Enhancing Motion Safety by Identifying Safety-critical Passageways

Christian Pek[1], Markus Koschi[2], Moritz Werling[1], and Matthias Althoff[2]

Abstract— **Safety is the most important aspect of systems which have to perform collision-free motions in dynamic environments. Formal verification methods, such as reachability analysis, are capable of guaranteeing safety for a given model and given assumptions (e. g. bounded velocity and acceleration). However, certain assumptions can be violated by dynamic obstacles during the execution of the verified motion plan, exposing the system to potential collisions. To compensate for the invalidated verification, this paper introduces the *Point of No Return* (PNR) and the *Point of Guaranteed Arrival* (PGA) by incorporating invariably safe sets. These concepts allow one to divide the planned trajectory into safe sections and safety-critical passageways. For the former, we are able to provide safety guarantees for an infinite time horizon. For the latter, we present a method to minimize such safety-critical passageways prior to execution and thus reduce the risk of potential collisions if assumptions are violated during execution. The safety benefits are highlighted by a numerical example of overtaking maneuvers of self-driving vehicles.**

I. Introduction

A. Motivation

Formal verification is a promising technique for assessing the safety of motion plans. It can prove whether a modeled system behaves correctly with respect to a given specification. However, these models are based on certain assumptions, e. g. that the velocity and acceleration of surrounding dynamic objects are bounded. Without assumptions, it is difficult to accomplish the provided task while ensuring safety, as the infinite number of possible behaviors of objects in the environment often results in collisions (cf. *freezing robot problem* [1]).

Using assumptions comes with the disadvantage that the safety of the system is no longer guaranteed if surrounding dynamic obstacles violate one or more of these assumptions. This unsafe situation has to be solved in a timely manner, since the system is exposed to potential collisions and must determine a feasible evasive trajectory to return to a safe state as fast as possible. Thus, advanced safety mechanisms have to recover safety even if certain assumptions are violated during the execution of the motion plan.

B. Literature Overview

In [2], three criteria for obtaining safe motion plans are introduced: a system should consider "*its own dynamics*", the "*environment objects' future behavior*", and "*reason over an infinite time horizon*" to avoid collisions at all times. For this purpose, the concept of *Inevitable Collision States* (ICS) was introduced [3]. ICS are states in which the system, regardless of which trajectory it follows, eventually collides with an obstacle [4], [5]. A motion plan of the system is safe if it avoids ICS at all times. To assess if a state is close to an ICS, *Regions of Near Collision* (RNC) and *Regions of Potential Collision* (RPC) are proposed in [6]. RNC contain states that will end in an ICS if the system does not change its current motion plan within a certain amount of time. On the other hand, RPC describe states which may end in an ICS due to uncertainties or faults in the control strategy. However, most ICS checkers are computationally costly and require deterministic motion predictions of dynamic obstacles [7].

Verifying the safety of systems can also be done by applying logical reasoning as presented in [8] for highway entry systems of self-driving vehicles or in [9] for the European train control system. Furthermore, some work defines application-specific logics, e. g. *Multi-lane Spatial Logic* (MLSL), which verifies the safety of a lane change controller [10], or *Quantified Differential Dynamic Logic*, which verifies an adaptive cruise control system [11]. Nevertheless, logical expressions for the verification of advanced systems are often complex and subject to the specific controller of the system.

Reachability analysis accounts for any feasible future motion of dynamic obstacles [12], [13]. By calculating the reachable set of each obstacle, i. e. the set of states reachable from their current state, and checking for intersections with the reachable set of the ego system, one can identify possible future collisions. Safety verification using reachability analysis has been proposed for several domains, e. g. self-driving vehicles [14] or robot manipulators [15].

Applying reachability analysis allows one to assess the feasibility of motion plans, e. g. as presented in [16] for overtaking maneuvers of self-driving vehicles with oncoming traffic. This technique can also be used to examine the existence of evasive trajectories by evaluating over-approximated reachable sets of the system. However, reachability analysis can be computationally costly, as one has to consider every possible control input for a given model and efficiently represent the resulting sets.

As a way to overcome these difficulties, the concepts *Invariant Sets* (IS) and *Controlled Invariant Sets* (CIS) [17] are becoming more popular in robotics. Invariant sets are sets of states which allow a system to remain within this set for an infinite time horizon. In [18]–[22], invariant sets are applied to motion planning of autonomous systems. Invariant sets are also used for safety verification. For instance, CIS

*The first two authors have contributed equally to this work.

[1]BMW Group, D-85748 Garching, Germany, E-mail: christian.pek@bmw.de and moritz.werling@bmw.de

[2]Department of Computer Science, Technical University of Munich, D-85748 Garching, Germany, E-mail: markus.koschi@tum.de and althoff@in.tum.de

are used to verify the safety of unmanned aerial vehicles (UAVs) [23], [24] or for safe controller design [25]. In combination with reachability analysis, invariant sets are used to verify the safety of adaptive cruise control systems [26], [27] or for predicitive threat assessment [28]. States within a CIS allow the system to stay in it indefinitely long. However, determining invariant sets is computationally costly, especially in dynamic environments.

C. Contribution

This paper presents a novel approach for assessing the safety of motion plans in dynamic environments and recovering the safety if a previously verified motion plan suddenly becomes invalidated due to the violation of assumptions. We derive invariably safe sets, which allow us to determine the *Point of No Return* (PNR) and the *Point of Guaranteed Arrival* (PGA) (cf. Def. in Sec. IV).

The properties of the PNR and PGA allow one to efficiently reason about safety. In time-critical situations in which a previously verified motion plan suddenly becomes unsafe during execution, our approach offers two advantages over existing work: (1) we are able to provide additional safety guarantees to find feasible trajectories to safe states, and (2) we can use the PNR and PGA to construct a utility function to reason about the safety of multiple motion hypotheses prior to their execution.

The remainder of this paper is organized as follows: In Sec. II, we model the system and define invariably safe sets. Sec. III covers the safety verification of planned trajectories using reachability analysis. In Sec. IV, the PNR and PGA are defined, and their safety properties are highlighted. The proposed concept is demonstrated by a numerical example in Sec. V using overtaking maneuvers of self-driving vehicles.

II. Preliminaries

Let us introduce $\mathcal{X} \subset \mathbb{R}^n$ as the set of feasible states x and $\mathcal{U} \subset \mathbb{R}^m$ as the set of admissible control inputs u of a system f, which is governed by the differential equation

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

We assume that the initial time is $t_0 = 0$ and adhere to the notation $u([0, t_h])$ to describe a trajectory $u(t) \in \mathcal{U}$ for $t \in [0, t_h]$, $0 < t_h$. Furthermore, $\chi\big(t_h, x(0), u([0, t_h])\big) \in \mathcal{X}$ denotes the solution of (1) at time t_h subject to $x(0) = x_0$ and $u([0, t_h])$.

Definition 1 (Safe States)
The set $\mathcal{F}^t$ describes the maximal set of safe states at the point in time t.

Please note that the definition of the set of safe states $\mathcal{F}^t$ depends on the system and its environment; in this work, we consider safe states to be collision-free, which describes the safety of many systems.

Definition 2 (Safe Input Trajectory)
An input trajectory $u([t_1, t_2])$ is called a safe input trajectory for the time interval $[t_1, t_2]$ if $\forall t \in [t_1, t_2]$: $\chi\big(t, x(t_1), u([t_1, t])\big) \in \mathcal{F}^t$.

By an abuse of notation, we use $u([t_1, t_2]) = \Phi\big(x([t_1, t_2]), r_{\text{ref}}\big)$ to emphasize that a trajectory is generated by a feedback control law Φ for a given reference r_{ref}, e. g. a desired velocity.

Definition 3 (Safe Feedback Control Law)
A feedback control law Φ is called a safe feedback control law if every produced input trajectory $u([t_1, t_2]) = \Phi\big(x([t_1, t_2]), r_{ref}\big)$ is a safe input trajectory.

We derive subsets of $\mathcal{F}^t$ which only contain invariably safe states, i. e. from these states, the system described in (1) is always able to be safe for an infinite time horizon, even in dynamic environments:

Definition 4 (Invariably Safe Set)
The Invariably Safe Set (ISS) $\mathcal{S}^t$ for a point in time t and a safe feedback control law Φ_{safe} is defined as

$$\mathcal{S}^t = \big\{x(t) \in \mathcal{F}^t \,\big|\, \forall \tau > t : \\ \chi\big(\tau, x(t), \Phi_{safe}(x([t, \tau]), r_{ref})\big) \in \mathcal{F}^\tau\big\}.$$

In contrast, states $x(t) \in (\mathcal{F}^t \setminus \mathcal{S}^t) := \{x \,|\, x \in \mathcal{F}^t \wedge x \notin \mathcal{S}^t\}$ are only regarded as safe for a finite time horizon, since they may inevitably lead to an unsafe state $x(\tau) \notin \mathcal{F}^\tau$, $\tau > t$. For the sake of clarity, we omit the notation of time in $\mathcal{F}^t$ and $\mathcal{S}^t$ if all points in time are considered.

III. Verification of Motion Plans

Let us consider tasks where the system (1) has to traverse from an initial state $x(0) \in \mathcal{S}^0_{\text{pre}}$ to a final state $x(t_h) \in \mathcal{S}^{t_h}_{\text{post}}$ (cf. Fig. 1). Both $\mathcal{S}^0_{\text{pre}} \subset \mathcal{S}^0$ and $\mathcal{S}^{t_h}_{\text{post}} \subset \mathcal{S}^{t_h}$ are ISSs according to Def. 4 for a given safe feedback control law Φ_{safe}. Often, one has situations in which $\forall t \in [0, t_h]$: $\mathcal{S}^t_{\text{pre}} \cap \mathcal{S}^t_{\text{post}} = \emptyset$, eliminating the possibility to use only this dedicated safe feedback control law. As a result, we cannot be sure that a planned trajectory $u([0, t_h])$ for the given task is safe (cf. Def. 2). To verify the traversing trajectory as collision-free with respect to the obstacles in the environment, we make use of reachability analysis:

Definition 5 (Reachable Set)
The reachable set $\mathcal{R} \subseteq \mathcal{X}$ of (1) is the set of states which are reachable at a certain point in time r from a set of initial states $\mathcal{X}^0$ at time t_0 and subject to the set of inputs $\mathcal{U}$:

$$\mathcal{R}(r) = \left\{ x(0) + \int_0^r f\big(x(t), u(t)\big) dt \,\middle|\, x(0) \in \mathcal{X}^0, \forall t : u(t) \in \mathcal{U} \right\}.$$

To realize efficient collision checking, we introduce a relation from the state space to the Euclidean space in world coordinates:

Definition 6 (Relation to Euclidean Space)
The operator $\mathrm{occ}(x)$ relates the state vector x to the set of occupied points in Euclidean space as

$$\mathrm{occ}(x) : \mathcal{X} \to \mathcal{P}(\mathbb{R}^\ell),$$

where $\mathcal{P}(\mathbb{R}^\ell)$ is the power set of $\mathbb{R}^\ell$. Given a set of states $\mathcal{X}$, we define $\mathrm{occ}(\mathcal{X}) := \{\mathrm{occ}(x) \,|\, x \in \mathcal{X}\}$.

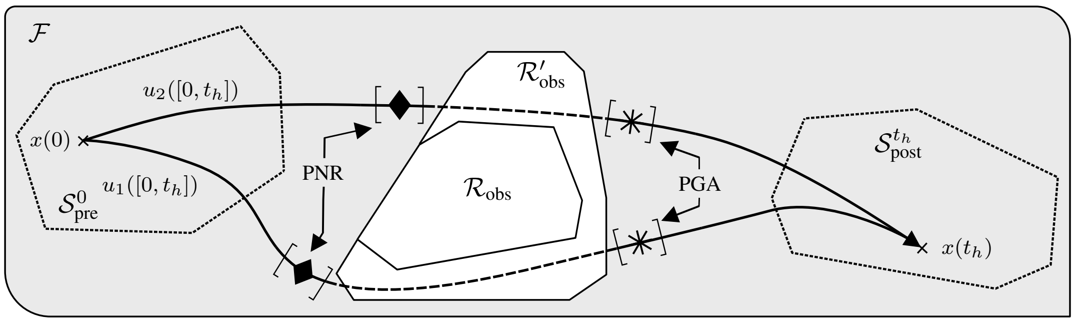

$\mathcal{F}$
$u_2([0,t_h])$
$x(0)$
$u_1([0,t_h])$
$\mathcal{S}^0_{\text{pre}}$
PNR
$\mathcal{R}'_{\text{obs}}$
$\mathcal{R}_{\text{obs}}$
PGA
$\mathcal{S}^{t_h}_{\text{post}}$
$x(t_h)$

Fig. 1. The trajectories $u_1([0,t_h])$ and $u_2([0,t_h])$, which start at an initial state $x(0) \in \mathcal{S}^0_{\text{pre}}$ and end in a final state $x(t_h) \in \mathcal{S}^{t_h}_{\text{post}}$, are verified as safe for $\mathcal{A}_{\text{viol}} = \emptyset$, which corresponds to the set of reachable states of all obstacles $\mathcal{R}_{\text{obs}}$. For a violation of assumptions (i. e. $\mathcal{A}_{\text{viol}} \neq \emptyset$) resulting in $\mathcal{R}'_{\text{obs}}$, we can determine the intervals of the PNR and the PGA along each trajectory. These points delimit the safety-critical passageway SCP, which is denoted by a dashed line.

Definition 7 (Occupancy Set)
Based on Def. 5 and Def. 6, the occupancy set $\mathcal{O}(t)$ describes the set of occupied points in Euclidean space at time t:

$$\mathcal{O}(t) = \operatorname{occ}\big(\mathcal{R}(t)\big).$$

We verify motion plans using occupancy sets:

Definition 8 (Collision-free Trajectory)
Given the possible occupancies of all surrounding obstacles $\mathcal{O}_{obs}(t) = \bigcup_{b\in\mathcal{B}} \mathcal{O}_b(t)$, $\mathcal{B} \subset \mathbb{N}$, and the occupancy of the ego system along its planned trajectory $\mathcal{O}_{ego}(t) := \operatorname{occ}\big(\chi(t, x(0), u([0,t])\big)$, this trajectory is collision-free if

$$\forall t \in [0, t_h] : \mathcal{O}_{ego}(t) \cap \mathcal{O}_{obs}(t) = \emptyset.$$

In order to obtain the occupancies $\mathcal{O}_{\text{obs}}(t)$ based on reachable states, we require assumptions on the bounds of the set of possible inputs for each obstacle (cf. $\mathcal{U}$ in Def. 5). These bounds constrain the behavior of the dynamic obstacles, since otherwise $\mathcal{O}_{\text{obs}}(t)$ would often intersect with $\mathcal{O}_{\text{ego}}(t)$ and thus the system is no longer able to safely accomplish a given task. We consider different types of assumptions:

Definition 9 (Assumptions)

- *Time-invariant assumptions $\mathcal{A}_\infty$ are assumptions which have to hold at any time.*
- *Violable assumptions $\mathcal{A}_\mathcal{B}$ are assumptions which constrain the motion of dynamic obstacles and might be violated at some point in time.*
- *Violated assumptions $\mathcal{A}_{viol} \subseteq \mathcal{A}_\mathcal{B}$ are the set of assumptions which have been violated by dynamic obstacles.*
- *Valid assumptions are defined as $\mathcal{A}_{valid} := \mathcal{A}_\infty \cup (\mathcal{A}_\mathcal{B} \setminus \mathcal{A}_{viol})$.*

For instance, $\mathcal{A}_\infty$ includes physical limitations, e. g. limited acceleration, or general assumptions on safety, e. g. that dynamic obstacles are not enforcing a collision with the ego system. Per definition (cf. Def. 9), the sets $\mathcal{S}_{\text{pre}}$ and $\mathcal{S}_{\text{post}}$ only result from $\mathcal{A}_\infty$. The set of violable assumptions $\mathcal{A}_\mathcal{B}$ may contain the assumption that the velocity of obstacles does not exceed a certain limit. From now on, we implicitly mean $\mathcal{A}_{\text{valid}}$ if we use the term assumptions.

Remark 1 (Assumptions for Verification)
The verification of motion plans according to Def. 8 is based on $\mathcal{A}_{valid}$.

IV. Enhancing Safety Using Safe Invariant Sets

If the set of assumptions changes during execution of the provided task, i. e. dynamic obstacles violate previously valid assumptions, the verification result is no longer applicable. Since a renewed verification of the motion plan according to the reduced set of assumptions often fails, we use $\mathcal{S}_{\text{pre}}$ and $\mathcal{S}_{\text{post}}$ (which are invariant to $\mathcal{A}_{\text{valid}}$) to propose safety-relevant points along the planned trajectory $u([0,t_h])$, which allow our system to regain safety (cf. Fig. 1):

Definition 10 (Point of No Return)
The Point of No Return (PNR) is the state $x(t_{PNR})$, $t_{PNR} \in [0,t_h]$, along $u([0,t_h])$ from which returning to $\mathcal{S}_{pre}$ is ultimately possible using a safe trajectory $u([t,r])$, $t < r$:

$$\forall t \in [0, t_{PNR}] : \exists u([t,r]) : \chi\big(r, x(t), u([t,r])\big) \in \mathcal{S}^r_{pre}$$
$$\wedge \forall t \in]t_{PNR}, t_h] : \nexists u([t,r]) : \chi\big(r, x(t), u([t,r])\big) \in \mathcal{S}^r_{pre}.$$

After a specific point along $u([0,t_h])$, the system is able to safely enter $\mathcal{S}_{\text{post}}$:

Definition 11 (Point of Guaranteed Arrival)
The Point of Guaranteed Arrival (PGA) is the state $x(t_{PGA})$, $t_{PGA} \in [0,t_h]$, along $u([0,t_h])$ from which point on safety is guaranteed using a safe trajectory $u([t,r])$, $t < r$:

$$\forall t \in [t_{PGA}, t_h] : \exists u([t,r]) : \chi\big(r, x(t), u([t,r])\big) \in \mathcal{S}^r_{post}.$$

By using Def. 10 and Def. 11, we define the safety-critical passageway along $u([0,t_h])$ as:

Definition 12 (Safety-critical Passageway)
The safety-critical passageway (SCP) between $\mathcal{S}^0_{pre}$ and $\mathcal{S}^{t_h}_{post}$ is defined as the set of states between the PNR and the PGA along $u([0,t_h])$:

$$SCP = \{x \,|\, x = \chi\big(t, x(0), u([0,t])\big), t_{PNR} < t < t_{PGA}\}.$$

A. Determining the PNR and PGA

We determine the PNR and PGA with respect to the remaining valid assumptions $\mathcal{A}_{\text{valid}}$. Based on the discussion in [29], the exact PNR and PGA along a trajectory cannot be determined, but rather a time interval $[\underline{t}, \overline{t}]$ of their possible locations. For the PNR, we can obtain an upper bound using reachability analysis and a lower bound using sampling methods as demonstrated subsequently. Please note that we focus on the basic concept of the search and not on specific implementation details.

Proposition 1 (Under-approximation)
A lower bound of the location of the PNR $\underline{t}_{PNR}$ is determined by obtaining witnesses of Def. 10 from sampling techniques.

Proof: Per definition, the set of sampled trajectories is a real subset of all feasible trajectories of (1). Thus, $\underline{t}_{\text{PNR}}$ represents an under-approximation. ■

Proposition 2 (Over-approximation)
By using over-approximated reachable sets of (1) *(cf. Def. 5), we define the upper bound as $\overline{t}_{PNR} \in [0, t_h]$ such that*

$$\forall t \in [\overline{t}_{PNR}, t_h] : \forall r \geq 0 : \mathcal{R}(r) \cap \mathcal{S}_{pre}^{t+r} = \emptyset$$
$$\textit{subject to } \mathcal{X}^0 = \{\chi(t, x(0), u([0,t]))\}.$$

Proof: Prop. 2 directly follows from the definition of over-approximated reachable sets of (1), which ensures that the system is not able to return to $\mathcal{S}_{\text{pre}}$ from the obtained upper bound. ■

The interval of the PGA can be obtained analogously.

Remark 2 (Precomputation)
One can precompute a sufficiently close approximation of the PNR and PGA intervals for predefined tasks and sets of violated assumptions. This precomputed approximation is used as an initial guess and further refined online. Additionally, both searches can be sped up by incorporating a binary search strategy to determine the optimal bound. The advantage of using this strategy is its anytime property.

B. Significance to Motion Safety

As mentioned before in Sec. III, assumptions are required for verifying the motion plan $u([0, t_h])$. Violation of assumptions during execution results in larger reachable sets of obstacles (cf. $\mathcal{R}'_{\text{obs}}$ in Fig. 1). Thus, the passageway SCP might contain unsafe states (i. e. SCP $\not\subseteq \mathcal{F}$):

Theorem 1 (Safe and Safety-critical Stages)
The motion plan $u([0, t_h])$ can be divided into safe and safety-critical stages using the PNR and PGA:

1) *$t \in [0, \underline{t}_{PNR}]$: A feasible and safe trajectory to a safe state $x \in \mathcal{S}_{pre}$ is guaranteed until the PNR. (In contrast, $\forall t > \overline{t}_{PNR}$: A feasible and safe trajectory reaching $\mathcal{S}_{pre}$ does not exist.)*
2) *$t \in]\underline{t}_{PNR}, \overline{t}_{PGA}[$: A feasible and safe trajectory to $\mathcal{S}_{post}$ may not exist within the SCP.*
3) *$t \in [\overline{t}_{PGA}, t_h]$: A feasible and safe trajectory to a safe state $x \in \mathcal{S}_{post}$ is guaranteed from the PGA onwards.*

Proof: Thm. 1 directly follows from Def. 10–12. As soon as the system enters $\mathcal{S}_{\text{pre}}$ or $\mathcal{S}_{\text{post}}$, it can switch to the designated safe feedback control law and remain safe for an infinite time horizon. ■

A motion planner can use safety-critical stages to evaluate trajectories:

Remark 3 (Safety Costs)
The safety of j different motion plans $u_i([0, t_h])$, $i \leq j$, can be assessed by using a cost function which assigns costs c_i to each passageway SCP$_i$.

Rmk. 3 follows from Def. 12 and allows one to characterize and compare the passageway of different motion plans $u_i([0, t_h])$. The cost function has to be modeled depending on the specific task and the utilized system. For example, the costs correspond to the time-span of the safety-critical passageway, and the safest motion plan to the one with the lowest costs.

Motion planners which do not consider these safety costs might determine trajectories with large safety-critical passageways. If we integrate the cost function of the passageway as a separate cost term into the optimization of the motion planner, the planner directly determines the safest trajectory. As a result, one may be able to obtain a trajectory with a passageway of size zero.

Remark 4 (Zero Passageway)
SCP $= \emptyset$ of a motion plan $u([0, t_h])$ guarantees that the system is always able to safely enter $\mathcal{S}_{pre}$ and $\mathcal{S}_{post}$.

V. Numerical Example

In this section, the proposed concept is demonstrated for the domain of self-driving vehicles. We consider highly safety-critical overtaking maneuvers on a two-lane road with oncoming traffic (cf. Fig. 2). The set of safe states $\mathcal{F}$ corresponds to the set of states which are collision-free and respect road boundaries. Given the time-invariant assumption that maximum absolute acceleration is limited to $a_{\max}$, we define $\mathcal{S}_{\text{pre}}$ and $\mathcal{S}_{\text{post}}$ using the safe feedback controller Φ_{safe} which keeps a formal safe distance to preceding vehicles [30] (cf. adaptive cruise control system (ACC) in [31]). We can infer that $\mathcal{S}_{\text{pre}} \cap \mathcal{S}_{\text{post}} = \emptyset$, since overtaking requires the ego vehicle to enter a lane with oncoming traffic.

The parameters of our numerical example are stated in Tab. I, where $(x, y, v)^T$ describes the x- and y-positions and velocity of a vehicle at the initial time $t_0 = 0\,\text{s}$. To obtain the motion plan of the ego system, we utilize the trajectory planner and the underlying vehicle model of [32]. Fig. 2 shows the resulting trajectory $u_1([0, t_h])$ of the overtaking maneuver.

A. Verification of the Overtaking Trajectory

The occupancy sets of all vehicles are predicted using our tool *SPOT*[1] [33]. This tool is based on reachability analysis and allows one to efficiently over-approximate the set of future occupancies of traffic participants under given assumptions.

In addition to $\mathcal{A}_\infty$, we consider the violable assumptions $\mathcal{A}_\mathcal{B}$ listed in Tab. II, which are based on a formalization of the Vienna Convention on Road Traffic [34], [35]. Based on $\mathcal{A}_{\text{valid}}$ and initially assuming $\mathcal{A}_{\text{viol}} = \emptyset$, we obtain the

[1]available at spot.in.tum.de

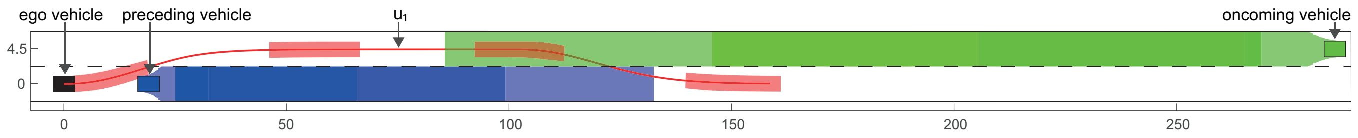


Fig. 2. Since the occupancy of the ego vehicle (red) along its trajectory $u_1([0,t_h])$ does not intersect with the occupancies of other vehicles (blue and green) in any time interval, the motion plan is verified as collision-free. Note that for the sake of clarity, the occupancy sets are only shown for the time intervals $[0,1]$, $[3,4]$, $[6,7]$, and $[9,10]$, and plotted transparently. The axes are in meters.

occupancy sets of each vehicle for consecutive time intervals with prediction step size $\Delta t = 0.1\,\mathrm{s}$ up to the time horizon $t_h = 10\,\mathrm{s}$ (cf. Fig. 2). The motion plan $u_1([0,t_h])$ is verified as collision-free, since none of the occupancies of other vehicles intersects with the occupancy of the ego vehicle along its planned trajectory in any time interval.

B. Determining the PNR and PGA

During the overtaking maneuver, we consider that the oncoming vehicle violates the assumptions $\mathcal{A}_{\text{viol}} = \{A_{v_{\max}}, A_{\text{back}}\}$ (cf. Tab. II). Thus, the previously verified overtaking maneuver is no longer collision-free. We determine the PNR and PGA based on the remaining valid assumptions $\mathcal{A}_{\text{valid}}$:

a) PNR interval: To compute the upper bound $\overline{t}_{\text{PNR}}$ of the PNR according to Prop. 2, we use our tool *SPOT*. For each state $x(k\Delta t)$, $k\Delta t \in [0,t_h]$, along the planned trajectory, we run the occupancy prediction and check from which k onwards the ego vehicle is not able to return to its initial lane and maintain a safe distance to the preceding vehicle.

After determining the upper bound, we can restrict the search of the lower bound to states $x(k\Delta t)$, $k\Delta t \in [0, \overline{t}_{\text{PNR}}[$. We use our sampling-based trajectory planner [32] to determine trajectories reaching $\mathcal{S}_{\text{pre}}$ and check if the ego vehicle maintains the necessary safe distance at all times of the resulting feasible trajectory. We obtain $\overline{t}_{\text{PNR}} = 0.7\,\mathrm{s}$ and $\underline{t}_{\text{PNR}} = 0.6\,\mathrm{s}$ for the upper and lower bound of the PNR, respectively. Fig. 4a visualizes the sampled trajectory and the occupancy sets for the time interval at which the ego vehicle is not able to maintain the safe distance.

TABLE I

PARAMETERS OF THE OVERTAKING SCENARIO.

Parameter	Description
Ego vehicle	$(x,y,v)^T_{\text{ego}} = (0\,\mathrm{m}, 0\,\mathrm{m}, 16.7\,\mathrm{m/s})^T$
Preceding vehicle	$(x,y,v)^T_{\text{pre}} = (19.0\,\mathrm{m}, 0\,\mathrm{m}, 11.1\,\mathrm{m/s})^T$
Oncoming vehicle	$(x,y,v)^T_{\text{onc}} = (285.6\,\mathrm{m}, 4.5\,\mathrm{m}, 16.7\,\mathrm{m/s})^T$
Speed limit	$v_{\text{lim}} = 16.7\,\mathrm{m/s}$
Maximum velocity	$v_{\max} = 1.2 v_{\text{lim}} = 20\,\mathrm{m/s}$
Switching velocity	$v_S = 5.0\,\mathrm{m/s}$
Maximum acceleration	$\lvert a_{\max}\rvert = 8.0\,\mathrm{m/s^2}$
Lateral distance between the lanes	$\Delta y = 4.5\,\mathrm{m}$
Time horizon	$t_h = 10.0\,\mathrm{s}$
Time step size	$\Delta t = 0.1\,\mathrm{s}$

b) PGA interval: Using SPOT, we obtain $\underline{t}_{\text{PGA}} = 8.4\,\mathrm{s}$ for the lower bound. The sampling method results in $\overline{t}_{\text{PGA}} = 8.5\,\mathrm{s}$ for the upper bound. Fig. 4b visualizes the sampled trajectory, which coincides with the overtaking trajectory $u_1([0,t_h])$, and the predicted occupancy sets starting at time $t = \underline{t}_{\text{PGA}}$.

C. Significance to Motion Safety

We validate Thm. 1 by sampling evasive trajectories for every state $x(k\Delta t)$, $k\Delta t \in [0,t_h]$, and checking them for collisions. The trajectory starting at $x(\underline{t}_{\text{PNR}})$ and returning to $\mathcal{S}_{\text{pre}}$ is visualized in Fig. 3 by a dotted line. The trajectory starting at $x(\overline{t}_{\text{PGA}})$ and ending in $\mathcal{S}_{\text{post}}$ coincides with the overtaking trajectory $u_1([0,t_h])$. All trajectories starting at a state $x(k\Delta t)$, $k\Delta t \in]\overline{t}_{\text{PNR}}, t_h]$, and ending in $\mathcal{S}_{\text{pre}}$ result in a collision, as the ego vehicle is not able to maintain the necessary safe distance when the preceding vehicle performs emergency braking.

Within the SCP, i. e. $\underline{t}_{\text{PNR}} < t < \overline{t}_{\text{PGA}}$, a collision-free evasive trajectory ending in $\mathcal{S}_{\text{post}}$ may not exist if assumptions are violated. To speed up the search for a feasible trajectory in such situations, one can make use of the fact that the ego vehicle has to reach the PGA to be safe again. This information allows the motion planner to exclude trajectories which end in $\mathcal{S}_{\text{pre}}$ or have velocities below the maximum reference velocity.

In our example, the oncoming vehicle violates the assumption of maximum speed (i. e. accelerating beyond $v_{\max}$) at time $t = 4.5\,\mathrm{s}$, where the ego vehicle has already passed the PNR and is located within the SCP. To avoid a potential collision, we must determine an evasive trajectory which exits the SCP as fast as possible. Using our novel concept, we are able to reduce the number of trajectory hypotheses

TABLE II

VIOLABLE ASSUMPTIONS ON THE BEHAVIOR OF OTHER VEHICLES.

Assumptions	Description
$A_{v_{\max}}$	When a parameterized speed $v_{\max}$ is reached, acceleration in driving direction is stopped.
A_{engine}	To model limited engine power, acceleration in driving direction is limited above a parameterized speed v_S.
A_{lane}	Leaving the lane is forbidden. Changing lanes is only allowed if the new lane has the same driving direction.
A_{back}	Driving backwards in a lane is not allowed.
A_{over}	If a vehicle is being overtaken, acceleration in driving direction is stopped.

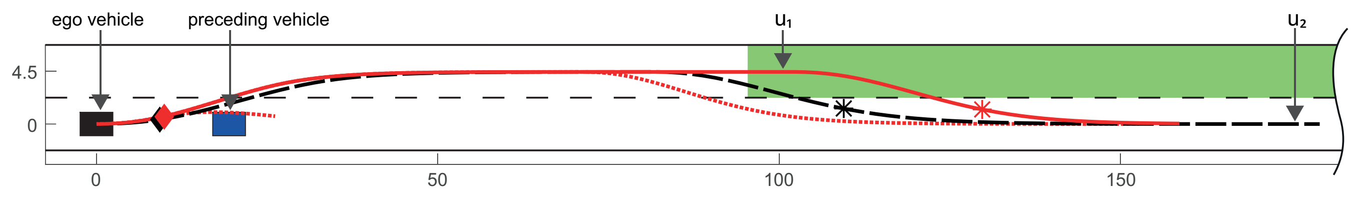


Fig. 3. Trajectory $u_1([0, t_h])$ with max. velocity of $16.7\,\mathrm{m/s}$ is not collision-free if the oncoming vehicle violates $\mathcal{A}_{\mathrm{viol}}$. Two evasive trajectories, denoted by dotted lines, branch off at the PNR and at $t = 4.5\,\mathrm{s}$ within the SCP. An alternative collision-free trajectory $u_2([0, t_h])$ with max. velocity of $19.4\,\mathrm{m/s}$ and shorter SCP is shown by a dashed line. The axes are in meters.

of our planner from 3500 down to 500, which shortens planning time by around $30\,\%$. The obtained trajectory with full acceleration allows the ego vehicle to enter $\mathcal{S}_{\mathrm{post}}$ without colliding with the speeding oncoming vehicle and is denoted by a dotted line in Fig. 3.

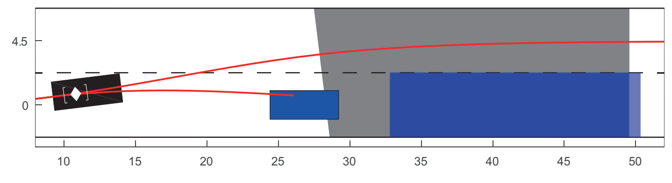
To assess the safety of the SCP for overtaking trajectories according to Rmk. 3, we model the cost function as $c = (\underline{t}_{\mathrm{PGA}} - \overline{t}_{\mathrm{PNR}})$. The costs for the initial overtaking trajectory $u_1([0, t_h])$ with max. velocity $16.7\,\mathrm{m/s}$ and for an alternative trajectory $u_2([0, t_h])$ with max. velocity $19.4\,\mathrm{m/s}$ (cf. Fig. 3) are $c_1 = 7.7\,\mathrm{s}$ and $c_2 = 6.0\,\mathrm{s}$, respectively. If the oncoming vehicle violates $A_{v_{\max}}$, $u_1([0, t_h])$ results in a collision with the oncoming vehicle (cf. occupancy set in Fig. 3). However, the trajectory $u_2([0, t_h])$ avoids a potential collision in this scenario, since the ego vehicle traverses the SCP faster due to the shorter passageway (indicated by lower costs $c_2 \ll c_1$). Incorporating the costs of the SCP into a motion planner allows it to minimize the SCP during optimization. In our example, this corresponds to a trajectory with the maximal feasible velocity profile during overtaking.

VI. Conclusions

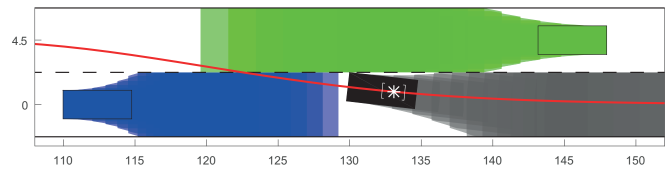
This paper considers situations in which a verified motion plan suddenly becomes unsafe due to the misbehavior of dynamic obstacles and provides a solution to this problem by introducing the PNR and PGA. These novel concepts allow one to derive additional safety guarantees for systems which have to perform collision-free motions in dynamic environments. The PNR and PGA divide motion plans into inherently safe sections and inherently safety-critical passageways.

Within the safety-critical passageway, the system is exposed to potential collisions if obstacles violate assumptions used in the verification. We show that one can minimize the SCP prior to execution by assigning costs to it and integrating the cost function into the optimization of the planner. Trajectories with SCP $= \emptyset$ guarantee safety for an infinite time horizon.

The presented concept is not restricted to self-driving vehicles and can also be applied to other systems, such as industrial robots or unmanned aerial vehicles (UAVs). To reduce computational costs, one may conservatively precompute the PNR and PGA for different tasks and switch between them during runtime.



(a) Upper and lower bound of the PNR.



(b) Upper and lower bound of the PGA.

Fig. 4. The intervals of the PNR and PGA are obtained using set-based prediction and trajectory sampling. The axes are in meters.

Acknowledgments

This work was funded by the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF and by the BMW Group through the CAR@TUM project.

References

[1] P. Trautman and A. Krause, "Unfreezing the robot: navigation in dense, interacting crowds," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 797–803.
[2] T. Fraichard, "A short paper about motion safety," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 1140–1145.
[3] T. Fraichard and H. Asama, "Inevitable collision states – a step towards safer robots?" in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 388–393.
[4] L. Martinez-Gomez and T. Fraichard, "An efficient and generic 2D inevitable collision state-checker," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 234–241.
[5] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 174–179.
[6] N. Chan, J. Kuffner, and M. Zucker, "Improved motion planning speed and safety using regions of inevitable collision," in *17th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control*, 2008, pp. 103–114.
[7] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2013, pp. 1028–1033.
[8] 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.
[9] W. Damm, A. Mikschl, J. Oehlerking, E.-R. Olderog, J. Pang, A. Platzer, M. Segelken, and B. Wirtz, "Automating verification of cooperation, control, and design in traffic applications," in *Formal Methods and Hybrid Real-Time Systems*, 2007, pp. 115–169.

[10] 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.

[11] 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.

[12] M. Althoff, C. L. Guernic, and B. H. Krogh, "Reachable set computation for uncertain time-varying linear systems," in *Proc. of Hybrid Systems: Computation and Control*, 2011, pp. 93–102.

[13] M. Althoff, "Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets," in *Proc. of Hybrid Systems: Computation and Control*, 2013, pp. 173–182.

[14] 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.

[15] A. Pereira and M. Althoff, "Safety control of robots under computed torque control using reachable sets," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 331–338.

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

[17] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[18] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *Proc. of the IEEE Int. Conf. on Decision and Control*, 2016, pp. 5986–5991.

[19] F. Blanchini, F. A. Pellegrino, and L. Visentini, "Control of manipulators in a constrained workspace by means of linked invariant sets," *Int. Journal of Robust and Nonlinear Control*, vol. 14, no. 1314, pp. 1185–1205, 2004.

[20] X. Qi, D. Theilliol, D. Song, and J. Han, "Invariant-set-based planning approach for obstacle avoidance under vehicle dynamic constraints," in *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, 2015, pp. 1692–1697.

[21] G. Franze and W. Lucia, "A receding horizon control strategy for autonomous vehicles in dynamic environments," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 695–702, 2016.

[22] 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.

[23] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Dissertation, Massachusetts Institute of Technology, 2006.

[24] 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 Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 3470–3477.

[25] N. Aréchiga and B. Krogh, "Using verified control envelopes for safe controller design," in *Proc. of the American Control Conference*. IEEE, 2014, pp. 2918–2923.

[26] R. Kianfar, P. Falcone, and J. Fredriksson, "Safety verification of automated driving systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4, pp. 73–86, 2013.

[27] S. Mammar, N. A. Oufroukh, Z. Yacine, D. Ichalal, and L. Nouveliere, "Invariant set based variable headway time vehicle longitudinal control assistance," in *Proc. of the American Control Conference*, 2012, pp. 2922–2927.

[28] 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.

[29] A. Platzer and E. M. Clarke, "The image computation problem in hybrid systems model checking," in *Proc. of Hybrid Systems: Computation and Control*, 2007, pp. 473–486.

[30] 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.

[31] S. Magdici and M. Althoff, "Adaptive cruise control with safety guarantees for autonomous vehicles," in *Proc. of the 20th World Congress of the Int. Federation of Automatic Control*, 2017, pp. 5939–5946.

[32] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet Frame," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 987–993.

[33] 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.

[34] United Nations Economic Commission for Europe, "Vienna Convention on Road Traffic," United Nations, 1968.

[35] A. Rizaldi and M. Althoff, "Formalising traffic rules for accountability of autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2015, pp. 1658–1665.