{(1)}), g_2(x_{\text{lon}}^{(1)}), \dots, g_h(x_{\text{lon}}^{(1)})\right).$$

Respecting the maximum of these h linear functions is equivalent to satisfying every single one of them due to convexity (cf. example point $\tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)})$ in Fig. 11).

We integrate the safe distance equation $\tilde{\Delta}_{\text{safe}}$ (cf. (4)) into the longitudinal position constraint (9) to obtain:

$$x_{\text{lon}}^{(0)}(t) + \tilde{\Delta}_{\text{safe}}(x_{\text{lon}}^{(1)}) \leq s_{\max}(t). \tag{26}$$

TABLE I
GENERAL PARAMETERS AND DATA OF THE DRIVING EXPERIMENTS.

Description	Parameter with value
Velocity range	$v_{\text{ego}} \in [0\,\text{m/s}, 15\,\text{m/s}]$
Desired velocity	$v_{\text{des}} = 13.9\,\text{m/s}$
Lon. acceleration range	$a_{\text{ego,lon}} \in [-4.0\,\text{m/s}^2, 2.0\,\text{m/s}^2]$
Lat. acceleration range	$a_{\text{ego,lat}} \in [-8.0\,\text{m/s}^2, 8.0\,\text{m/s}^2]$
Jerk range	$j_{\text{ego}} \in [-10\,\text{m/s}^3, 10\,\text{m/s}^3]$
Curvature range	$\kappa_{\text{ego}} \in [-0.2/\text{m}, 0.2/\text{m}]$
Curvature change range	$\dot{\kappa}_{\text{ego}} \in [-0.2/\text{m}, 0.2/\text{m}]$
Dimensions ego vehicle	length= 5.238 m, width= 2.169 m
Circle approx. ego vehicle	$\ell = 3.5\,\text{m}, r = 1.4\,\text{m}$
Reaction time braking	$\delta_{\text{brake}} = 0.3\,\text{s}$
Reaction time steering	$\delta_{\text{steer}} = 0.3\,\text{s}$
Time step size	$\Delta t = 0.25\,\text{s}$
Lane width	width= 3.5 m
Evasive distance	$d_{\text{eva}} = 3.5\,\text{m}$
SPOT parameters	$a_{\text{max,veh}} = 5\,\text{m/s}^2$, $v_{\text{max,veh}} = 13.9\,\text{m/s}$, $f_S = 1.2$ $a_{\text{max,ped}} = 0.6\,\text{m/s}^2$, $a_{\text{max,ped,stop}} = 0.6\,\text{m/s}^2$, $v_{\text{max,ped}} = 2\,\text{m/s}$, $d_{\text{perp}} = 1.5\,\text{m}$ $b_{\text{cross}} = b_{\text{stop}} = \text{False}$
Weights in J_{lon}	$w_a = 1, w_j = 2, w_{\varsigma_1} = 5, w_{\varsigma_2} = 10$
Weights in J_{lat}	$w_d = 0.2, w_\theta = 2, w_\kappa = 20, w_{\dot{\kappa}} = 20$
Time for verification Sce. A	13 ms
Time for verification Sce. B	26 ms
Time for verification Sce. C	23 ms

In case the utilized solver of the optimization problem cannot handle constraints with a maximum function, one can also add h linear constraints in the form of:

$$x_{\text{lon}}^{(0)}(t) + \left(g_i(x_{\text{lon}}^{(1)}) + \delta_{\text{brake}} x_{\text{lon}}^{(1)}\right) \leq s_{\max}(t). \qquad (27)$$

It should be noted that larger numbers of linear functions h decrease the approximation error, but increase the computational time of solving the optimization problem.

VI. EXPERIMENTAL RESULTS

To demonstrate the drivability of fail-safe trajectories and the proposed safety benefits, we have implemented our online verification technique in Python and C++ (for computational efficiency) for the use in a real test-vehicle. The computer in the vehicle is equipped with an Intel i7-6900k processor and 64GB of memory. The frequency of the processor is underclocked from 3.2GHz to 1.2GHz to improve energy consumption and heat management. We use discrete time versions of our planners and the convex optimization packages CVXPY [89], ECOS [90] and CVXPY-CODEGEN to generate embedded code. Sec. VI-A to VI-C present selected results of our 127 conducted driving experiments. Since we are only able to consider simpler traffic scenarios in the driving experiment, we have also validated our verification technique by post-processing recorded scenarios with dense traffic in Sec. VI-D. Videos of our experiments can be found in the supplementary files of this paper. The illustrated scenarios are included in the CommonRoad benchmark suite (Version 2018b) for reproducibility [19]. The parameters of the planners and the set-based prediction tool SPOT [71] are summarized in Tab. I.

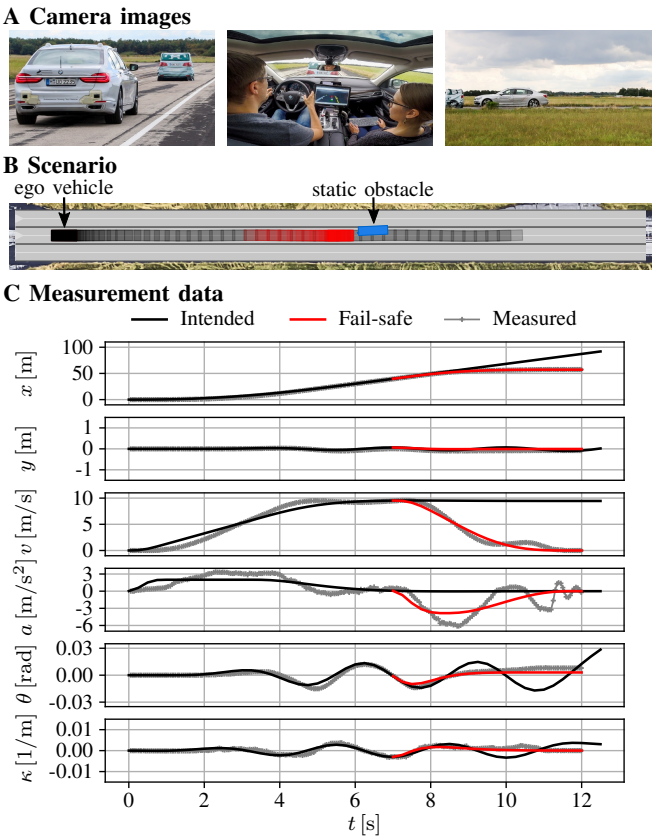


Fig. 12. Braking maneuver to avoid collisions with a static obstacle (ZAM_Urban-2_1). (A) Camera images of the experiment. (B) The planned trajectories and the occupancy set of the static obstacle. (C) The measured data from this experiment.

A. Avoiding collisions with static obstacles

In our first experiment, we show that the proposed fail-safe motion planning technique is designed to ensure safety for any given intended trajectory. This property is especially important when the intended motion planner of the ego vehicle is changed or machine learning techniques are employed. To demonstrate this property under extreme conditions, we have created a malicious intended trajectory planner, which tries to reach random desired velocities, and performs oscillating lateral motions with random frequency and amplitude. Nevertheless, all generated trajectories are kinematically feasible.

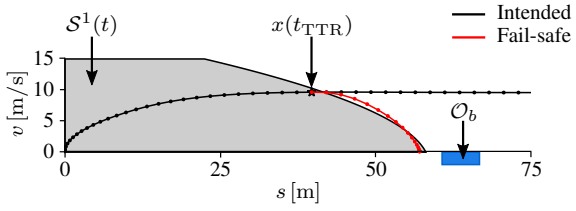


Fig. 13. Invariably safe set of the scenario in Fig. 12. The computed fail-safe trajectory starts at the state $x(t_{\text{TTR}})$, which is the last state along the intended trajectory that is still enclosed in the invariably safe set $\mathcal{S}^1(t)$. The set is shown as a projection onto the s-v plane.

We consider a two-lane scenario (with an additional shoulder), in which we have randomly placed a foam vehicle dummy in the lane of the ego vehicle. Fig. 12 shows the results of an intended trajectory (black occupancies) and computed fail-safe trajectory (red occupancies), which avoids a collision by braking. When applying the intended trajectory, the ego vehicle accelerates to a velocity of about $10\,\mathrm{m/s}$ without reacting to the static obstacle. Our verification technique automatically computes the safe part of the malicious trajectory and a subsequent fail-safe trajectory. This fail-safe trajectory with a horizon of $t_{\mathrm{fs}} = 5\,\mathrm{s}$ starts at the time-to-react of $t_{\mathrm{TTR}} = 7\,\mathrm{s}$. By automatically executing this trajectory, the ego vehicle avoids a collision and comes to a standstill directly in front of the static obstacle.

Fig. 13 illustrates the invariably safe set of the scenario as a projection onto the s-v plane. Since this scenario is static, the resulting invariably safe sets are also time-invariant. The fail-safe trajectory starts at the last state of the intended trajectory, which is still enclosed in $\mathcal{S}^1(t)$ (light gray set in Fig. 13). The set $\mathcal{S}^1(t)$ has been computed with a reaction time of $\delta_{\mathrm{brake}} = 0.3\,\mathrm{s}$ to indicate when fail-safe trajectories need to start.

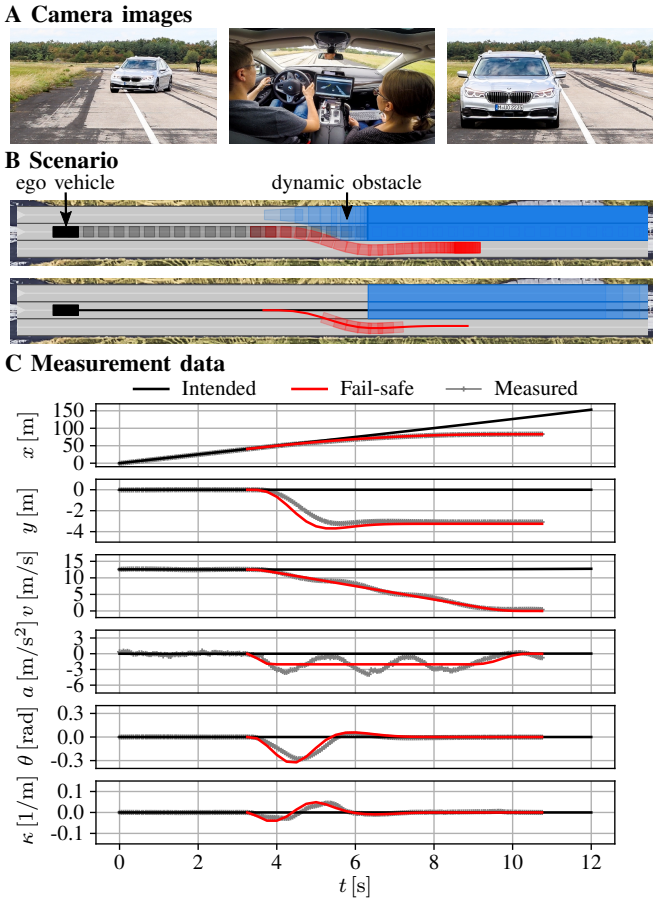
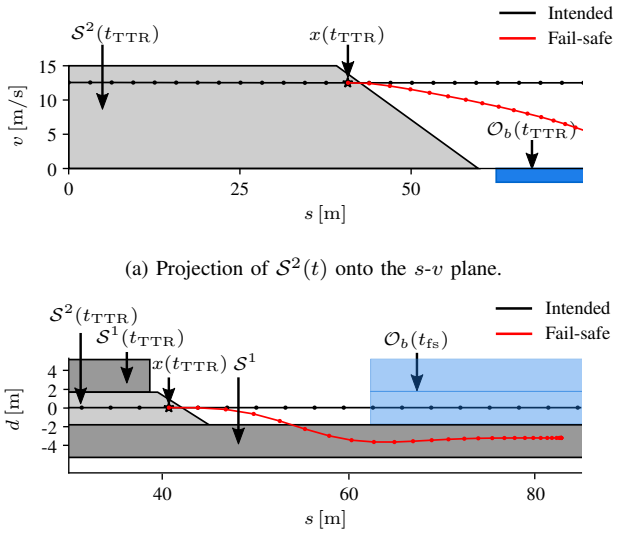


Fig. 14. Avoiding collisions with a vehicle that has cut in by swerving to the adjacent shoulder (ZAM_Urban-7_1_S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the dynamic obstacle over the whole time horizon and a selected interval. (C) The measured data from this experiment.



(a) Projection of $\mathcal{S}^2(t)$ onto the s-v plane.

(b) Projection of $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$ onto the s-d plane.

Fig. 15. Invariably safe set of the scenario in Fig. 14. (a) The computed safe set $\mathcal{S}^2(t)$ is shown as a projection onto the s-v plane. (b) The fail-safe trajectory starts in $\mathcal{S}^2(t)$ and ends in $\mathcal{S}^1(t)$, shown as projections onto the s-d plane.

B. Reacting to cut-in vehicle

In the following experiment, we consider a dynamic environment in which a vehicle in an adjacent lane cuts into the lane of the ego vehicle and then performs emergency braking. The simulated dynamic vehicle is randomly placed in the adjacent left lane with an initial velocity of $v = 13.89\,\mathrm{m/s}$ in the environment model of the ego vehicle.

Fig. 14 illustrates the results of the experiment. The ego vehicle is travelling at a constant velocity of $v = 12.56\,\mathrm{m/s}$ and the initial distance between the ego vehicle and the other vehicle is approximately $45\,\mathrm{m}$. The time-to-react is computed as $t_{\mathrm{TTR}} = 3.25\,\mathrm{s}$. The generated fail-safe trajectory lets the ego vehicle swerve to the adjacent shoulder lane to avoid colliding with the vehicle that has cut the ego vehicle off. The maximum lateral acceleration during the evasive maneuver is measured at $4.1\,\mathrm{m/s^2}$. Fig. 14B shows the top view of the scenario for the entire time horizon and selected time steps $t \in [4.5, 5.75]$.

The computed invariably safe sets are shown in Fig. 15 in two different projections. Fig. 15a visualizes $\mathcal{S}^2(t)$ for the time step t_{TTR} in the s-v plane together with the intended and fail-safe trajectory. The s-d plane projections of the invariably safe sets $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$ are presented in Fig. 15b for the time step t_{TTR} and velocity slice $v(t_{\mathrm{TTR}}) = 12.51\,\mathrm{m/s}$. The computed fail-safe trajectory starts in $\mathcal{S}^2(t)$ and ends in $\mathcal{S}^1(t)$ of the shoulder lane.

In this scenario, an evasive maneuver can be executed at a later point in time compared to a braking maneuver. We illustrate this fact by making use of the computed invariably safe sets. As a reference, Fig. 15b also illustrates $\mathcal{S}^1(t)$ for the adjacent left lane at t_{TTR}. Here, the set $\mathcal{S}^1(t)$ has the same size as for the lane of the ego vehicle, because the minimum longitudinal positions of vehicle b in the occupancy

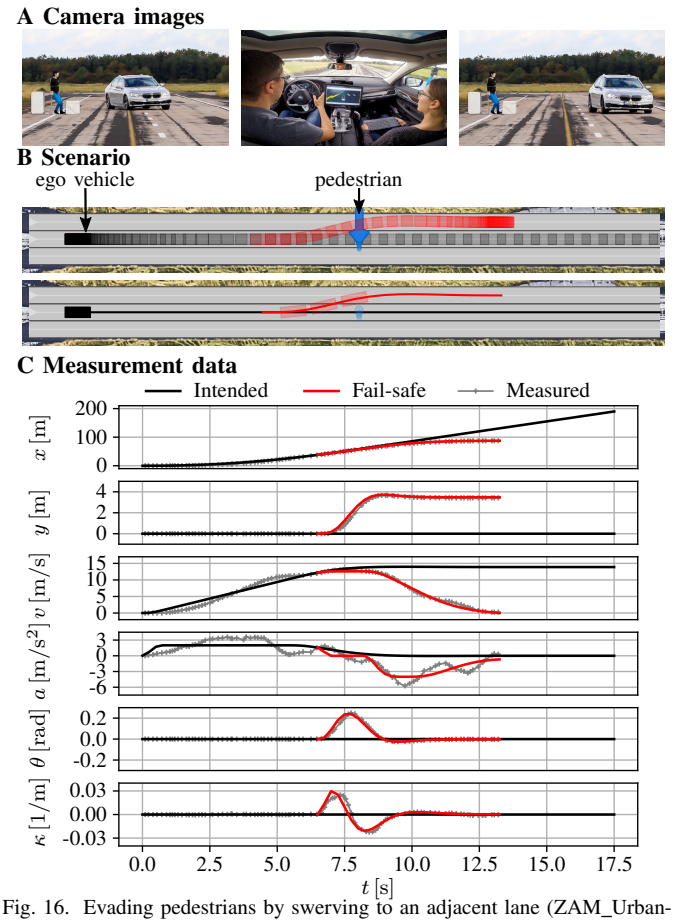


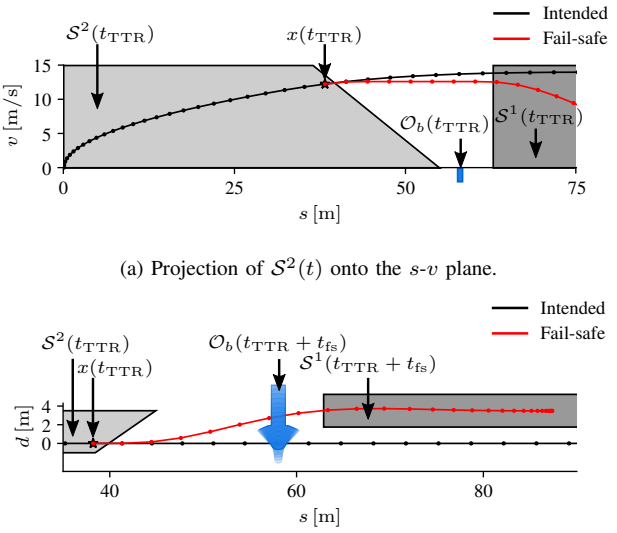
Fig. 16. Evading pedestrians by swerving to an adjacent lane (ZAM_Urban-5_1_S-1). (A) Camera images of the experiment. (B) The planned trajectories and the predicted occupancy set of the pedestrian over the whole time horizon and a selected time steps $t \in \{0.5\,\mathrm{s}, 1.0\,\mathrm{s}, 1.5\,\mathrm{s}\}$. (C) The measured data from this experiment.

set $\mathcal{O}_b(t_{\mathrm{TTR}})$ are equal. Since $\mathcal{S}^2(t)$ is larger than $\mathcal{S}^1(t)$ on this lane and it encloses a state at a later point in time, evading can be performed one step later than braking.

C. Avoiding collisions with jaywalking pedestrians

The last driving experiment highlights how our verification technique copes with pedestrians that suddenly enter the lane of the ego vehicle. We have placed a foam pedestrian close to the right border of the lane of the ego vehicle. The on-board sensors of the ego vehicle detect the pedestrian and the set-based prediction computes the set of future behaviors based on simulated initial dynamics of the pedestrian that we choose for creating critical situations. In particular, we assume that the pedestrian does not react to the oncoming ego vehicle and thus continues crossing the lane.

In our scenario (cf. Fig. 16), the intended trajectory accelerates the ego vehicle to the desired velocity of $13.9\,\mathrm{m/s}$. The pedestrian enters the lane of the ego vehicle with a velocity of $v_{\mathrm{ped}} = 1.5\,\mathrm{m/s}$. In our simulation, we set the time when the pedestrian enters the lane to the time-to-react of $t_{\mathrm{TTR}} = 6.5\,\mathrm{s}$. This choice enables us to enforce an evasive instead of a braking maneuver for demonstration; otherwise, the pedestrian will already be blocking the lane as the ego vehicle approaches. The computed fail-safe trajectory lets the



(a) Projection of $\mathcal{S}^2(t)$ onto the s-v plane.

(b) Projection of $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$ onto the s-d plane.

Fig. 17. Invariably safe set of the scenario in Fig. 16. (a) The computed safe set $\mathcal{S}^2(t)$ is shown as a projection onto the s-v plane. (b) The fail-safe trajectory starts in $\mathcal{S}^2(t)$ and ends in $\mathcal{S}^1(t)$, shown as projections onto the s-d plane.

ego vehicle swerve into the adjacent left lane with a velocity of $v(t_{\mathrm{TTR}}) = 12.22\,\mathrm{m/s}$. After fully entering the adjacent lane and passing the pedestrian, the ego vehicle performs a braking maneuver to come to a standstill. During this experiment, we measured a maximum lateral acceleration of $4.8\,\mathrm{m/s^2}$, which is the highest among all of our experiments.

Fig. 17 illustrates the computed invariably safe sets in two different projections. The invariably safe sets $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$ are visualized in Fig. 17a as a projection onto the s-v plane. Similar to previous pedestrian scenario, the fail-safe trajectory starts in $\mathcal{S}^2(t)$ and lets the ego vehicle swerve to the left adjacent lane. As soon as the fail-safe trajectory enters $\mathcal{S}^1(t)$ from the adjacent lane, the ego vehicle initiates a braking maneuver to safely stop. Both sets, $\mathcal{S}^1(t)$ and $\mathcal{S}^2(t)$, are visualized in Fig. 17b as a projection onto the s-d plane.

D. Intervention assessment in urban traffic

In this section, we assess the intervention rate of our verification technique in typical urban traffic situations. We recorded urban scenarios with dense traffic. Our approach is used to verify the safety of the current control input of the human driver. Thus, the intended trajectory of the vehicle corresponds to the currently chosen input of the human driver. Due to safety reasons, we postprocess the data after the test drives.

Fig. 18 shows the $17\,\mathrm{km}$ long route of the driving experiment, which covers different urban (speed limit of $8.3\,\mathrm{m/s}$) and country road situations (speed limit of $27.8\,\mathrm{m/s}$). For most of the roads along this route, the human driver has the right of way. We conduct four test drives (two in each direction) with a BMW 7-series test vehicle on Wednesday, 13 March 2019, from 1:30PM until 5PM (usual afternoon commuter traffic). Each drive takes $23\,\mathrm{min}$ on average, which implies a mean

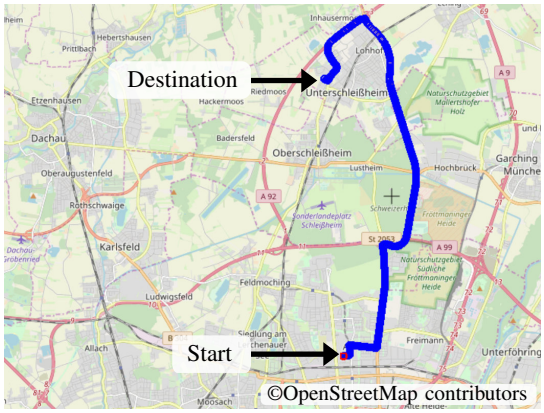


Fig. 18. The intervention assessment study was conducted on the 17 km long route between the BMW Autonomous Driving Campus (ADC) in Unterschleißheim and the BMW Research and Innovation Center (FIZ) in Munich.

velocity of approximately $12.32\,\mathrm{m/s}$. We sample different traffic densities for our study. The planning horizon of our fail-safe planner is set to $t_{\mathrm{fs}} = 5\,\mathrm{s}$ with a step size of $\Delta t = 0.25\,\mathrm{s}$. The average computation times of the prediction and the fail-safe planner are $20.1\,\mathrm{ms}$ and $16.1\,\mathrm{ms}$ per call, respectively.

Since we are interested in the intervention rate of the safety layer, we present the results of the test drive with the most executions of fail-safe trajectories. In total, $N_{\mathrm{attempt}} = 6,157$ verification attempts are performed during this test drive. Among these attempts, $N_{\mathrm{N}} = 6056$ situations ($98.36\,\%$) are verified as safe by successfully computing a fail-safe trajectory (example shown in Fig. 1 and 19a). In only $N_{\mathrm{P}} = 101$ cases ($1.64\,\%$) is the current traffic scenario not verified and the ego vehicle has to execute a fail-safe trajectory. We investigate each verification attempt manually in detail.

Tab. II summarizes the analysis results of the alleged fail-safe trajectory executions. Half of the fail-safe trajectory executions amount to true positives, i.e., the situation is in fact unsafe and the execution of a fail-safe trajectory would improve safety. Most fail-safe trajectory executions are caused by the driver violating the safe distance to preceding vehicles ($55.3\,\%$). The second major reason for unsafe situations is the high degree of uncertainty in the environment model ($38.3\,\%$). Nevertheless, even in uncertain scenarios, verification techniques need to account for these uncertainties to prevent collisions. The last reason for justified fail-safe executions is a situation in which a pedestrian suddenly enters the road ($6.4\,\%$).

Considering the 0.87% of unjustified fail-safe trajectory executions (cf. Tab. II), the majority of false positives (i.e., the situation is actually safe) amount to unmodeled priority rules in the environment model which cannot be used in the set-based prediction ($61.1\,\%$). More specifically, right-of-way rules are not yet included. For instance, Fig. 19b illustrates a situation from the test drive in which the ego vehicle begins to turn right; however, an oncoming vehicle is also allowed to turn left. Another major cause for fail-safe executions lies in the utilized solver ($25.9\,\%$), which sometimes fails to obtain fail-safe trajectories. The errors are likely caused by numerical instabilities of the solver and the Python interface, which processes the data from the C++ implementation of the solver. The conversion of the matrices and constraints from Python to embedded code might result in inaccuracies. Lastly, $13\,\%$ of false positives are caused in one traffic situation in which a preceding vehicle enters a parking area, leaving the map area. In this situation, the set-based prediction cannot map the traffic participant to a certain road and thus assumes that it is allowed to drive anywhere.

Note that even though the ego vehicle has to execute a fail-safe trajectory, this trajectory does not necessarily need to be fully executed. In our experiment, the duration of the longest unjustified execution of a fail-safe trajectory is $1.75\,\mathrm{s}$, which would result in slowing down the vehicle by only 30%. In addition, the vehicle quickly recovered after this situation, since we were able to verify an intended trajectory again. Thus, this situation would not be immediately recognized by passengers in the vehicle.

VII. Conclusions

This paper proposes fail-safe motion planning to ensure that autonomous vehicles never cause accidents under the premise that other traffic participants are allowed to perform any legal behavior. If a collision occurs nonetheless, we can verifiably argue that another traffic participant has violated traffic rules.

In contrast to exisiting verification techniques, our approach is the first verification technique that can be used in arbitrary traffic situations. It is real-time capable with computation times of less than $40\,\mathrm{ms}$ and works with any provided intended motion plan, even if it has been generated using machine-learning techniques. Furthermore, our technique ensures that the vehicle always has a provably safe plan to follow even if the intended motion is potentially unsafe. By making use of invariably safe sets, we are able to ensure that fail-safe trajectories intervene at the optimal point in time in unsafe sitations and guarantee safety for an infinite time horizon.

In 127 driving experiments with a BMW 7-series test-vehicle, we showed for the first time that the safety benefits of our verification technique prove true in reality. In all experiments, the ego vehicle remained safe through the execution of fail-safe trajectories. Our fail-safe planner generates drivable trajectories that can be tracked by a vehicle controller, even in highly dynamic stuations, validating our linearized kinematic model. Invariably safe sets further allow us to efficiently determine the existence of fail-safe trajectories and to ensure that these trajectories remain safe at all times.

In recorded scenarios with dense traffic in the Munich area, we performed the first detailed intervention assessment study for online verification of autonomous vehicles. The results of our study indicate that our verification technique has low intervention rates. Even if the ego vehicle has to execute a fail-safe trajectory, it can recover to its intended motion when the situation resolves itself. Consequently, employing our technique does not result in overly conservative behaviors of the autonomous vehicle.

Our online verification technique has the potential to drastically reduce the number of traffic accidents. However, to

TABLE II
ANALYSIS RESULTS OF ALLEGED FAIL-SAFE EXECUTIONS.

Type	Reason	Number	Comment
TP	Safe distance	26	The driver violated the safe distance to preceding vehicles.
	Pedestrian	3	A pedestrian suddenly entered the ego vehicle's lane.
	Uncertainties	18	High uncertainties in the environment model led to rejecting intended trajectories.
FP	Solver error	14	The solver failed to obtain a fail-safe trajectory even though the situation was safe. This error might be a result of inaccuracies in the embedded code generation.
	Map information	7	A vehicle entered a parking area and left current map area. In these situations, the set-based prediction cannot consider lanes and driving directions anymore.
	Unmodeled priority rules	33	Right-of-way rules are not included in the environment model. As a result, the set-based prediction predicts that other vehicles turn in front of the ego vehicle.

(a) Verified traffic situation

(b) Unsafe traffic situation

Fig. 19. Snapshots of intervention study. (a) The verification successfully computed a fail-safe trajectory (red regions). (b) Priority traffic rules are not yet implemented in the environment model. The ego vehicle cannot compute a fail-safe trajectory, since the green vehicle is predicted to turn left, merging into the ego vehicle's lane.

realize our verification technique in series production, authorities must first legislate regulations governing autonomous vehicle safety. Afterward, manufacturers can certify our technique in their vehicles. In the event that authorities extend the specification of legal safety, our verification technique automatically adapts to this new specification through the provided occupancy sets that capture all the legal behaviors of other traffic participants.

ACKNOWLEDGMENT

The authors thank Markus Koschi for the development of the tool SPOT and BMW for supporting our driving experiments. Moreover, we thank Stefanie Manzinger, Anna-Katharina Rettinger, Sebastian Kaster, and Julia Kabalar, Dr. Sebastian Gnatzig, and Dr. Tobias Rehder for their assistance during our experiments. This work is partially funded by the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF (https://www.ko-haf.de/) and the project interACT within the EU Horizon 2020 programme under grant agreement No 723395.

REFERENCES

[1] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[2] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The Int. Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[3] S. Vaskov, S. Kousik, H. Larson, F. Bu, J. R. Ward, S. Worrall, M. Johnson-Roberson, and R. Vasudevan, "Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments," in *Proc. of Robotics: Science and Systems*, 2019, pp. 1–10.

[4] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[5] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[6] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020.

[7] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 3530–3538.

[8] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 2641–2646.

[9] X. Ma, K. Driggs-Campbell, and M. J. Kochenderfer, "Improved robustness and safety for autonomous vehicle control with adversarial reinforcement learning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1665–1671.

[10] M. Mukadam, A. Cosgun, A. Nakhaei, and K. Fujimura, "Tactical decision making for lane changing with deep reinforcement learning," in *NIPS Workshop on Machine Learning for Intelligent Transportation Systems*, 2017, pp. 1–7.

[11] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. M. Zöllner, "Adaptive behavior generation for autonomous driving using deep reinforcement learning with compact semantic states," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 993–1000.

[12] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, pp. 1–9, 2016.

[13] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring Drivability of Planned Motions Using Formal Methods," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–8.

[14] D. Heß, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 1474–1481.

[15] D. J. Grymin, C. B. Neas, and M. Farhood, "A hierarchical approach for primitive-based motion planning and control of autonomous vehicles," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 214–228, 2014.

[16] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 1649–1654.

[17] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Algorithmic Foundations of Robotics X*, 2013, pp. 543–558.

[18] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 5883–5890.

[19] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.

[20] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[21] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proc. of the American Control Conference*, 2001, pp. 43–49.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[24] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Int. Journal of Field Robotics*, vol. 25, no. 9, pp. 640–673, Sept. 2008.

[25] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Systems and Robots*, 2009, pp. 1879–1884.

[26] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 4889–4895.

[27] M. Pivtoraiko and A. Kelly, "Kinodynamic motion planning with state lattice motion primitives," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 2172–2179.

[28] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The Int. Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[29] Z. Ajanovic, B. Lacevic, B. Shyrokau, M. Stolz, and M. Horn, "Search-based optimal motion planning for automated driving," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 4523–4530.

[30] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. Journal of Robotic Research*, vol. 31, no. 3, pp. 346–359, 2012.

[31] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[32] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. of the IEEE European Control Conference*, 2001, pp. 2603–2608.

[33] Y. Du, Y. Wang, and C. Chan, "Autonomous lane-change controller via mixed logical dynamical," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2014, pp. 1154–1159.

[34] F. Molinari, Nguyen Ngoc Anh, and L. Del Re, "Efficient mixed integer programming for autonomous overtaking," in *Proc. of the American Control Conference*, 2017, pp. 2303–2308.

[35] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. of the American Control Conference*, 2002, pp. 1936–1941.

[36] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 42–49.

[37] B. Landry, R. Deits, P. R. Florence, and R. Tedrake, "Aggressive quadrotor flight through cluttered environments using mixed integer programming," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 1469–1475.

[38] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha – A local, continuous method," in *Proc. of the IEEE Int. Symposium on Intelligent Vehicles*, 2014, pp. 450–457.

[39] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *The Int. Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[40] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars," in *Proc. of the European Control Conference*, 2018, pp. 1341–1348.

[41] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering using nonlinear model predictive control," in *Proc. of the American Control Conference*, 2018, pp. 4772–4777.

[42] F. Gritschneder, K. Graichen, and K. Dietmayer, "Fast trajectory planning for automated vehicles using gradient-based nonlinear model predictive control," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 7369–7374.

[43] D. Bertsekas, *Nonlinear Programming*, ser. Athena scientific optimization and computation series. Athena Scientific, 2016.

[44] B. Yi, S. Gottschling, J. Ferdinand, N. Simm, F. Bonarens, and C. Stiller, "Real time integrated vehicle dynamics control and trajectory planning with MPC for critical maneuvers," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 584–589.

[45] V. Jain, U. Kolbe, G. Breuel, and C. Stiller, "Reacting to multi-obstacle emergency scenarios using linear time varying model predictive control," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1822–1829.

[46] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. of the IEEE Int. Conf. on Decision and Control*, 2007, pp. 2980–2985.

[47] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *Int. Journal of Vehicle Autonomous Systems*, vol. 8, no. 2–4, pp. 190–216, 2010.

[48] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg, "Predictive manoeuvre generation for automated driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2013, pp. 418–423.

[49] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2016.

[50] J. Schulz, K. Hirsenkorn, J. Löchner, M. Werling, and D. Burschka, "Estimation of collective maneuvers through cooperative multi-agent planning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 624–631.

[51] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[52] C. Miller, C. Pek, and M. Althoff, "Efficient mixed-integer planning for longitudinal and lateral control of autonomous vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, accepted.

[53] W. Damm, H.-J. Peter, J. Rakow, and B. Westphal, "Can we build it: formal synthesis of control strategies for cooperative driver assistance systems," *Mathematical Structures in Computer Science*, vol. 23, no. 04, pp. 676–725, 2013.

[54] M. Hilscher, S. Linker, and E.-R. Olderog, "Proving safety of traffic manoeuvres on country roads," in *Theories of Programming and Formal Methods*. Springer, 2013, pp. 196–212.

[55] S. M. Loos, A. Platzer, and L. Nistor, "Adaptive cruise control: hybrid, distributed, and now formally verified," in *Proc. of the Int. Symposium on Formal Methods*, 2011, pp. 42–56.

[56] S. Mitsch, S. M. Loos, and A. Platzer, "Towards formal verification of freeway traffic control," in *Proc. of the IEEE Int. Conf. on Cyber-Physical Systems*, 2012, pp. 171–180.

[57] T. Fraichard and H. Asama, "Inevitable collision states. A step towards safer robots?" in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 388–393.

[58] T. Fraichard, "A short paper about motion safety," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 1140–1145.

[59] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.

[60] A. Lawitzky, A. Nicklas, D. Wollherr, and M. Buss, "Determining states of inevitable collision using reachability analysis," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 4142–4147.

[61] L. Martinez-Gomez and T. Fraichard, "An efficient and generic 2D inevitable collision state-checker," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 234–241.

[62] D. Althoff, M. Buss, A. Lawitzky, M. Werling, and D. Wollherr, "Online trajectory generation for safe and optimal vehicle motion planning," in *Autonomous Mobile Systems*, 2012, pp. 99–107.

[63] S. Söntges and M. Althoff, "Determining the nonexistence of evasive trajectories for collision avoidance systems," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 956–961.

[64] ——, "Computing the Drivable Area of Autonomous Road Vehicles in Dynamic Road Scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1855–1866, June 2018.

[65] K. Berntorp, A. Weiss, C. Danielson, and S. Di Cairano, "Automated driving: Safe motion planning using positively invariant sets," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 2247–2252.

[66] D. Althoff, M. Althoff, and S. Scherer, "Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 3470–3477.

[67] M. Jalalmaab, B. Fidan, S. Jeon, and P. Falcone, "Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 843–848.

[68] C. Pek and M. Althoff, "Efficient computation of invariably safe states for motion planning of self-driving vehicles," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 3523–3530.

[69] P. Falcone, M. Ali, and J. Sjöberg, "Predictive threat assessment via reachability analysis and set invariance theory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1352–1361, 2011.

[70] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: a modular framework for fast and guaranteed safe motion planning," in *Proc. of the IEEE Conference on Decision and Control*, 2017, pp. 1517–1522.

[71] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 1679–1686.

[72] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv:1708.06374 [cs.RO]*, pp. 1–37, 2017.

[73] P. F. Orzechowski, K. Li, and M. Lauer, "Towards responsibility-sensitive safety of automated vehicles with reachable set analysis," in *Proc. of the IEEE Int. Conf. on Connected Vehicles and Expo*, 2019, pp. 1–6.

[74] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1447–1454.

[75] S. Steyer, G. Tanzmeister, and D. Wollherr, "Grid-based environment estimation using evidential mapping and particle tracking," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 384–396, 2018.

[76] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016.

[77] A. Tamke, T. Dang, and G. Breuel, "A flexible method for criticality assessment in driver assistance systems," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2011, pp. 697–702.

[78] Economic Comission for Europe: Inland Transport Committee, "Vienna Convention on Road Traffic," Nov. 1968. [Online]. Available: http://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf

[79] A. Rizaldi, F. Immler, and M. Althoff, "A formally verified checker of the safe distance traffic rules for autonomous vehicles," in *NASA Formal Methods Symposium*, 2016, pp. 175–190.

[80] C. Pek, P. Zahn, and M. Althoff, "Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 1477–1483.

[81] A. Rizaldi, F. Immler, and M. Althoff, "A formally verified checker of the safe distance traffic rules for autonomous vehicles," in *NASA Formal Methods Symposium*, 2016, pp. 175–190.

[82] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf, and T. Nipkow, "Formalising and monitoring traffic rules for autonomous vehicles in Isabelle/HOL," in *Integrated Formal Methods*, 2017, pp. 50–66.

[83] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[84] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: investigating ride comfort measures in autonomous cars," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 4–17, 2015.

[85] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2010, pp. 518–522.

[86] E. Velenis and P. Tsiotras, "Optimal velocity profile generation for given acceleration limits; the half-car model case," in *Proc. of the IEEE Int. Symposium on Industrial Electronics*, 2005, pp. 355–360.

[87] A. Eckert, B. Hartmann, M. Sevenich, and P. Rieth, "Emergency steer & brake assist: A systematic approach for system integration of two complementary driver assistance systems," in *Proc. of the Int. Technical Conf. on Enhanced Safety of Vehicles*, 2011, pp. 1–9.

[88] S. Ovchinnikov, "Max-min representation of piecewise linear functions," *Contributions to Algebra and Geometry*, vol. 43, no. 1, pp. 297–302, 2002.

[89] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[90] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *European Control Conference (ECC)*, 2013, pp. 3071–3076.

Christian Pek is a postdoctoral researcher in the Division of Robotics, Perception and Learning at KTH Royal Institute of Technology. Before joining KTH, he was a PhD student in the Cyber-Physical Systems Group at the Technical University of Munich under Prof. Dr.-Ing. Matthias Althoff. He was a research assistant in the motion planning group at BMW Group from 2015 until 2019. Christian graduated with the Master of Science degree in computer science and robotics from the Technical University of Braunschweig, Germany, and the University of Auckland, New Zealand, in 2015. His vision is a future of robots which robustly and safely accomplish tasks with and around humans.

Matthias Althoff is an associate professor in computer science at Technische Universität München, Germany. He received his diploma engineering degree in Mechanical Engineering in 2005, and his Ph.D. degree in Electrical Engineering in 2010, both from Technische Universität München, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Technische Universität Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, and power systems.