Department of Informatics
Technical University of Munich

Set-based Prediction of Traffic Participants and its Applications to Safe Motion Planning

Markus Matthias Koschi

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:
Prof. Dr. Helmut Seidl

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Matthias Althoff
2. Prof. Dr.-Ing. Markus Lienkamp

Die Dissertation wurde am 09.10.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 26.02.2021 angenommen.

Abstract

This dissertation presents methods for guaranteeing safety of autonomous vehicles. A major challenge in safe motion planning is the unknown future evolution of prior unseen traffic situations, which are usually only partially observable. To cope with these uncertainties, we develop novel solutions for predicting both detected and undetected traffic participants and for planning collision-free maneuvers in a fail-safe manner.

We propose a set-based prediction that computes all acceptable future behaviors of other traffic participants. By performing reachability analysis based on formalized traffic rules and nondeterministic motion models, we predict the set of all possible states of vehicles, pedestrians, and cyclists. Even if traffic participants violate traffic rules or if sensor measurements are noisy or incomplete, safety is not compromised due to our constraint management. To also consider occluded traffic participants, we create phantom objects at all safety-relevant parts outside of the observed area.

Subsequently, we develop methods that ensure safe motions for autonomous vehicles. Since all acceptable behaviors of other traffic participants are captured by our set-based prediction, we can plan maneuvers for the autonomous vehicle that do not cause accidents. In particular, our approach determines the latest point in time at which a collision can still be avoided and computes evasive maneuvers that safeguard the autonomous vehicle. To reveal safety gaps in motion planners, conversely, we propose an efficient testing method based on falsification.

Extensive real-world experiments with test vehicles validate our solutions. For example, we verify the safety of the autonomous vehicle online in various traffic scenarios, including jaywalking pedestrians and taxis braking suddenly. Overall, our methods can be directly used as a safety layer for existing planning frameworks to drastically reduce the number of traffic accidents.

Summary: Our novel methods capture all acceptable behaviors of other traffic participants and prevent autonomous vehicles from causing accidents.

Zusammenfassung

Diese Dissertation präsentiert Methoden, um die Sicherheit von autonomen Fahrzeugen zu garantieren. Bei der sicheren Bewegungsplanung stellen besonders unbekannte und meist nur teilweise einsehbare Verkehrsszenarien, deren zukünftiger zeitlicher Verlauf sehr ungewiss ist, eine große Herausforderung dar. Diese Unsicherheiten werden durch neuartige Lösungen bewältigt, indem sowohl erfasste als auch verdeckte Verkehrsteilnehmer prädiziert und so kollisionsfreie, ausfallsichere Manöver für das eigene Fahrzeug geplant werden.

Eine mengenbasierte Prädiktion wird vorgestellt, die für jeden Verkehrsteilnehmer alle zulässigen Bewegungen vorausberechnet. Auf Basis von formalisierten Verkehrsregeln und nichtdeterministischen Bewegungsmodellen werden Erreichbarkeitsanalysen durchgeführt. Dadurch können alle möglichen Zustände von Fahrzeugen, Fußgängern und Fahrradfahrern vorhergesagt werden. Auch wenn sich Verkehrsteilnehmer nicht an die Verkehrsregeln halten oder die Sensormessungen ungenau oder unvollständig sind, wird die Sicherheit durch automatisches Anpassen der Prädiktionsparameter gewährleistet. Um verdeckte Verkehrsteilnehmer zu berücksichtigen, werden Phantomobjekte in allen sicherheitsrelevanten Bereichen außerhalb der erfassten Umgebung erstellt.

Als Nächstes werden Methoden zur garantiert sicheren Bewegungsplanung von autonomen Fahrzeugen entwickelt. Da alle zulässigen Bewegungen anderer Verkehrsteilnehmer in der mengenbasierten Prädiktion enthalten sind, können Manöver für das autonome Fahrzeug geplant werden, die keine Unfälle verursachen. Der vorgestellte Ansatz findet insbesondere den spätmöglichsten Zeitpunkt, an dem eine Kollision noch vermieden werden kann, und berechnet Ausweichmanöver, die das autonome Fahrzeug jederzeit in einen sicheren Zustand bringen können. Zuletzt wird eine effiziente Testmethode entwickelt, die durch Falsifikation Sicherheitslücken von Bewegungsplanern aufdeckt.

Die ausgearbeiteten Lösungen werden mit Testfahrzeugen in umfassenden Versuchen validiert. Zum Beispiel wird die Sicherheit des autonomen Fahrzeugs online verifiziert, wobei in diversen Verkehrsszenarien etwa unachtsam querende Fußgänger oder plötzlich abbremsende Taxis berücksichtigt werden. Zusammenfassend lässt sich feststellen, dass die vorgestellten Methoden zur Absicherung von bestehenden Systemen eingesetzt werden können, um die Häufigkeit von Verkehrsunfällen drastisch zu reduzieren.

Kurzdarstellung: Die entwickelten Methoden berücksichtigen alle zulässigen Bewegungen anderer Verkehrsteilnehmer und verhindern, dass autonome Fahrzeuge Unfälle verursachen.

Acknowledgments

Many people have directly or indirectly contributed to this dissertation. I have expressed my sincere and immense gratitude to all of them directly. For you, dear reader, I will simply list them: my supervisor Matthias Althoff; my fellow PhD students, especially Christian Pek and Stefanie Manzinger; my colleagues at BMW, especially Moritz Werling; my colleagues at Zenuity, especially Mattias Brännström; my second examiner Markus Lienkamp; the chairs' staff, especially Alex Lenz, Ute Lomp, and Daniel Renjewski; the students I supervised for their thesis, seminar, or practical course; my parents and my sister; and my friends, especially Greta, Martin, and Joh.

Munich, September 2020 Markus Koschi

Contents

1 Introduction

Autonomous vehicles will drive modern society to completely new means of mobility. Human drivers get relieved from driving tasks and can enjoy other things, like reading a dissertation. If eventually no driver is required, even people who cannot drive, such as elderly people and children, will be able to reach more places by car. Also public transportation systems are expected to become more flexible. Besides the change of mobility solutions, autonomous vehicles can significantly reduce the number and severity of traffic accidents. However, to realize these benefits, we need to ensure that autonomous vehicles are safe in all traffic situations, such as the critical day-to-day situations pictured in Figure 1.1.

It is commonly requested that autonomous vehicles have to be more reliable than human drivers [1]. Since we want autonomous vehicles to prevent accidents, at least severe ones, let us describe the risk of driving by the number of accidents involving causalities per distance traveled. For example, in the United States of America, 1.747 million accidents involving

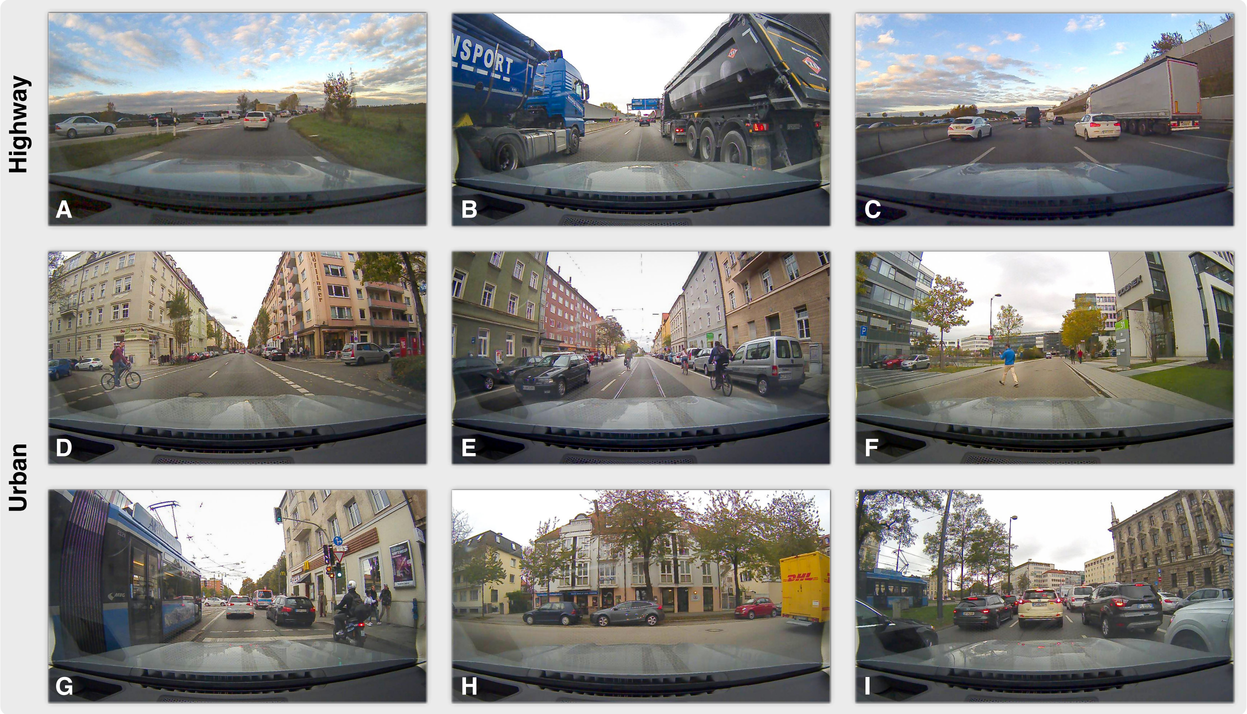


Figure 1.1: Autonomous vehicles have to cope with various situations, such as the following ones we recorded during a single test drive in Germany on October 25, 2018, from 1 p.m. to 6 p.m. (A) Entering a highway. (B) Sudden, close lane change of a truck into the own lane. (C) Driving on a highway with dense traffic. (D) Cyclist who unexpectedly crosses the road. (E) Multiple cyclists in different lanes. (F) Pedestrian who is jaywalking. (G) Interacting with trams and motorcycles. (H) Turning at an intersection despite occlusions. (I) Driving in dense urban traffic.

casualties have been reported in 2015 [2], while a total of $3,095,373$ million km has been driven [3]. Thus, the probability of an accident involving casualties was $5.64 \cdot 10^{-7}$ per km. However, it is challenging to optimize autonomous driving systems so that their collision risk converges to such a low value. Alternatively, 7.99 billion km can be driven to demonstrate that the system is better by 20 % than the human driver fatality rate (with 95 % confidence) [1]. Yet, a fleet of $1,000$ autonomous test vehicles requires about 12.2 years for this distance when driven 24 hours every day at an average speed of 75 km per hour. Instead of real driving, we can use driving simulations, but they are also no remedy due to the vast amount of test cases [4]. Overall, conventional approaches are not sufficient to achieve desired levels of safety. In consequence, we require a paradigm shift to new solutions that can eliminate even residual collision risks of autonomous vehicles in a fail-safe manner. One of the main challenges in avoiding collisions is the unknown future behavior of other traffic participants, such as of surrounding vehicles and pedestrians. Only if considering the future evolution of the traffic scenario, autonomous vehicles can plan collision-free motions.

This dissertation proposes novel solutions for predicting all future behaviors of other traffic participants, which enables safe motion planning for autonomous vehicles. In particular, we develop a formal prediction that captures all possible evolutions of any traffic scenario. Subsequently, we demonstrate that this prediction can be used online by autonomous vehicles to compute safe fallback plans that prevent causing accidents. Furthermore, we identify behaviors of other traffic participants that lead to safety gaps in existing motion planners. All our solutions are designed both for vehicles without a human driver and for driver assistance systems.

1.1 Motion safety

Approaches for safe motion planning often originate from concepts applied to mobile robots. Yet, the application to autonomous road vehicles poses specific challenges but also allows certain optimizations, e. g., due to the structure of traffic scenarios and traffic rules. In this dissertation, we review only selected works on general robotics and mainly focus on approaches applicable to autonomous vehicles in traffic environments. Note that we do not rely on an explicit communication between traffic participants; for discussions on connected vehicles, we refer to [5–7].

1.1.1 Safety specifications

Many works on safe motion planning have been proposed, even though they often do not describe or fulfill desired safety properties. In fact, it is usually relatively easy to find an acceptable behavior for another traffic participant so that the proposed system eventually causes a collision. In contrast, even residual collision risks can be eliminated by using formal methods [8–13]. These methods propose a safety specification and ensure that this formal specification is fulfilled. In line with these works, we want to encourage a strict handling of safety. We believe that it is important to carefully specify safety properties and to only claim what can be proven.

To begin with, we need to determine which situations we regard as safe and which future evolutions we want to consider. *Absolute safety* requires that the ego vehicle, i. e., the autonomous vehicle under control, is *not involved* in any accident. This is clearly not possible, since other traffic participants can easily cause collisions inevitable for the ego vehicle, e. g., by crashing into the back of the ego vehicle. In contrast, self-inflicted accidents can and should be eliminated. Thus, sophisticated safety specifications require that the ego vehicle does *not cause* any accident:

- *Passive safety* [8] requires the ego vehicle to be at rest when a collision occurs. Passive safety is suitable for mobile robotics, but has limited applicability to road traffic, since it does not require the ego vehicle to stop such that other traffic participants are still able to avoid collisions.

- *Legal safety* [9,11] requires the ego vehicle to be collision-free against all legal behaviors of other traffic participants. The legal behaviors are defined based on traffic rules, i. e., other traffic participants are allowed to perform any behavior that conforms with traffic rules. If another traffic participant severely violates traffic rules and hence a collision occurs, the ego vehicle is not considered responsible for the collision.

- *Responsibility-Sensitive Safety* [14] requires the ego vehicle to perform proper responses in case longitudinal or lateral safe distances to other traffic participants are violated. The safe distances and proper responses are defined based on common sense rules and behaviors.

- *Not-at-fault driving* [15] requires the ego vehicle to be collision-free against other traffic participants while moving and allows the ego vehicle to be at rest anywhere. In this specification, the behaviors of other traffic participants are not specified, but required as input.

To fulfill any of these safety specifications, a prediction of other traffic participants is required. This prediction must provide at least all the future behaviors that need to be considered according to the safety specification. However, we need to decide which future behaviors must be accounted for and which can be disregarded.

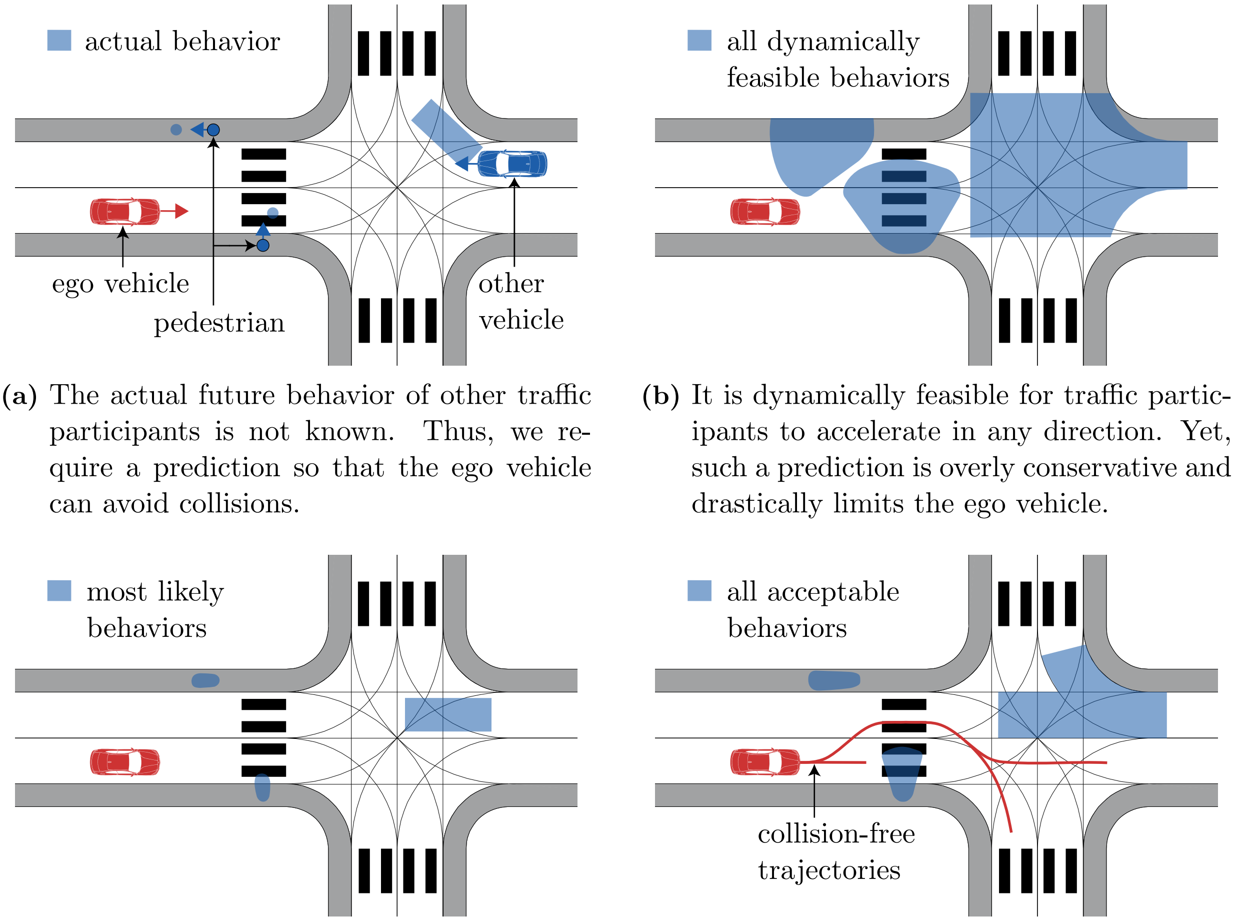
1.1.2 Acceptable behaviors

Let us discuss different perspectives on which behaviors to consider using the scenario in Figure 1.2 as a running example.

(a) Actual behavior Ideally, we want to exactly know the behavior each other traffic participant will perform in the future, i. e., its *actual behavior* (cf. Figure 1.2a). This is not possible, since even the traffic participant itself might not be aware of its intended behavior and can also suddenly change its behavior.

Thus, we need to make a sophisticated forecast about the future behaviors of other traffic participants.

(b) Dynamically feasible behaviors On the one hand, we can consider traffic environments as adversarial. Then, possible evolutions of the environment are *all dynamically feasible*


■ actual behavior
ego vehicle
pedestrian
other vehicle
■ all dynamically feasible behaviors
■ most likely behaviors
■ all acceptable behaviors
collision-free trajectories

(a) The actual future behavior of other traffic participants is not known. Thus, we require a prediction so that the ego vehicle can avoid collisions.

(b) It is dynamically feasible for traffic participants to accelerate in any direction. Yet, such a prediction is overly conservative and drastically limits the ego vehicle.

(c) The most likely behaviors of other traffic participants might be to continue with constant velocity. Yet, when only relying on such a prediction, the ego vehicle might cause a collision if the actual future behavior is missed.

(d) Acceptable behaviors are those allowed by traffic rules, e. g., pedestrians may walk but not run across the road and other vehicles respect the right of way. Such a prediction allows the ego vehicle to obtain collision-free trajectories.

Figure 1.2: The ego vehicle approaches an oncoming vehicle and pedestrians who are walking on the sidewalk (cf. initial states in Figure 1.2a). To obtain safe motions for the ego vehicle, which future behaviors of the other traffic participants do we need to anticipate?

behaviors of every traffic participant (cf. Figure 1.2b). These behaviors can be computed as described in [16–18].

Yet, this prediction is overly conservative, e. g., since pedestrians are considered to suddenly jump on the road at any time. As a result, the maneuverability of the ego vehicle is often limited too drastically. Even passing an oncoming vehicle on a two-lane road would not be possible safely, since oncoming vehicles are always allowed to drive into the lane of the ego vehicle (if accounting for all dynamically feasible behaviors).

(c) Most likely behaviors On the other hand, we can try to only consider *most likely behaviors* of other traffic participants (cf. Figure 1.2c). Using probabilistic methods, we can infer behaviors that other traffic participants are most likely performing currently or will perform in the near future [19].

The prediction of most likely behaviors can be used to optimize the comfort of motions for the ego vehicle. Yet, the safety of planned motions may be derogated, if the prediction does not contain the actual future behavior of other traffic participants, e. g., since rather unlikely behaviors have been disregarded. In fact, if traffic participants behave differently than predicted, the ego vehicle may no longer be able to avoid a collision.

(d) Acceptable behaviors We believe that the solution for achieving safety lies in between the above two perspectives (b) and (c). In particular, future behaviors should be predicted considering traffic rules; otherwise, the prediction often either misses the actual behavior or is too conservative, which would lead to an overestimation or underestimation of the actual collision risk [9, Sec. IV-A]. For example, we do not have to expect other vehicles to drive 120 km per hour in urban areas, but we have to expect critical behaviors allowed by traffic rules, such as emergency braking, full acceleration, or rapid lane changes. Note that in court proceedings, traffic rules are essential in deciding who is responsible for an accident.

Thus, we request that all behaviors allowed according to traffic rules should be considered by the ego vehicle (cf. legal safety of Section 1.1.1). Let us denote these behaviors as the *acceptable behaviors* of other traffic participants (cf. Figure 1.2d). In other words, acceptable behaviors are all dynamically feasible behaviors that do not violate traffic rules.

Yet, some behaviors are forbidden by traffic rules but common for human drivers, such as slight overspeeding. Thus, the specification of acceptable behaviors should be parameterizable to user preferences, and the prediction should be able to automatically adapt to other traffic participants violating traffic rules.

Our specification of acceptable behaviors (cf. formal specification later in Chapter 3) can now be included in the desired safety specification (cf. Section 1.1.1). As a result, safety can be ensured if the prediction includes all acceptable behaviors.

1.2 Overview of related literature

This section provides a brief overview of the state of the art without discussing individual works. Yet, we provide references to thorough literature reviews in later sections of this dissertation or in existing surveys. Section 1.2.1 introduces the different categories of prediction methods, and Section 1.2.2 introduces concepts to ensure safety based on such a prediction.

1.2.1 Prediction of traffic participants

Approaches that predict future behaviors of other traffic participants can be categorized according to different aspects:

- by the objective: the most likely behavior (e. g., for motion planners requiring high accuracy for long prediction horizons), all dynamically feasible behaviors (e. g., for ensuring safety of mobile robots in environments shared with pedestrians), all acceptable behaviors (e. g., for ensuring safety of autonomous vehicles in traffic environments), or for other specific use cases (e. g., computationally efficient for driver assistance systems on tightly restricted hardware);

- by the type of considered traffic participants: motorized vehicles, pedestrians, or cyclists;
- by the applied methodology: e. g., Bayesian filtering, neural networks, or reachability analysis;
- by the abstraction of underlying motion models (also known as evolution or propagation models), if applicable: e. g., physics-based, maneuver-based, or interaction-aware;
- by the consideration of traffic rules: e. g., explicitly formalized as constraints or learned by observations; or
- by the type of the prediction result and its representation: a classification into maneuvers (which represent intentions), a single trajectory, a finite number of trajectories, a probability distribution (e. g., continuous distribution over state variables or discretized distribution in an occupancy grid), or set-based (e. g., occupancy polygons, velocity intervals, or other bounded sets of states).

The literature on predicting other traffic participants is extensively reviewed in Section 3.1 with regards to these aspects. Section 3.2 puts particular emphasis on predicting pedestrians and Section 3.3 on considering interaction between traffic participants. A contemporary survey on motion prediction of road vehicles does not exist; we refer to [19] for the most recent one. For surveys on the prediction of pedestrians and their interaction with the ego vehicle, we refer to [20–22].

The vast amount of works on predicting other traffic participants reveals that research in this area has been intensifying in the last few years. Especially, many approaches for predicting most likely behaviors have been proposed. However, a sophisticated prediction of detected and undetected traffic participants containing all their acceptable behaviors based on traffic rules does not yet exist.

1.2.2 Safe motion planning

To ensure safety of motions of the ego vehicle, we can make use of different techniques: (a) assess the risk of the traffic situation, (b) plan safe trajectories for the ego vehicle, (c) verify the safety of planned trajectories, and (d) falsify the safety of a motion planner. In all these areas, predicting other traffic participants is required.

(a) Risk assessment Risk assessment or threat assessment determines the criticality of the current traffic situation for the ego vehicle [19, 23]. It can be used to trigger warnings or interventions in driver assistance systems or to make decisions in motion planning that are the least critical. Since the criticality highly depends on which options are available for the ego vehicle, a prediction of the future behavior of other traffic participants is usually required.

Prominent risk assessment approaches are reviewed in Section 4.1, and we refer to [23] for an extensive, contemporary survey. While most approaches determine a probabilistic risk measure, an upper bound of the risk is usually not provided. However, such a worst-case analysis is required for the ego vehicle to remain safe even if the traffic situation evolves in the worst possible way.

(b) Motion planning Motion planning for autonomous vehicles is usually separated into the prediction of other traffic participants and subsequent decision making and trajectory planning for the ego vehicle [24–26]. Thus, the future behavior of other traffic participants is regarded as independent from the decision of the ego vehicle during one planning cycle, and the interaction is implicitly modeled by replanning. In contrast, interactive motion planning directly includes the prediction in the decision making [27]. Overall, predicting other traffic participants is an integral part of motion planning.

Existing motion planning approaches are reviewed in Section 4.2, and we refer to [24] for the most contemporary survey. Most planning approaches assume a given prediction. However, when considering all acceptable behaviors, the solution space for the ego vehicle often becomes small and convoluted, which poses challenges for trajectory planners. In addition, the interplay between trajectory planning and prediction needs to be considered for consecutive planning cycles.

(c) Safety verification Formal verification allows us to mathematically prove that the ego vehicle always complies with a desired specification (cf. Section 1.1.1), i. e., the ego vehicle always remains outside of unsafe sets [13].

Approaches for safety verification make use of different techniques and are reviewed in Sections 4.2 and 4.3. Yet, most of these works require a prediction that provides all acceptable future behaviors of other traffic participants, since these behaviors are regarded as the time-variant unsafe sets according to the desired safety specification. Furthermore, it is unclear how the ego vehicle can react if traffic participants perform behaviors not considered to be acceptable and whether the approaches generalize to situations that have not been tested or considered during development.

(d) Safety falsification Falsification is a testing method [28] and aims to disprove a desired property of a given system [29, 30]. Instead of proving the safety as in safety verification, we challenge the system by trying to find counter-examples. Such counter-examples constitute of a valid behavior for another traffic participant that led to a safety violation of the ego vehicle.

Approaches for safety falsification are reviewed in Section 4.4, and we refer to [28, 30] for contemporary surveys. However, existing approaches are often computationally expensive or do not exploit specific domain knowledge.

1.3 Contributions

This dissertation proposes a novel set-based prediction that encloses all acceptable behaviors of both detected and undetected traffic participants (see Figure 1.3 for an example). As a result, other traffic participants may perform any acceptable behavior in the future, but the prediction is guaranteed to already contain this behavior; i. e., the prediction is over-approximative. At the same time, all behaviors not contained in the prediction are guaranteed to be not acceptable behaviors; i. e., the prediction is bounded. In other words, the actual future behavior is included in the prediction with a probability of 1 and is not included in the prediction with a probability of strictly 0 (under the premise that other traffic participants are allowed to perform any acceptable behavior).

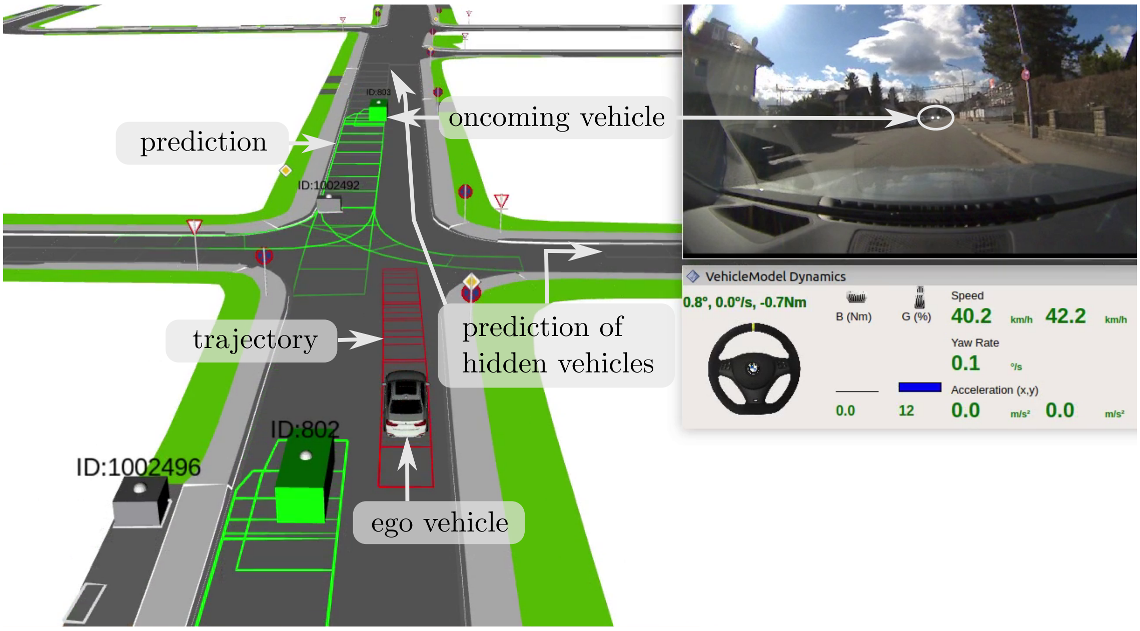
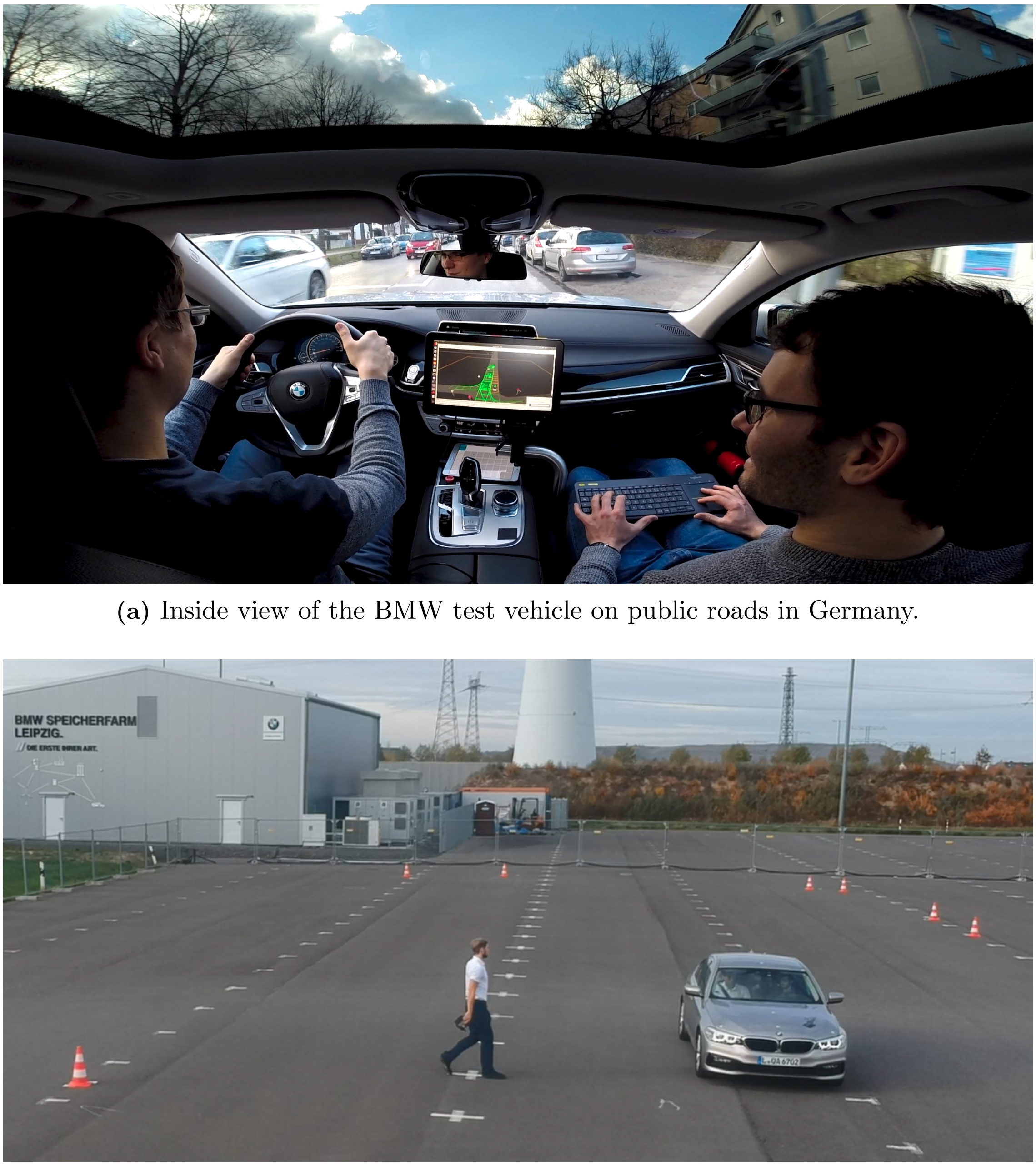


Figure 1.3: The ego vehicle is approaching an intersection. For the detected, oncoming vehicle, we predict all acceptable behaviors, including turning at the intersection. Due to limited observability, we also determine where vehicles could be potentially hidden and predict all their acceptable behaviors. Based on the prediction result, the ego vehicle can obtain a collision-free trajectory.

Previous works [11, 31] presents fundamentals for the reachability analysis of other traffic participants. Reachability analysis computes the set of states that can be reached by a traffic participant (cf. formal introduction later in Section 2.2) and thus can be used to compute all acceptable behaviors. This dissertation relaxes some simplifications made in the previous works and proposes various, novel solutions:

- We ensure an over-approximation by additionally considering occlusions, all measurement uncertainties, changes of the heading of vehicles, and reversing of vehicles.
- We reduce the over-approximation by improving existing models; developing new models, e. g., for interaction and the minimum turning radius; and formalizing traffic rules, e. g., on the safe distance between vehicles, on priorities at intersections, and regarding pedestrians crossing the road.
- We validate our prediction in several real-world experiments to demonstrate that the proposed prediction is real-time capable and applicable for different use cases. Figure 1.4 shows exemplary experiments with our test vehicles.

In addition, we propose a specification for the prediction that defines the acceptable behaviors based on dynamic constraints and traffic rules. In consequence, the prediction enables motion planners to fulfill desired safety specification. Our specification also allows users to tune the degree of conformity to traffic rules. Furthermore, behaviors that are not acceptable (according to the specification and are thus not necessarily included in our prediction) get included as soon as another traffic participants actually performs such a behavior, as postulated by [9]. Therefore, we propose a constraint management that makes the prediction



(a) Inside view of the BMW test vehicle on public roads in Germany.

(b) Overview of a fenced test track, in which the BMW test vehicle avoids collisions with pedestrians.

Figure 1.4: Real-world experiments of the proposed set-based prediction and safe motion planning.

robust against traffic participants violating traffic rules, large measurement uncertainties, and if information about the environment is missing.

Since a method to predict all acceptable behaviors is now available, this dissertation subsequently examines the impact of the prediction on ensuring safety and answers open questions (cf. Section 1.2.2) like:

- What applications does set-based prediction have in motion planning and do they generalize to arbitrary traffic situations?

- When is the latest time at which the ego vehicle needs to react or at which the driver assistance system needs to intervene?
- Given a motion planning framework, how can we construct a safety layer that ensures safety over consecutive planning cycles and copes with convoluted solution spaces? How useful are motions that result from this safety layer?
- How can we ensure safety despite traffic participants violating traffic rules?
- Where are safety gaps of existing motion planners and which future behaviors of other traffic participants lead to them? How can we obtain these counter-examples in a computationally efficient way when performing black-box testing (i. e., without knowledge about the system under test)?

1.4 Outline

The contributions of this dissertation have been developed in eleven publications [61–71]. All of them have been published in peer-reviewed, international journals or conferences.

1.4.1 Included publications

This cumulative dissertation includes a selection of those publications. The author of this dissertation is the first author or one of the first authors of each included publication. Together with the included reprint of each publication, the content of the publication is summarized and related to the other included publications. As required by the regulations for the award of doctoral degrees, the main contributions of M. K., the author of this dissertation, are listed. Nonetheless, also the co-authors have significantly contributed to each publication, which is gratefully acknowledged, but their individual contributions are not listed. The contributions of the publication compared to the literature are discussed within the publication.

Prior to presenting the publications, the underlying, general methodology is briefly introduced in Chapter 2. The subsequently included publications are structured in two parts: the set-based prediction of traffic participants in Chapter 3 and the applications of this prediction to safe motion planning in Chapter 4.

Chapter 3 presents the theory of the set-based prediction as well as experimental results. This chapter is organized as follows. Section 3.1 presents [70], which holistically describes the set-based prediction of other traffic participants. In particular, this journal article defines the motion models for all different types of traffic participants, considers occlusions, and presents real-world experiments. Section 3.2 details the prediction of pedestrians by presenting [67], and Section 3.3 extends the prediction to consider interactions between vehicles by presenting [63].

Chapter 4 presents methods for safe motion planning that are enabled by the set-based prediction and is organized as follows. Section 4.1 determines the maximum Time-to-React for risk assessment of autonomous vehicles by presenting [66]. Section 4.2 presents [71], which develops a safety layer for existing motion planning frameworks to prevent autonomous vehicles from causing accidents. Thus, this journal article demonstrates the effectiveness of formal safety verification for autonomous driving on real-world data, resulting in legally safe and not overly conservative motions. Section 4.3 investigates the influence on safety when traffic participants violate traffic rules that have been an assumption for predicting their future behavior, i. e., when traffic participants perform behaviors that have not been considered to be acceptable, by presenting [64]. Section 4.4 efficiently tests and falsifies the safety of motion planners using rapidly-exploring random trees by presenting [69].

Chapter 5 closes this dissertation by discussing conclusions and suggestions for future research.

1.4.2 Excluded publications

The following publications of the author are within the scope of this dissertation but not included:

[61] presents SPOT, which is a publicly available MATLAB toolbox for the set-based prediction of traffic participants.[1]

[68] presents the integration of [61] as a safety layer into existing motion planning frameworks.

[62] presents CommonRoad, which is a benchmark suite for trajectory planners. The composable benchmarks consist of traffic scenarios, cost functions, and vehicle models. As a result, CommonRoad enables reproducible experiments and comparable results.[2]

[65] presents the conversion of road networks from OpenDRIVE to lanelets, which are both commonly used map formats, as publicly available Python modules.[2]

Lastly, we acknowledge the students [72–89], which have completed their Bachelor Thesis or Master Thesis at the Technical University of Munich under supervision of the author of this dissertation and have thereby contributed to this dissertation.

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

[2] available at commonroad.in.tum.de

2 Preliminaries for Prediction and Motion Planning

In this chapter, we introduce methodology that is relevant for prediction and motion planning. First, we mathematically describe the problem statements of this dissertation in Section 2.1. Please note that these problem statements are defined for our purposes but can also be defined differently. Subsequently, we introduce the admissible and reachable sets for the reachability analysis in Section 2.2, and we describe our model of the traffic environment in Section 2.3.

2.1 Problem statements

Let us introduce the state $\boldsymbol{x}^{(p)} \in \mathbb{R}^n$ and a set of states $\mathcal{X}^{(p)} \subseteq \mathbb{R}^n$ of a traffic participant $p \in \mathcal{P}$, where $\mathcal{P}$ is the set of all other traffic participants, which may be detected but can also be occluded. We distinguish between different types of states:

$\boldsymbol{x}^{(p)}_{\text{actual}}(t)$ denotes the actual state, i. e., the ground-truth, at time t.

$\mathcal{X}^{(p)}_{\text{meas}}(t)$ denotes the set of states obtained from an uncertain measurement at t that contains at least the actual state, i. e., $\boldsymbol{x}^{(p)}_{\text{actual}}(t) \in \mathcal{X}^{(p)}_{\text{meas}}(t)$ (cf. Figure 2.1).

$\mathcal{X}^{(p)}_{\text{pred}}(t;t_0)$ denotes the set of states predicted for time t based on information at an initial time t_0.

$\boldsymbol{x}^{(p)}_{\text{plan}}(t;t_0)$ denotes the state at time t resulting from executing a (dynamically feasible) trajectory that was planned based on information at t_0.

$\mathcal{X}^{(p)}_{\text{accept}}(t;t_0)$ denotes the set of states at time t that would result from performing all acceptable behaviors when starting at t_0.

To denote that a state describes the ego vehicle and not another traffic participant, we use $\boldsymbol{x}^{(\text{ego})}$, which can be of the same types as introduced above for p. We further introduce the operator $\texttt{occ}\big(\boldsymbol{x}^{(p)}\big) : \mathbb{R}^n \to \texttt{Pow}(\mathbb{R}^2)$ returning the set of points in the two-dimensional Cartesian frame that are occupied by the traffic participant p (or the ego vehicle if using $\boldsymbol{x}^{(\text{ego})}$ instead of $\boldsymbol{x}^{(p)}$), where $\texttt{Pow}(\mathbb{R}^2)$ denotes the power set of $\mathbb{R}^2$. For a set of states $\mathcal{X}^{(p)}$, the occupancy operator is defined as $\texttt{occ}\big(\mathcal{X}^{(p)}\big) := \{\texttt{occ}\big(\boldsymbol{x}^{(p)}\big) \,|\, \boldsymbol{x}^{(p)} \in \mathcal{X}^{(p)}\}$.

Problem statement 1 (Prediction) Based on the measurement $\mathcal{X}^{(p)}_{\text{meas}}(t_0)$, the goal of the prediction for traffic participant p is to determine a set $\mathcal{X}^{(p)}_{\text{pred}}(t;t_0)$ for a desired future time $t \geq t_0$ that over-approximates all acceptable behaviors, i. e.,

$$\forall t \geq t_0 : \mathcal{X}^{(p)}_{\text{pred}}(t;t_0) \supseteq \mathcal{X}^{(p)}_{\text{accept}}(t;t_0),$$

while containing as little over-approximation as possible, i. e., the size of $\mathcal{X}^{(p)}_{\text{pred}}(t;t_0)$ shall be minimal.

As a result, a prediction solving Problem statement 1 is guaranteed to contain the actual state for any future time, i. e., $\forall t \geq t_0 : \boldsymbol{x}^{(p)}_{\text{actual}}(t) \in \mathcal{X}^{(p)}_{\text{pred}}(t;t_0)$, if the other traffic participant p is only performing acceptable behaviors. However, if p is misbehaving, i. e., $\exists t_1 > t_0 : \boldsymbol{x}^{(p)}_{\text{actual}}(t_1) \notin \mathcal{X}^{(p)}_{\text{accept}}(t_1;t_0)$, the actual state might be missed by the prediction. Thus, we require a constraint management that adapts the prediction.

Problem statement 2 (Constraint management) If we detect at $t_1 > t_0$ that a traffic participant p is performing a behavior that is not considered acceptable, i. e., $\boldsymbol{x}^{(p)}_{\text{actual}}(t_1) \notin \mathcal{X}^{(p)}_{\text{accept}}(t_1;t_0)$, the goal of the constraint management is to modify the prediction parameters so that this unacceptable behavior gets included in subsequent prediction results for this traffic participant, i. e.,

$$\forall t \geq t_1 : \boldsymbol{x}^{(p)}_{\text{actual}}(t) \in \mathcal{X}^{(p)}_{\text{pred}}(t;t_1),$$

under the assumption that p does not perform other unacceptable behaviors (i. e., p may continue to perform the detected unacceptable behavior or may perform any acceptable behavior).

Problem statement 3 (Motion planning) The goal of the motion planning for the ego vehicle is to determine a trajectory of states that are collision-free against the prediction of all other traffic participants from t_0 until the final planning time t_f, i. e.,

$$\forall t \in [t_0, t_f], \forall p \in \mathcal{P} : \mathtt{occ}\big(\boldsymbol{x}^{(\text{ego})}_{\text{plan}}(t;t_0)\big) \cap \mathtt{occ}\big(\mathcal{X}^{(p)}_{\text{pred}}(t;t_0)\big) = \emptyset.$$

If a trajectory is planned such that it satisfies Problem statement 3 and the utilized prediction satisfies Problem statement 1, the ego vehicle will not cause a collision with any other traffic participant according to our safety specification of legal safety (cf. Section 1.1.1), since the prediction over-approximates all acceptable behaviors of other traffic participants. Also note that Problem statement 3 is applicable for both planning and verification (cf. Section 1.2.2).

Problem statement 4 (Risk assessment) The goal of the risk assessment at the current time t' is to determine the latest point in time $t_0 \in [t', t' + t_f]$ at which Problem statement 3 can still be solved.

Problem statement 5 (Falsification) The goal of the falsification is to determine a trajectory of states from t_0 until t_f for a specific traffic participant p_1 (using only acceptable behaviors) so that a trajectory planned by the motion planner of the ego vehicle subsequently at $t_1 \in [t_0, t_f]$ eventually causes a collision, i. e.,

$$\begin{aligned}&\forall t \in [t_0, t_f] : \boldsymbol{x}^{(p_1)}_{\text{plan}}(t;t_0) \in \mathcal{X}^{(p_1)}_{\text{accept}}(t;t_0) \wedge \\ &\exists t \in [t_1, t_f] : \mathtt{occ}\big(\boldsymbol{x}^{(\text{ego})}_{\text{plan}}(t;t_1)\big) \cap \mathtt{occ}\big(\boldsymbol{x}^{(p_1)}_{\text{plan}}(t;t_0)\big) \neq \emptyset.\end{aligned}$$

2.2 Reachability analysis

To solve these problems, we make use of reachability analysis [32–34]. In addition to the set of states $\mathcal{X}^{(p)} \subseteq \mathbb{R}^n$, let us introduce the set of inputs $\mathcal{U}^{(p)} \subseteq \mathbb{R}^m$.

Definition 1 (Model $\boldsymbol{M}$) A model $M^{(p)}$ for the dynamics of traffic participant p is defined as the tuple $M^{(p)} := \langle \boldsymbol{f}_M^{(p)}, \mathcal{X}_M^{(p)}, \mathcal{U}_M^{(p)} \rangle$, where $\boldsymbol{f}_M^{(p)}$ is the right-hand side of the differential equation describing the motion of a traffic participant by

$$\dot{\boldsymbol{x}}^{(p)}(t) = \boldsymbol{f}_M^{(p)}\big(\boldsymbol{x}^{(p)}(t), \boldsymbol{u}^{(p)}(t)\big), \tag{2.1}$$

the set of admissible states $\mathcal{X}_M^{(p)}(t)$ bounds the states, i. e., $\forall t : \boldsymbol{x}^{(p)}(t) \in \mathcal{X}_M^{(p)}(t)$, and the set of admissible inputs bounds the inputs, i. e., $\forall t : \boldsymbol{u}^{(p)}(t) \in \mathcal{U}_M^{(p)}(t)$.

When starting at an initial state $\boldsymbol{x}^{(p)}(t_0)$ and using an input trajectory $\boldsymbol{u}^{(p)}(\cdot)$, a possible solution of (2.1) at time $t \geq t_0$ is denoted by $\boldsymbol{\chi}^{(p)}\big(t; \boldsymbol{x}^{(p)}(t_0), \boldsymbol{u}^{(p)}(\cdot)\big)$.

Definition 2 (Reachable set $\boldsymbol{\mathcal{R}}$) The reachable set $\mathcal{R}^{(p)}$ of model $M^{(p)}$ is the set of states that are reachable at time $t \geq t_0$ from the initial set $\mathcal{X}^{(p)}(t_0)$ when applying all admissible inputs $\mathcal{U}_M^{(p)}(t)$ while staying within $\mathcal{X}_M^{(p)}(t)$:

$$\mathcal{R}^{(p)}(t; M^{(p)}, \mathcal{X}^{(p)}(t_0)) := \Big\{ \boldsymbol{\chi}^{(p)}\big(t, \boldsymbol{x}^{(p)}(t_0), \boldsymbol{u}^{(p)}(\cdot)\big) \,\Big|\, \boldsymbol{x}^{(p)}(t_0) \in \mathcal{X}^{(p)}(t_0), \forall t^\star \in [t_0, t] : \\ \boldsymbol{\chi}^{(p)}\big(t^\star; \boldsymbol{x}^{(p)}(t_0), \boldsymbol{u}^{(p)}(\cdot)\big) \in \mathcal{X}_M^{(p)}(t^\star), \boldsymbol{u}^{(p)}(t^\star) \in \mathcal{U}_M^{(p)}(t^\star) \Big\}.$$

In Chapter 3, we use reachability analysis for prediction to determine all possible future states that can be reached by a traffic participant when performing any acceptable behavior (cf. Problem statement 1). In Sections 4.1 and 4.2, we use reachability analysis for motion planning to determine the drivable area of the ego vehicle and to ensure that a planned trajectory of the ego vehicle is collision-free (cf. Problem statements 3 and 4). In Section 4.4, we sample only a few future states instead of determining all reachable states (cf. Problem statement 5).

Key challenges in performing reachability analysis are (a) developing an appropriate model of a real system, which includes deriving differential equations and defining the set of admissible states and the set of admissible inputs, (b) solving these differential equations, and (c) choosing an efficient set representation.

2.3 Environment model

The most important input for the prediction and motion planning is a model of the traffic environment. This environment model contains current information about the map and other traffic participants, as illustrated in Figure 2.1.

The map is usually generated offline and describes the road, which is partitioned in lanes and specific areas for pedestrians. The map may also contain information about traffic rules, such as speed limits or priorities at intersections.

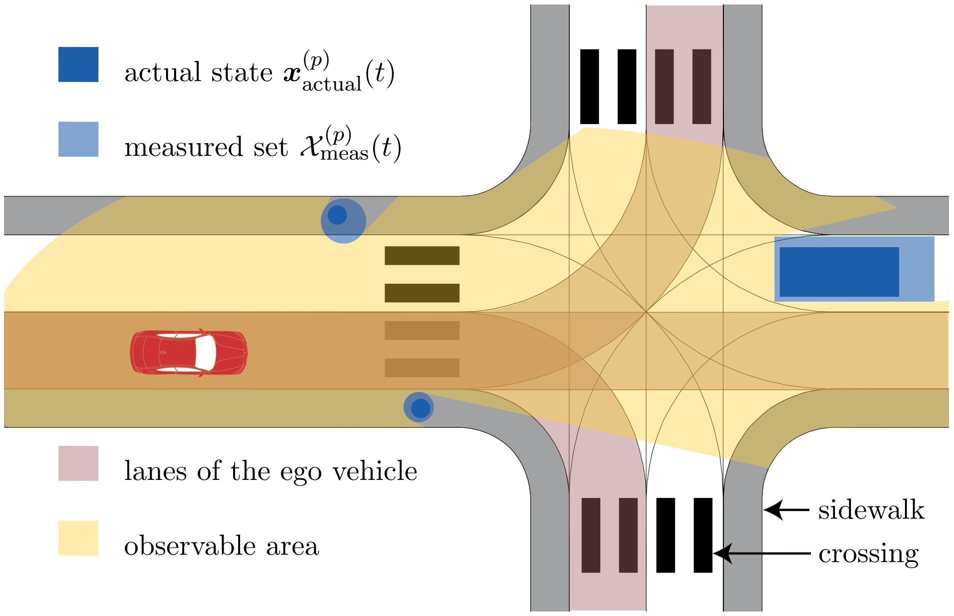

actual state $\boldsymbol{x}^{(p)}_{\text{actual}}(t)$
measured set $\mathcal{X}^{(p)}_{\text{meas}}(t)$
lanes of the ego vehicle
observable area
sidewalk
crossing

Figure 2.1: Our environment model describes the road and all other traffic participants within the observable area. The set $\mathcal{X}^{(p)}_{\text{meas}}(t)$ estimating the current state of a traffic participant contains the actual state $\boldsymbol{x}^{(p)}_{\text{actual}}(t)$ and allows for bounded measurement uncertainties.

The information about other traffic participants is gathered online by on-board sensors of the ego vehicle or obtained by communication with infrastructure, for example. However, the complete environment can usually not be detected, and we need to handle restricted observability. Thus, the environment is classified into either unobservable areas or observable areas (see Figure 2.1). All objects present within observable areas are included in our environment model. For each observed traffic participant, the model contains its type and its current state. The type can be passenger car, truck, bus, motorcycle, bicycle, pedestrian, static, or a combination of these types. For the state estimation, we require that it contains the actual state and that measurement uncertainties are strictly bounded (see Figure 2.1). Our requirements on the object detection are common and are already mostly met by contemporary approaches, such as [35–44], especially when using set-based observers [45, 46].

3 Set-based Prediction of Traffic Participants

In this chapter, we develop a formal set-based prediction that solves Problem statement 1, i. e., our prediction contains all acceptable future behaviors of detected and undetected traffic participants in arbitrary traffic environments. Section 3.1 fully introduces the set-based prediction of other traffic participants, proposes an algorithm to tackle occlusions, and presents real-world experiments. Section 3.2 focuses on the prediction of pedestrians, and Section 3.3 focuses on the interaction between vehicles.

3.1 TIV 2020: Set-based Prediction of Traffic Participants Considering Occlusions and Traffic Rules [70]

Summary A major challenge in provably safe motion planning is the unknown future behavior of other traffic participants. We propose a set-based prediction that enables the ego vehicle to anticipate all acceptable behaviors of other traffic participants. Therefore, we perform reachability analysis based on formalized traffic rules and nondeterministic models, which over-approximate the real dynamics of vehicles and pedestrians. Each model is formally defined, and its reachable set is computed to efficiently obtain the maximum possible positions and velocities. As prediction features, we use longitudinal and lateral dynamics, the motion history, and contextual information.

Yet, many traffic participants cannot be predicted directly, since they are hidden due to occlusions. To capture this risk, we create phantom traffic participants at all safety-relevant boundaries of the field of view of the ego vehicle. These phantom traffic participants are then predicted together with the detected traffic participants.

For the first time, our set-based prediction is validated in test vehicles. Real-world experiments in various traffic situations demonstrate that our over-approximative prediction is applicable for both online verification and fail-safe motion planning. We perform online verification in the presence of pedestrians in a parking environment. As a result, the ego vehicle only executes trajectories that are collision-free against all acceptable behaviors of pedestrians. In further experiments, we execute our prediction while driving on public roads. Our constraint management successfully deals with traffic participants violating traffic rules, large measurement uncertainties, and incomplete environment models (cf. Problem statement 2). Even in congested, complex traffic situations, our approach enables the ego vehicle to obtain collision-free fail-safe trajectories.

Contributions of M. K. M. K. developed the legal specification, the algorithm to consider occlusions, the abstractions that extend previous work, and the constraint management.

M. K. designed and conducted the experiments (together with C. P., S. K., and F. S.). M. K. evaluated the experiments. M. K. wrote the article.

Journal article ©2020 IEEE. Reprinted, with permission, from Markus Koschi and Matthias Althoff, Set-based Prediction of Traffic Participants Considering Occlusions and Traffic Rules, IEEE Transactions on Intelligent Vehicles.

Attachments The video attachment of this publication is available at go.tum.de/812843.

Set-based Prediction of Traffic Participants Considering Occlusions and Traffic Rules

Markus Koschi and Matthias Althoff

***Abstract*—Provably safe motion planning for automated road vehicles must ensure that planned motions do not result in a collision with other traffic participants. This is a major challenge in autonomous driving, since the future behavior of other traffic participants is not known and traffic participants are often hidden due to occlusions. In this work, we propose a formal set-based prediction that contains all acceptable future behaviors of both detected and potentially hidden traffic participants. Based on formalized traffic rules and nondeterministic motion models, we perform reachability analysis to predict the set of possible occupancies and velocities of vehicles, pedestrians, and cyclists. Real-world experiments with a test vehicle in various traffic situations demonstrate the applicability and real-time capability of our over-approximative prediction for both online verification and fail-safe trajectory planning. Even in congested, complex traffic scenarios, our forecasting approach enables self-driving vehicles to never cause accidents.**

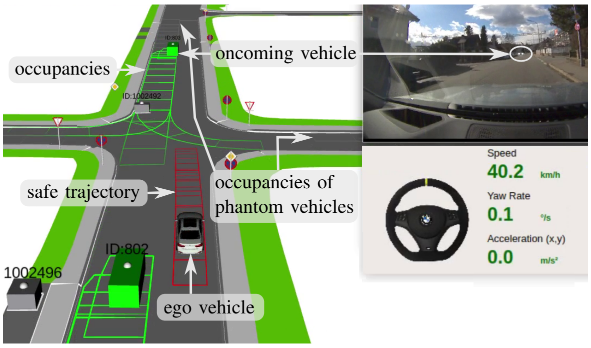


Fig. 1. Snapshot of our real-world experiments with a BMW 7 series test vehicle. The motion of the ego vehicle is provably safe if its trajectory never intersects with any predicted occupancy of detected and phantom (i. e., potentially hidden) traffic participants.

I. Introduction

By accounting for safety in a rigorous and formal manner, we verify that autonomous vehicles do not cause any accident, which is referred to as legal safety [1]–[5]. Absolute safety is not possible, since other traffic participants can easily cause inevitable collisions, e. g., by crashing into the back of an autonomous vehicle. If every traffic participant adheres to legal safety, which most traffic participants do, no collisions will occur. Related safety concepts are passive safety [6], which requires the autonomous vehicle to be at rest when a collision occurs, and Responsibility-Sensitive Safety (RSS) [7], which determines the traffic participant responsible for a collision based on safe distances for specific driving situations.

However, if traffic participants behave differently than predicted by the autonomous vehicle, a collision for which the autonomous vehicle is responsible might be inevitable. Therefore, we propose a set-based prediction that formally encloses all acceptable future behaviors of other traffic participants. A legal specification defines which behaviors are considered to be acceptable. It explicitly represents our assumptions based on traffic rules, while the degree of conformity to traffic rules can be parameterized by the user. Some people might argue that one cannot restrict acceptable behaviors; however, these behaviors are based on applicable law, and we believe that it is better to provide guarantees under these legal assumptions than to provide no guarantees (which is the case for most probabilistic approaches).

The planned motion of the ego vehicle, i. e., the autonomous vehicle under control, is safe if its motion does not intersect with any predicted occupancy of all detected and potentially hidden traffic participants. For example, consider a situation where the ego vehicle intends to turn left at an intersection but has to yield to oncoming traffic (cf. Fig. 1). Set-based prediction allows the ego vehicle to obtain a trajectory that is provably collision-free against all oncoming and crossing traffic. In [8], we have shown that this does not result in overly conservative behaviors for the ego vehicle. Our proposed method has several applications for autonomous vehicles and driver assistance systems:

a) Safe states: Based on the predicted occupancies, we can determine the maximum drivable area [9], the maximum Time-To-React [10], and the Point of No Return [11]. By additionally considering the predicted velocity, we can compute safe states for the ego vehicle, e. g., to maintain a safe distance to other vehicles [12]. To guarantee safety for an infinite time horizon, the planned motion of the ego vehicle must end in a state that is safe forever. Such invariably safe states can be determined using our set-based prediction [13].

b) Trajectory planning: Several trajectory planners for provably safe motions without being overly conservative use our prediction tool (SPOT [14]) [15]–[18] or assume the existence of a set-based prediction [19], [20].

c) Verification: Verification of a trajectory means that we check whether this trajectory complies with a given specification. Online verification of automated vehicles using set-based prediction is shown in [3], [8]. It can be extended to an anytime approach [21] and be embedded in any given vehicle framework [22]. For industrial robots, set-based prediction of human body parts has also been successfully used for verification [23].

Markus Koschi and Matthias Althoff are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany (email: *markus.koschi@tum.de* and *althoff@tum.de*).

Digital Object Identifier 10.1109/TIV.2020.3017385

A. Related work

We solely focus on motion prediction of other traffic participants [24]–[26], which is an integral part of motion planning [27]–[29] and risk assessment [24], [30]. The following related aspects are beyond the scope of this paper: extracting the information of surrounding traffic participants from sensor measurements [31]–[33], the uncertainty of these measurements [34]–[36], and implications on the prediction for connected vehicles [37], [38].

We categorize prominent early or most recent works by whether they compute *a)* a finite number of future trajectories, *b)* a probability distribution, or *c)* a bounded set of states. Since our proposed prediction considers occlusions, unlike most of the reviewed works, we subsequently present works on motion planning in the presence of occlusions[1].

a) Trajectories: Early works consider single trajectories of other traffic participants for collision avoidance [39]. To obtain a probabilistic prediction, multiple trajectory hypotheses can be weighted by probabilities obtained from Monte Carlo sampling [40]. Alternatively, intention estimation, i. e., a probabilistic classification into discrete, semantically interpretable maneuver classes, is often performed based on support vector machines [41], hidden Markov models [42], or Bayesian networks [43]–[45]; particularly for pedestrians, Gaussian process dynamical models are often used [46]. In most of these works, motion models generate a trajectory for each distinct maneuver class. In contrast, recurrent neural networks often directly predict a trajectory [47], [48]. Predicted trajectories can be compared using validation metrics [49] or similarity measures [50].

b) Probability distribution: To consider that other traffic participants have infinitely many future behaviors, we can compute a probability distribution, e. g., of kinematic variables using dynamic Bayesian networks [51]–[53]. Furthermore, neural networks have been proposed to predict most likely behaviors of vehicles on highways [54], [55], of pedestrians [56], and of cyclists [57]. For pedestrians, also linear quadratic regulator-based models are used [58]. Probability distributions can be represented as occupancy grids, which are obtained through machine learning [59]–[62] or Markov chains [63]. Overall, probability distributions can be used for motion planning [64]–[66], but they usually do not strictly bound all possible future behaviors as required for provably safe motions.

c) Bounded sets: Set-based prediction utilizes reachability analysis to compute all future behaviors of other traffic participants in accordance with the assumptions made [67]. Instead of specifying the input constraints for the reachability analysis in the assumptions, the constraints can also be estimated from Gaussian processes [68]. The work of [67] is extended in [16] by considering occlusions. Set-based prediction is also able to consider interaction between traffic participants [69] and formalized traffic rules [14], [70]. The predicted occupancy sets can also be weighted by probabilities [71], [72]

d) Occlusion: The risk from occlusions is tackled either by shrinking the field of view over the prediction horizon [73]–[76] or by introducing and predicting individual, potentially present obstacles (aka *phantom* or *virtual objects*) [1], [16], [77]–[85]. Early works considering occlusions are motion planners for mobile robots [86], [73]–[75]. Later, risk assessment systems for road vehicles have included occluded intersections [77]–[80]. In recent motion planners, a partially observable Markov decision process optimizes the behavior of the ego vehicle such that the collision risk due to occlusions is reduced [81]–[84]. In a pedestrian collision avoidance system, a partially observable Markov decision process propagates the belief states of occluded pedestrians based on reachable sets [85]. The occlusion-aware motion planner in [87] remains collision-free in specific traffic situations for which the authors have manually defined the worst-case. In contrast, the planners in [16], [88] generalize to arbitrary traffic situations, since they use a set-based prediction. In particular, [16] introduces phantom vehicles that could have right of way, and [88] extends [16] by optimizing comfort while keeping safety guarantees. Using reachability analysis, [76] guarantees passive safety for autonomous vehicles despite occlusions.

B. Contributions

This work significantly extends our previous work on set-based prediction [14], [67], [69], [70] and other previous works, especially [16], by considering 1) all safety-relevant occluded vehicles, pedestrians, and static obstacles, 2) priorities of traffic participants at intersections, 3) safe distances to the ego vehicle, 4) limited turning radii of vehicles, and 5) by validating the prediction in real-world experiments.

Overall, we present a holistic, formal prediction that enables provably safe motions for the ego vehicle. In particular, our prediction offers the following properties:

- uncertainty-aware, i. e., we consider all uncertainties from sensor measurements as well as of the future evolution of the environment;
- complete, i. e., our over-approximative prediction is guaranteed to contain any acceptable behavior;
- occlusion-aware, i. e., risks due to occlusions are considered by formally creating phantom objects;
- interaction-aware, i. e., interactions between the ego vehicle and other vehicles and between other vehicles are considered;
- considering traffic rules, i. e., restrictions due to the internationally applicable convention on road traffic [89];
- robust against traffic participants violating traffic rules, high measurement uncertainties, and incomplete environment models in the conducted experiments;
- designed for both structured and non-structured environments and not restricted to predefined behaviors;
- computes predictions for arbitrary time intervals without having to consider predictions of previous time steps; and
- real-time capable for a replanning rate of 50 Hz.

The remainder of this paper is organized as follows. Sec. II introduces the required formalization and our problem statement. In Sec. III, we describe our legal specification and

[1] By the term occlusion, we mean that the environment model of the ego vehicle misses information from non-observable parts outside of its field of view.

provide an overview of the prediction algorithm. Sec. IV presents our extension for occlusions, and Sec. V details all used models for the prediction. We continue with our constraint management in Sec. VI and evaluate our prediction by numerical and real-world experiments in Sec. VII. Finally, Sec. VIII concludes this paper and proposes future work.

II. Preliminaries

Throughout this paper, we will describe our method for the current planning cycle starting at t_0 when receiving an updated environment model from the ego vehicle. The initial time of the planning cycle before t_0 is denoted by t_{c-1}. The environment model $\Omega := \langle \mathcal{P}, \mathcal{N}, \mathcal{D}^{\mathcal{P}}, \mathcal{F} \rangle$ is formalized by its elements in the following subsections.

A. Notation

Vectors and matrices are written in bold and sets using a calligraphic font. For a vector $\boldsymbol{\nu} \in \mathbb{R}^n$, the operator $\texttt{proj}_{\square}(\boldsymbol{\nu})$ projects $\boldsymbol{\nu}$ to its element(s) $\square$. The lower and upper limits of an interval $[\nu] \subset \mathbb{R}$ are written with overlines and underlines, respectively, i. e., $[\nu] := [\underline{\nu}, \overline{\nu}]$, and the comparison operators for intervals are defined as $[\nu] > a \Leftrightarrow \underline{\nu} > a$.

The operator $\texttt{conv}(\mathcal{C}_1, \mathcal{C}_2)$ returns the convex hull of the sets $\mathcal{C}_1$ and $\mathcal{C}_2$, and $\mathcal{C}_1 \oplus \mathcal{C}_2$ denotes the Minkowski addition of $\mathcal{C}_1$ and $\mathcal{C}_2$. The set of the Boolean values is denoted by $\mathcal{B} := \{\textit{true}, \textit{false}\}$. The power set of $\mathbb{R}^n$ is denoted by $\texttt{Pow}(\mathbb{R}^n)$. A disk, i. e., a circular area, with center $[c_x, c_y]^T$ and radius r is denoted by $\mathcal{C}\big([c_x, c_y]^T, r\big) := \big\{[x, y]^T \,|\, (x - c_x)^2 + (y - c_y)^2 \leq r^2\big\}$. The 2-dimensional rotation matrix is defined as

$$\boldsymbol{R}(\alpha) := \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \tag{1}$$

B. Formalization of traffic participants

The state vector of a traffic participant in a Cartesian coordinate frame is $\boldsymbol{s}(t) := \big[x(t), y(t), v(t), \psi(t)\big]^T \in \mathbb{R}^4$ and consists of the position in x-direction and y-direction $\big[x(t), y(t)\big]^T$, the scalar velocity $v(t)$, and the heading $\psi(t)$.

The set of all traffic participants is $\mathcal{P}$. Each traffic participant $p \in \mathcal{P}$ is described by the tuple $p := \langle \boldsymbol{c}^p, \mathcal{S}_0^p, \mathcal{A}^p, \mathcal{Q}^p \rangle$, where

- $\boldsymbol{c}^p \in \mathbb{C}$ is the classification consisting of the type, which is either *ego vehicle*, *pedestrian*, or *vehicle* (with subtypes *car*, *truck*, *bus*, *motorcycle*, and *bicycle*), the attribute *detected* or *phantom* (see Sec. IV), and the attribute *dynamic* or *static*. Thus, $\mathbb{C} := \big\{\{\mathrm{ego}, \mathrm{ped}, \{\mathrm{veh} \times \{\mathrm{car}, \mathrm{truck}, \mathrm{bus}, \mathrm{motcyc}, \mathrm{cyc}\}\}\} \times \{\mathrm{detected}, \mathrm{phantom}\} \times \{\mathrm{dyn}, \mathrm{static}\}\big\}$.
- $\mathcal{S}_0^p := \big[[x_0], [y_0], [v_0], [\psi_0]\big]^T \subset \mathbb{R}^4$ is the set of uncertain initial states at t_0. Bounded measurement uncertainties can be provided by set-based observers [90], [91].
- $\mathcal{A}^p$ is the uncertain size of p. For the ego vehicle and other vehicles, we use rectangles with length $[\ell]$ and width $[w]$, and for pedestrians, we use circles with radius $[r]$. The reference point of a traffic participant is its geometric center.
- $\mathcal{Q}^p$ is the tuple of parameters for p (see Tab. I).

The superscript $\square$ in $\nu^{\square}$ denotes that variable ν describes traffic participant $\square \in \mathcal{P}$ or all traffic participants with classification $\square \subset \mathbb{C}$, e. g., we write ν^{veh} for all vehicles except the ego vehicle. For the sake of clarity, we write ν instead of $\nu^{\square}$ unless a distinction is necessary.

The operator $\texttt{occ}(\boldsymbol{s}(t), \mathcal{A}) : \mathbb{R}^4 \times \mathbb{R}^2 \rightarrow \texttt{Pow}(\mathbb{R}^2)$ returns the set of points in the two-dimensional Cartesian frame that are occupied by the traffic participant. For a set of states $\mathcal{S}(t)$, the occupancy operator is defined as $\texttt{occ}(\mathcal{S}(t), \mathcal{A}) := \{\texttt{occ}(\boldsymbol{s}(t), \mathcal{A}) \,|\, \boldsymbol{s}(t) \in \mathcal{S}(t)\}$.

To account for the limited sensor range of the ego vehicle and occlusions from other objects, we introduce the field of view:

Definition 1 (Field of view $\mathcal{F}$): The field of view $\mathcal{F} \subset \mathbb{R}^2$ is the maximum area in which all other traffic participants are guaranteed to be detected at the initial time.

C. Formalization of the road network

The road network $\mathcal{N} := \langle \mathcal{W}_{\mathrm{road}}, \mathcal{W}_{\mathrm{prio}}(t), \mathbb{D} \rangle$ describes the environment in separate layers for vehicles ($\mathcal{N}^{\mathrm{veh}}$), bicycles ($\mathcal{N}^{\mathrm{cyc}}$), and pedestrians ($\mathcal{N}^{\mathrm{ped}}$) and is formalized by its elements as follows.

Definition 2 (Allowed positions $\mathcal{W}_{\mathrm{road}}$): $\mathcal{W}_{\mathrm{road}} \subset \mathbb{R}^2$ describes all positions in the road network that the corresponding types of traffic participants may occupy.

For example, $\mathcal{W}_{\mathrm{road}}^{\mathrm{cyc}}$ can be restricted to bicycle lanes or also contain the rest of the carriageway (cf. [89, 25§1(a), 27§4]). The allowed positions $\mathcal{W}_{\mathrm{road}}^{\mathrm{ped}}$ for pedestrians consist of all sidewalks and pedestrian crossings and, if desired, other parts of the environment, e. g., parking areas or unclassified areas.

Definition 3 (Priority-based positions $\mathcal{W}_{\mathrm{prio}}$): $\mathcal{W}_{\mathrm{prio}}(t) \subset \mathcal{W}_{\mathrm{road}}$ describes the time-dependent positions that the corresponding types of traffic participants may occupy at time t without violating the priority of other traffic participants. This especially includes restrictions due to traffic lights and when turning at intersections.

In each layer[2], the road network is modeled by lanelets [93], which are atomic, interconnected, and drivable/walkable road segments:

Definition 4 (Lanelet l): A lanelet l is defined by its left and right bound, where each bound is represented by an array of points, as shown in Fig. 2a for l_1.

The bounds of a lanelet should be constructed so that the lanelet is at least as wide as the real lane; to anticipate that traffic participants slightly violate lane markings, the width of a lanelet can be enlarged by a user-defined margin. The driving direction of a lanelet is implicitly defined by its left and right bound; for pedestrian lanelets, we do not make a distinction of the driving direction. If two lanelets have a drivable/walkable connection, their relation is modeled as either longitudinally adjacent (i. e., predecessor and successor) or laterally adjacent.

[2] Instead of separate layers, one can also use the concept in Lanelet2 [92].

We construct a graph of the road network (for each of its layers), where a node represents a set of laterally adjacent lanelets depending on the two Boolean constraint parameters $b_{\text{lane}_1} \in \{\mathit{noLat}, \mathit{lat}\}$ and $b_{\text{lane}_2} \in \{\mathit{drivDir}, \mathit{anyDir}\}$:

- if $b_{\text{lane}_1} = \mathit{noLat}$, a node contains only one lanelet and no laterally adjacent lanelets (see graph in Fig. 2a);
- if $b_{\text{lane}_1} = \mathit{lat} \wedge b_{\text{lane}_2} = \mathit{drivDir}$, a node contains all laterally adjacent lanelets with the same driving direction (see graph in Fig. 2b or 2c);
- if $b_{\text{lane}_1} = \mathit{lat} \wedge b_{\text{lane}_2} = \mathit{anyDir}$, a node contains all laterally adjacent lanelets (see graph in Fig. 2d).

Two nodes are connected in the graph, if at least one lanelet in the one node is longitudinally adjacent to at least one lanelet in the other node.

Definition 5 (Driving corridor D): A driving corridor D is a union of lanelets along a path through the graph of the road network, as shown in Fig. 2.

If a lanelet or its laterally adjacent lanelets have multiple successors/predecessors, as in the case of road forks/merges, multiple driving corridors are created, e. g., l_2 is included in D_2 describing a right turn (see Fig. 2b) and also in D_3 describing a left turn (see Fig. 2c). Furthermore, each driving corridor provides a speed limit $v_{\text{speedLim}} > 0$, and the operator $\texttt{occ}(D) : D \rightarrow \texttt{Pow}(\mathbb{R}^2)$ returns the occupancy of D.

Definition 6 (All driving corridors $\mathbb{D}$): The set of all driving corridors $\mathbb{D}(b_{\text{lane}_1}, b_{\text{lane}_2})$ is obtained by performing breadth-first graph search on the graph of the road network constructed for the given values of b_{lane_1} and b_{lane_2}. The initial nodes are all nodes that contain only lanelets with no predecessor, and the goal nodes are all nodes that contain only lanelets with no successor.

Definition 7 (Corridors of a traffic participant $\mathcal{D}^p$): The set of driving corridors of traffic participant p is denoted by $\mathcal{D}^p(b_{\text{lane}_1}, b_{\text{lane}_2}) \subset \mathbb{D}(b_{\text{lane}_1}, b_{\text{lane}_2})$ and is provided by the environment model.

For example, the set of driving corridors of the vehicle in Fig. 2 can be $\mathcal{D}^p(\mathit{noLat}, \mathit{drivDir}) = \{D_1\}$ or $\mathcal{D}^p(\mathit{lat}, \mathit{drivDir}) = \{D_2, D_3\}$. When using the parameters $b^p_{\text{lane}_1}, b^p_{\text{lane}_2}$ of a traffic participant p, we only write $\mathcal{D}^p$ for brevity. Furthermore, let the forward driving corridor $\vec{\mathcal{D}}$ be the part of $\mathcal{D}$ that is not behind $\texttt{occ}(\mathcal{S}_0, \mathcal{A})$ with respect to the driving direction (cf. $\vec{\mathcal{D}}^{\text{ego}}_{\text{reach}}$ in Fig. 3 later).

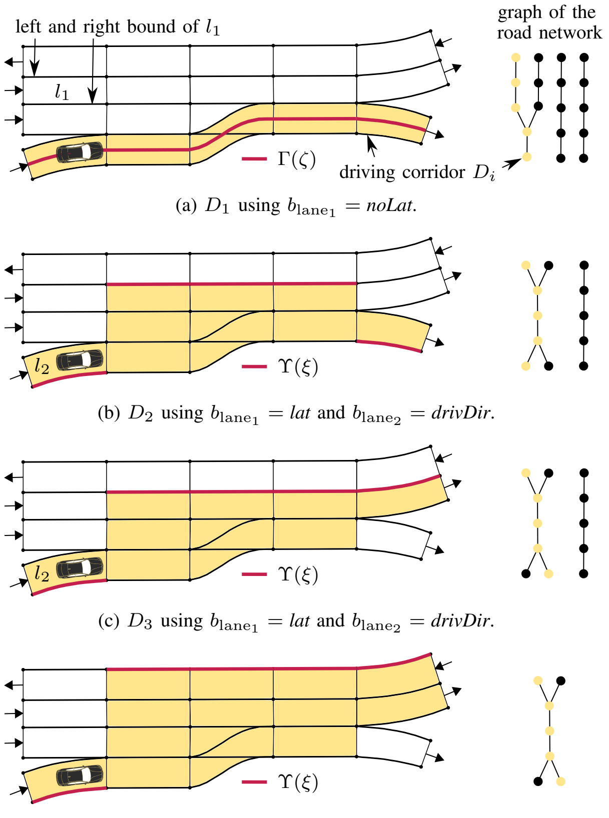
D. Reachable set of traffic participants

Let us define the prerequisites for the reachability analysis based on [67, Sec. IV].

Definition 8 (Model M): A model M is defined as the tuple $M := \langle \boldsymbol{f}_M, \mathcal{S}_M, \mathcal{U}_M \rangle$, where $\boldsymbol{f}_M$ is the right-hand side of the differential equation describing the motion of a traffic participant by

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{f}_M\big(\boldsymbol{s}(t), \boldsymbol{u}(t)\big), \qquad (2)$$

and $\mathcal{S}_M(t) \subseteq \mathbb{R}^n$ and $\mathcal{U}_M(t) \subseteq \mathbb{R}^m$ denote the admissible sets bounding the states $\boldsymbol{s}(t)$ and inputs $\boldsymbol{u}(t)$ of the traffic participant, respectively.



(a) D_1 using $b_{\text{lane}_1} = \mathit{noLat}$.

(b) D_2 using $b_{\text{lane}_1} = \mathit{lat}$ and $b_{\text{lane}_2} = \mathit{drivDir}$.

(c) D_3 using $b_{\text{lane}_1} = \mathit{lat}$ and $b_{\text{lane}_2} = \mathit{drivDir}$.

(d) D_4 using $b_{\text{lane}_1} = \mathit{lat}$ and $b_{\text{lane}_2} = \mathit{anyDir}$.

Fig. 2. The road network $\mathcal{N}$ (here, we only show the layer for vehicles) is modeled by lanelets l (see left part). (a)–(d) Given the values for b_{lane_1} and b_{lane_2}, we construct the graph of the road network (see right part) and show a possible driving corridor D_i (yellow in both left and right part).

When starting at a state $\boldsymbol{s}(t_0) \in \mathcal{S}_0$ and using an input trajectory $\boldsymbol{u}(\cdot)$, a possible solution of (2) at time $t \geq t_0$ is denoted by $\boldsymbol{\chi}\big(t; \boldsymbol{s}(t_0), \boldsymbol{u}(\cdot)\big)$.

Definition 9 (Reachable set $\mathcal{R}$): The reachable set $\mathcal{R}$ of model M is the set of states that are reachable at time $t \geq t_0$ from the initial set $\mathcal{S}_0$ when applying all admissible inputs $\mathcal{U}_M(t)$ while staying within $\mathcal{S}_M(t)$:

$$\mathcal{R}(t; M, t_0) := \Big\{ \boldsymbol{\chi}\big(t, \boldsymbol{s}(t_0), \boldsymbol{u}(\cdot)\big) \,\Big|\, \boldsymbol{s}(t_0) \in \mathcal{S}_0, \forall t^\star \in [t_0, t] : \\ \boldsymbol{\chi}\big(t^\star; \boldsymbol{s}(t_0), \boldsymbol{u}(\cdot)\big) \in \mathcal{S}_M(t^\star), \boldsymbol{u}(t^\star) \in \mathcal{U}_M(t^\star) \Big\}.$$

To over-approximate the reachable set of a model, we introduce abstractions:

Definition 10 (Abstraction): Model M_2 is an abstraction of model M_1, if $\forall t \geq t_0 : \mathcal{R}(t; M_1, t_0) \subseteq \mathcal{R}(t; M_2, t_0)$.

To efficiently minimize the over-approximation caused by an abstraction, we use several abstractions:

Lemma 1 (Combining abstractions): If M_i, $i = 2, \ldots, m$, are abstractions of model M_1, the intersection of their reachable sets remains an over-approximation of the reachable set of the original model M_1:

$$\forall t \geq t_0 : \mathcal{R}(t; M_1, t_0) \subseteq \bigcap_{i=2}^{m} \mathcal{R}(t; M_i, t_0). \qquad \square$$

Proof: The over-approximation directly follows from [3, Prop. V.1]. ■

If considering the reachable set only at distinct points in time, we cannot provide any safety guarantees for the ego vehicle between these points in time. Thus, we need to compute the reachable set for a time interval $[t] := [\underline{t}, \overline{t}] \geq t_0 : \mathcal{R}([t]; M, t_0) := \bigcup_{t \in [t]} \mathcal{R}(t; M, t_0)$.

E. Problem statement

Let M_{real} be the model that exactly describes the motions of a traffic participant that can be performed in the real world and comply with all applicable traffic rules. Our goal is to predict the future reachable set of a model M_{pred} that is an abstraction of M_{real}, i. e., $\mathcal{R}(t; M_{\text{real}}, t_0) \subseteq \mathcal{R}(t; M_{\text{pred}}, t_0)$ for any $t \in [t]$, with as little over-approximation as possible.

III. Specification and Overall Algorithm

Instead of trying to explicitly describe all acceptable behaviors in abstraction M_{pred}, we define constraints in our specification that lead to an over-approximation of acceptable behaviors. Our specification is chosen such that the prediction conforms to legal safety based on traffic rules. Thus, it is in line with RSS [7] and rulebooks [95], which both specify acceptable behaviors for the ego vehicle, while we, from the prediction perspective, focus on the acceptable behaviors of other traffic participants. Note that our approach has the benefit that even if we do not model all traffic rules, our prediction remains over-approximative.

Our parameterizable specification consists of independent constraints C that are listed in Tab. I. Each constraint is defined by its parameters, textual description, formalization, and source. The Boolean parameters b allow us to enable or disable constraints individually, and the parameters Δ allow us to tune our reaction to violations of constraints (see Sec. VI later). The longitudinal direction is described with respect to the driving direction. In summary, our specification either constrains the dynamics of other traffic participants (see upper part of Tab. I) or constrains the allowed regions in the environment (see lower part of Tab. I).

Alg. 1 provides an overview of our prediction running in every planning cycle. At the current initial time t_0, we receive as input an updated environment model $\Omega_0 = \langle \mathcal{P}, \mathcal{N}, \mathcal{D}^{\mathcal{P}}, \mathcal{F} \rangle$ of the ego vehicle. If available, the environment model from the previous planning cycle can also be provided (cf. optional input of Alg. 1). The parameters $\mathcal{Q}$ (cf. Tab. I) are initialized as desired by the user (cf. Tab. IV later).

First, we create phantom traffic participants that capture the risks from potentially undetected traffic participants (line 1 of Alg. 1; cf. Sec. IV). For each traffic participant (except the ego vehicle), we validate its constraint parameters $\mathcal{Q}^p$ (line 3; cf. Sec. VI) and choose all valid abstractions $M^p_\diamond$ (line 4; cf. Tab. II). Next, for each given time interval $[t]$, we compute the reachable set of each valid abstraction (line 8; cf. Sec. V) and intersect them to obtain a tight over-approximative reachable set (line 9; cf. Sec. V-F).

Algorithm 1 Set-basedPrediction

Input: environment model $\Omega_0 = \langle \mathcal{P}, \mathcal{N}, \mathcal{D}^{\mathcal{P}}, \mathcal{F} \rangle$ at t_0 (containing $p = \langle \boldsymbol{c}^p, \mathcal{S}^p_0, \mathcal{A}^p, \mathcal{Q}^p \rangle$ for each $p \in \mathcal{P}$), default parameters $\mathcal{Q}$, and set τ of arbitrary time intervals $[t] \geq t_0$
Optional input: environment model Ω_{c-1} from previous cycle
Output: over-approximative reachable set $\mathcal{R}^p$ for each $p \in \mathcal{P}$

```
 1: $\mathcal{P}$.addPhantoms($\mathcal{N}$, $\mathcal{F}$, $\mathcal{Q}$)          ▷ consider occlusions
 2: for all $p \in \mathcal{P}$ do
 3:   $\mathcal{Q}^p \leftarrow$ validateConstraints($\Omega_0$, $\Omega_{c-1}$)
 4:   $\mathcal{Q}^p \leftarrow$ selectValidAbstractions($\boldsymbol{c}^p$, $\mathcal{Q}^p$)
 5:   $\mathcal{R}^p(\cdot; M^p_{\text{pred}}, t_0) \leftarrow \mathbb{R}^4$          ▷ initialize
 6:   for all $M^p_\diamond \in \mathcal{Q}^p$ do
 7:     for all $[t] \in \tau$ do
 8:       $\mathcal{R}^p([t]; M^p_\diamond, t_0) \leftarrow$ reach($[t]$, $M^p_\diamond$, $p$, $\mathcal{N}$, $\mathcal{D}^p$, $\mathcal{Q}^p$)
 9:       $\mathcal{R}^p([t]; M^p_{\text{pred}}, t_0) \leftarrow \mathcal{R}^p([t]; M^p_{\text{pred}}, t_0) \cap \mathcal{R}^p([t]; M^p_\diamond, t_0)$
10:     end for
11:   end for
12: end for
13: interaction($\mathcal{R}^p(\cdot)$ for all $p \in \mathcal{P}$, $\mathcal{N}$)          ▷ optional
14: return $\mathcal{R}^p([t]; M^p_{\text{pred}}, t_0)$ for all $p \in \mathcal{P}$ and $[t] \in \tau$
```

The time complexity of our algorithm is linear in the number of traffic participants and the number of time intervals. Our algorithm can be parallelized for each traffic participant and each abstraction. Line 13 of Alg. 1 optionally considers the interaction between vehicles as described in [69], e. g., that a vehicle cannot tunnel through a stationary vehicle.

IV. Occlusion

To consider traffic participants that are hidden due to occlusions and therefore cannot be predicted directly, we create all phantom traffic participants $p = \langle \boldsymbol{c}, \mathcal{S}_0, \mathcal{A}, \mathcal{Q} \rangle$ that could be relevant for the motion of the ego vehicle, as summarized in Alg. 2, visualized in Fig. 3, and described subsequently.

Def. 1 implies that no traffic participant can suddenly appear within the field of view, but may enter the field of view at any time $t > t_0$. Thus, we intersect the boundary of the field of view with all driving corridors $\mathbb{D}(\mathit{lat}, \mathit{drivDir})$ of each layer and split the boundary at each intersection point into border segments (or edges) e (lines 1–3 of Alg. 2; cf. Fig. 3). The resulting set $\mathcal{E} := \{e_1, \dots, e_i\}$ contains all border segments e of the field of view through which phantom traffic participants can emerge. To consider additional sources of traffic participants, e. g., doors where pedestrians can appear, each source can be modeled as an additional driving corridor.

Border segment e is relevant for the motion of the ego vehicle, if the ego vehicle can be influenced by a phantom traffic participant that is positioned at e and performs any acceptable behavior in accordance with our legal specification. Therefore, we require all forward driving corridors of the ego vehicle $\vec{\mathcal{D}}^{\text{ego}}_{\text{reach}} := \vec{\mathcal{D}}^{\text{ego}}(b^{\text{ego}}_{\text{lane}_1}, b^{\text{ego}}_{\text{lane}_2})$, as shown in Fig. 3. When using $b^{\text{ego}}_{\text{lane}_1} = \mathit{lat}$ and $b^{\text{ego}}_{\text{lane}_2} = \mathit{anyDir}$, we will create phantom traffic participants considering all possible behaviors of the ego vehicle. In case we know that the ego vehicle will

TABLE I
LEGAL SPECIFICATION CONSTRAINING THE ACCEPTABLE BEHAVIORS OF OTHER TRAFFIC PARTICIPANTS.

Constraint	Parameters	Description and formalization (based on state variables $\forall t \geq t_0$)	Source
$C_{a_{\max}}$	$a_{\max} > 0$, $\Delta_{a_{\max}} \geq 0$	Absolute acceleration, i. e., accelerating and braking, does not exceed $a_{\max}$: $\lvert\dot{v}(t)\rvert \leq a_{\max}$.	physical law (friction circle)
$C_{v_{\max}}$	$v_{\max} > 0$, $\Delta_{v_{\max}} \geq 0$	Absolute velocity does not exceed $v_{\max}$: $\lvert v(t)\rvert \leq v_{\max}$.	physical law and [94]
C_{speedLim}	$f_{\mathrm{speed}} \geq 1$, $\Delta_{f_{\mathrm{speed}}} \geq 0$	For vehicles, longitudinal velocity does not exceed the official speed limit v_{speedLim} multiplied by a speeding factor f_{speed}:$\lvert v_\xi(t)\rvert \leq v_{\mathrm{speedLim}} \cdot f_{\mathrm{speed}}$.	[89, 13§1–2]
C_{engine}	$v_S > 0$	For vehicles, above the switching velocity v_S, longitudinal acceleration is decreasing inversely proportional to longitudinal velocity due to limited engine power: $\lvert v_\xi(t)\rvert < v_S \vee \lvert\dot{v}_\xi(t)\rvert \leq a_{\max} \cdot \frac{v_S}{\lvert v_\xi(t)\rvert}$.	physical law
C_{reverse}	$b_{\mathrm{reverse}} \in \mathcal{B}$, $\Delta_{v_{\mathrm{reverse}}} \leq 0$	For vehicles, it is forbidden to reverse, i. e., to drive backwards in longitudinal direction: $v_\xi(t) \geq 0$.	[89, 14§2]
$C_{v_{\min}}$	$v_{\min} \in \mathbb{R}$, $\Delta_{v_{\min}} \geq 0$	For vehicles, longitudinal velocity does not fall below $v_{\min}$: $v_\xi(t) \geq v_{\min}$.	[89, 13§4, 23§1]
C_{turn}	$0 \leq \delta_{\max} \leq \pi/2$, $\underline{\ell}_{\mathrm{wb}} > 0$, $\underline{\ell}_{\mathrm{ovr}} \geq 0$, $\overline{\ell}_{\mathrm{ovr}} \geq 0$	For vehicles, the steering angle does not exceed $\delta_{\max}$, and turning within lanes is forbidden: $\mathtt{occ}\big(\boldsymbol{s}(t), \mathcal{A}\big) \cap \mathcal{O}_{\mathrm{turn}}(t_{c-1}) = \emptyset$.	physical law and [89, 14§2]
C_{road}	$b_{\mathrm{road}} \in \mathcal{B}$	It is forbidden to leave $\mathcal{W}_{\mathrm{road}}$, which are the allowed positions for this type of traffic participant (cf. Def. 2): $\mathtt{occ}\big(\boldsymbol{s}(t), \mathcal{A}\big) \subseteq \mathcal{W}_{\mathrm{road}}$.	[89, 1§(d)–(j)]
C_{prio}	$b_{\mathrm{prio}} \in \mathcal{B}$	It is forbidden to occupy parts of the road network that intersect with other lanes (including forks and merging lanes) for which other traffic participants currently have priority: $\mathtt{occ}\big(\boldsymbol{s}(t), \mathcal{A}\big) \subseteq \mathcal{O}_{\mathrm{prio}}(t; t_0)$.	[89, 18§1–7, 20§6(b), 21§2]
C_{lane}	$b_{\mathrm{lane}_1} \in \{\mathit{noLat}, \mathit{lat}\}$, $b_{\mathrm{lane}_2} \in \{\mathit{drivDir}, \mathit{anyDir}\}$	For vehicles, changing lanes is restricted: $\vec{\mathcal{D}}(t) \subseteq \vec{\mathcal{D}}(t_{c-1})$ using the same $\mathcal{N}$; if $b_{\mathrm{lane}_1} = \mathit{noLat}$: It is forbidden to change to any other lane. if $b_{\mathrm{lane}_2} = \mathit{drivDir}$: It is forbidden to change to a lane that is not appropriate with respect to the direction of traffic.	[89, 10§4–5, 11§1–11]
C_{safe}	$T^{\mathrm{ego}} \geq 0$, $a^{\mathrm{ego}}_{\mathrm{comfort}} \geq 0$	For vehicles, a safe distance (measured along the centerline of the lanes) to the ego vehicle must be kept when driving behind the ego vehicle or merging in front of it.	[89, 13§5, 11§2(d)]

not overtake in a lane not appropriate to the direction of traffic, we can use $b^{\mathrm{ego}}_{\mathrm{lane}_2} = \mathit{drivDir}$, and if we know that the ego vehicle will not change to any laterally adjacent lane, we can use $b^{\mathrm{ego}}_{\mathrm{lane}_1} = \mathit{noLat}$; this minimizes the set $\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}}$ to reduce computation costs.

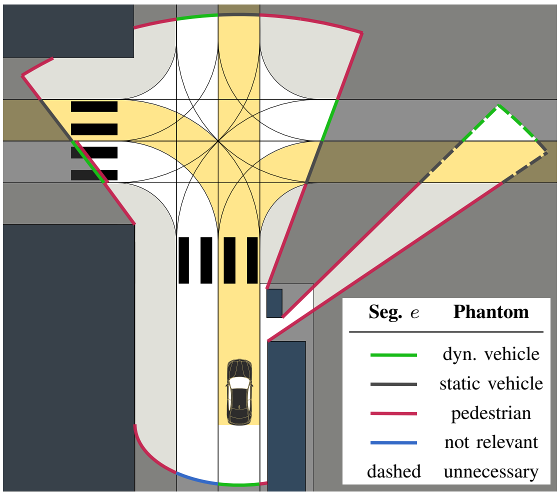


Fig. 3. When intersecting the field of view (bright area) of the ego vehicle with all driving corridors of each layer of the road network, we obtain border segments e. At each e, we introduce phantom traffic participants (see legend) if they could be relevant for the motion of the ego vehicle, which is determined using the forward driving corridors of the ego vehicle $\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}} = \vec{\mathcal{D}}^{\mathrm{ego}}(\mathit{lat}, \mathit{drivDir})$ (yellow area) and extends [16, Fig. 2].

Let us denote the forward driving corridors when starting at e by $\vec{\mathcal{D}}(e) \subset \mathbb{D}(\mathit{lat}, \mathit{drivDir})$. By comparing $\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}}$ with $\vec{\mathcal{D}}(e)$ as described in lines 5–17 of Alg. 2, we determine whether e is relevant and what classification $\boldsymbol{c}$ for a phantom traffic participant at e is required (cf. Fig. 3). An example for a border segment that is not relevant for the motion of the ego vehicle is the blue segment in Fig. 3.

Next, in lines 18–28 of Alg. 2, we set the initial positions as the border segment e (which spans across all laterally adjacent lanelets with the same driving direction), the initial velocities as all admissible velocities in the driving corridor of e, the initial heading aligned with the driving direction, and the size to the values given in Tab. IV so that $e \subset \mathtt{occ}(\mathcal{S}_0, \mathcal{A})$. As a result, the phantom traffic participant is modeled as an abstraction of any possibly appearing traffic participant. Finally, we add the phantom traffic participant to $\mathcal{P}$ (line 29 of Alg. 2); thus, it will be predicted analogously to the detected traffic participants (cf. Alg. 1).

We might have added multiple phantom vehicles in the same driving corridor, as shown in the right part of Fig. 3 (dashed segments). If the forward driving corridor of a dynamic phantom vehicle is completely enclosed by the forward driving corridor of another dynamic phantom vehicle, we can remove the latter phantom vehicle, since it is further away from the ego vehicle and its threat is already considered by the other, former phantom vehicle (line 31 and 40 of Alg. 2).

Algorithm 2 addPhantoms()

```
Input: road network $\mathcal{N}$, field of view $\mathcal{F}$, default parameters $\mathcal{Q}$
Output: set of phantom traffic participants $\mathcal{P}$

$\mathcal{E}^{\mathrm{veh}} \leftarrow \mathcal{F} \cap \mathbb{D}^{\mathrm{veh}}(lat, drivDir)$
$\mathcal{E}^{\mathrm{cyc}} \leftarrow \mathcal{F} \cap \mathbb{D}^{\mathrm{cyc}}(lat, drivDir)$
$\mathcal{E}^{\mathrm{ped}} \leftarrow \mathcal{F} \cap \mathbb{D}^{\mathrm{ped}}(lat, drivDir)$
for all $e \in \{\mathcal{E}^{\mathrm{veh}} \cup \mathcal{E}^{\mathrm{cyc}} \cup \mathcal{E}^{\mathrm{ped}}\}$ do
  if $e \in \mathcal{E}^{\mathrm{veh}}$ and $\vec{\mathcal{D}}(e) \subseteq \vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}}$ then
    $\boldsymbol{c} \leftarrow \{\mathrm{veh}, \mathrm{phantom}, \mathrm{static}\}$          ▷ vehicle ahead
  else if $e \in \mathcal{E}^{\mathrm{veh}}$ and $\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}} \subseteq \vec{\mathcal{D}}(e)$ then
    $\boldsymbol{c} \leftarrow \{\mathrm{veh}, \mathrm{phantom}, \mathrm{dyn}\}$          ▷ vehicle behind
  else if $e \in \mathcal{E}^{\mathrm{veh}}$ and $\mathsf{occ}(\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}}) \cap \mathsf{occ}\big(\vec{\mathcal{D}}(e)\big) \neq \emptyset$ then
    $\boldsymbol{c} \leftarrow \{\mathrm{veh}, \mathrm{phantom}, \mathrm{dyn}\}$          ▷ crossing vehicle
  else if $e \in \mathcal{E}^{\mathrm{cyc}}$ and $\mathsf{occ}(\vec{\mathcal{D}}^{\mathrm{ego}}_{\mathrm{reach}}) \cap \mathsf{occ}\big(\vec{\mathcal{D}}(e)\big) \neq \emptyset$ then
    $\boldsymbol{c} \leftarrow \{\mathrm{cyc}, \mathrm{phantom}, \mathrm{dyn}\}$          ▷ crossing cyclist
  else if $e \in \mathcal{E}^{\mathrm{ped}}$ then
    $\boldsymbol{c} \leftarrow \{\mathrm{ped}, \mathrm{phantom}, \mathrm{dyn}\}$          ▷ pedestrian
  else
    continue  ▷ not relevant, as no interaction with ego vehicle
  end if
  if ped $\in \boldsymbol{c}$ then          ▷ initial state for pedestrian
    $[v_0] \leftarrow [0, v^{\mathrm{ped}}_{\max}]$          ▷ from $\mathcal{Q}$
    $[\psi_0] \leftarrow [-\pi, \pi]$
    $\mathcal{A} \leftarrow$ createCircle($\mathcal{Q}$)
  else          ▷ initial state for vehicle (incl. cyclist)
    $[v_0] \leftarrow [0, v_{\max,\xi}]$          ▷ from $\mathcal{Q}$ and (5)
    $\psi_0 \leftarrow$ getDrivingDirection($\vec{\mathcal{D}}(e)$)
    $\mathcal{A} \leftarrow$ createRectangle($\mathcal{Q}$)
  end if
  $[[x_0], [y_0]]^T \leftarrow$ createBoundingBox($e$)
  $\mathcal{D} \leftarrow \mathcal{D}(e)$
  $\mathcal{P}$.add($\langle \boldsymbol{c}, \big[[x_0], [y_0], [v_0], [\psi_0]\big]^T, \mathcal{A}, \mathcal{Q}\rangle$)
end for
$\mathcal{P} \leftarrow$ removeUnnecessaryPhantoms($\mathcal{P}$, $\mathcal{N}$)          ▷ optional
return $\mathcal{P}$

function removeUnnecessaryPhantoms($\mathcal{P}$, $\mathcal{N}$)
  for all $i, j \in \mathcal{P}$ do
    if $i = j$ or veh $\notin \boldsymbol{c}^i$ or phantom $\notin \boldsymbol{c}^i$ or $\boldsymbol{c}^i \neq \boldsymbol{c}^j$ then
      break
    else if static $\in \boldsymbol{c}^i$ and $\vec{\mathcal{D}}^i \subseteq \vec{\mathcal{D}}^j$ then
      $\mathcal{P}$.remove($i$)          ▷ $j$ is behind $i$ and the ego vehicle is behind both
    else if dyn $\in \boldsymbol{c}^i$ and $\vec{\mathcal{D}}^i \subseteq \vec{\mathcal{D}}^j$ then
      $\mathcal{P}$.remove($j$)          ▷ $i$ is in front of $j$ and either both are behind the ego vehicle or both are approaching the ego vehicle
    end if
  end for
  return $\mathcal{P}$
end function
```

V. Abstractions

We minimize the over-approximation of our prediction by using several abstractions (cf. Lemma 1). Tab. II provides an overview of the proposed abstractions and their covered constraints so that all constraints of Tab. I are considered. Some abstractions require that other constraints have not been violated, i. e., the Boolean parameters given in Tab. II must be *true*; otherwise, this abstraction cannot be computed and gets disabled, e. g., M_{long} is omitted if $b_{\mathrm{road}} = \mathit{false}$. In the following subsections, we define these abstractions and present how to compute their reachable set and occupancy.

TABLE II
Overview of the abstractions.

Abstraction	Covers constraints	Requires	See
M_{acc}	$C_{a_{\max}}$	n/a	Sec. V-A
M_{vel}	$C_{v_{\max}}$	n/a	Sec. V-A
M_{long}	C_{speedLim}, C_{engine}, C_{reverse}, $C_{v_{\min}}$, C_{road}, C_{lane} (and both $C_{a_{\max}}$ and $C_{v_{\max}}$ only in longitudinal direction)	b_{road}	Sec. V-B
M_{turn}	C_{turn}	$b_{\mathrm{road}} \wedge b_{\mathrm{reverse}}$	Sec. V-D
M_{prio}	C_{prio}	$b_{\mathrm{road}} \wedge b_{\mathrm{prio}}$	Sec. V-E
M_{safe}	C_{safe}	$b_{\mathrm{road}} \wedge b_{\mathrm{reverse}}$	Sec. V-C

A. Abstractions based on point-mass model (M_{acc} and M_{vel})

To describe a point-mass model, let us rewrite the state vector as $\boldsymbol{s}(t) = [x(t), y(t), v_x(t), v_y(t)]^T \in \mathbb{R}^4$ with $v_x(t) = v(t) \cdot \cos(\psi(t))$ and $v_y(t) = v(t) \cdot \sin(\psi(t))$. Analogously, the set of initial states is $\mathcal{S}_0 = \big[[x_0], [y_0], [v_{x_0}], [v_{y_0}]\big]^T \subset \mathbb{R}^4$. The input for the abstractions based on a point-mass model consists of the acceleration in x-direction and y-direction, i. e., $\boldsymbol{u}(t) = [u_x(t), u_y(t)]^T \in \mathbb{R}^2$.

Definition 11 (Acceleration-bounded abstraction M_{acc}): Abstraction $M_{\mathrm{acc}} := \langle \boldsymbol{f}_{M_{\mathrm{acc}}}, \mathcal{S}_{M_{\mathrm{acc}}}, \mathcal{U}_{M_{\mathrm{acc}}} \rangle$ is an acceleration-bounded point-mass model ($C_{a_{\max}}$), where

$$\dot{x}(t) = v_x(t), \dot{y}(t) = v_y(t), \dot{v}_x(t) = u_x(t), \dot{v}_y(t) = u_y(t),$$
$$\mathcal{S}_{M_{\mathrm{acc}}} := \mathbb{R}^4,$$
$$\mathcal{U}_{M_{\mathrm{acc}}} := \big\{[u_x(t), u_y(t)]^T \mid \sqrt{u_x(t)^2 + u_y(t)^2} \leq a_{\max}\big\}.$$

Proposition 1 (Reachable set of M_{acc}): The reachable set of M_{acc} for a time interval $[t] \geq t_0$ is

$$\mathcal{R}([t]; M_{\mathrm{acc}}, t_0) = \mathtt{conv}\big(\boldsymbol{T}_{\mathrm{hom}}(\underline{t}) \cdot \mathcal{S}_0, \boldsymbol{T}_{\mathrm{hom}}(\overline{t}) \cdot \mathcal{S}_0\big) \oplus \boldsymbol{T}_{\mathrm{inp}}(\overline{t}) \cdot \mathcal{U}_{M_{\mathrm{acc}}},$$

as shown in the blue part of Fig. 4 and where

$$\boldsymbol{T}_{\mathrm{hom}}(t) = \begin{bmatrix} 1 & 0 & t - t_0 & 0 \\ 0 & 1 & 0 & t - t_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$\boldsymbol{T}_{\mathrm{inp}}(t) = \begin{bmatrix} 1/2 \cdot (t - t_0)^2 & 0 \\ 0 & 1/2 \cdot (t - t_0)^2 \\ t - t_0 & 0 \\ 0 & t - t_0 \end{bmatrix}. \qquad \square$$

Proof: The reachable set directly follows from [70, Prop. 2].■

To compute the occupancy of $\mathcal{R}([t]; M_{\mathrm{acc}}, t_0)$ for vehicles, we require the heading. However, due to the state representation of the point-mass model, the reachable set does not contain a bound for the heading. In our previous work [67], we have assumed that the heading is constant over the prediction horizon. In this work, we do not make this assumption. We can bound the heading until the earliest point in time $t_{v=0}$ at

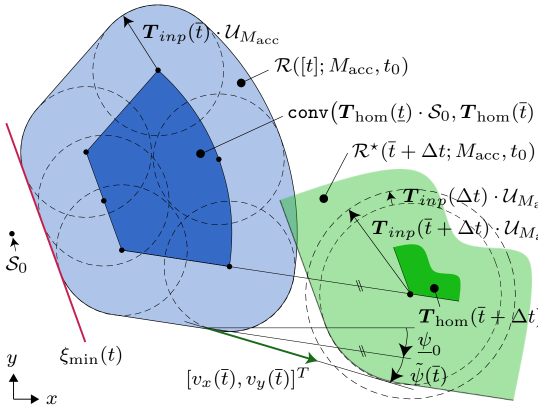

$\boldsymbol{T}_{inp}(\bar{t})\cdot\mathcal{U}_{M_{\text{acc}}}$
$\mathcal{R}([t];M_{\text{acc}},t_0)$
$\texttt{conv}\big(\boldsymbol{T}_{\text{hom}}(\underline{t})\cdot\mathcal{S}_0,\boldsymbol{T}_{\text{hom}}(\bar{t})\cdot\mathcal{S}_0\big)$
$\mathcal{R}^\star(\bar{t}+\Delta t;M_{\text{acc}},t_0)$
$\boldsymbol{T}_{inp}(\Delta t)\cdot\mathcal{U}_{M_{\text{acc}}}$
$\boldsymbol{T}_{inp}(\bar{t}+\Delta t)\cdot\mathcal{U}_{M_{\text{acc}}}$
$\mathcal{S}_0$
$\boldsymbol{T}_{\text{hom}}(\bar{t}+\Delta t)\cdot\mathcal{S}_0$
$\underline{\psi}_0$
y
$\xi_{\min}(t)$
$[v_x(\bar{t}),v_y(\bar{t})]^T$
$\tilde{\psi}(\bar{t})$
x

Fig. 4. Blue part (upper left): The reachable set $\mathcal{R}([t];M_{\text{acc}},t_0)$ is bounded by the Minkowski addition of the homogeneous solution $\boldsymbol{T}_{\text{hom}}([t])\cdot\mathcal{S}_0$ with the input solution $\boldsymbol{T}_{\text{inp}}(\bar{t})\cdot\mathcal{U}_{M_{\text{acc}}}$ (cf. Prop. 1 and [70, Fig. 3]). Green part (lower right): A bound on the heading $\psi([t])$ for M_{acc} is obtained from the velocity vector $[v_x(\bar{t}),v_y(\bar{t})]^T$ that has to point within $\mathcal{R}^\star(\bar{t}+\Delta t;M_{\text{acc}},t_0)$ (cf. Lemma 2). Red line (left): To prevent reversing, we restrict the minimum positions in M_{long} to $\xi_{\min}(t)$ based on $\mathcal{R}(t_{v=0};M_{\text{acc}},t_0)$ (cf. (8)). Note that all sets are projected onto the position domain.

which the vehicle can come to a standstill when fully braking:

$$t_{v=0}:=\begin{cases}\frac{\underline{v}_0}{a_{\max}}+t_0 & \text{if } \underline{v}_0\geq 0\\ -\infty & \text{otherwise.}\end{cases}\tag{3}$$

Lemma 2 (Bounds for ψ of M_{acc}): Due to the limited acceleration in M_{acc}, the heading of a vehicle for $[t]\geq t_0$ is

$$\psi([t])\in\begin{cases}\left[\underline{\psi}_0-\tilde{\psi}(\bar{t}),\overline{\psi}_0+\tilde{\psi}(\bar{t})\right] & \text{if } \bar{t}<t_{v=0}\\ \mathbb{R} & \text{otherwise,}\end{cases}$$

with $\tilde{\psi}(\bar{t}):=\sin^{-1}\Big(\frac{a_{\max}}{\underline{v}_0}\cdot(\bar{t}-t_0)\Big)$. □

Proof: Let $\Delta t>0$, $\bar{t}\geq t_0$, and $\bar{t}+\Delta t<t_{v=0}$. A velocity vector $[v_x(\bar{t}),v_y(\bar{t})]^T$ at $\bar{t}$ in $\mathcal{R}(\bar{t};M_{\text{acc}},t_0)$ has to point to a position in $\mathcal{R}^\star(\bar{t}+\Delta t;M_{\text{acc}},t_0):=\boldsymbol{T}_{\text{hom}}(\bar{t}+\Delta t)\cdot\mathcal{S}_0\oplus\big(\boldsymbol{T}_{\text{inp}}(\bar{t}+\Delta t)\cdot\mathcal{U}_{M_{\text{acc}}}-\boldsymbol{T}_{\text{inp}}(\Delta t)\cdot\mathcal{U}_{M_{\text{acc}}}\big)$, since we can accelerate by $\mathcal{U}_{M_{\text{acc}}}$ during Δt but must satisfy Prop. 1 at $\bar{t}+\Delta t$ (see Fig. 4). The maximum angle of this velocity vector can be described by the tangent against $\texttt{proj}_{x,y}\big(\mathcal{R}(\bar{t};M_{\text{acc}},t_0)\big)$ and $\texttt{proj}_{x,y}\big(\mathcal{R}^\star(\bar{t}+\Delta t;M_{\text{acc}},t_0)\big)$. The angle of a tangent on two circles is the inverse of the sine function of the difference of their radii divided by the distance of their center points [96]; for our case (see Fig. 4), $\tilde{\psi}(\bar{t})=\sin^{-1}\left(\frac{\texttt{proj}_{x,y}\left((\boldsymbol{T}_{\text{inp}}(\bar{t}+\Delta t)-\boldsymbol{T}_{\text{inp}}(\Delta t))\cdot\mathcal{U}_{M_{\text{acc}}}-\boldsymbol{T}_{\text{inp}}(\bar{t})\cdot\mathcal{U}_{M_{\text{acc}}}\right)}{\left\|\texttt{proj}_{x,y}\left(\boldsymbol{T}_{\text{hom}}(\bar{t}+\Delta t)\cdot\mathcal{S}_0-\boldsymbol{T}_{\text{hom}}(\bar{t})\cdot\mathcal{S}_0\right)\right\|_2}\right)$. Using $t^\star:=\bar{t}-t_0$ and a $v_0\in[v_0]$, this evaluates to $\tilde{\psi}(\bar{t})=\sin^{-1}\left(\frac{1/2\cdot((t^\star+\Delta t)^2-\Delta t^2)\cdot a_{\max}-1/2\cdot t^\star\cdot a_{\max}}{v_0\cdot(t^\star+\Delta t)-v_0\cdot t^\star}\right)$. After simplifying the term and by selecting the $v_0\in[v_0]$ that maximizes $\tilde{\psi}(\bar{t})$, we obtain $\tilde{\psi}(\bar{t})=\sin^{-1}\Big(\frac{a_{\max}}{\underline{v}_0}\cdot(\bar{t}-t_0)\Big)$. Since the inverse of the sine function is monotonic, $\forall t\in[\underline{t},\bar{t}]:\tilde{\psi}(t)\leq\tilde{\psi}(\bar{t})$. Finally, we add the initial heading $[\psi_0]$ and obtain the bound on $\psi([t])$. ■

Definition 12 (Velocity-bounded abstraction M_{vel}): Abstraction $M_{\text{vel}}:=\langle\boldsymbol{f}_{M_{\text{acc}}},\mathcal{S}_{M_{\text{vel}}},\mathcal{U}_{M_{\text{vel}}}\rangle$ is a velocity-bounded point-mass model ($C_{v_{\max}}$), where

$$\mathcal{S}_{M_{\text{vel}}}:=\Big\{[x(t),y(t),v_x(t),v_y(t)]^T\ \Big|\ \sqrt{v_x(t)^2+v_y(t)^2}\leq v_{\max}\Big\},$$
$$\mathcal{U}_{M_{\text{vel}}}:=\mathbb{R}^2.$$

When using M_{acc} and M_{vel} at the same time, the constraint on acceleration is more restrictive than the constraint on velocity until the earliest point in time $t_{v_{\max}}$ at which $v_{\max}$ or $-v_{\max}$ can be reached:

$$t_{v_{\max}}=\frac{v_{\max}-\max\left(|\underline{v}_0|,|\overline{v}_0|\right)}{a_{\max}}+t_0.\tag{4}$$

Thus, if $t_{v_{\max}}>t_0$, we can reduce the over-approximation in the reachable set of M_{vel} by initializing it at $t_{v_{\max}}$ with the result of M_{acc} (instead of at t_0 with $\mathcal{S}_0$):

Proposition 2 (Reachable set of M_{vel}): The reachable set of M_{vel} for $[t]>t_{v_{\max}}$ is

$$\mathcal{R}([t];M_{\text{vel}},t_{v_{\max}})=\bigg\{[x,y,v,\psi]^T\ \bigg|\ [x,y]^T\in \texttt{proj}_{x,y}\big(\mathcal{R}(t_{v_{\max}};M_{\text{acc}},t_0)\big)\oplus\mathcal{C}\big([0,0]^T,v_{\max}\cdot(\bar{t}-t_{v_{\max}})\big), v\in[-v_{\max},v_{\max}],\psi\in\mathbb{R}\bigg\}.$$ □

Proof: The reachable set directly follows from [70, (9)]. ■

B. Abstraction in longitudinal direction (M_{long})

So far, we have covered constraints on absolute acceleration and absolute velocity. With abstraction M_{long}, we restrict the motion of vehicles in longitudinal direction and to the road. According to C_{road} and C_{lane}, the admissible positions on the road are obtained from the driving corridors of vehicle p as $\texttt{occ}(\mathcal{D}^p)$.

For each driving corridor $D\in\mathcal{D}^p$, we define a curvilinear coordinate frame along a reference path $\Upsilon(\xi):\mathbb{R}\to\mathbb{R}^2$, where the path variable ξ represents the arc length. Since we want to over-approximate the behavior of vehicles when accelerating in driving direction, we require that $\Upsilon(\xi)$ is the shortest possible path through the driving corridor. This shortest path is obtained by following the inner bound of the driving corridor (i. e., the bound in the inside of the curve), while jumping at inflection points instantaneously to the new inner bound, as described in [67, Def. 8] and illustrated in Fig. 2b–2d.

To describe motions along $\Upsilon(\xi)$, we rewrite $\boldsymbol{s}(t)=[\xi(t),v_\xi(t)]^T\in\mathbb{R}^2$ and $\mathcal{S}_0=\big[[\xi_0],[v_{\xi_0}]\big]^T\subset\mathbb{R}^2$ by using $v_\xi(t)=v(t)$, i. e., we over-approximate the longitudinal velocity by the absolute velocity. The maximum longitudinal velocity is determined by the more restrictive constraint of C_{speedLim} and $C_{v_{\max}}$ (cf. Tab. I) as

$$v_{\max,\xi}:=\min(v_{\text{speedLim}}\cdot f_{\text{speed}},v_{\max}),\tag{5}$$

and the minimum longitudinal velocity is determined by the more restrictive constraint of C_{reverse}, $C_{v_{\min}}$, and $C_{v_{\max}}$ as

$$v_{\min,\xi} := \begin{cases} \max(v_{\min}, 0) & \text{if } b_{\text{reverse}} = \textit{true} \\ \max(v_{\min}, -v_{\max,\xi}) & \text{otherwise.} \end{cases} \tag{6}$$

In combination with $C_{a_{\max}}$ (only in longitudinal direction) and C_{engine}, we describe the maximum longitudinal acceleration $a_{\max,\xi}$ (i. e., the limit on increasing the signed velocity) and the minimum longitudinal acceleration $a_{\min,\xi}$ (i. e., the limit on decreasing the signed velocity) as

$$a_{\max,\xi}\big(v_\xi(t)\big) := \tag{7a}$$
$$\begin{cases} 0 & \text{if } v_\xi(t) \geq v_{\max,\xi} \\ a_{\max} \cdot \frac{v_S}{|v_\xi(t)|} & \text{if } v_S \leq v_\xi(t) < v_{\max,\xi} \\ a_{\max} & \text{if } 0 \leq v_\xi(t) < \min(v_S, v_{\max,\xi}) \\ \infty & \text{if } v_\xi(t) < 0, \end{cases}$$

$$a_{\min,\xi}\big(v_\xi(t)\big) := \tag{7b}$$
$$\begin{cases} -\infty & \text{if } v_\xi(t) > 0 \\ -a_{\max} & \text{if } 0 \geq v_\xi(t) > \max(-v_S, v_{\min,\xi}) \\ -a_{\max} \cdot \frac{v_S}{|v_\xi(t)|} & \text{if } -v_S \geq v_\xi(t) > v_{\min,\xi} \\ 0 & \text{if } v_\xi(t) \leq \min(0, v_{\min,\xi}), \end{cases}$$

which extends [67, $a_{\text{c2,long}}$] by considering reversing. Note that in (7a) and (7b), the braking acceleration (i. e., decreasing the absolute velocity) is set to infinity, since braking behaviors cannot be over-approximated using the shortest path. However, braking behaviors are already considered by M_{acc}. Since M_{acc} does not consider $v_{\min,\xi}$, we restrict the minimum reachable position (see red line in Fig. 4) to

$$\xi_{\min}(t) := \tag{8}$$
$$\begin{cases} \texttt{proj}_{\underline{x}}\big(\mathcal{R}(t_{v=0}; M_{\text{acc}}, t_0)\big) & \text{if } v_{\min,\xi} \geq 0 \wedge t \geq t_{v=0} \geq t_0 \\ -\infty & \text{otherwise,} \end{cases}$$

when assuming without loss of generality that the mean heading is aligned with the x-axis and by transforming $\xi_{\min}(t)$ to $\Upsilon(\xi)$. Using the above definitions, we define our abstraction:

Definition 13 (Abstraction M_{long} for driving corridors): Abstraction $M_{\text{long}} := \langle \boldsymbol{f}_{M_{\text{long}}}, \mathcal{S}_{M_{\text{long}}}, \mathcal{U}_{M_{\text{long}}} \rangle$ is defined along the shortest path $\Upsilon(\xi)$ of each driving corridor D:

$$\begin{aligned} &\dot\xi(t) = v_\xi(t), \dot v_\xi(t) = u_\xi(t), \\ &\mathcal{S}_{M_{\text{long}}} := \big\{[\xi(t), v_\xi(t)]^T \,\big|\, \xi(t) \geq \xi_{\min}(t), \\ &\qquad\qquad\qquad v_\xi(t) \in [v_{\min,\xi}, v_{\max,\xi}]\big\}, \\ &\mathcal{U}_{M_{\text{long}}} := \big\{u_\xi(t) \in \big[a_{\min,\xi}\big(v_\xi(t)\big), a_{\max,\xi}\big(v_\xi(t)\big)\big]\big\}. \end{aligned}$$

Proposition 3 (Reachable set of M_{long}): The reachable set of M_{long} for $[t] \geq t_0$ is

$$\begin{aligned} \mathcal{R}([t]; M_{\text{long}}, t_0) = \Bigg\{ [\xi, v_\xi]^T \,\Bigg|\, \\ \xi \in \Bigg[\max\Bigg(\int_{t_0}^{\overline{t}}\int_{t_0}^{\overline{t}} a_{\min,\xi}\big(\underline{v}_\xi(t)\big)\, \mathrm{d}^2t, \xi_{\min}(\overline{t})\Bigg), \\ \int_{t_0}^{\overline{t}}\int_{t_0}^{\overline{t}} a_{\max,\xi}\big(\overline{v}_\xi(t)\big)\, \mathrm{d}^2t\Bigg], \\ v_\xi \in \Bigg[\int_{t_0}^{\overline{t}} a_{\min,\xi}\big(\underline{v}_\xi(t)\big)\, \mathrm{d}t, \int_{t_0}^{\overline{t}} a_{\max,\xi}\big(\overline{v}_\xi(t)\big)\, \mathrm{d}t\Bigg]\Bigg\}, \end{aligned}$$

where the integrals can be solved stepwise according to the discontinuities in (7). □

Proof: The reachable set directly follows from [67, Thm. 2].■

To compute the occupancy of $\mathcal{R}([t]; M_{\text{long}}, t_0)$, we enlarge $[\xi([t])]$ by $\pm\big(\overline{\ell}^2 + \overline{w}^2\big)^{1/2}$ so that all headings $\psi(t) \in \mathbb{R}$ are considered, and we restrict the lateral positions such that the occupancy remains within $\texttt{occ}(D)$.

C. Abstraction based on safe distance (M_{safe})

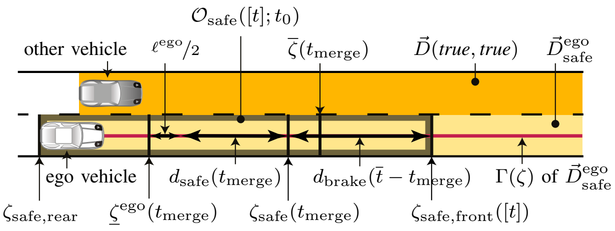
To consider that vehicles have to maintain a safe distance to the ego vehicle (C_{safe}), we determine the area $\mathcal{O}_{\text{safe}}$ that has to be kept free by other vehicles. In contrast to the other abstractions, we need to construct $\mathcal{O}_{\text{safe}}$ such that it is under-approximative, since $\mathcal{O}_{\text{safe}}$ is subtracted from the prediction via set difference.

We apply this abstraction M_{safe} for each forward driving corridor of the ego vehicle without laterally adjacent lanelets, i. e., $\forall \vec{D}^{\text{ego}}_{\text{safe}} \in \vec{\mathcal{D}}^{\text{ego}}(\textit{noLat}, \textit{drivDir})$ (cf. Fig. 5 and Def. 7). Vehicles driving in front of the ego vehicle are excluded for M_{safe}, since it is the responsibility of the ego vehicle to maintain a safe distance in this case. Thus, we only consider vehicles for M_{safe} that drive behind or next to the ego vehicle with the same driving direction or that can eventually merge into the lane of the ego vehicle, i. e., $\big(\vec{\mathcal{D}}(\textit{noLat}, \textit{drivDir}) \not\subseteq \vec{D}^{\text{ego}}_{\text{safe}}\big) \wedge \big(\vec{\mathcal{D}}(\textit{lat}, \textit{drivDir}) \cap \vec{D}^{\text{ego}}_{\text{safe}} \neq \emptyset\big)$ (cf. Fig. 2 and 5).

To compute the safe distance, we assume that vehicles brake until standstill and do not reverse, i. e., $b_{\text{reverse}} = \textit{true}$. We further assume that the ego vehicle may accelerate with $a^{\text{ego}}_{\text{comfort}} \geq 0$ until $v^{\text{ego}}_{\max,\xi}$ (cf. (5)), i. e., its velocity is at least $\underline{v}^{\text{ego}}(t) := \min(\underline{v}^{\text{ego}}_0 + a^{\text{ego}}_{\text{comfort}} \cdot (t - t_0), v^{\text{ego}}_{\max,\xi})$. If another vehicle merges in front of the ego vehicle and performs emergency braking, we assume that the ego vehicle is able to react by braking with $-a^{\text{ego}}_{\max}$ after its reaction delay T^{ego}.

Lemma 3 (Relative safe distance): A vehicle is only allowed to merge in front of the ego vehicle if it maintains at least the safe distance d_{safe}:

$$d_{\text{safe}}([t]) := \begin{cases} d_{\text{safe},1} & \text{if } \big(a_{\max} < a^{\text{ego}}_{\max}\big) \wedge \left(\frac{v^{\text{ego}}}{a^{\text{ego}}_{\max}} < \frac{v_\star}{a_{\max}}\right) \wedge \\ & \big(v_\star < v^{\text{ego}}\big) \\ d_{\text{safe},2} & \text{otherwise,} \end{cases}$$



$\mathcal{O}_{\text{safe}}([t]; t_0)$
other vehicle
$\ell^{\text{ego}}/2$
$\overline{\zeta}(t_{\text{merge}})$
$\vec{D}(\textit{true}, \textit{true})$
$\vec{D}^{\text{ego}}_{\text{safe}}$
ego vehicle
$d_{\text{safe}}(t_{\text{merge}})$
$d_{\text{brake}}(\overline{t} - t_{\text{merge}})$
$\Gamma(\zeta)$ of $\vec{D}^{\text{ego}}_{\text{safe}}$
$\zeta_{\text{safe,rear}}$
$\underline{\zeta}^{\text{ego}}(t_{\text{merge}})$
$\zeta_{\text{safe}}(t_{\text{merge}})$
$\zeta_{\text{safe,front}}([t])$

Fig. 5. The safe distance occupancy $\mathcal{O}_{\text{safe}}([t]; t_0)$ is constructed from $\zeta_{\text{safe,rear}}$ to $\zeta_{\text{safe,front}}([t])$ along $\Gamma(\zeta)$ of each $\vec{D}^{\text{ego}}_{\text{safe}}$ and considers that the other vehicle may legally be allowed to merge in front of the ego vehicle.

where

$$d_{\mathrm{safe},1} := \frac{\big(v - a_{\max} \cdot T^{\mathrm{ego}} - v^{\mathrm{ego}}\big)^2}{2 \cdot (a^{\mathrm{ego}}_{\max} - a_{\max})} + {}^1\!/_2 \cdot a_{\max} \cdot T^{\mathrm{ego}2} + \big(v^{\mathrm{ego}} - v\big) \cdot T^{\mathrm{ego}},$$

$$d_{\mathrm{safe},2} := \frac{v^{\mathrm{ego}2}}{2 \cdot a^{\mathrm{ego}}_{\max}} - \frac{v^2}{2 \cdot a_{\max}} + v^{\mathrm{ego}} \cdot T^{\mathrm{ego}},$$

with, for $[t] \geq t_0$,

$$v := \max\big(\mathtt{proj}_{\overline{v}_\xi}(\mathcal{R}(\overline{t}; M_{\mathrm{long}}, t_0)), 0\big),$$
$$v_\star := \max\big(\mathtt{proj}_{\overline{v}_\xi}(\mathcal{R}(\overline{t}; M_{\mathrm{long}}, t_0)) - a_{\max} \cdot T^{\mathrm{ego}}, 0\big),$$
$$v^{\mathrm{ego}} := \max\big(\underline{v}^{\mathrm{ego}}(\underline{t} + T^{\mathrm{ego}}), 0\big). \qquad \square$$

Proof: The safe distance for exact velocities, a single point in time, and constant velocity of the ego vehicle during T^{ego} is provided in [97, Thm. 2.8]. Since the safe distance is monotonic with respect to v and v^{ego} (which can be easily shown by computing the derivative of $d_{\mathrm{safe},1}$ and $d_{\mathrm{safe},2}$) and both v and v^{ego} are monotonic with respect to t, we can select the bound of each interval such that the safe distance is under-approximated, i. e., $\operatorname{argmin}\big(d_{\mathrm{safe}}(\cdot)\big)$, and we can allow the ego vehicle to accelerate during its reaction delay. ■

To describe the safe distance along the centerline of the road and relative to the minimum position of the ego vehicle for an under-approximation, we define a curvilinear coordinate frame along the reference path $\Gamma(\zeta)$ for each driving corridor $\vec{D}^{\mathrm{ego}}_{\mathrm{safe}}$, where $\Gamma(\zeta)$ corresponds to the centerline (cf. Fig. 2a and 5). Thus, we rewrite the state vector as $\boldsymbol{s}(t) = [\zeta(t), v_\zeta(t)]^T$ and the safe distance in front of the ego vehicle (see Fig. 5) as

$$\zeta_{\mathrm{safe}}([t]) := \underline{\zeta}^{\mathrm{ego}}(\underline{t}) + {}^{\ell^{\mathrm{ego}}}\!/_2 + d_{\mathrm{safe}}([t]), \tag{9}$$

where $\underline{\zeta}^{\mathrm{ego}}(\underline{t})$ is obtained from $\underline{v}^{\mathrm{ego}}(\underline{t})$. However, a vehicle can merge in front of the ego vehicle while maintaining the safe distance at

$$t_{\mathrm{merge}} := \min\big(\{t \geq t_0 \,|\, \overline{\zeta}(t) - {}^{\underline{\ell}}\!/_2 \geq \zeta_{\mathrm{safe}}(t)\}\big), \tag{10}$$

where $\overline{\zeta}(t)$ is obtained by transforming $\mathtt{proj}_{\overline{\xi}}\big(\mathcal{R}(t; M_{\mathrm{long}}, t_0)\big)$ to $\Gamma(\zeta)$ of $\vec{D}^{\mathrm{ego}}_{\mathrm{safe}}$ (see Fig. 5).

Proposition 4 (Safe distance in front of the ego vehicle): The under-approximative safe distance in front of the ego vehicle (see Fig. 5) is

$$\zeta_{\mathrm{safe,front}}([t]) := \begin{cases} \zeta_{\mathrm{safe}}([t]) & \text{if } \overline{t} < t_{\mathrm{merge}} \\ \zeta_{\mathrm{safe}}(t_{\mathrm{merge}}) + d_{\mathrm{brake}}(\overline{t} - t_{\mathrm{merge}}) & \text{if } t_{\mathrm{merge}} \leq \overline{t} < t_{\mathrm{standstill}} \\ \zeta_{\mathrm{safe}}(t_{\mathrm{merge}}) + d_{\mathrm{brake}}(t_{\mathrm{standstill}} - t_{\mathrm{merge}}) & \text{otherwise,} \end{cases}$$

where $d_{\mathrm{brake}}(t) := -{}^1\!/_2 \cdot a_{\max} \cdot t^2 + v_{\mathrm{merge}} \cdot t$, $v_{\mathrm{merge}} := \max(\mathtt{proj}_{\overline{v}_\xi}(\mathcal{R}(t_{\mathrm{merge}}; M_{\mathrm{long}}, t_0)), 0)$, and $t_{\mathrm{standstill}} := {}^{v_{\mathrm{merge}}}\!/_{a_{\max}} + t_0$. □

Proof: For $\overline{t} < t_{\mathrm{merge}}$, (9) holds (cf. Lemma 3). At t_{merge}, the other vehicle can legally merge into $\vec{D}^{\mathrm{ego}}_{\mathrm{safe}}$ and can brake with $-a_{\max}$. Thus, for $t_{\mathrm{merge}} \leq \overline{t} < t_{\mathrm{standstill}}$, the minimum distance between the ego vehicle and the other vehicle is $\zeta_{\mathrm{safe}}(t_{\mathrm{merge}})$ plus its braking distance $d_{\mathrm{brake}}(\overline{t} - t_{\mathrm{merge}})$. For $\overline{t} \geq t_{\mathrm{standstill}}$, the safe distance is no longer increasing, since the other vehicle could have come to a standstill. ■

For the case that the other vehicle remains behind the ego vehicle, the safe distance is the initial position of the ego vehicle (see Fig. 5):

$$\zeta_{\mathrm{safe,rear}} := \underline{\zeta}^{\mathrm{ego}}_0 - {}^{\ell^{\mathrm{ego}}}\!/_2, \tag{11}$$

since this over-approximates a legally allowed emergency braking maneuver by the ego vehicle.

Finally, the safe distance occupancy $\mathcal{O}_{\mathrm{safe}}([t]; t_0)$ is obtained by transforming $[\zeta_{\mathrm{safe,rear}}, \zeta_{\mathrm{safe,front}}([t])]$ to the Cartesian coordinate frame and limiting the lateral positions to $\mathtt{occ}(\vec{D}^{\mathrm{ego}}_{\mathrm{safe}})$, as shown in Fig. 5.

D. Abstraction for kinematic constraints (M_{turn})

So far, we have only covered dynamic constraints that do not consider the nonholonomic constraints of vehicles. In particular, we are interested in the minimum turning radius (C_{turn}):

Definition 14 (Turning radius abstraction M_{turn}): Derived from the kinematic single-track model [98, Sec. 2.2], abstraction M_{turn} removes the maximum area a vehicle does not penetrate when turning with positive velocity and steering angle up to $\delta_{\max}$, as shown in Fig. 6:

$$\mathcal{S}_{M_{\mathrm{turn}}} := \mathbb{R}^4 \setminus (\mathcal{C}_{\mathrm{turn,left}} \cup \mathcal{C}_{\mathrm{turn,right}}),$$

where

$$\mathcal{C}_{\mathrm{turn,left}} := \mathcal{C}\big(\boldsymbol{R}(\psi) \cdot [x_{\mathrm{turn}}, R_{\mathrm{turn}}]^T + [x, y]^T, r_{\mathrm{turn}}\big),$$
$$\mathcal{C}_{\mathrm{turn,right}} := \mathcal{C}\big(\boldsymbol{R}(\psi) \cdot [x_{\mathrm{turn}}, -R_{\mathrm{turn}}]^T + [x, y]^T, r_{\mathrm{turn}}\big),$$

with $x_{\mathrm{turn}} := -{}^\ell\!/_2 + \ell_{\mathrm{ovr}}$, $R_{\mathrm{turn}} := \ell_{\mathrm{wb}} \cdot \tan({}^\pi\!/_2 - \delta_{\max})$, and $r_{\mathrm{turn}} := R_{\mathrm{turn}} - {}^w\!/_2$. The rear overhang ℓ_{ovr} and the wheelbase ℓ_{wb} are vehicle parameters.

Note that the turning radius is often referred to as the radius of the path the outside front wheel is describing during turning. In contrast, our definition of r_{turn} describes the smaller radius of the path of the inside rear wheel (cf. Fig. 6). Moreover, since it is possible to enter the turning circle $\mathcal{C}_{\mathrm{turn}}$ when performing a full turn, constraint C_{turn} assumes that vehicles do not turn within lanes (cf. Tab. I and [89, 14§2]).

Given a set of initial states and uncertain vehicle parameters, we under-approximate the minimum turning radius:

Proposition 5 (Non-reachable occupancy of M_{turn}): As illustrated in Fig. 6, the time-independent area not reachable due to M_{turn} for any $[t] \geq t_0$ is

$$\mathcal{O}_{\mathrm{turn}} = \mathcal{O}_{\mathrm{turn,left}} \cup \mathcal{O}_{\mathrm{turn,right}},$$

where $\mathcal{O}_{\mathrm{turn,left}} =$

$$\bigcap_{[x_0, y_0, \psi_0, x_{\mathrm{turn}}]^T \in \mathbb{S}} \mathcal{C}\big(\boldsymbol{R}(\psi_0) \cdot [x_{\mathrm{turn}}, \underline{R}_{\mathrm{turn}}]^T + [x_0, y_0]^T, \underline{r}_{\mathrm{turn}}\big),$$

with $\mathbb{S} := \big\{\{\underline{x}_0, \overline{x}_0\} \times \{\underline{y}_0, \overline{y}_0\} \times \{\underline{\psi}_0, \overline{\psi}_0\} \times \{\underline{x}_{\mathrm{turn}}, \overline{x}_{\mathrm{turn}}\}\big\}$, $\underline{x}_{\mathrm{turn}} = -{}^{\overline{\ell}}\!/_2 + \underline{\ell}_{\mathrm{ovr}}$, $\overline{x}_{\mathrm{turn}} = \min(-{}^{\underline{\ell}}\!/_2 + \overline{\ell}_{\mathrm{ovr}}, 0)$, $\underline{R}_{\mathrm{turn}} = \underline{\ell}_{\mathrm{wb}} \cdot \tan({}^\pi\!/_2 - \delta_{\max})$, and $\underline{r}_{\mathrm{turn}} = \max(|\underline{R}_{\mathrm{turn}}| - {}^{\overline{w}}\!/_2, 0)$.

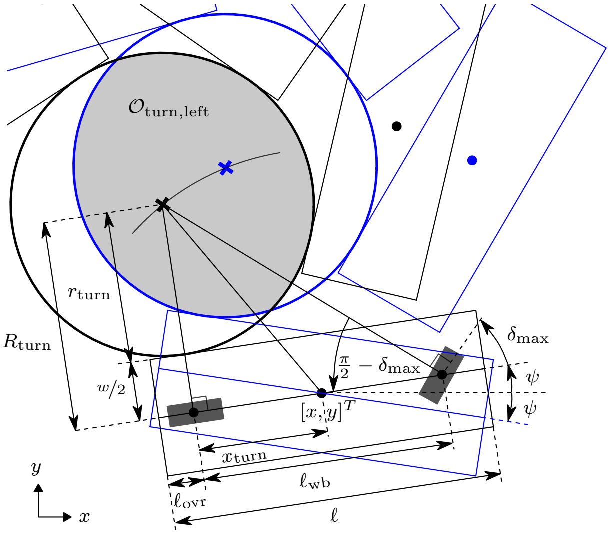


Fig. 6. $\mathcal{O}_{\text{turn}}$ is constructed by intersecting the minimum turning circles of the kinematic single-track vehicle model for all initial states and uncertain vehicle parameters. For the sake of clarity, we only show the vehicle traces and turning circles for $\pm\psi$ (black and blue).

$\mathcal{O}_{\text{turn,right}}$ is constructed analogous as $\mathcal{O}_{\text{turn,left}}$ except that $\underline{R}_{\text{turn}}$ is multiplied by -1. □

Proof: To under-approximate Def. 14 for all intervals $[x_0]$, $[y_0]$, $[\psi_0]$, $[\ell]$, $[w]$, $[\ell_{\text{ovr}}]$, and $[\ell_{\text{wb}}]$, we would require infinitely many intersections of all possible combinations with all interval values. However, we can reduce the solution to a finite amount of intersections. The intersection of the solution using $\underline{x}_0$ and the one using $\overline{x}_0$ contains the solution for all $[x_0]$, since x_0 only linearly translates the solution over a closed interval. In addition, $[x_0]$ has no influence on the other variables. Both properties also apply to $[y_0]$ as well as $[\ell]$ and $[\ell_{\text{ovr}}]$. We only need to consider the upper bound of $[w]$, since $\overline{w}$ is minimizing r_{turn} and all intersections of arbitrary circles with the same center always contain the circle with the minimum radius. Due to the same reason, the lower bound of $[\ell_{\text{wb}}]$ suffices. The centers (see crosses in Fig. 6) of the turning circles with $\psi_0 \in [\underline{\psi}_0, \overline{\psi}_0]$ lie on a circular arc with radius $\|[x_{\text{turn}}, R_{\text{turn}}]^T\|_2$. Since $\underline{r}_{\text{turn}} \leq \|[x_{\text{turn}}, R_{\text{turn}}]^T\|_2$ for all possible x_{turn} and R_{turn}, the intersection of the solution using $\underline{\psi}_0$ and the one using $\overline{\psi}_0$ contains the solution for all $[\psi_0]$. By intersecting the solution of all possible combinations of the remaining extreme values given in $\mathbb{S}$, we obtain the result. ■

In summary, abstraction M_{turn} especially reduces the over-approximation in the prediction for low initial velocities and small initial heading intervals. For high measurement uncertainties, however, $\mathcal{O}_{\text{turn}}$ can also be empty.

E. Abstraction based on priority traffic rules (M_{prio})

The only constraint we have not yet considered is C_{prio}.

Definition 15 (Priority-based abstraction M_{prio}): Based on priority traffic rules, abstraction M_{prio} restricts the occupancy to $\mathcal{W}_{\text{prio}}(t)$, which is provided by the environment model (cf. Def. 3), without constraining the dynamics.

The occupancy of M_{prio} for $[t] \geq t_0$ is $\mathcal{O}_{\text{prio}}([t];t_0) = \bigcup_{t\in[t]} \mathcal{W}_{\text{prio}}(t)$. Since pedestrians often do not observe the priority of vehicular traffic, e. g., by jaywalking, $\mathcal{O}^{\text{ped}}_{\text{prio}}(t;t_0)$ can be extended to a more sophisticated prediction of pedestrians stepping on the road and potentially crossing it as described in [70, Sec. III-B].

F. Summary of abstractions

After introducing all abstractions and the computation of their reachable set and occupancy, we summarize the prediction for each type of traffic participant for a time interval $[t] \geq t_0$ in accordance with Lemma 1 and such that all applicable constraints of Tab. I are considered (cf. Tab. II). For vehicles, the reachable occupancy is

$$\begin{aligned}\mathcal{O}^{\text{veh}}([t];t_0) := \mathtt{occ}\big(\mathcal{R}([t];M_{\text{acc}},t_0),\mathcal{A}\big) \cap \mathcal{O}^{\complement}_{\text{turn}} \\ \cap\, \mathtt{occ}\big(\mathcal{R}([t];M_{\text{long}},t_0),\mathcal{A}\big) \\ \cap\, \mathcal{O}^{\complement}_{\text{safe}}([t];t_0) \cap \mathcal{O}_{\text{prio}}([t];t_0),\end{aligned} \tag{12}$$

where $\mathcal{O}^{\complement}$ denotes the complement of $\mathcal{O}$. For pedestrians, the reachable occupancy is

$$\begin{aligned}\mathcal{O}^{\text{ped}}([t];t_0) := \mathtt{occ}\big(\mathcal{R}([t];M_{\text{acc}},t_0),\mathcal{A}\big) \\ \cap\, \mathtt{occ}\big(\mathcal{R}([t];M_{\text{vel}},t_0),\mathcal{A}\big) \cap \mathcal{O}_{\text{prio}}([t];t_0),\end{aligned} \tag{13}$$

since the other abstractions are only applicable to vehicles.

VI. Constraint Management

Our assumptions can become violated, if other traffic participants misbehave, i. e., perform an unacceptable behavior, or if measurement uncertainties are very high. To enable the ego vehicle to react to these violations, we validate the constraint parameters $\mathcal{Q}^p$ of each traffic participant based on the current environment model Ω_0 and, if available, on the environment model Ω_{c-1} of the previous planning cycle.

We adjust the constraint parameters in case of violations such that observed but unacceptable behavior gets no longer excluded from our prediction, as described in Tab. III, which extends [14, Tab. III]. Numerical parameters are updated to the measured state plus a threshold, where we use thresholds $\Delta_{a_{\max}}, \Delta_{v_{\max}}, \Delta_{f_{\text{speed}}}, \Delta_{v_{\min}}$ to prevent an updated constraint from directly being violated again, and threshold $\Delta_{v_{\text{reverse}}}$ to prevent noisy velocity measurements slightly below 0 from being considered as reversing. Boolean parameters are updated to *false* so that violated constraints get disabled. C_{engine} and C_{turn} also get disabled in case of a violation by setting their parameter to the maximum value (cf. Tab. III).

Our default set of parameters $\mathcal{Q}$ is provided in Tab. IV. Note that these values are suggestions to over-approximate the real and legal motions of traffic participants in accordance with M_{real}, but that they can be adjusted to user preferences. Especially, the parameters for C_{turn} to under-approximate the turning radius should be adapted to the applicable legal regulations of the target country (cf. [89, 30§4–5]). The default values for C_{lane} forbid vehicles to overtake in a lane not

TABLE III
CONSTRAINT MANAGEMENT.

Constraint of Tab. I	**If formalization of Tab. I evaluates to *false* for $t = t_0$, update parameters as**
$C_{a_{\max}}$	$a_{\max} \leftarrow \overline{a}_0 + \Delta_{a_{\max}}$
$C_{v_{\max}}$	$v_{\max} \leftarrow \max\left(\lvert\underline{v}_0\rvert, \lvert\overline{v}_0\rvert\right) + \Delta_{v_{\max}}$
C_{speedLim}	$f_{\text{speed}} \leftarrow \frac{\overline{v}_0}{v_{\text{speedLim}}} + \Delta_{f_{\text{speed}}}$
C_{engine}	$v_S \leftarrow \infty$
C_{reverse}	if $\underline{v}_0 < \Delta_{v_{\text{reverse}}}$: $b_{\text{reverse}} \leftarrow$ *false*
$C_{v_{\min}}$	$v_{\min} \leftarrow \underline{v}_0 - \Delta_{v_{\min}}$
C_{turn}	$\delta_{\max} \leftarrow \pi/2$
C_{road}	$b_{\text{road}} \leftarrow$ *false*
C_{prio}	$b_{\text{prio}} \leftarrow$ *false*

C_{lane}	if b_{lane_1} = *noLat* : $b_{\text{lane}_1} \leftarrow$ *lat*
	else if b_{lane_2} = *drivDir* : $b_{\text{lane}_2} \leftarrow$ *anyDir*
	else: $b_{\text{road}} \leftarrow$ *false*

TABLE IV
DEFAULT PARAMETERS.

Constraint of Tab. I	**Parameter and its default value**					
$C_{a_{\max}}$	$a_{\max}^{\text{veh}}$	$8.0\,\text{m/s}^2$	$a_{\max}^{\text{ped}}$	$1.0\,\text{m/s}^2$	$a_{\max}^{\text{cyc}}$	$3.5\,\text{m/s}^2$
	$\Delta_{a_{\max}}$	$0.5\,\text{m/s}^2$				
$C_{v_{\max}}$	$v_{\max}^{\text{veh}}$	$70.0\,\text{m/s}$	$v_{\max}^{\text{ped}}$	$2.0\,\text{m/s}$	$v_{\max}^{\text{cyc}}$	$12.0\,\text{m/s}$
	$\Delta_{v_{\max}}^{\text{veh}}$	$0.5\,\text{m/s}$				
C_{speedLim}	f_{speed}	1.2	$\Delta_{f_{\text{speed}}}$	0.1		
C_{engine}	v_S^{veh}	$7.0\,\text{m/s}$	v_S^{cyc}	∞		
C_{reverse}	b_{reverse}	*true*	$\Delta_{v_{\text{reverse}}}$	$-1.0\,\text{m/s}$		
$C_{v_{\min}}$	$v_{\min}$	$-10.0\,\text{m/s}$	$\Delta_{v_{\min}}$	$1.0\,\text{m/s}$		
C_{turn}	$\underline{\ell}_{\text{wb}}^{\text{car}}$	$1.8\,\text{m}$	$\underline{\ell}_{\text{wb}}^{\text{motcyc}}$	$1.1\,\text{m}$	$\underline{\ell}_{\text{wb}}^{\text{cyc}}$	$0.8\,\text{m}$
	$\underline{\ell}_{\text{wb}}^{\text{truck}}$	$3.0\,\text{m}$	$\underline{\ell}_{\text{wb}}^{\text{bus}}$	$3.0\,\text{m}$		
	$\delta_{\max}$	$1.0\,\text{rad}$	$\underline{\ell}_{\text{ovr}}$	0		
	$\overline{\ell}_{\text{ovr}}^{\text{car}}$	$3.7\,\text{m}$	$\overline{\ell}_{\text{ovr}}^{\text{motcyc}}$	$1.0\,\text{m}$	$\overline{\ell}_{\text{ovr}}^{\text{cyc}}$	$1.0\,\text{m}$
	$\overline{\ell}_{\text{ovr}}^{\text{truck}}$	$3.7\,\text{m}$	$\overline{\ell}_{\text{ovr}}^{\text{bus}}$	$4.9\,\text{m}$		
C_{road}	b_{road}	*true*				
C_{prio}	b_{prio}	*true*				
C_{lane}	b_{lane_1}	*lat*	b_{lane_2}	*drivDir*		
C_{safe}	T^{ego}	$1.0\,\text{s}$	$a_{\text{comfort}}^{\text{ego}}$	$1.0\,\text{m/s}^2$		
$\mathcal{A}^{\text{phantom}}$	w	0	ℓ	$0.5\,\text{m}$	r	$0.25\,\text{m}$

appropriate to the direction of traffic, since such a behavior is only allowed if not endangering or interfering with oncoming traffic [88, 11§2(c)], and thus it is forbidden in the vicinity of the ego vehicle.

VII. EXPERIMENTAL RESULTS

For a prediction that claims to be over-approximative (cf. our problem statement in Sec. II-E), it is crucial to demonstrate this property. In our previous work, we have already shown conformance of the prediction on recorded data of 1074 vehicles in [67, Sec. V-C] and of 400 pedestrians in [70, Sec. IV-A], and we have evaluated how conservative the prediction is against a high-fidelity vehicle model in [67, Sec. V-B]. These results demonstrate that the ground-truth trajectories were always contained in the prediction and that the over-approximation was not unreasonably conservative.

In this paper, we want to demonstrate that our prediction works on complicated, real-world scenarios and, despite being over-approximative, allows the ego vehicle to obtain collision-free trajectories. Therefore, we simulate an urban intersection with occlusions in Sec. VII-A, and, for the first time, we present real-world experiments with test vehicles in Sec. VII-B and VII-C. The video attachment of this paper contains further results. For all experiments, we used the parameters of Tab. IV if not noted otherwise and implemented Alg. 1 without considering interaction, i. e., we omitted the optional line 13. Initial positions are over-approximated either by rectangles aligned with the mean heading of the traffic participant or by circles to ease the consideration of the traffic participant's size. As representation for the predicted set, we choose polygons for the position domain and intervals for the other states. Thus, the states are not coupled with each other to allow for efficient computations despite some over-approximations.

A. Intersection with occlusions and priorities

Fig. 7 presents an urban intersection with different detected traffic participants. The road network is provided with one layer for vehicles and one for bicycles, and the speed limit of all lanes is $v_{\text{speedLim}} = 13.89\,\text{m/s}$. Due to occlusions and a limited sensor range with radius of 33 m, the field of view $\mathcal{F}_0$ is restricted. To capture this risk, our approach creates 3 phantom vehicles, 2 phantoms cyclists, and 24 static phantom obstacles. The prediction result is shown for a time horizon of 1.0 s with a time step size of 0.1 s. The oncoming phantom vehicle (from the top) is forbidden to make a left turn, since the ego vehicle has the right of way, which is modeled by $\mathcal{W}_{\text{prio}}(t)$. Based on the predicted occupancies, the ego vehicle can decide when to safely proceed into the intersection.

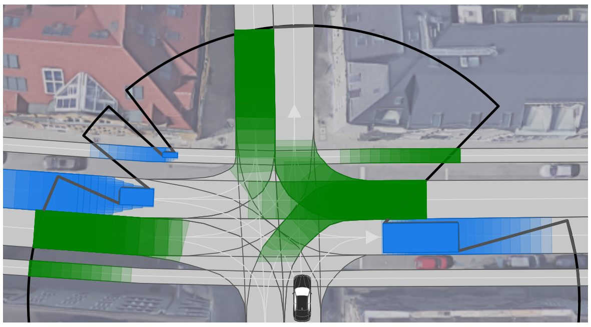


Fig. 7. Urban intersection (CommonRoad ID: S=DEU_Muc-30_1_S-1:2018b [99]): the ego vehicle (black) has to yield to crossing traffic. Since two vehicles (blue) and one cyclist (blue) cause occlusions, we create phantom traffic participants (dynamic: green, static: grey) at the boundary of the field of view (black). (background image: Google, GeoBasis-DE/BKG)

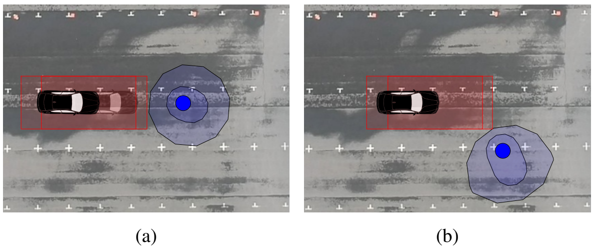


Fig. 8. Online verification of parking maneuvers for the ego vehicle (black) considering a pedestrian (blue). The predicted occupancy of the pedestrian (light blue) and the occupancy of the verified maneuver of the ego vehicle including safety margins (red) are both shown for two time intervals, $t \in [0\,\mathrm{s}, 0.8\,\mathrm{s}]$ and $[0.8\,\mathrm{s}, 1.6\,\mathrm{s}]$. (a) The ego vehicle executed a verified braking maneuver such that it definitely will come to a stop in front of the pedestrian. The recorded stopping position of the ego vehicle $1.6\,\mathrm{s}$ later is shown transparently. (b) Since the occupancies did not intersect anymore, a new maneuver for the ego vehicle has been verified as safe. A video of this real-world experiment is attached to this paper.

B. Online verification considering pedestrians

We have performed online verification of maneuvers in the presence of pedestrians. Online verification ensures that the ego vehicle only executes trajectories that have been verified as safe [3], [8]. For our experiments, we want to achieve passive safety, i. e., a trajectory is verified as safe if the maneuver is collision-free against all acceptable future behaviors of surrounding traffic participants and brings the ego vehicle to a standstill. In particular, our self-driving BMW 5 series test vehicle has to avoid collisions with pedestrians in a parking lot, i. e., an unstructured environment. The ego vehicle receives trajectories that are following a predefined path with constant velocity $v_{\mathrm{des}}^{\mathrm{ego}} = 2.0\,\mathrm{m/s}$ for a planning horizon of $t_h = 1.6\,\mathrm{s}$ (with constant time offset to be robust against processing time delays). These intended trajectories are not aware of pedestrians. Thus, we append a path-consistent braking profile to the given intended trajectory such that the ego vehicle comes to a stop within t_h, and we predict the pedestrians using $a_{\mathrm{max}}^{\mathrm{ped}} = 2.0\,\mathrm{m/s^2}$. If the new trajectory does not intersect with the predicted occupancies, the ego vehicle will execute it; otherwise, it will keep executing the trajectory that has been verified in the previous planning cycle.

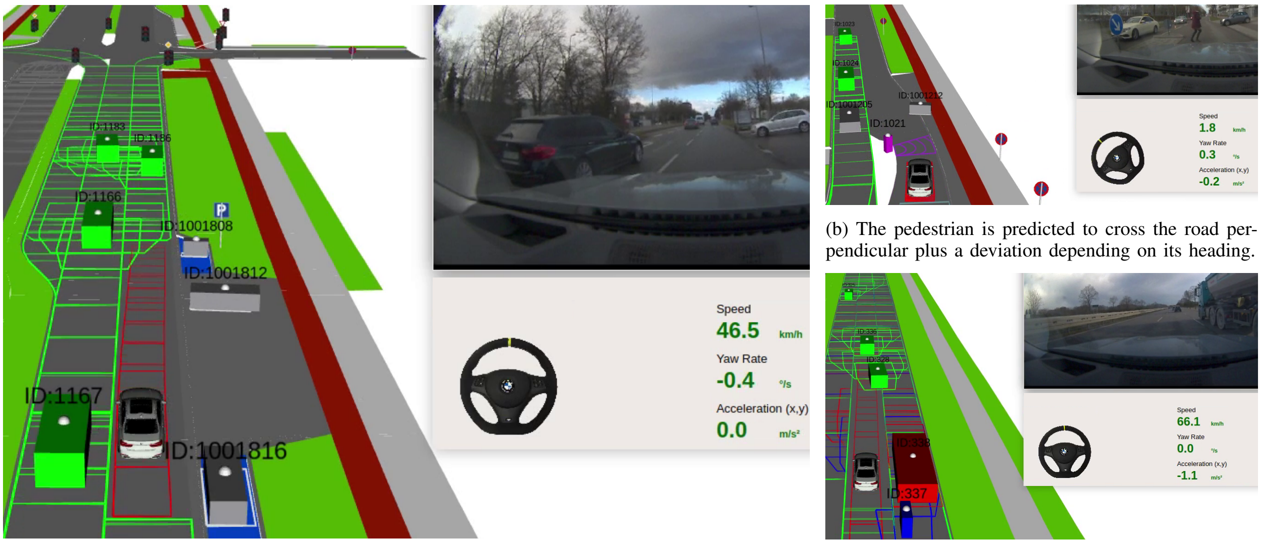
This online verification has been executed on our test vehicle on November 09, 2018, and Fig. 8 shows recordings of these real-world experiments. Since the pedestrian was blocking the path of the ego vehicle, the ego vehicle eventually could not verify a new trajectory and, by executing the previously verified trajectory, came to a stop (see Fig. 8a). A few seconds later, the pedestrian walked away and a new trajectory has been verified as safe (see Fig. 8b).

C. Online experiments on public roads

We have executed our prediction online in a test vehicle on public roads. Therefore, we implemented our approach in C++ on a BMW 7 series test vehicle. The environment model provides the initial states of surrounding traffic participants based on [33] and the rectangular field of view without occlusions that extends $100\,\mathrm{m}$ in longitudinal and $60\,\mathrm{m}$ in lateral direction of the current pose of the ego vehicle. We use the planner of [15] to obtain trajectories for the ego vehicle that are collision-free against all predicted occupancies and bring the ego vehicle to a standstill; for the few cases the initial velocity is too high to come to a standstill within the planning horizon, we constrain the final state to comply with safe distances to predicted traffic participants. The prediction and planning horizon is $3.0\,\mathrm{s}$ with a time step size of $0.25\,\mathrm{s}$.

We conducted four test drives in Germany from 1.30 p.m. to 5 p.m. on Wednesday, March 13, 2019. Each test drive was along the $17\,\mathrm{km}$ long route between the BMW Autonomous Driving Campus in Unterschleißheim and the BMW Research and Innovation Center in Munich and contains both urban and rural multi-lane roads with speed limits ranging from $8.3\,\mathrm{m/s}$ to $27.8\,\mathrm{m/s}$. While we have performed the prediction online, we did not perform the trajectory planning closed-loop but offline in a postprecessing step, since approval by authorities has not yet been given. In all test drives combined, we have predicted $163,715$ detected and $211,863$ phantom traffic participants (dynamic and static) in $29,818$ replanning steps. Fig. 1 and 9 show exemplary results. Predicted occupancies and planned trajectories are shown for the full time horizon. The visualization of the ego vehicle, its trajectory, and other traffic participants can have a slight time offset to each other due to the asynchronous updates. Overall, the results demonstrate that the prediction performs well in arbitrary road networks and with vast numbers of traffic participants. Even in crowded environments, the prediction incorporates the interaction with the ego vehicle and allows to obtain collision-free trajectories, while containing all acceptable behaviors of other traffic participants. Only in a few situations, a new safe trajectory for the ego vehicle could not be obtained, as shown in Fig. 10; since the prediction was not provided with $\mathcal{W}_{\mathrm{prio}}(t)$, it could not consider the right of way for the ego vehicle.

During the real-world experiments, our legal specification has been violated a few times by the recorded traffic participants. Tab. V evaluates how often the constraint management had to update the values of the constraints according to Tab. III, when using as initial values the ones of Tab. IV except for v_{min}. For each parameter, we present its relative number of updates for all detected, dynamic traffic participants in our test drives (i. e., for $90,779$ motorized vehicles, $15,650$ pedestrians, and $4,770$ cyclists), the maximum value it has been updated to, and the mean value of all updated values. Note that the maximum and mean values are the measured values plus our thresholds (cf. Tab. III). In most cases, the violations were caused by high measurement uncertainties or an incomplete environment model, e. g., when no driving corridor was provided for a traffic participant (see Fig. 10). In other cases, a traffic participant indeed violated our specification. Since the mean values of all violations are only slightly above the initial values, the initial parameterization seems reasonable, but can be adjusted to user preferences. To reduce the influence of violated constraints on the safety of motion plans, we refer to [11]. Legal safety can be ensured despite constraint violations by planning fail-safe trajectories [15] and switching to a reactive mode for collision mitigation in case of inevitable collisions.



(a) Even on multi-lane roads, the ego vehicle has enough free space, since the safe distance forbids passing vehicles from merging directly in front of the ego vehicle.

(b) The pedestrian is predicted to cross the road perpendicular plus a deviation depending on its heading.

(c) While overtaking a truck, a vehicle ahead is merging into the lane of the ego vehicle.

Fig. 9. Set-based prediction of various traffic participants (car: green, truck/bus: red, cyclist: turquoise, motorcyclist: blue, pedestrian: magenta, static: grey box, phantom: grey area) in different urban and rural scenarios of our real-world experiments. Based on the predicted occupancies, we successfully obtained collision-free trajectories (red) for the ego vehicle (silver-colored vehicle). Videos of further real-world experiments are attached.

Let us finally evaluate the required computation times for the prediction, i. e., for the loop over all traffic participants in Alg. 1. The test vehicle is equipped with an Intel i7 6900K processor and 64 GB memory; the frequency of the processor is underclocked from 3.2 GHz to 1.2 GHz to improve the energy consumption and heat management. The mean computation time for one planning cycle was 9.86 ms with a standard deviation of 12.02 ms for a prediction horizon of 2.0 s. Note that the outliers mostly occurred due to high computational load caused by other software modules. Further experiments showed that the computation time is linear with the prediction horizon.

TABLE V
EVALUATION OF THE CONSTRAINT MANAGEMENT.

Constraint of Tab. I	Parameter	Initial value	Mean value of updates	Max. value of updates	Num. of updates
$C_{a_{\max}}$	$a_{\max}^{\text{veh}}$	$8.0\,\text{m}/\text{s}^2$	$9.50\,\text{m}/\text{s}^2$	$15.14\,\text{m}/\text{s}^2$	0.03 %
	$a_{\max}^{\text{ped}}$	$1.0\,\text{m}/\text{s}^2$	$2.18\,\text{m}/\text{s}^2$	$7.41\,\text{m}/\text{s}^2$	6.91 %
	$a_{\max}^{\text{cyc}}$	$3.5\,\text{m}/\text{s}^2$	$4.57\,\text{m}/\text{s}^2$	$7.89\,\text{m}/\text{s}^2$	0.59 %
$C_{v_{\max}}$	$v_{\max}^{\text{veh}}$	$70.0\,\text{m}/\text{s}$	n/a	n/a	0.00 %
	$v_{\max}^{\text{ped}}$	$2.0\,\text{m}/\text{s}$	$3.39\,\text{m}/\text{s}$	$6.78\,\text{m}/\text{s}$	5.83 %
	$v_{\max}^{\text{cyc}}$	$12.0\,\text{m}/\text{s}$	$12.92\,\text{m}/\text{s}$	$13.48\,\text{m}/\text{s}$	0.27 %
C_{speedLim}	f_s^{veh}	1.2	1.43	3.36	0.21 %
C_{engine}	v_s^{veh}	$7.0\,\text{m}/\text{s}$	n/a	∞	0.83 %
C_{reverse}	$b_{\text{reverse}}^{\text{veh}}$	*true*	n/a	n/a	2.01 %
	$b_{\text{reverse}}^{\text{cyc}}$	*true*	n/a	n/a	0.27 %
$C_{v_{\min}}$	$v_{\min}^{\text{veh}}$	$-1.0\,\text{m}/\text{s}$	$-1.64\,\text{m}/\text{s}$	$-10.93\,\text{m}/\text{s}$	2.01 %
	$v_{\min}^{\text{cyc}}$	$-1.0\,\text{m}/\text{s}$	$-2.10\,\text{m}/\text{s}$	$-5.84\,\text{m}/\text{s}$	0.27 %
$C_{\text{lane}}/\ C_{\text{road}}$	$b_{\text{road}}^{\text{veh}}$	*true*	n/a	n/a	2.27 %
	$b_{\text{road}}^{\text{cyc}}$	*true*	n/a	n/a	31.07 %

VIII. Conclusions and Future Work

We have presented a set-based prediction for provably safe motion planning based on legal safety. Our prediction is guaranteed to contain all acceptable behaviors in accordance with a legal specification. This is achieved by rigorous computations in a formal manner, nondeterministic models that over-approximate the dynamics of the traffic participants, and conservative parameterization. As prediction features, we use longitudinal and lateral dynamics, the motion history, and the types of traffic participants in combination with contextual information and the field of view.

For the first time, we have validated our prediction in test vehicles. These real-world experiments demonstrate that our

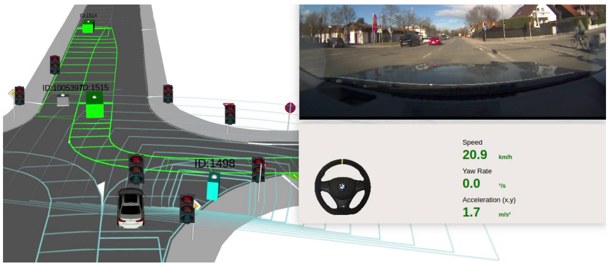


Fig. 10. Situation of our real-world experiments (cf. Fig. 9) in which a safe trajectory could not be obtained. Since the environment model did not restrict the priority-based positions $\mathcal{W}_{\text{prio}}(t)$ for the oncoming vehicle (ID 1515), the prediction allows this vehicle to traverse the lane of the ego vehicle. In addition, since the environment model did not provide a driving corridor for the cyclist (ID 1498) next to the ego vehicle, the constraint management updated $b_{\text{road}} \leftarrow$ *false* and the prediction of this cyclist can only use M_{acc}.

prediction runs online in arbitrary traffic scenarios and that motion planners are able to obtain collision-free trajectories despite the over-approximative prediction and even in congested environments. In addition, our constraint management successfully dealt with traffic participants that violate traffic rules, high measurement uncertainties, and incomplete environment models.

For a good performance of the prediction, we require a detailed and precise environment model with strictly bounded measurement uncertainties. Future work includes more restrictive bounds on the admissible velocity by considering the curvature of the road and on the admissible lateral acceleration (e. g., based on [100]) while remaining over-approximative. It also seems interesting to use our proposed set-based prediction as propagation model for object tracking.

Acknowledgments

The authors thank Christian Pek, Sebastian Kaster, and the BMW Group for making the experiments of Sec. VII-C possible and Fabian Schönert for conducting the experiments of Sec. VII-B. The authors also thank Lukas Willinger, Maxime Allard, and Alexander Gaul for their valuable contributions to improving the prediction. Furthermore, the authors gratefully acknowledge financial support received from the Free State of Bavaria, the BMW Group within the CAR@TUM project, and the European Research Council (ERC) project justITSELF under grant agreement number 817629.

References

[1] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 333–347, 2013.

[2] R. Kianfar, P. Falcone, and J. Fredriksson, "Safety verification of automated driving systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, pp. 73–86, 2013.

[3] 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.

[4] S. Mitsch, K. Ghorbal, D. Vogelbacher, and A. Platzer, "Formal verification of obstacle avoidance and navigation of ground robots," *Int. Journal of Robotics Research*, vol. 36, no. 12, pp. 1312–1340, 2017.

[5] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018.

[6] K. Maček, D. Vasquez, T. Fraichard, and R. Siegwart, "Towards safe vehicle navigation in dynamic urban scenarios," *Automatika*, vol. 50, no. 3-4, pp. 184–194, 2009.

[7] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv:1708.06374v1 [cs.RO]*, pp. 1–25, 2017.

[8] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, 2020, [in press].

[9] S. Söntges and M. Althoff, "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, 2018.

[10] S. Söntges, M. Koschi, and M. Althoff, "Worst-case analysis of the time-to-react using reachable sets," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1891–1897.

[11] C. Pek, M. Koschi, M. Werling, and M. Althoff, "Enhancing motion safety by identifying safety-critical passageways," in *Proc. of the 56th IEEE Conference on Decision and Control*, 2017, pp. 320–326.

[12] M. Koschi, C. Pek, S. Maierhofer, and M. Althoff, "Computationally efficient safety falsification of adaptive cruise control systems," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*,

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

[14] 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.

[15] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1447–1454.

[16] P. F. Orzechowski, A. Meyer, and M. Lauer, "Tackling occlusions & limited sensor range with set-based safety verification," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1729–1736.

[17] 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.

[18] S. Manzinger, C. Pek, and M. Althoff, "Using reachable sets for trajectory planning of automated vehicles," *IEEE Transactions on Intelligent Vehicles*, 2020, [in press].

[19] W. Zhan, C. Liu, C. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proc. of the 19th IEEE Int. Conf. on Intelligent Transportation Systems*, 2016, pp. 459–464.

[20] S. Vaskov, H. Larson, S. Kousik, M. Johnson-Roberson, and R. Vasudevan, "Not-at-fault driving in traffic: A reachability-based approach," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2785–2790.

[21] F. Gruber and M. Althoff, "Anytime safety verification of autonomous vehicles," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1708–1714.

[22] C. Pek, M. Koschi, and M. Althoff, "An online verification framework for motion planning of self-driving vehicles with safety guarantees," in *AAET - Automatisiertes und vernetztes Fahren*, 2019, pp. 260–274.

[23] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira, "Effortless creation of safe robots from modules through self-programming and self-verification," *Science Robotics*, vol. 4, no. 31, 2019.

[24] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[25] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. Journal of Robotics Research*, 2020, [available online].

[26] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, G. Markkula, A. Schieben, F. Tango, N. Merat, and C. Fox, "Pedestrian models for autonomous driving part II: High level models of human behaviour," *IEEE Transactions on Intelligent Transportation Systems*, 2020, [available online].

[27] 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, 2019.

[28] D. González, J. Pérez, V. Milanés, 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.

[29] 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.

[30] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: A literature review on threat-assessment techniques," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 101–113, 2019.

[31] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[32] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[33] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr, "Grid-based object tracking with nonlinear dynamic state and shape estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2874–2893, 2019.

[34] Y. Emzivat, J. Ibanez-Guzman, H. Illy, P. Martinet, and O. H. Roux, "A formal approach for the design of a dependable perception system for autonomous vehicles," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2452–2459.

[35] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3266–3273.

[36] M. T. Le, F. Diehl, T. Brunner, and A. Knoll, "Uncertainty estimation for deep neural object detectors in safety-critical applications," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3873–3878.

[37] D. Elliott, W. Keen, and L. Miao, "Recent advances in connected and automated vehicles," *Journal of Traffic and Transportation Engineering*, vol. 6, no. 2, pp. 109–131, 2019.

[38] S. W. Loke, "Cooperative automated vehicles: A review of opportunities and challenges in socially intelligent vehicles beyond networking," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 509–518, 2019.

[39] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[40] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 137–147, 2008.

[41] C. Lienke, C. Wissing, M. Keller, T. Nattermann, and T. Bertram, "Predictive driving: Fusing prediction and planning for automated highway driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 456–467, 2019.

[42] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[43] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model- and learning-based framework for interaction-aware maneuver prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1538–1550, 2016.

[44] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[45] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 3999–4006.

[46] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. A. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2019.

[47] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 353–359.

[48] H. Xiong, F. B. Flohr, S. Wang, B. Wang, J. Wang, and K. Li, "Recurrent neural network architectures for vulnerable road user trajectory prediction," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 171–178.

[49] T. A. Wheeler, P. Robbel, and M. J. Kochenderfer, "Analysis of microscopic behavior models for probabilistic modeling of driver behavior," in *Proc. of the 19th IEEE Int. Conf. on Intelligent Transportation Systems*, 2016, pp. 1604–1609.

[50] J. Quehl, H. Hu, O. Ş. Taş, E. Rehder, and M. Lauer, "How good is my prediction? Finding a similarity measure for trajectory prediction evaluation," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–6.

[51] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[52] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, 2018.

[53] L. Sun, W. Zhan, D. Wang, and M. Tomizuka, "Interactive prediction for multiple, heterogeneous traffic participants with multi-agent hybrid dynamic bayesian network," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 1025–1031.

[54] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, "Deep neural networks for markovian interactive scene prediction in highway scenarios," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 685–692.

[55] C. Tang, J. Chen, and M. Tomizuka, "Adaptive probabilistic vehicle trajectory prediction through physically feasible bayesian recurrent neural network," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 3846–3852.

[56] S. Zernetsch, H. Reichert, V. Kress, K. Doll, and B. Sick, "Trajectory forecasts with uncertainties of vulnerable road users by means of neural networks," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 810–815.

[57] E. A. I. Pool, J. F. P. Kooij, and D. M. Gavrila, "Context-based cyclist path prediction using recurrent neural networks," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 824–830.

[58] I. Batkovic, M. Zanon, N. Lubbe, and P. Falcone, "A computationally efficient model for pedestrian motion prediction," in *Proc. of the European Control Conference*, 2018, pp. 374–379.

[59] P. Nadarajan and M. Botsch, "Probability estimation for predicted-occupancy grids in vehicle safety applications based on machine learning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 1285–1292.

[60] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 399–404.

[61] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 2056–2063.

[62] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2052–2059.

[63] J. Wu, J. Ruenz, and M. Althoff, "Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1285–1292.

[64] W. Zhan, A. de La Fortelle, Y. Chen, C. Chan, and M. Tomizuka, "Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1150–1156.

[65] M. Brännström, F. Sandblom, and L. Hammarstrand, "A probabilistic framework for decision-making in collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 637–648, 2013.

[66] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[67] 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.

[68] M. Hartmann and D. Watzenig, "Optimal motion planning with reachable sets of vulnerable road users," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 891–898.

[69] M. Koschi and M. Althoff, "Interaction-aware occupancy prediction of road vehicles," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1885–1892.

[70] M. Koschi, C. Pek, M. Beikirch, and M. Althoff, "Set-based prediction of pedestrians in urban environments considering formalized traffic rules," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2704–2711.

[71] K. Driggs-Campbell, R. Dong, and R. Bajcsy, "Robust, informative human-in-the-loop predictions via empirical reachable sets," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 300–309, 2018.

[72] P. Zechel, R. Streiter, K. Bogenberger, and U. Göhner, "Pedestrian occupancy prediction for autonomous vehicles," in *Proc. of the 3rd IEEE Int. Conf. on Robotic Computing*, 2019, pp. 230–235.

[73] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. Moon, and J. Song, "Safe navigation of a mobile robot considering visibility of environment," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3941–3950, 2009.

[74] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 267–283, 2012.

[75] D. Phan, J. Yang, R. Grosu, S. A. Smolka, and S. D. Stoller, "Collision avoidance for mobile robots with limited sensing and limited information about moving obstacles," *Formal Methods in System Design*, vol. 51, no. 1, pp. 62–86, 2017.

[76] Y. Nager, A. Censi, and E. Frazzoli, "What lies in the shadows? Safe and computation-aware motion planning for autonomous vehicles using intent-aware dynamic shadow regions," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 5800–5806.

[77] R. Matsumi, P. Raksincharoensak, and M. Nagai, "Study on autonomous intelligent drive system based on potential field with hazard anticipation," *Journal of Robotics and Mechatronics*, vol. 27, no. 1, pp. 5–11, 2015.

[78] F. Damerow, T. Puphal, Y. Li, and J. Eggert, "Risk-based driver assistance for approaching intersections of limited visibility," in *Proc. of the IEEE Int. Conf. on Vehicular Electronics and Safety*, 2017, pp. 178–184.

[79] M. Lee, K. Jo, and M. Sunwoo, "Collision risk assessment for possible collision vehicle in occluded area based on precise map," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–6.

[80] M. Yu, R. Vasudevan, and M. Johnson-Roberson, "Occlusion-aware risk assessment for autonomous driving in urban environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2235–2241, 2019.

[81] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *Proc. of the 17th IEEE Int. Conf. on Intelligent Transportation Systems*, 2014, pp. 392–399.

[82] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Scalable decision making with sensor occlusions for autonomous driving," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 2076–2081.

[83] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A POMDP maneuver planner for occlusions in urban scenarios," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1909–1916.

[84] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, and X. Zhang, "Decision making through occluded intersections for autonomous driving," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2449–2455.

[85] M. Schratter, M. Bouton, M. J. Kochenderfer, and D. Watzenig, "Pedestrian collision avoidance system for scenarios with occlusions," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1054–1060.

[86] M. Sadou, V. Polotski, and P. Cohen, "Occlusions in obstacle detection for safe navigation," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2004, pp. 716–721.

[87] O. Ş. Taş and C. Stiller, "Limited visibility and uncertainty aware motion planning for automated driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1171–1178.

[88] M. Naumann, H. Konigshof, M. Lauer, and C. Stiller, "Safe but not overcautious motion planning under occlusions and limited sensor range," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 140–145.

[89] United Nations Economic Commission for Europe, "Convention on road traffic," United Nations Conference on Road Traffic, 1968, (consolidated version of 2006), https://www.unece.org/fileadmin/DAM/trans/conventn/Conv_road_traffic_EN.pdf.

[90] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.

[91] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, "Consistent outdoor vehicle localization by bounded-error state estimation," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1211–1216.

[92] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1672–1679.

[93] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[94] *Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body (ISO 13855:2010)*, ISO Std., 2010.

[95] A. Censi, K. Slutsky, T. Wongpiromsarn, D. Yershov, S. Pendleton, J. Fu, and E. Frazzoli, "Liability, ethics, and culture-aware behavior specification using rulebooks," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 8536–8542.

[96] J. Casey, *A Sequel to the First Six Books of the Elements of Euclid, Containing an Easy Introduction to Modern Geometry with Numerous Examples*. Dublin: Hodges, Figgis, & Co., 1888.

[97] A. Rizaldi, "Formal specification, monitoring, and verification of autonomous vehicles with Isabelle/HOL," Dissertation, Technische Universität München, 2019, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20191218-1484146-1-4.

[98] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2012.

[99] 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.

[100] P. Zechel, R. Streiter, K. Bogenberger, and U. Göhner, "Assumptions of lateral acceleration behavior limits for prediction tasks in autonomous vehicles," in *Proc. of the 7th Int. Conf. on Mechatronics Engineering*, 2019, pp. 1–6.

Markus Koschi is a research associate in the Cyber-Physical Systems Group at the Department of Informatics of the Technical University of Munich, Germany, since 2016. He received the Master of Science degree in mechanical engineering from the Technical University of Munich in 2016. His research interests include motion planning, behavior prediction, safety verification, and falsification of autonomous vehicles towards a future of zero traffic accidents.

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.

3.2 ITSC 2018: Set-Based Prediction of Pedestrians in Urban Environments Considering Formalized Traffic Rules [67]

Summary A set-based prediction tailored to the acceptable behaviors of pedestrians does not yet exist. Thus, this section extends the general set-based prediction, which has been presented in the previous Section 3.1, especially by formalizing traffic rules applicable for pedestrians. By incorporating the dynamics of pedestrians, contextual information, and traffic rules related to the interaction between vehicles and pedestrians, we obtain tight over-approximations of pedestrians' reachable occupancy (cf. Problem statement 1), which significantly improves predictions compared to a solely dynamical model. Particular focus lies on the models to predict pedestrians disregarding traffic rules, such as jaywalking pedestrians, to solve Problem statement 2. We introduce atomic constraints that automatically adapt to possible violations so that all behaviors relevant for approaching vehicles are included in the prediction. In particular, we distinguish between behavior that does not enter the road, that stops as quickly as possible, that crosses the road perpendicular, and that occupies only the edge of the road.

Using datasets containing 400 recorded pedestrians, we validate our proposed method. Conformance of the over-approximation is achieved, since the ground-truth trajectories are fully contained in the predicted occupancy sets. We also demonstrate the usefulnesses of our set-based prediction for evasive maneuver planning. Our results show that by using the predicted occupancies, the ego vehicle is able to avoid the jaywalking pedestrian. Thus, our approach can improve systems for collision avoidance with pedestrians. Real-world vehicle experiments, which are proposed as future work at the end of the publication, are realized in Section 4.2.

Contributions of M. K. M. K. developed the motion models and their reachability analysis (together with M. A.), the formalization of traffic rules (together with M. B.), and the constraint management. M. K. designed, conducted, and evaluated the experiments (together with C. P.). M. K. wrote most of the article.

Conference paper ©2018 IEEE. Reprinted, with permission, from Markus Koschi, Christian Pek, Mona Beikirch, and Matthias Althoff, Set-Based Prediction of Pedestrians in Urban Environments Considering Formalized Traffic Rules, in Proc. of the 21st IEEE International Conference on Intelligent Transportation Systems.

Attachments The video attachment of this publication is available at go.tum.de/074008.

2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Set-Based Prediction of Pedestrians in Urban Environments Considering Formalized Traffic Rules

Markus Koschi[1], Christian Pek[1,2], Mona Beikirch[1,2], and Matthias Althoff[1]

***Abstract*— Set-based predictions can ensure the safety of planned motions, since they provide a bounded region which includes all possible future states of nondeterministic models of other traffic participants. However, while autonomous vehicles are tested in urban environments, a set-based prediction tailored to pedestrians does not exist yet. This paper addresses this problem and presents an approach for set-based predictions of pedestrians using reachability analysis. We obtain tight over-approximations of pedestrians' reachable occupancy by incorporating the dynamics of pedestrians, contextual information, and traffic rules. In addition, since pedestrians often disregard traffic rules, our constraints automatically adapt so that such behaviors are included in the prediction. Using datasets of recorded pedestrians, we validate our proposed method and demonstrate its use for evasive maneuver planning of automated vehicles.**

I. Introduction

A. Motivation

Automated vehicles may endanger other traffic participants in the event that they misjudge a traffic situation. In urban environments in particular, vulnerable road users such as pedestrians impose strict safety requirements. For example, if autonomous vehicles do not consider that an approaching pedestrian might try to cross the road at the last second (cf. Fig. 1), a fatal collision could be inevitable.

To prevent such situations at an early stage, the future motion of pedestrians needs to be accurately predicted [1], [2]. Current probabilistic approaches are limited when predicting all feasible and legal future motion, since they are not designed to enclose all behaviors given an uncertain pedestrian model. In contrast, set-based predictions guarantee that all planned motions are safe, even when traffic participants deviate from the most likely prediction [3]. Recently, a prediction approach using reachability analysis to account for any feasible future motion of other traffic participants in a set-based fashion was proposed [4]. However, a set-based prediction method for pedestrians considering both structured and unstructured environments does not yet exist, making it difficult to provide advanced safety systems which ensure the safety of vulnerable road users.

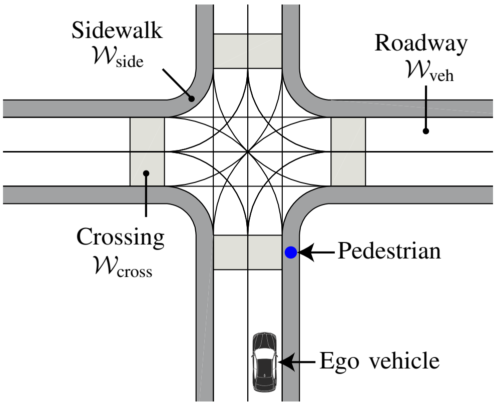

Sidewalk
$\mathcal{W}_{\text{side}}$
Roadway
$\mathcal{W}_{\text{veh}}$
Crossing
$\mathcal{W}_{\text{cross}}$
Pedestrian
Ego vehicle

Fig. 1. The safety of the ego vehicle's planned motions can be guaranteed for given model assumptions by predicting all possible future behaviors of the pedestrian, which may include crossing the road even when traffic rules (e.g., a red light for the pedestrian) forbid such behavior.

B. Related Work

We review existing work on pedestrian prediction for automated vehicles in unknown environments categorized by whether they compute *a)* a single behavior, *b)* a probability distribution of multiple behaviors, or *c)* a bounded set of future behaviors. In order to apply such predictions, we require the current state of pedestrians from sensor data, which can be obtained as described in [5]–[7]; however, this process itself is beyond the scope of this work.

a) Single behavior: The probability of whether pedestrians intend to cross the roadway is computed in [8]–[12] using one or more of the following sources: motion information (previous path and current position), situation awareness (e.g., head pose), and contextual information (e.g., proximity to curb or intersection). Based on the predicted intention, the most likely behavior can be inferred, while other works directly compute a single trajectory [13], [14] or the time until the pedestrian will most likely cross [15].

b) Probability distribution: Predicting only a single behavior may suffice for short-term prediction; however, since many possible maneuvers exist, it is beneficial to compute a probability distribution of future behaviors by considering the possible goals of pedestrians [16]–[20].

c) Set of future behaviors: To verify that one does not collide with a pedestrian, a bounded set containing its possible future behaviors must be considered. While dynamic-based models have been used in [21]–[23], set-based models which integrate map-based information or traffic rules have not yet been developed, to the best of our knowledge.

[1] Department of Informatics, Technical University of Munich, 85748 Garching, Germany.
[2] BMW Group, 85716 Unterschleissheim, Germany.
markus.koschi@tum.de, christian.pek@tum.de, mona.beikirch@tum.de, matthias.althoff@tum.de

This work was partially supported by the BMW Group within the CAR@TUM project, the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF, and the German Research Foundation (DFG) under grant number AL 1185/7-1.

978-1-7281-0323-5/18/$31.00 ©2018 IEEE

C. Contribution

This paper significantly extends previous work on set-based predictions [4], [23] by considering not only motorized traffic participants but also pedestrians, while exploiting traffic rules based on the given environment map. This extension will be available in the next version of our open-source prediction tool SPOT[1]. More specifically, our method is the first that can:

1) predict the feasible future motion of pedestrians in a formal and set-based manner,
2) obtain tight over-approximative occupancies by making use of formalized traffic rules and contextual information,
3) explicitly consider measurement uncertainties in the initial state of pedestrians, and
4) guarantee the safety of planned motions according to our assumptions.

The remainder of this paper is organized as follows. Sec. II introduces the required models and definitions, and Sec. III explains the set-based prediction of pedestrians. Sec. IV demonstrates our approach by using different datasets of recorded pedestrians and by evasive planning for autonomous vehicles. Finally, Sec. V concludes the paper.

II. Preliminaries

A. Road Model

We model our environment in $\mathbb{R}^2$ using lanelets, which are atomic, interconnected, and drivable road segments [24]. Lanelets are defined using a left and right bound represented by a linearly interpolated list of points. As Fig. 1 shows, we distinguish two types of lanelets: vehicular lanelets (i.e., roadways) and pedestrian lanelets (i.e., sidewalks/pavements and crossings).

Definition 1 (Road networks)
We define the following types of road networks:

- *The vehicular network is the union of all vehicular lanelets and is denoted by $\mathcal{W}_{veh} \subset \mathbb{R}^2$.*
- *The pedestrian network $\mathcal{W}_{ped} \subset \mathbb{R}^2$ is the union of all pedestrian lanelets, i.e., sidewalks $\mathcal{W}_{side}$ and crossings $\mathcal{W}_{cross}$. We use $\mathcal{W}_{cross}^{prio}(t)$ to denote the crossings a pedestrian is allowed to cross at time t (cf. Sec. III-B).*
- *The forbidden network $\mathcal{W}_{forbid} := \mathcal{W}_{veh} \cap \mathcal{W}_{ped}^{\complement}$ is the part of $\mathcal{W}_{veh}$ pedestrians are not allowed to enter (where $\mathcal{W}_{ped}^{\complement}$ denotes the complement of $\mathcal{W}_{ped}$, cf. Fig. 4). The boundary of $\mathcal{W}_{forbid}$ is denoted by $\delta\mathcal{W}_{forbid}$.*

Let a disk, i.e., a circular area, with center $[c_x, c_y]^T$ and radius r be denoted as $\mathcal{C}\big([c_x, c_y]^T, r\big) := \big\{[s_x, s_y]^T \,|\, (s_x - c_x)^2 + (s_y - c_y)^2 \leq r^2\big\}$. If $c_x = c_y = 0$, we just write $\mathcal{C}(r)$. The following predicates are defined using first-order logic to argue about the position of pedestrians:

Definition 2 (Not intruding $\mathcal{W}_{\mathbf{forbid}}$)
The predicate $\mathrm{notInWf}(\mathcal{X}_s, r)$ evaluates to true if all points $[s_x, s_y]^T \in \mathcal{X}_s \subset \mathbb{R}^2$ intrude the vehicular network by at most the distance r:

$$\mathrm{notInWf}(\mathcal{X}_s, r) \Leftrightarrow \big(\mathcal{W}_{forbid} \ominus \mathcal{C}(r)\big) \cap \mathcal{X}_s = \emptyset,$$

where $\ominus$ denotes the Minkowski difference defined for sets $\mathcal{A}$ and $\mathcal{B}$ as $\mathcal{A} \ominus \mathcal{B} := (\mathcal{A}^{\complement} \oplus \mathcal{B})^{\complement}$ using the Minkowski addition $(\mathcal{A} \oplus \mathcal{B} := \{a + b \,|\, a \in \mathcal{A}, b \in \mathcal{B}\})$.

Definition 3 (Conforming to crossing priority)
The predicate $\mathrm{confPrio}(\mathcal{X}_s, t)$ evaluates to true if none of the points $[s_x, s_y]^T \in \mathcal{X}_s$ are located in a forbidden crossing at time t:

$$\mathrm{confPrio}(\mathcal{X}_s, t) \Leftrightarrow \mathcal{W}_{cross} \cap \mathcal{W}_{cross}^{prio}(t)^{\complement} \cap \mathcal{X}_s = \emptyset.$$

B. Reachable Set of Pedestrians

The motion of a pedestrian can be described by the differential equation

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

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, and t is the time. The possible states and inputs are bounded by the sets $\mathcal{X}$ and $\mathcal{U}$, respectively. We denote the initial time by t_0, the final time by $t_f > t_0$, an input trajectory by $u(\cdot)$, and a possible solution of (1) at time t by $\chi\big(t, x(t_0), u(\cdot)\big)$.

Definition 4 (Reachable set)
The reachable set $\mathcal{R} \subseteq \mathcal{X}$ of (1) is the set of states which are reachable at time t from an initial set $\mathcal{X}^0 \subseteq \mathcal{X}$ at time t_0 and subject to the set of inputs $\mathcal{U}$:

$$\mathcal{R}(t) = \Big\{\chi\big(t, x(t_0), u(\cdot)\big) \,\Big|\, x(t_0) \in \mathcal{X}^0, \\ \forall t^\star \in [t_0, t] : \chi\big(t^\star, x(t_0), u(\cdot)\big) \in \mathcal{X}, u(t^\star) \in \mathcal{U}\Big\}.$$

Our state vector is $x = [s_x, s_y, v_x, v_y]^T$, where s_x and s_y denote the position, v_x and v_y the velocity, each in x- and y-direction, respectively. We define the occupancy of a state as:

Definition 5 (Occupancy of a state)
The operator $\mathrm{occ}(x)$ returns the set of points in the two-dimensional Cartesian space occupied by the pedestrian in state x due to its circular dimensions with radius r_{ped}:

$$\mathrm{occ}(x) := \big\{Px \oplus \mathcal{C}(r_{ped})\big\},$$

where P is the projection matrix $P = [I\,\mathbf{0}] \in \mathbb{R}^{2\times 4}$, I the identity matrix, and $\mathbf{0}$ a matrix of zeros, both with proper dimensions. Given a set of states $\mathcal{X}$, the operator is defined as $\mathrm{occ}(\mathcal{X}) := \{\mathrm{occ}(x) \,|\, x \in \mathcal{X}\}$.

To obtain the future occupancy of pedestrians efficiently, we over-approximate their reachable occupancy:

Definition 6 (Over-approximative occupancy set)
Based on Def. 4 and Def. 5, the occupancy set $\mathcal{O}(t)$ over-approximates the set of occupied points which are reachable by the pedestrian: $\mathcal{O}(t) \supseteq \mathrm{occ}\big(\mathcal{R}(t)\big)$.

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

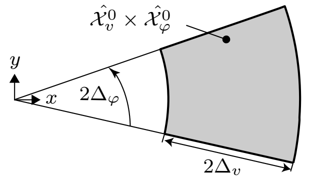


Fig. 2. The set of initial velocities, i.e., $\hat{\mathcal{X}}_v^0 \times \hat{\mathcal{X}}_\varphi^0$, is modeled by an annulus sector, i.e., a circular ring sector.

Since an occupancy $\mathcal{O}(t)$ can be non-convex, we represent it by a polygon, and since a collision check with the intended trajectory of the ego vehicle requires an infinite number of points in time to be checked, we compute occupancies for consecutive time intervals $\tau_k = [t_k, t_{k+1}] \subseteq [t_0, t_f]$ with time step size $\Delta t = t_{k+1} - t_k$.

The initial set $\mathcal{X}^0$ in Def. 4 contains measurement uncertainties. Using a polar coordinate system to describe v_x and v_y by the radius v and the polar angle φ, we introduce the following initial sets:

$$\hat{\mathcal{X}}_s^0 := \mathcal{C}(s_0, \Delta_s), \tag{2}$$

$$\hat{\mathcal{X}}_v^0 := [v_0 - \Delta_v, v_0 + \Delta_v], \tag{3}$$

$$\hat{\mathcal{X}}_\varphi^0 := [\varphi_0 - \Delta_\varphi, \varphi_0 + \Delta_\varphi], \tag{4}$$

where $s_0 := [s_{x_0}, s_{y_0}]^T$, and Δ_s, Δ_v, and Δ_φ denote the measurement uncertainty of the corresponding variable. As Fig. 2 shows, the set of initial velocities is bounded by an annulus sector. The initial set $\hat{\mathcal{X}}^0$ is constructed by the Cartesian product of the partial initial sets $\hat{\mathcal{X}}^0 := \hat{\mathcal{X}}_s^0 \times \hat{\mathcal{X}}_v^0 \times \hat{\mathcal{X}}_\varphi^0$; the set $\hat{\mathcal{X}}^0$ in Cartesian coordinates is denoted by $\mathcal{X}^0$. The initial occupancy is $\mathcal{O}^0 := \mathrm{occ}(\mathcal{X}^0)$, which accounts for uncertainties in the pedestrian's dimensions by choosing r_{ped} as the maximum of the measured radii.

III. Prediction of Pedestrians

To efficiently compute a tight over-approximative occupancy of (1), we use two types of occupancies: 1) the occupancy $\mathcal{O}_{\text{dyn}}(t)$ considering the dynamics of the pedestrian and 2) the occupancy $\mathcal{O}_{\text{rule}}(t)$ considering possible states according to traffic rules, as described in Sec. III-A and Sec. III-B, respectively. Then, the over-approximative occupancy is the intersection of both over-approximations:

$$\forall \tau_k \subseteq [t_0, t_f] : \mathcal{O}(\tau_k) = \mathcal{O}_{\text{dyn}}(\tau_k) \cap \mathcal{O}_{\text{rule}}(\tau_k). \tag{5}$$

A. Dynamic-Based Occupancy

We use a kinematic model for pedestrians:

Definition 7 (Dynamic model of pedestrians) *The dynamics of a pedestrian are described by a velocity- and acceleration-bounded point mass:*

$$\ddot{s}_x = u_x,\ \ddot{s}_y = u_y, \tag{6a}$$

$$\sqrt{|u_x|^2 + |u_y|^2} \leq a_{\max}, \tag{6b}$$

$$\sqrt{|v_x|^2 + |v_y|^2} \leq v_{\max}, \tag{6c}$$

where u_x and u_y denote the acceleration input in the x- and y-direction, respectively, $a_{\max}$ the maximum allowed acceleration, and $v_{\max}$ the maximum allowed velocity.

To efficiently obtain the occupancy, we do not directly perform reachability analysis on (6), but use the approach proposed by [23]: We separately compute an acceleration-constrained occupancy $\mathcal{O}_{\text{acc}}(t)$ considering only the constraint (6b) and a velocity-constrained occupancy $\mathcal{O}_{\text{vel}}(t)$ considering only the constraint (6c), as explained in Sec. III-A.1 and III-A.2, respectively.

1) Acceleration-constrained occupancy:

Proposition 1 (Reachable positions $\mathcal{R}_{\text{acc}}^{\text{pos}}(t)$ for point in time) *The reachable positions $\mathcal{R}_{acc}^{pos}(t)$ of* (6a) *with* (6b) *are*

$$\mathcal{R}_{acc}^{pos}(t) = \Gamma_{hom}(t)\mathcal{X}^0 \oplus \Gamma_{inp}(t)\mathcal{U},$$

where

$$\Gamma_{hom}(t) = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \end{bmatrix}, \qquad \Gamma_{inp}(t) = \frac{1}{2}t^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Proof: Let us first write (6a) in state-space form:

$$\underbrace{\begin{bmatrix} \dot{s}_x \\ \dot{s}_y \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} s_x \\ s_y \\ v_x \\ v_y \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_x \\ u_y \end{bmatrix}}_{u}. \tag{7}$$

In general, the exact reachable set of linear systems cannot be computed, except for when A is nilpotent or the eigenvalues are purely real or imaginary [25]. Since A is nilpotent (A^2 is a matrix of zeros), we can compute the exact reachable set as presented in [26, Sec. 3.2]:

$$\mathcal{R}_{\text{acc}}(t) = e^{At}\mathcal{X}^0 \oplus \bigoplus_{i=0}^{\infty} \frac{A^i t^{i+1}}{(i+1)!} B\mathcal{U}$$

$$\overset{nilpotence}{=} (I + At)\mathcal{X}^0 \oplus (B\mathcal{U}t) \oplus \frac{1}{2}At^2 B\mathcal{U}.$$

Since we are only interested in the reachable set of the positions, we multiply the above solution with the projection

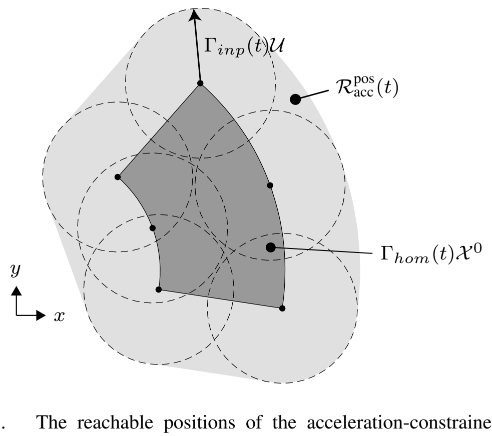

$\Gamma_{inp}(t)\mathcal{U}$
$\mathcal{R}_{\text{acc}}^{\text{pos}}(t)$
$\Gamma_{hom}(t)\mathcal{X}^0$
y
x

Fig. 3. The reachable positions of the acceleration-constrained model $\mathcal{R}_{\text{acc}}^{\text{pos}}(t)$ are bounded by the Minkowski addition of $\Gamma_{hom}(t)\mathcal{X}^0$ with $\Gamma_{inp}(t)\mathcal{U}$ (cf. Prop. 1).

TABLE I

DEFINITION OF THE OCCUPANCIES BASED ON FORMALIZED TRAFFIC RULES, WHICH ARE COMPUTED DEPENDING ON THEIR BOOLEAN VARIABLE.

Constraint	Boolean	Description	Traffic rule [29]	Occupancy
C_{prio}	b_{prio}	Within $\mathcal{W}_{\text{cross}}$, pedestrians are allowed to cross if they have priority over vehicular traffic, i.e., at pedestrian crossings, at intersections when pedestrians have green traffic lights, and at intersections without traffic lights when vehicles take a turn[2].	20§6(b), 21§2	$\mathcal{O}_{\text{prio}}(t) = \begin{cases} \mathcal{W}^{\text{prio}}_{\text{cross}}(t), & b_{\text{prio}}, \\ \mathcal{W}_{\text{cross}}, & \neg b_{\text{prio}}. \end{cases}$
C_{stop}	b_{stop}	When (carelessly) stepping on the roadway outside $\mathcal{W}_{\text{cross}}$ (i.e., onto $\mathcal{W}_{\text{forbid}}$), the pedestrian immediately slows down with a_{stop} to come to a stop as soon as possible in order to not impede vehicular traffic.	20§6(a,c)	$\mathcal{O}_{\text{stop}} = \begin{cases} \emptyset, & b_{\text{stop}}, \\ \mathcal{C}\big(s_0, r_{\text{stop}} + r_{\text{ped}}\big), & \neg b_{\text{stop}} \wedge \text{notInWf}(\mathcal{O}^0, 0), \\ \mathcal{C}\big(p_{\mathcal{W}}(s_0), r_{\text{stop}} + r_{\text{ped}}\big), & \neg b_{\text{stop}} \wedge \neg\text{notInWf}(\mathcal{O}^0, 0). \end{cases}$
C_{perp}	b_{perp}	Crossing the roadway outside $\mathcal{W}_{\text{cross}}$ is not allowed. If crossing nevertheless, the shortest path of width ξ_{perp}, which is perpendicular to the driving direction, must be chosen.	20§6(c,d)	$\mathcal{O}_{\text{perp}} = \begin{cases} \emptyset, & b_{\text{perp}}, \\ \mathcal{W}_{\text{perp}} \cap \mathcal{W}_{\text{forbid}}, & \neg b_{\text{perp}}. \end{cases}$
C_{slack}	b_{slack}	Walking on the roadway is not allowed; $\mathcal{W}_{\text{forbid}}$ may be entered by the margin ξ_{slack} only if no usable sidewalk is provided.	20§2(a), 20§3, 20§4	$\mathcal{O}_{\text{slack}} = \begin{cases} \emptyset, & b_{\text{slack}}, \\ (\mathcal{W}^{\complement}_{\text{forbid}} \oplus \mathcal{C}(\xi_{\text{slack}})) \cap \mathcal{W}_{\text{forbid}}, & \neg b_{\text{slack}}. \end{cases}$

matrix P (cf. Fig. 3):

$$\mathcal{R}^{\text{pos}}_{\text{acc}}(t) = P\mathcal{R}_{\text{acc}}(t) = \underbrace{\begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \end{bmatrix}}_{\Gamma_{\text{hom}}} \mathcal{X}^0 \oplus \underbrace{\frac{1}{2}t^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\Gamma_{\text{inp}}} \mathcal{U}.$$

■

Next, we consider the reachable set for time intervals.

Proposition 2 (Reachable positions $\mathcal{R}^{\text{pos}}_{\text{acc}}(\tau_k)$ for time interval) *The reachable positions of the acceleration-bounded model for a time interval $\tau_k = [t_k, t_{k+1}]$ are*

$$\mathcal{R}^{pos}_{acc}(\tau_k) = \operatorname{conv}\big(\Gamma_{hom}(t_k)\mathcal{X}^0, \Gamma_{hom}(t_{k+1})\mathcal{X}^0\big) \oplus \Gamma_{inp}(t_{k+1})\mathcal{U},$$

where $\operatorname{conv}(\mathcal{A}, \mathcal{B})$ *returns the convex hull of the sets $\mathcal{A}$ and $\mathcal{B}$.*

Proof: The proof follows directly from [27, Alg. 1], where the matrix $\mathcal{F}$ in that algorithm is a matrix of zeros due to the nilpotence of A. ■

Finally, the acceleration-constrained occupancy for a single point in time is $\mathcal{O}_{\text{acc}}(t) = \operatorname{occ}\big(\mathcal{R}^{\text{pos}}_{\text{acc}}(t)\big)$ and $\mathcal{O}_{\text{acc}}(\tau_k) = \operatorname{occ}\big(\mathcal{R}^{\text{pos}}_{\text{acc}}(\tau_k)\big)$ for a time interval.

2) Velocity-constrained occupancy: Up until now, we have ignored the maximum velocity constraint in (6c). Let us first determine the earliest point in time when the maximum velocity is reached:

$$v_{\max} = v_0 + \Delta_v + a_{\max} t_{v_{\max}} \Leftrightarrow t_{v_{\max}} = \frac{v_{\max} - (v_0 + \Delta_v)}{a_{\max}}.$$

When starting at the origin with the maximum velocity in all directions, the reachable set is a disk centered at the origin with radius $v_{\max}(t - t_{v_{\max}})$. Thus, an over-approximation of the reachable positions considering the velocity constraint for $t > t_{v_{\max}}$ is

$$\mathcal{O}_{\text{vel}}(t) = \mathcal{O}_{\text{acc}}(t_{v_{\max}}) \oplus \mathcal{C}\big(v_{\max}(t - t_{v_{\max}})\big). \quad (8)$$

Due to the monotonic growth of $\mathcal{C}\big(v_{\max}(t - t_{v_{\max}})\big)$, it follows that $\mathcal{C}\big(v_{\max}(t_{k+1} - t_{v_{\max}})\big) \supseteq \mathcal{C}\big(v_{\max}(t_k - t_{v_{\max}})\big)$, and thus

$$\mathcal{O}_{\text{vel}}(\tau_k) = \mathcal{O}_{\text{acc}}(t_{v_{\max}}) \oplus \mathcal{C}\big(v_{\max}(t_{k+1} - t_{v_{\max}})\big). \quad (9)$$

Since $\mathcal{O}_{\text{vel}}(t)$ is not required for $t \leq t_{v_{\max}}$, the overall dynamic-based occupancy is

$$\mathcal{O}_{\text{dyn}}(\tau_k) = \begin{cases} \mathcal{O}_{\text{acc}}(\tau_k), & t_k \leq t_{v_{\max}}, \\ \mathcal{O}_{\text{acc}}(\tau_k) \cap \mathcal{O}_{\text{vel}}(\tau_k), & t_k > t_{v_{\max}}. \end{cases} \quad (10)$$

B. Rule-Based Occupancy

We incorporate formalized traffic rules into our prediction [28]. As a legal source, we use the Vienna Convention on Road Traffic [29]. We assume that pedestrians adhere to traffic rules and do not obstruct vehicular traffic [29, 7§1]. However, if pedestrians violate rules, we have to take necessary precautions to avoid endangering pedestrians [29, 21§1].

Pedestrians are generally not allowed to leave the pedestrian network and enter the roadway [29, 20§2]. Since this rule has different cases of violations, we deduce four atomic constraints described in Tab. I; these constraints each have a Boolean variable, denoted by b, which allows us to enable and disable this constraint individually.

The occupancies resulting from these constraints are illustrated in Fig. 4 and are computed as presented in Tab. I, where we use the following variables: The distance required

[2] In order to automatically consider that pedestrians have priority over turning vehicles at intersections, it is necessary to compute this occupancy depending on the intended motion of the ego vehicle.

TABLE II

THE CONSTRAINT MANAGEMENT DEACTIVATES OR ADAPTS THE CONSTRAINTS IF UNDERLYING ASSUMPTIONS ARE VIOLATED.

Constraint	**Parameter with default values**	**Condition adapting the constraint variable**
C_{prio}	—	$b_{\text{prio}} \Leftrightarrow \text{confPrio}(\mathcal{O}^0, t_0) \wedge b_{\text{perp}}$
C_{stop}	$a_{\text{stop}} = 0.6\,\text{m}/\text{s}^2$	$b_{\text{stop}} \Leftrightarrow \forall t \in [t_0, t_f] : \big(\mathcal{O}_{\text{dyn}}(t) \cap \mathcal{O}_{\text{rule}}(t)\big) \ominus \mathcal{C}(r_{\text{ped}}) \neq \emptyset$
C_{perp}	$\xi_{\text{perp}} = 2.0\,\text{m}$	$b_{\text{perp}} \Leftrightarrow \text{notInWf}\big(\mathcal{O}^0, \max(\xi_{\text{slack}}, r_{\text{stop}} + r_{\text{ped}})\big) \wedge \big(b_{\text{stop}} \vee \forall t \in [t_0, t_f] :$ $\big(\mathcal{O}_{\text{dyn}}(t) \cap \mathcal{O}_{\text{rule}}(t)\big) \ominus \mathcal{C}(r_{\text{ped}}) \neq \emptyset\big)$
C_{slack}	$\xi_{\text{slack}} = 1.0\,\text{m}$	$b_{\text{slack}} \Leftrightarrow \text{notInWf}(\mathcal{O}^0, 0)$
$C_{a_{\max}}$	$a_{\max} = 0.6\,\text{m}/\text{s}^2$, $\Delta_{a_{\max}} = 0.05\,\text{m}/\text{s}^2$	$a_{\max} \leftarrow \max(a_{\max}, a_0 + \Delta_a + \Delta_{a_{\max}})$
$C_{v_{\max}}$	$v_{\max} = 2\,\text{m}/\text{s}$, $\Delta_{v_{\max}} = 0.1\,\text{m}/\text{s}$	$v_{\max} \leftarrow \max(v_{\max}, v_0 + \Delta_v + \Delta_{v_{\max}})$

for the pedestrian to stop with deceleration a_{stop} from its current velocity is $r_{\text{stop}} := \frac{1}{2a_{\text{stop}}}(v_0 + \Delta_v)^2 + \Delta_s$. The point on $\delta\mathcal{W}_{\text{forbid}}$ closest to the center of $\mathcal{O}^0$ is $p_{\mathcal{W}}(s_0) := \underset{p \in \delta\mathcal{W}_{\text{forbid}}}{\text{argmin}} \|p - s_0\|_2$ (cf. Fig. 4), and the unit vector at $p_{\mathcal{W}}(s_0)$ tangential to $\delta\mathcal{W}_{\text{forbid}}$ is denoted by $t_{\mathcal{W}}(s_0)$. We choose s_0 as the center of $\mathcal{O}_{\text{stop}}$, since we assume that pedestrians immediately slow down to avoid entering the road (i.e., entering $\mathcal{W}_{\text{forbid}}$); however, for a pedestrian already located in $\mathcal{W}_{\text{forbid}}$, we assume that the pedestrian had started slowing down when entering $\mathcal{W}_{\text{forbid}}$, and thus choose $p_{\mathcal{W}}(s_0)$ as the center of $\mathcal{O}_{\text{stop}}$ (cf. Tab. I). For $\mathcal{O}_{\text{perp}}$, we compute the area perpendicular to the roadway by

$$\mathcal{W}_{\text{perp}} := \left\{ \begin{bmatrix} s_x \\ s_y \end{bmatrix} \,\middle|\, \left\| t_{\mathcal{W}}(s_0)^T \left(\begin{bmatrix} s_x \\ s_y \end{bmatrix} - p_{\mathcal{W}}(s_0) \right) \right\| \leq \frac{\xi_{\text{perp}}}{2} \right\},$$

where the parameter ξ_{perp} describes the width of this corridor (cf. Fig. 4).

Finally, we define the rule-based occupancy $\mathcal{O}_{\text{rule}}(t)$ as the area of the whole vehicular and pedestrian network respecting the constraints of Tab. I:

Definition 8 (Rule-based occupancy)
Using the partial occupancies from Tab. I, the rule-based occupancy of a pedestrian is

$$\mathcal{O}_{rule}(t) = \mathcal{W}_{side} \cup \mathcal{O}_{prio}(t) \cup \mathcal{O}_{stop} \cup \mathcal{O}_{perp} \cup \mathcal{O}_{slack}.$$

Note that all partial occupancies of $\mathcal{O}_{\text{rule}}(t)$, except $\mathcal{O}_{\text{prio}}(t)$, are constant over the prediction horizon, and $\mathcal{W}_{\text{side}}$, $\mathcal{W}_{\text{cross}}$, and $\mathcal{O}_{\text{slack}}$ can be precomputed offline for given road networks.

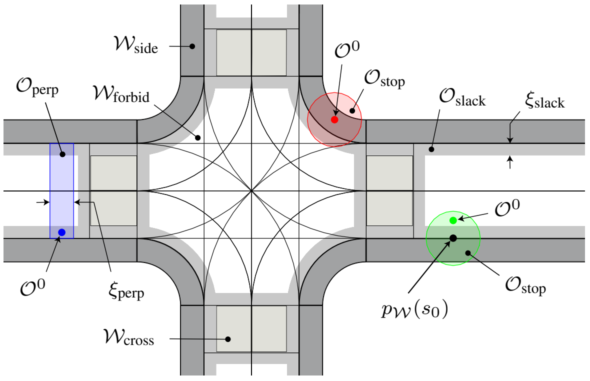

$\mathcal{W}_{\text{side}}$
$\mathcal{O}^0$
$\mathcal{O}_{\text{stop}}$
$\mathcal{O}_{\text{perp}}$
$\mathcal{W}_{\text{forbid}}$
$\mathcal{O}_{\text{slack}}$
ξ_{slack}
$\mathcal{O}^0$
$\mathcal{O}^0$
ξ_{perp}
$\mathcal{O}_{\text{stop}}$
$p_{\mathcal{W}}(s_0)$
$\mathcal{W}_{\text{cross}}$

Fig. 4. Visualization of the occupancies based on formalized traffic rules of Tab. I.

C. Constraint Management

The prediction of each pedestrian is based on traffic rules, which are represented by the constraints introduced in Tab. I. The Boolean variables are initialized with $b = \text{true}$ and then set according to the conditions listed in the last column of Tab. II. Thus, constraints are automatically deactivated as soon as assumptions on the traffic rules are violated. Let us explain the constraint management for C_{stop} and C_{perp} in more detail. For C_{stop}, pedestrians are anticipated to step on the roadway if they cannot stop before; in this case, $b_{\text{stop}} = \text{false}$. The condition of C_{perp} further anticipates that the pedestrian crosses the road if stopping within $\mathcal{O}_{\text{stop}}$ is not possible or if $\mathcal{W}_{\text{forbid}}$ is intruded by more than the maximum of ξ_{slack} and $r_{\text{stop}} + r_{\text{ped}}$; in these cases, $b_{\text{perp}} = \text{false}$.

Note that the proposed conditions for the constraints are deduced from traffic rules, but may be adapted to obtain more or less conservative behavior, e.g., a sophisticated intention prediction may directly set b_{perp} and b_{slack} to false (and increase ξ_{perp} and ξ_{slack}) for a child playing at the side of the road. Thus, our approach offers the possibility of deactivating constraints based on the specifications of users while still remaining formally valid.

For the dynamic-based occupancy, the constraints (6b) and (6c) are not deactivated if we measure higher values; instead, their maximum allowed values are adjusted as presented in the last two rows of Tab. II, where the parameters $\Delta_{a_{\max}}$ and $\Delta_{v_{\max}}$ are thresholds to anticipate that the measured values might be exceeded and thus, the updated maximum values will not be directly violated again.

IV. EVALUATION WITH REAL-WORLD DATA

We evaluate our approach using recorded data with measurement noise obtained from a moving vehicle [30]. Fig. 5 shows the view from the front camera of the autonomous vehicle and three tracked pedestrians crossing the street. The default values for the constraints in our prediction are listed in Tab. II. For the dynamic-based occupancy, the maximum acceleration and velocity must be parametrized. As in [23], we use as the default $a_{\max} = 0.6\,\text{m}/\text{s}^2$ (based on a labeled video source [31]) and $v_{\max} = 2.0\,\text{m}/\text{s}$ (which

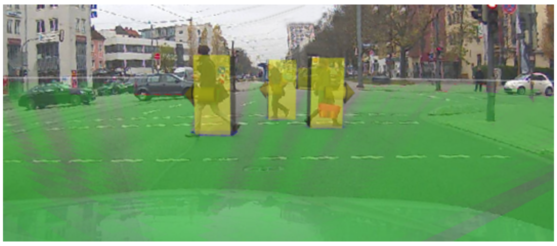
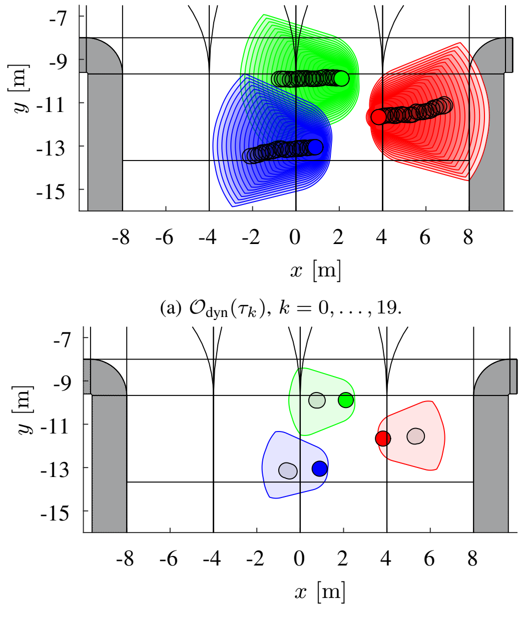


Fig. 5. View from the front camera of the autonomous vehicle approaching three pedestrians crossing the street. The recorded data is provided by [30].



(a) $\mathcal{O}_{\text{dyn}}(\tau_k)$, $k = 0, \ldots, 19$.

(b) $\mathcal{O}_{\text{dyn}}(\tau_9)$.

Fig. 6. The predicted dynamic-based occupancy contains the recorded occupancy of the three pedestrians at the crossing of Fig. 5.

is the transition speed between walking and running and is suggested by [32]). One can also choose different values, e.g., from extensive physiological experiments on walking, running, and stopping [33]. Our results have been obtained using MATLAB 2016a on a machine with a 2.6 GHz Intel Core i7 processor with 20 GB 1600 MHz DDR3 memory.

A. Conformance of Dynamic-Based Occupancy

We validate our dynamic-based occupancy by checking whether our model over-approximates the real behavior of walking-only pedestrians, using $\Delta t = 0.1\,\text{s}$, $t_f - t_0 = 2.0\,\text{s}$, and $r_{\text{ped}} = 0.35\,\text{m}$. From [30], we predict 11 pedestrians for a total of 7008 s with $\Delta_s \in [0.19\,\text{m}, 0.96\,\text{m}]$, $\Delta_v \in [0.21\,\text{m/s}, 1.5\,\text{m/s}]$, and $\Delta_\varphi \in [0.21\,\text{rad}, 0.88\,\text{rad}]$. We achieve a coverage of 100 %, since all recorded occupancies, i.e., ground-truth trajectories without uncertainty enlarged by r_{ped}, were fully contained within the predicted $\mathcal{O}_{\text{dyn}}(\tau_k)$. As an example, Fig. 6a depicts our result of the three pedestrians from Fig. 5. A snapshot in Fig. 6b for $\tau_9 = [0.9\,\text{s}, 1.0\,\text{s}]$ shows that our set-based prediction is not unreasonably conservative. Note that our prediction remains over-approximative for longer time horizons (tested for up to 5.0 s).

The computation time for each pedestrian was 23 ms; however, since the set intersection in (10) requires the most resources, the prediction only required 5 ms for $\Delta t = 0.5\,\text{s}$.

Furthermore, we also validated our model using ground truth trajectories of 389 pedestrians from the publicly available BIWI Walking Pedestrians dataset of a street scene in Zurich, Switzerland [31]. Again, we achieved 100 % coverage using $\Delta_s \in [0.15\,\text{m}, 0.3\,\text{m}]$, $\Delta_v = 0.15\,\text{m/s}$, and $\Delta_\varphi \in [0.2\,\text{rad}, 0.5\,\text{rad}]$.

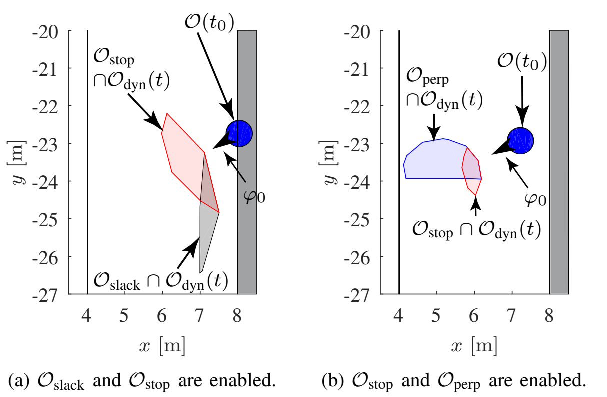
B. Evaluation of Rule-Based Occupancy

Next, we demonstrate the influence of the constraints deduced from the traffic rules. As shown in Fig. 7a, the pedestrian just stepped onto the roadway (with $v_0 = 1.40\,\text{m/s}$ at $t_0 = 0\,\text{s}$). According to Tab. II, $b_{\text{slack}} = \text{false}$ and since $\exists t \in [t_0, t_f] : \big(\mathcal{O}_{\text{dyn}}(t) \cap \mathcal{O}_{\text{rule}}(t)\big) \ominus \mathcal{C}(r_{\text{ped}}) = \emptyset$, $b_{\text{stop}} = \text{false}$. The resulting occupancy $(\mathcal{O}_{\text{slack}} \cup \mathcal{O}_{\text{stop}}) \cap \mathcal{O}_{\text{dyn}}(t)$ restricts the pedestrian from completely crossing the street. For $t_f - t_0 = 3.0\,\text{s}$, the computation time was 66 ms for $\Delta t = 0.1\,\text{s}$ and was reduced by a factor of 3 for $\Delta t = 0.5\,\text{s}$.

When the initial state is updated at $t_0 = 0.5\,\text{s}$, the pedestrian had made another step onto the roadway ($v_0 = 1.58\,\text{m/s}$, cf. Fig. 7b). Since the constraint management again detects that the occupancy is empty, $b_{\text{perp}} = \text{false}$ and we predict the pedestrian crossing the street perpendicular to the driving direction, as shown in Fig. 7b.

C. Application to Evasive Motion Planning

To demonstrate how the obtained prediction can be used for evasive trajectory planning of autonomous vehicles, we make use of a trajectory planner based on convex optimization techniques [3]. The scenarios presented next are available in the CommonRoad benchmark suite including all simulation parameters[3] [34].



(a) $\mathcal{O}_{\text{slack}}$ and $\mathcal{O}_{\text{stop}}$ are enabled.

(b) $\mathcal{O}_{\text{stop}}$ and $\mathcal{O}_{\text{perp}}$ are enabled.

Fig. 7. The rule-based occupancy intersected with $\mathcal{O}_{\text{dyn}}(t)$ for different stages of a pedestrian crossing the roadway.

[3] commonroad.in.tum.de

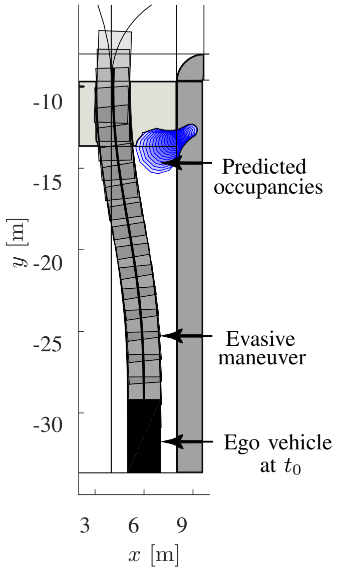


Fig. 8. By making use of our set-based prediction, planned maneuvers of the ego vehicle are guaranteed to be collision-free. As an example, we show an evasive trajectory and the predicted occupancies with b_{stop} = false for $t \in [0\,\text{s}, 1.8\,\text{s}]$.

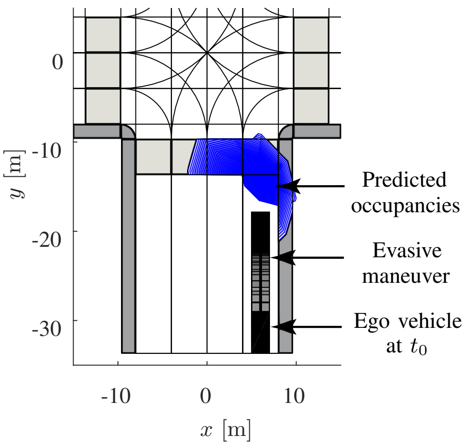


Fig. 9. The pedestrian has just entered the crossing and is predicted with $b_{\text{stop}} = b_{\text{prio}}$ = false and $t_f - t_0 = 5.0\,\text{s}$. If the ego vehicle reduces its speed from $13.8\,^{\text{m}}/_{\text{s}}$ (in Fig. 8) to $12.5\,^{\text{m}}/_{\text{s}}$, it is also able to perform a collision-free braking maneuver.

The ego vehicle in our first scenario (cf. Fig. 8, CommonRoad ID: S=ZAM_Intersect-1_1_S-1:2018a) is approaching the intersection with a velocity of $13.8\,^{\text{m}}/_{\text{s}}$, while the pedestrian approaches a forbidden crossing (without priority due to a red traffic light) with $v_0 = 1.35\,^{\text{m}}/_{\text{s}}$. Similar to the example in Sec. IV-B, the pedestrian cannot stop before entering the roadway and thus is predicted with b_{stop} = false. We plan an evasive maneuver which involves swerving to the left adjacent lane to avoid a collision. The obtained evasive trajectory is guaranteed to be collision-free given our assumptions (cf. Fig. 8).

Furthermore, the prediction can be used to proactively evaluate evasive options so that the number of available evasive maneuvers is increased. As an example, we consider the situation $0.5\,\text{s}$ later when the pedestrian has already entered the forbidden crossing ($v_0 = 1.40\,^{\text{m}}/_{\text{s}}$), implying b_{prio} = false (cf. prediction in Fig. 9; CommonRoad ID: S=ZAM_Intersect-1_2_S-1:2018a). Considering the current velocity of the ego vehicle, a collision with the crossing pedestrian can only be avoided by swerving to the left adjacent lane (similar to the maneuver of the previous scenario, cf. Fig. 8). However, this may not be an option in the presence of other vehicles. Thus, we simultaneously plan evasive trajectories for different velocities of the vehicle. As a result, we observe that the ego vehicle is still able to avoid a collision using emergency braking for a velocity of $12.5\,^{\text{m}}/_{\text{s}}$ (cf. trajectory in Fig. 9).

V. Conclusions

This paper proposes a formal prediction of the possible and legal future motion of pedestrians in an over-approximative, set-based fashion. By considering contextual information and the traffic rules pedestrians should adhere to, the prediction is significantly improved compared to a solely dynamical model. Nevertheless, our approach anticipates that pedestrians disregard rules and automatically adapts the prediction to ignore violated rules. We have validated our method using recorded motions of pedestrians and highlighted its use for evasive maneuver planning.

Future work includes further studies on pedestrian behavior to validate and parameterize the constraints based on the traffic rules. Furthermore, we are currently preparing real-world vehicle experiments for evasive maneuver planning considering pedestrians crossing the road.

Acknowledgments

The authors thank Sascha Steyer for the recordings of real-world data and Carmella Schürmann for the voice-over in the video attachment.

References

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[2] M. S. Shirazi and B. T. Morris, "Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2017.

[3] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *Proc. of the 21th IEEE International Conference on Intelligent Transportation Systems*, 2018.

[4] 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.

[5] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "A probabilistic framework for joint pedestrian head and body orientation estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1872–1882, 2015.

[6] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proc. of the IEEE International Conference on Computer Vision Workshop*, 2017, pp. 206–213.

[7] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[8] C. G. Keller and D. M. Gavrila, “Will the pedestrian cross? A study on pedestrian path prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[9] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmuedderich, “Pedestrian crossing prediction using multiple context-based models,” in *Proc. of the 17th IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 378–385.

[10] A. T. Schulz and R. Stiefelhagen, “Pedestrian intention recognition using latent-dynamic conditional random fields,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2015, pp. 622–627.

[11] R. Quintero, I. Parra, J. Lorenzo, D. Fernández-Llorca, and M. A. Sotelo, “Pedestrian intention recognition by means of a hidden Markov model and body language,” in *Proc. of the 20th IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1–7.

[12] S. Neogi, M. Hoy, W. Chaoqun, and J. Dauwels, “Context based pedestrian intention prediction using factored latent dynamic conditional random fields,” in *Proc. of the IEEE Symposium Series on Computational Intelligence*, 2017, pp. 1–8.

[13] J. F. P. Kooij, N. Schneider, and D. M. Gavrila, “Analysis of pedestrian dynamics from a vehicle perspective,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 1445–1450.

[14] M. Goldhammer, M. Gerhard, S. Zernetsch, K. Doll, and U. Brunsmann, “Early prediction of a pedestrian’s trajectory at intersections,” in *Proc. of the 16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 237–242.

[15] B. Völz, H. Mielenz, R. Siegwart, and J. Nieto, “Predicting pedestrian crossing using quantile regression forests,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 426–432.

[16] E. Rehder and H. Kloeden, “Goal-directed pedestrian prediction,” in *Proc. of the IEEE International Conference on Computer Vision Workshop*, 2015, pp. 139–147.

[17] D. Vasquez, “Novel planning-based algorithms for human motion prediction,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 3317–3322.

[18] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, “Intent-aware long-term prediction of pedestrian motion,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 2543–2549.

[19] P. Vasishta, D. Vaufreydaz, and A. Spalanzani, “Natural vision based method for predicting pedestrian behaviour in urban environments,” in *Proc. of the 20th IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1–6.

[20] J. Wu, J. Ruenz, and M. Althoff, “Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1285–1292.

[21] K. C. Fuerstenberg and J. Scholz, “Reliable pedestrian protection using laserscanners,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 142–146.

[22] M. Meinecke, M. Roehder, T. Nguyen, M. Obojski, M. Heuer, B. Giesler, and B. Michaelis, “Motion model estimation for pedestrians in street-crossing scenarios,” in *Proc. of the International Workshop on Intelligent Transportation*, vol. 7, 2010.

[23] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, “Provably safe motion of mobile robots in human environments,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1351–1357.

[24] P. Bender, J. Ziegler, and C. Stiller, “Lanelets: Efficient map representation for autonomous driving,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[25] G. Lafferriere, G. J. Pappas, and S. Yovine, “A new class of decidable hybrid systems,” in *Hybrid Systems: Computation and Control*, ser. LNCS 1569. Springer, 1999, pp. 137–151.

[26] M. Althoff, “Reachability analysis and its application to the safety assessment of autonomous cars,” Dissertation, Technische Universität München, 2010, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[27] M. Althoff, C. Le Guernic, and B. H. Krogh, “Reachable set computation for uncertain time-varying linear systems,” in *Hybrid Systems: Computation and Control*, 2011, pp. 93–102.

[28] A. Rizaldi and M. Althoff, “Formalising traffic rules for accountability of autonomous vehicles,” in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 1658–1665.

[29] United Nations Economic Commission for Europe, “Convention on road traffic,” United Nations Conference on Road Traffic, 1968, (consolidated version of 2006). [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/conventn/Conv_road_traffic_EN.pdf

[30] S. Steyer, G. Tanzmeister, and D. Wollherr, “Object tracking based on evidential dynamic occupancy grids in urban environments,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 1064–1070.

[31] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, “You’ll never walk alone: modeling social behavior for multi-target tracking,” in *Proc. of the 12th IEEE International Conference on Computer Vision*, 2009, pp. 261–268.

[32] *Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body (ISO 13855:2010)*, ISO Std., 2010.

[33] N. Tiemann, “Ein Beitrag zur Situationsanalyse im vorausschauenden Fußgängerschutz,” Dissertation, Universität Duisburg-Essen, 2012, https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-31259/Tiemann_Diss.pdf.

[34] 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.

3.3 ITSC 2017: Interaction-Aware Occupancy Prediction of Road Vehicles [63]

Summary The set-based prediction introduced so far considers traffic rules to reduce the set of predicted behaviors. However, it neglects mutual influences between detected vehicles. This section presents an extension to consider interaction in set-based prediction so that the over-approximation of the prediction is reduced (cf. Problem statement 1). Instead of explicitly modeling mutual dependencies between vehicles, we remove unreachable occupancy regions, which is computationally much more efficient. Therefore, we sort all vehicles based on their current position and determine all vehicles that have to follow another one. A vehicle has to follow another one if overtaking is not possible or allowed, until the preceding vehicle reaches a road fork, and after a vehicle has merged into the lane of another vehicle. In these cases, the order of two vehicles is given ambiguously. In consequence, we can determine the areas that are unreachable by the following vehicle, since its maximum reachable position can never be greater than the maximum reachable position of the preceding vehicle.

The usefulness of the proposed anytime algorithm is demonstrated in four traffic scenarios from the CommonRoad benchmarks. If considering interaction in scenarios where overtaking is not possible, the drivable area of the ego vehicle is significantly increased. On a multi-lane road, vehicles can easily overtake each other; thus, almost all occupancy regions are reachable and our method would not be very beneficial. Overall, by using the proposed extension to set-based prediction, the quality of prediction result is improved due to a reduced over-approximation.

Note that the CommonRoad IDs given in the publication are referring to version 2017a of CommonRoad. They differ to the IDs in contemporary versions of CommonRoad, since the construction of IDs has been unified in version 2018a. Thus, we also provide the updated IDs. The presented scenarios with IDs S=GER_B471_1a, S=GER_Ffb_1b, S=Z_Merge_1a, and S=GER_Muc_2b of version 2017a, are now available under the IDs S=DEU_B471-1_1_T-1:2018a, S=DEU_Ffb-1_1_T-1:2018a, S=ZAM_Merge-1_1_T-1:2018a, and S=DEU_Muc-4_1_T-1:2018a, respectively.

Contributions of M. K. M. K. developed the sorting of vehicles. M. K. designed and conducted the experiments (together with H. B. and V. B.). M. K. evaluated the experiments. M. K. wrote the article (together with M. A.).

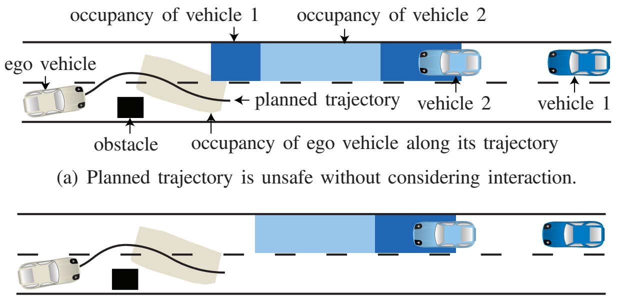
Conference paper ©2017 IEEE. Reprinted, with permission, from Markus Koschi and Matthias Althoff, Interaction-Aware Occupancy Prediction of Road Vehicles, in Proc. of the 20th IEEE International Conference on Intelligent Transportation Systems.

2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)

Interaction-Aware Occupancy Prediction of Road Vehicles

Markus Koschi and Matthias Althoff

***Abstract*— A crucial capability of autonomous road vehicles is the ability to cope with the unknown future behavior of surrounding traffic participants. This requires using nondeterministic models for prediction. While stochastic models are useful for long-term planning, we use set-valued nondeterminism capturing all possible behaviors in order to verify the safety of planned maneuvers. To reduce the set of solutions, our earlier work considers traffic rules; however, it neglects mutual influences between traffic participants. This work presents the first solution for establishing interaction within set-based prediction of traffic participants. Instead of explicitly modeling dependencies between vehicles, we trim reachable occupancy regions to consider interaction, which is computationally much more efficient. The usefulness of our approach is demonstrated by experiments from the CommonRoad benchmark repository.**



(a) Planned trajectory is unsafe without considering interaction.

(b) Planned trajectory can be verified as safe when considering interaction.

Fig. 1. Occupancies of interacting vehicles for a selected time interval.

I. Introduction

It is commonly agreed that purely reactive controllers for collision avoidance only considering the current situation are insufficient for avoiding collisions in road traffic. Integrating a prediction of other traffic participants facilitates much better solutions [1].

Depending on the purpose of the vehicle motion planner or driving assistant system, different types of prediction are appropriate. For driving assistant systems, simple predictions only producing a single behavior are sufficient [2]–[6], since warnings are not necessarily safety-critical. However, for long-term planning of automated vehicles, simple predictions are insufficient, since they do not explicitly consider the growing uncertainty when one increases the prediction horizon. Stochastic approaches account for this shortcoming [7]–[11]. To guarantee safe movement, however, one cannot rely on stochastic approaches, since ensuring safety or a very small crash probability (around 10^{-10} for a $5\,\mathrm{s}$ prediction horizon) is necessary in order to obtain motions which are superior to those of humans. Such small probabilities are difficult to verify, so we propose set-based predictions as developed in our previous work [12], [13]. Set-based prediction, based on models with uncertain yet bounded inputs and parameters, contains all possible movements of traffic participants.

Clearly, set-based prediction considering all possible behaviors can block unnecessarily large sections of a road network for the motion planner. To manage this issue, we predict behaviors that comply with traffic rules only, which can be individually deactivated in case of violation. In addition, one can restrict the prediction horizon by computing fail-safe maneuvers [14]. However, we have not yet exploited mutual influences between traffic participants to improve the quality of the prediction.

One possibility for considering interaction would be to use a concrete model of dependencies. However, such models are typically unknown [15] and result in large combined systems, which are hard to analyze. Instead, we consider interaction between traffic participants on a more abstract level: e.g. when two cars drive in a lane as shown in Fig. 1, the maximum reachable position of the following vehicle 1 can never be greater than the maximum reachable position of the leading vehicle 2. Fig. 1(a) shows the occupancies of both vehicles, i.e. the region they can occupy in the selected time interval (see Def. 9 later), without considering their interaction. When taking this into account, the region occupied by vehicle 1 can be shortened for all consecutive time intervals such that it no longer reaches in front of the occupancy of vehicle 2, as shown in Fig. 1(b).

There is only very little work considering interactions between traffic participants for prediction, as pointed out in [1]. For single behavior prediction, one can assume that other vehicles avoid collisions and thus penalize the trajectories which result in a collision [16], [17]. In terms of stochastic prediction, the work in [18] considers interaction by adjusting the acceleration and lane-change behavior of following vehicles. Since modeling the pairwise dependencies between traffic participants grows with the number of entities, one can reduce the complexity by assuming unidirectional influence [19], [20]. Instead of considering the dependencies pairwise, the authors of [21] model mutual influences as a function of the local situational context. Based on [21], a fully probabilistic model is presented in [22]. The work in [23] presents experience-based data on the interaction between the ego and surrounding vehicles during lane changes. In order to

Markus Koschi and Matthias Althoff are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany. {markus.koschi, matthias.althoff}@tum.de

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

consider interaction in situation assessment, one can compute an interaction-aware joint probability distribution [24] or detect conflicting intentions at intersections by comparing what vehicles intend to do with what they are expected to do [25].

This work is the first which incorporates interaction into set-based prediction of other traffic participants. Our paper is organized as follows: After providing relevant definitions in Sec. II, we introduce set-based prediction in Sec. III. Sec. IV defines our concept for considering interaction and describes our algorithm for removing unreachable occupancy regions. Numerical experiments are presented in Sec. V and discussed in Sec. VI.

II. Preliminaries

A. Road Network

Our road network model is composed by *lanelets* [26], which are atomic, interconnected, and drivable road segments:

Definition 1 (Lanelets [26]): A lanelet is defined by its left and right bound, where each bound is represented by an array of points (a polyline), as shown in Fig. 2. The driving direction of a lanelet is implicitly defined by its left and right bound.

To represent the road network as a directed graph, we introduce relations between two lanelets: successor, left, and right.

Definition 2 (Lanes): We define lanes as the union of lanelets which are longitudinally adjacent, i.e. are successors of each other.

Note that a lanelet which has multiple successors, as in the case of road forks, becomes an element of multiple lanes (e.g. see $\mathrm{lanelet}_2$ in Fig. 2).

Definition 3 (Merging Lanes): Two lanes are merging into one lane, if they are constructed from distinct lanelets which eventually have a common successor lanelet (e.g. see Fig. 5). The geometric condition for merging lanes is that the start points of the two lanes must be different and their end points must be equal (see Fig. 2 for the definition of these points).

Definition 4 (Current Lanes of a Vehicle): The current lanes of a vehicle are defined as all lanes in which the vehicle is currently positioned (e.g. in Fig. 2, lane_2 and lane_3 are the current lanes of the vehicle).

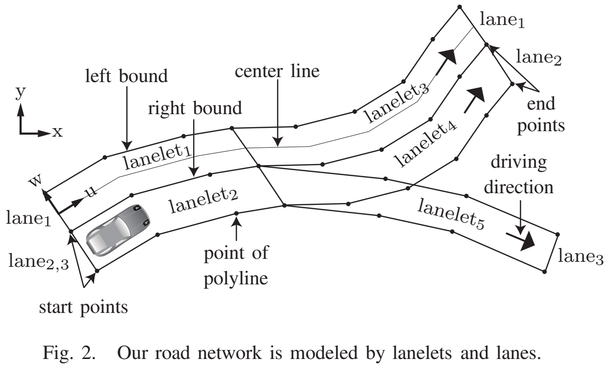


Fig. 2. Our road network is modeled by lanelets and lanes.

In addition to the Cartesian space in world coordinates xy, we require a lane coordinate system uw:

Definition 5 (Curvilinear Lane Coordinate System): A curvilinear lane coordinate system uw is defined for each lane such that the u-axis is parallel to the center line of the lane and the w-axis is perpendicular to u, as shown in Fig. 2. The origin of the lane coordinate system is in the start point of each center line, and the positive u-axis points in the driving direction. The u-coordinate of a point p in lane l_i is denoted by $\mathrm{u}_p^{l_i}$.

Definition 6 (Front-Most and Rear-Most Point): For a set of points $\mathcal{P}$, the point with the maximum u-coordinate in lane l_i is defined as

$$\max(\mathrm{u}_{\mathcal{P}}^{l_i}) := \max(\mathrm{u}_p^{l_i} | p \in \mathcal{P}).$$

The rear-most point of $\mathcal{P}$ in l_i, $\min(\mathrm{u}_{\mathcal{P}}^{l_i})$, is defined analogously.

For the sake of clarity, we omit the lane's notation by using only u_p or $\mathrm{u}_{\mathcal{P}}$ if the point or the set of points is defined in only one lane.

B. Occupancy of a Vehicle

The dynamics of a vehicle can be described by the differential equation

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

where $x \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}^m$ is the input. The possible initial states and the possible inputs are bounded by sets: $x(0) \in \mathcal{X}_0, \forall t : u(t) \in \mathcal{U}$.

Definition 7 (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 $\mathcal{U}$:

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

Furthermore, we introduce a relation from a state vector x to the Cartesian coordinate system xy:

Definition 8 (Relation to Cartesian Space): The operator $\mathrm{state2occ}(x)$ relates the state of a vehicle to the set of points in Cartesian space occupied by the vehicle (including its dimensions) as

$$\mathrm{state2occ}(x) : \mathcal{X} \rightarrow \mathcal{P}(\mathbb{R}^2),$$

where $\mathcal{P}(\mathbb{R}^2)$ is the power set of $\mathbb{R}^2$. Given a set of states $\mathcal{X}$, the relation is defined as $\mathrm{state2occ}(\mathcal{X}) := \{\mathrm{state2occ}(x) | x \in \mathcal{X}\}$.

Definition 9 (Over-approximative Occupancy Set): Based on Def. 7 and Def. 8, the occupancy set $\mathcal{O}(t)$ over-approximates the set of occupied points in Cartesian space which are reachable by the vehicle:

$$\forall t : \mathcal{O}(t) \supseteq \mathrm{state2occ}\big(\mathcal{R}(t)\big).$$

We can use over-approximative occupancy sets to describe the unknown future behavior of vehicles.

2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)

III. Set-Based Prediction

The set of future occupancies according to Def. 9 can be obtained with set-based prediction [12]. Using reachability analysis, we predict occupancies for consecutive time intervals as shown in Fig. 3, where we use polygons as set representation. Given the predicted occupancies of other vehicles and the occupancy of the ego vehicle along its planned trajectory, the planned trajectory can be verified as safe [27]: If none of the computed occupancies intersects with the occupancy of the ego vehicle for all points in time, one can guarantee that the ego vehicle does not cause a collision.

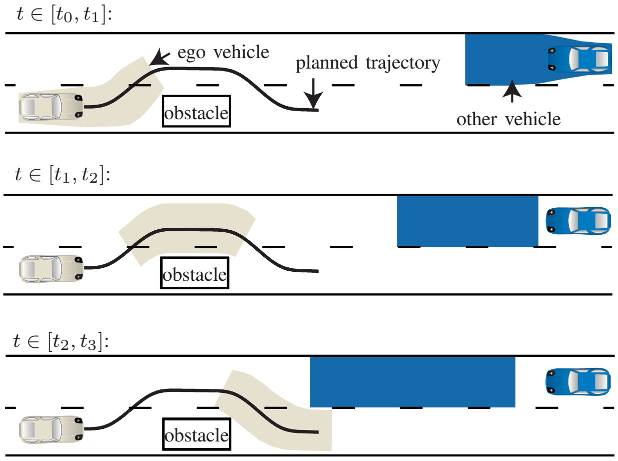


Fig. 3. Snapshots of the predicted occupancy of the other vehicle for selected consecutive time intervals.

Set-based prediction is designed to verify motion plans of short time horizons. Due to the full consideration of uncertainties, the future occupancy of other vehicles grows over time and thus limits the solution space for the ego vehicle. For this reason, we suggest performing trajectory planning for two time horizons in parallel [12]: While non-formal prediction techniques help to find long-term motion plans, set-based occupancy prediction can be used to guarantee the safety of short-term motion plans.

We compute over-approximative occupancies including all possible behaviors under given constraints, which are listed in Tab. I. All assumptions are taken from [13] and are either physical constraints ($C_{a_{\max}}$ and C_{engine}) or a formalization of the Vienna Convention on Road Traffic [28], [29]. Please note that we deactivate constraints individually during online execution if traffic rules are violated. For more details on our constraint management, please see [13].

TABLE I

Vehicle constraints.

Constraint	Description
$C_{a_{\max}}$	Maximum absolute acceleration is limited by $a_{\max}$.
$C_{v_{\max}}$	Positive longitudinal acceleration is stopped when a parameterized speed $v_{\max}$ is reached.
C_{engine}	Above a parameterized speed v_S, acceleration in the driving direction is $a_{\text{long}} = a_{\max}\frac{v_S}{v}$, which models limited engine power.
C_{back}	Driving backwards in a lane is not allowed.
C_{lane}	Leaving the lane is forbidden. Changing lanes is only allowed if the new lane has the same driving direction as the previous one.

IV. Interactions Between Vehicles

Since vehicles share the same road, their presence and actions constantly influence other vehicles. As an example, Fig. 1 shows a traffic scenario in which considering interaction is important. While the ego vehicle plans an overtaking maneuver similar to the situation in Fig. 3, it has to consider two oncoming vehicles with different initial velocities. The following vehicle 1 moves faster than the preceding vehicle 2. Hence, the independently predicted occupancy of vehicle 1 is larger than the occupancy of vehicle 2, as shown in Fig. 1(a). For a certain time interval, the ego vehicle might crash into vehicle 1 when following its planned trajectory. However, vehicle 1 cannot reach the part of its occupancy where it ranges in front of the occupancy of vehicle 2, since it cannot surpass vehicle 2. When considering the interactions, we can remove the unreachable region and thus the plan of the ego vehicle can be verified as safe (see Fig. 1(b)).

Set-based occupancy prediction has neglected dependencies between traffic participants so far. In this section, we describe our extension to consider the interactions between vehicles. Our rule-based approach focuses on two-lane roads with only one lane per driving direction. We do not include roads with multiple lanes per driving direction, since vehicles can easily overtake others in the left and right lanes, as demonstrated later in Sec. V. Thus, only small regions are not reachable. For set-based prediction, which must include all reachable occupancies (see Def. 9), considering interactions in multi-lane scenarios is not beneficial. Instead, we incorporate dependencies between vehicles which are in the same lane of two-lane roads, i.e. the considered vehicles are either in the same current lane or in merging lanes, as described later in Sec. IV-B. Since we only handle vehicles in the same lane, it is sufficient to compare vehicles pairwise to consider their interaction.

A. Overall Algorithm

Alg. 1 gives an overview of the computation steps to consider interaction. From the set-based prediction, we require the independently computed occupancies of all vehicles for all time intervals τ_k from the initial time t_0 until the prediction horizon t_f, where $\tau_k = [t_k, t_{k+1}]$ with a time step size of $\Delta t = t_{k+1} - t_k$. First, we sort all vehicles which are in the same lane based on their initial position (see line 1 of Alg. 1 and Sec. IV-B). The returned list [v] contains the pairwise sorted vehicles as a tuple $(\mathrm{v}_i, \mathrm{v}_{i+1})^{[t_{s_i}, t_{e_i}]}$, which represents the fact that v_i is behind v_{i+1} in the time interval $[t_{s_i}, t_{e_i}]$. Please note that due to road forks and merging lanes, the order of the vehicles in each tuple is only valid from the specified start time t_{s_i} to the end time t_{e_i}, as explained in more detail later. An example of the list [v] is

$$[\mathrm{v}] = [(\mathrm{v}_1, \mathrm{v}_2)^{[t_{s_1}, t_{e_1}]}, (\mathrm{v}_3, \mathrm{v}_4)^{[t_{s_3}, t_{e_3}]}, \ldots, \\ (\mathrm{v}_n, \mathrm{v}_{n+1})^{[t_{s_n}, t_{e_n}]}].$$

Second, we consider the interaction of all vehicle pairs $(\mathrm{v}_i, \mathrm{v}_{i+1})^{[t_{s_i}, t_{e_i}]}$ from front to back in each lane (see line 2 to 4 of Alg. 1). The occupancy of all following vehicles v_i is trimmed $\forall \tau_k \subseteq [t_{s_i}, t_{e_i}]$ such that unreachable areas are removed, as described in Sec. IV-C. In the following, we denote an element of [v] without loss of generality by $(\mathrm{v}_1, \mathrm{v}_2)^{[t_s, t_e]}$ and omit further indices for the sake of clarity.

Algorithm 1 Consider Interaction in Two-Lane Roads

```
Require: vehicles (incl. their occupancies $\mathcal{O}(\tau_k)$), lanes
1: [v] ← SORTVEHICLESINLANES(vehicles, lanes)
2: for all $(\mathrm{v}_i, \mathrm{v}_{i+1})^{[t_{s_i}, t_{e_i}]} \in$ [v] do
3:     for all $\tau_k \subseteq [t_{s_i}, t_{e_i}]$ do
4:         $\mathcal{O}_{\mathrm{v}_i}(\tau_k) \leftarrow$ TRIMREACHABLE($\mathcal{O}_{\mathrm{v}_i}(\tau_k)$, $\mathcal{O}_{\mathrm{v}_{i+1}}(\tau_k)$)
5:     end for
6: end for
```

B. Sort Vehicles in the Same Lane

Through a pairwise comparison of all vehicles, the function SORTVEHICLESINLANES() of Alg. 1 sorts vehicles in their lanes in ascending u-coordinates and returns them in the list [v]. If we cannot guarantee that one vehicle will precede another due to the growing uncertainty in the prediction, this vehicle pair is omitted in [v]. When sorting, we distinguish three different cases:

1) Sort Vehicles in the Same Current Lane: Two vehicles are in the same current lane if they are in only one current lane (see Def. 4) and this lane is the same for both vehicles. Then, their order is unambiguously given by their u-coordinates in the lane coordinate system (see Def. 5). As an example, Fig. 1 shows the sorted vehicles 1 and 2, which can be added to [v] as $(\mathrm{v}_1, \mathrm{v}_2)^{[t_0, t_f]}$ since $\mathrm{u}_{\mathrm{v}_1} < \mathrm{u}_{\mathrm{v}_2}$.

2) Sort Vehicles in Forking Lanes: In the case of road forks, where more than one current lane is identical, we can also sort two vehicles until their reachable occupancies split onto two different lanes after the road fork. Since the assumption of one lane per driving direction is invalid after road forks, we can no longer guarantee that one vehicle will precede another. To determine the time until we can sort vehicles in forking lanes, we introduce the point p_{fork} as the intersection of the corresponding lane bounds of the bifurcating lanes, as shown in Fig. 4.

Definition 10 (Not Passed the Lane Fork): We formulate the predicate NOT_PASSED_FORK$(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t)$ using first-order logic:

$$\text{NOT\_PASSED\_FORK}(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t) \Leftrightarrow \\ \max\big(\mathrm{u}^{l_1}_{\mathcal{O}_{\mathrm{v}_2}}(t)\big) \leq \mathrm{u}^{l_1}_{p_{\mathrm{fork}}} \wedge \max\big(\mathrm{u}^{l_2}_{\mathcal{O}_{\mathrm{v}_2}}(t)\big) \leq \mathrm{u}^{l_2}_{p_{\mathrm{fork}}}.$$

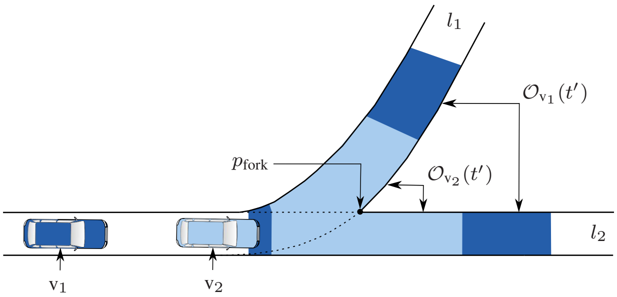

l_1
$\mathcal{O}_{\mathrm{v}_1}(t')$
p_{fork}
$\mathcal{O}_{\mathrm{v}_2}(t')$
l_2
v_1
v_2

Fig. 4. The vehicles v_1 and v_2 cannot be unambiguously sorted for $t \geq t'$, since $\mathcal{O}_{\mathrm{v}_2}(t')$ splits after the road fork.

It evaluates to true at time t if the front-most point of $\mathcal{O}_{\mathrm{v}_2}(t)$ (see Def. 6) has not passed p_{fork} in both bifurcating lanes l_1 and l_2.

Definition 11 (Time until the Lane Fork): We define t_{fork} as the latest time at which the predicate NOT_PASSED_FORK$(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t)$ still evaluates to true:

$$t_{\mathrm{fork}} := \max_{T \in \mathbb{R}} \big(\forall t : t \leq T : \text{NOT\_PASSED\_FORK}(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t)\big)$$

Proposition 1 (Precedence in Forking Lanes): $\forall t : \tilde{t} \leq t \leq t_{\mathrm{fork}}$, vehicle v_1 cannot precede vehicle v_2 no matter what vehicle v_2 is doing, where $\tilde{t}$ is some time within $[t_0, t_{\mathrm{fork}}[$.

Proof: If NOT_PASSED_FORK$(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t)$ evaluates to true (see Def. 10), vehicle v_1 cannot precede vehicle v_2 at time t, since they are still in the same current lane. In combination with Def. 11, Prop. 1 follows. The uncertainty of $\tilde{t}$ originates from the unspecified road network traversed before reaching the road fork. ■

Thus, we can sort vehicles in forking lanes analogously to vehicles in the same current lane (see previous paragraph), but only in the time interval $t_s = \tilde{t}$ to $t_e = t_{\mathrm{fork}}$.

3) Sort Vehicles in Merging Lanes: If two vehicles are not yet in the same current lane, but their current lanes are merging (see Fig. 5), we also consider their interaction. All merging lanes can be detected by evaluating all pairs of lanes according to Def. 3. In order to argue about the order of vehicles in merging lanes, we must first define some distances, which are shown in Fig. 5. The distance along the u-axis of lane l_2 between the rear-most point of $\mathcal{O}_{\mathrm{v}_2}(t)$ and the intersection point of the merging lanes p_{merge} is defined as

$$d_{\mathrm{merge}}(t) = \mathrm{u}^{l_2}_{p_{\mathrm{merge}}} - \min\big(\mathrm{u}^{l_2}_{\mathcal{O}_{\mathrm{v}_2}}(t)\big). \tag{2}$$

After describing the intersection of the occupancy of v_1 with the lane l_2 in which the other vehicle v_2 is positioned as

$$\mathcal{O}^{l_2}_{\mathrm{v}_1}(t) := \mathcal{O}_{\mathrm{v}_1}(t) \cap l_2, \tag{3}$$

the distance between the front-most point of $\mathcal{O}^{l_2}_{\mathrm{v}_1}(t)$ and the rear-most point of $\mathcal{O}_{\mathrm{v}_2}(t)$ in l_2 can be defined as

$$d_{\mathrm{bounds}}(t) = \max\big(\mathrm{u}^{l_2}_{\mathcal{O}^{l_2}_{\mathrm{v}_1}}(t)\big) - \min\big(\mathrm{u}^{l_2}_{\mathcal{O}_{\mathrm{v}_2}}(t)\big). \tag{4}$$

Please note that both $d_{\mathrm{merge}}(t)$ and $d_{\mathrm{bounds}}(t)$ can be negative.

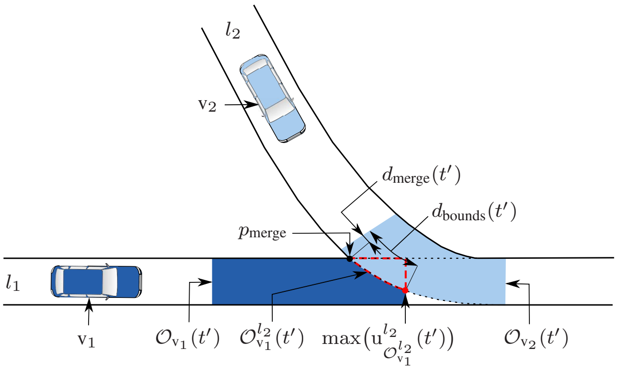

l_2
v_2
$d_{\text{merge}}(t')$
$d_{\text{bounds}}(t')$
p_{merge}
l_1
v_1
$\mathcal{O}_{\mathrm{v}_1}(t')$
$\mathcal{O}^{l_2}_{\mathrm{v}_1}(t')$
$\max(\mathrm{u}^{l_2}_{\mathcal{O}^{l_2}_{\mathrm{v}_1}}(t'))$
$\mathcal{O}_{\mathrm{v}_2}(t')$

Fig. 5. The vehicles v_1 and v_2 in the merging lanes can be sorted at t', since v_1 cannot precede v_2 no matter what v_2 is doing.

Definition 12 (Passed the Lane Merge): We define the predicate PASSED_MERGE$(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, t)$ as

$$\text{PASSED\_MERGE}(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, t) \Leftrightarrow$$
$$d_{\text{merge}}(t) < \text{length}_{\mathrm{v}_2} \wedge$$
$$\left(\mathcal{O}^{l_2}_{\mathrm{v}_1}(t) = \emptyset \vee d_{\text{bounds}}(t) < \text{length}_{\mathrm{v}_2}\right),$$

where $\text{length}_{\mathrm{v}_2}$ denotes the length of the vehicle's enclosing rectangle.

Definition 13 (Time after the Lane Merge): We introduce t_{merge} as the earliest time at which PASSED_MERGE$(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, t)$ evaluates to true:

$$t_{\text{merge}} := \min_{T \in \mathbb{R}} \Big(\forall t : t \geq T :$$
$$\text{PASSED\_MERGE}(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, t)\Big).$$

Proposition 2 (Precedence in Merging Lanes): $\forall t : t_{\text{merge}} \leq t \leq \hat{t}$, vehicle v_1 cannot precede vehicle v_2 no matter what v_2 is doing; $\hat{t} \in]t_{\text{merge}}, t_f]$.

Proof: Vehicle v_1 can precede v_2 at time t, either if v_2 has not passed p_{merge} but v_1 has, or if $d_{\text{bounds}}(t) \geq \text{length}_{\mathrm{v}_2}$. Thus at t', if $d_{\text{merge}}(t') < \text{length}_{\mathrm{v}_2}$, v_2 certainly passed the intersection point p_{merge}. If, in addition, either $\mathcal{O}^{l_2}_{\mathrm{v}_1}(t') = \emptyset$ (i.e. v_1 has not passed p_{merge}) or $d_{\text{bounds}}(t') < \text{length}_{\mathrm{v}_2}$, the predicate PASSED_MERGE$(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, t')$ evaluates to true (see Def. 12), and it is not possible to shift v_1 in $\mathcal{O}_{\mathrm{v}_1}(t')$ so that it is in front of v_2 in $\mathcal{O}_{\mathrm{v}_2}(t')$. Since v_2 certainly precedes v_1 at time t', v_2 is the preceding vehicle $\forall t : t' \leq t \leq \hat{t}$. Using Def. 13, Prop. 2 follows. The uncertainty of $\hat{t}$ originates from the unspecified road network after the lane merge. ■

Consequently, at t_{merge}, we can formally guarantee for the first time that v_1 cannot precede v_2 in the merging lanes, no matter what vehicle v_2 is doing (see Fig. 5). Note that before t_{merge}, we cannot eliminate the possibility that v_1 can precede v_2 in the future.

We sort vehicles in merging lanes by determining t_{merge} by evaluating PASSED_MERGE$(\mathcal{O}_{\mathrm{v}_1}, \mathcal{O}_{\mathrm{v}_2}, p_{\text{merge}}, \tau_k)$ for all vehicles pairwise and for all $\tau_k \subseteq [t_0, t_f]$. If the time t_{merge} exists, the vehicles v_1 and v_2 can be included in the list [v] as $(\mathrm{v}_1, \mathrm{v}_2)^{[t_{\text{merge}}, \hat{t}]}$. Otherwise, we omit the currently compared vehicles in [v], since it is not possible to determine that v_1 cannot precede v_2 for any $\tau_k \subseteq [t_0, t_f]$.

Please note that we do not include $\text{length}_{\mathrm{v}_2}$ in Def. 10 to determine whether vehicle v_1 can precede v_2 in forking lanes (unlike Def. 12 for merging lanes), since the earliest point where v_1 can possibly pass v_2 on its left or right side depends much on the lane geometry. To obtain a simple and provable over-approximative solution, we choose p_{fork}.

C. Remove Unreachable Occupancies

After sorting the vehicles in the same lane (for the vehicles where an order can be determined), we remove the occupancy regions of each following vehicle v_1 which are not reachable due to the preceding vehicle v_2. The function TRIMREACHABLE() of Alg. 1 trims the reachable occupancy as shown in Fig. 6: The occupancy $\mathcal{O}_{\mathrm{v}_1}(t)$ is shortened such that it is not ahead of the trim line, i.e. after trimming it holds that $\max(\mathrm{u}_{\mathcal{O}_{\mathrm{v}_1}}(t)) = \max(\mathrm{u}_{\mathcal{O}_{\mathrm{v}_2}}(t)) - \text{length}_{\mathrm{v}_2}$.

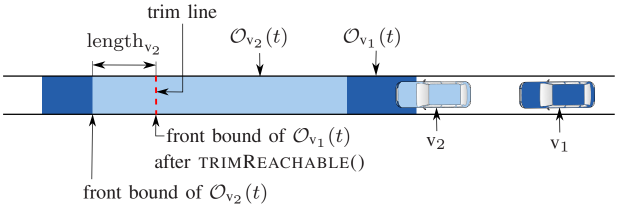

trim line
$\text{length}_{\mathrm{v}_2}$
$\mathcal{O}_{\mathrm{v}_2}(t)$
$\mathcal{O}_{\mathrm{v}_1}(t)$
front bound of $\mathcal{O}_{\mathrm{v}_1}(t)$
after TRIMREACHABLE()
v_2
v_1
front bound of $\mathcal{O}_{\mathrm{v}_2}(t)$

Fig. 6. Removing unreachable occupancies of the following vehicle v_1.

V. Numerical Examples

We demonstrate our interaction-aware occupancy prediction in hand-crafted scenarios from the CommonRoad benchmarks[1] [30]. Each benchmark has a unique ID, which are mentioned later. For the sake of clarity, we have so far extracted only two other vehicles, v_1 and v_2, besides the ego vehicle (and a static obstacle in Scenario I). All results are obtained first by independently predicting the occupancies of v_1 and v_2 using our tool SPOT[2] [13] and then by considering their dependencies as described in Alg. 1. In order to evaluate the benefit for the ego vehicle, we compute its drivable area as presented in [31] for the occupancies without considering interaction (case A) and with considering interaction (case B). The drivable area is the area which a vehicle can reach without causing a collision.

Tab. II lists the parameters of the numerical examples, in which we use different initial velocities for the following and preceding vehicle (i.e. $v_{\mathrm{v}_{1,0}} \neq v_{\mathrm{v}_{2,0}}$). To evaluate different road conditions, we vary the values for the maximum acceleration $a_{\max}$, which are obtained by choosing a friction coefficient of $\mu = 1.0$ and $\mu = 0.82$ for a dry, good road, and $\mu = 0.25$ for a road covered with snow (and a gravity constant of $g = 9.81\,{}^{\mathrm{m}}\!/\!{}_{\mathrm{s}^2}$) [32]. We use a time step size of $\Delta t = 0.1\,\mathrm{s}$ and a prediction horizon of $t_f = 2.3\,\mathrm{s}$ for Scenario III and $t_f = 5.0\,\mathrm{s}$ for the other scenarios.

[1] commonroad.in.tum.de

[2] spot.in.tum.de

In all following figures, the following vehicle v_1 and the preceding vehicle v_2 are depicted in blue and green, respectively. Their predicted occupancies are plotted in their vehicle color and such that the shorter occupancy region is on top of the other one. For Scenarios I and II, the occupancies $\mathcal{O}_{\mathrm{v}}(t)$ are shown for the entire prediction interval, i.e. $t \in [t_0, t_f]$, while we set $t \in [(t_f - \Delta t), t_f]$ for Scenarios III and IV. We mark the initial state of the ego vehicle at $t = t_0$ with a red circle and its drivable area at $t = t_f$ with a red region.

TABLE II
PARAMETERS FOR THE SCENARIOS (S.) I TO IV

Parameter	S. Ia	S. Ib	S. II	S. III	S. IV
$v_{\mathrm{v}_1,0}$	$28\,\mathrm{m/s}$	$28\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$
$v_{\mathrm{v}_2,0}$	$8.3\,\mathrm{m/s}$	$8.3\,\mathrm{m/s}$	$0\,\mathrm{m/s}$	$10.0\,\mathrm{m/s}$	$6.0\,\mathrm{m/s}$
$v_{\mathrm{v}_{\mathrm{ego}},0}$	$18.0\,\mathrm{m/s}$	$18.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	—	$14.0\,\mathrm{m/s}$
$v_{\mathrm{v}_1,\mathrm{max}}$	$28.0\,\mathrm{m/s}$	$28.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	$28.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$
$v_{\mathrm{v}_2,\mathrm{max}}$	$17.0\,\mathrm{m/s}$	$17.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	$28.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$
$v_{\mathrm{v}_{\mathrm{ego}},\mathrm{max}}$	$28.0\,\mathrm{m/s}$	$28.0\,\mathrm{m/s}$	$14.0\,\mathrm{m/s}$	—	$14.0\,\mathrm{m/s}$
a_{max}	$8.0\,\mathrm{m/s^2}$	$2.5\,\mathrm{m/s^2}$	$2.5\,\mathrm{m/s^2}$	$10.0\,\mathrm{m/s^2}$	$2.5\,\mathrm{m/s^2}$

A. Two-Lane Road (Scenario I)

Scenario I (CommonRoad ID: S=GER_B471_1a) features a rural road with one lane per driving direction and a static obstacle (displayed as a gray box) in the lane of the ego vehicle, as shown in Fig. 7. Thus, the ego vehicle requires an overtaking maneuver but also has to avoid a collision with the two oncoming vehicles. Fig. 8 shows the predicted occupancies of v_1 and v_2 in Scenario Ia. The occupancies of the following vehicle v_1 (shown in blue) reach in front of the slower preceding truck v_2 (shown in green) (see Fig. 8(a)). As mentioned before, we plot the occupancy sets such that the shorter occupancy is on top of the other one. The result of removing the unreachable occupancy regions after sorting the two vehicles in the same current lane is shown in Fig. 8(b). It can be seen that the difference is not much more than the length of the preceding truck. In Scenario Ib, we use $a_{\mathrm{max}} = 2.5\,\mathrm{m/s^2}$ and plot the occupancy sets in Fig. 9. When comparing case A in Fig. 9(a) and case B in Fig. 9(b), it can be observed that the effect of the interaction is significant and greater than in Scenario Ia.

In Fig. 10, the benefit for the ego vehicle from the interaction-aware prediction is evaluated using the drivable area of the ego vehicle under the given velocity and acceleration limits (see Tab. II). Since no drivable area exists in front of the static obstacle in case A, overtaking is only safely possible when removing unreachable occupancies (case B).

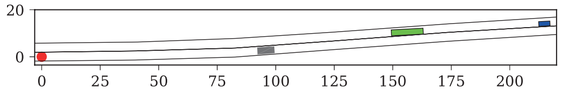
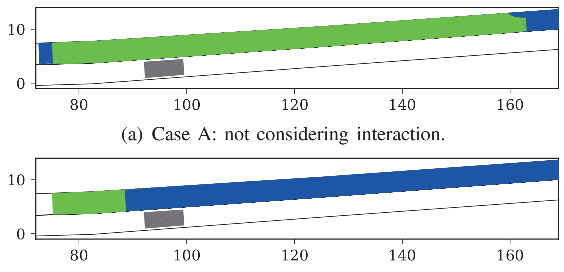


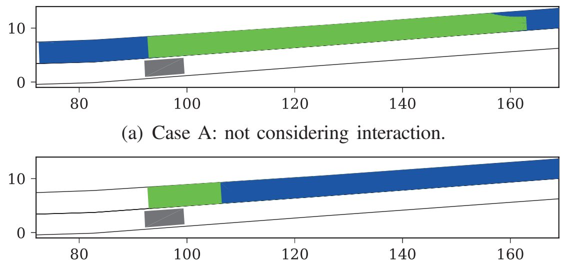
Fig. 7. Initial configuration of Scenario I.



(a) Case A: not considering interaction.

(b) Case B: considering interaction.

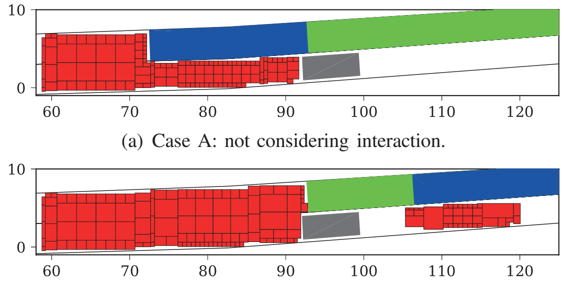
Fig. 8. Occupancies in Scenario Ia ($a_{\mathrm{max}} = 8.0\,\mathrm{m/s^2}$).



(a) Case A: not considering interaction.

(b) Case B: considering interaction.

Fig. 9. Occupancies in Scenario Ib ($a_{\mathrm{max}} = 2.5\,\mathrm{m/s^2}$).



(a) Case A: not considering interaction.

(b) Case B: considering interaction.

Fig. 10. Drivable area of the ego vehicle in Scenario Ib.

B. Intersection of Multiple Two-Lane Roads (Scenario II)

Scenario II presents an urban intersection, where four two-lane roads cross (CommonRoad ID: S=GER_Ffb_1b). As depicted in Fig. 11, the two vehicles v_1 and v_2 are driving south, while the ego vehicle is approaching the intersection from east. The independently predicted occupancy sets are plotted in Fig. 12(a) together with the drivable area of the ego vehicle. Fig. 12(b) shows the trimmed occupancies after removing the unreachable area of v_1. It can be seen that when considering interaction in the occupancy prediction, the ego vehicle can safely cross the intersection. At time t_f, the difference of the drivable area between cases A and B is larger than $100\,\mathrm{m}^2$. Please note that in this intersection scenario, which consists of multiple road forks, we can sort the vehicles to consider their interaction, since the occupancy of the preceding vehicle v_2 does not split onto several lanes yet, i.e. $\mathrm{NOT\_PASSED\_FORK}(\mathcal{O}_{\mathrm{v}_2}, p_{\mathrm{fork}}, t_f)$ evaluates to true.

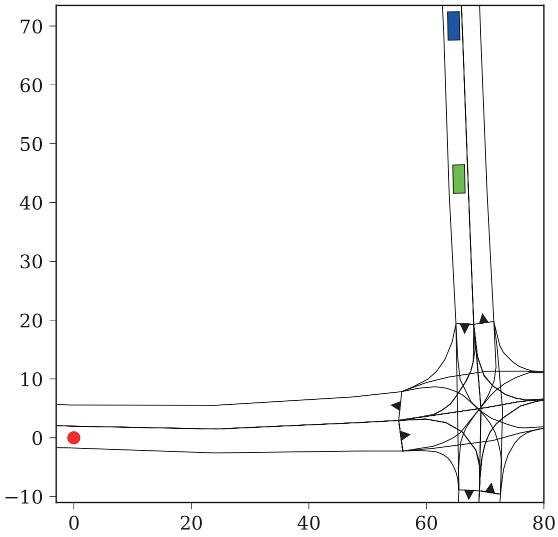
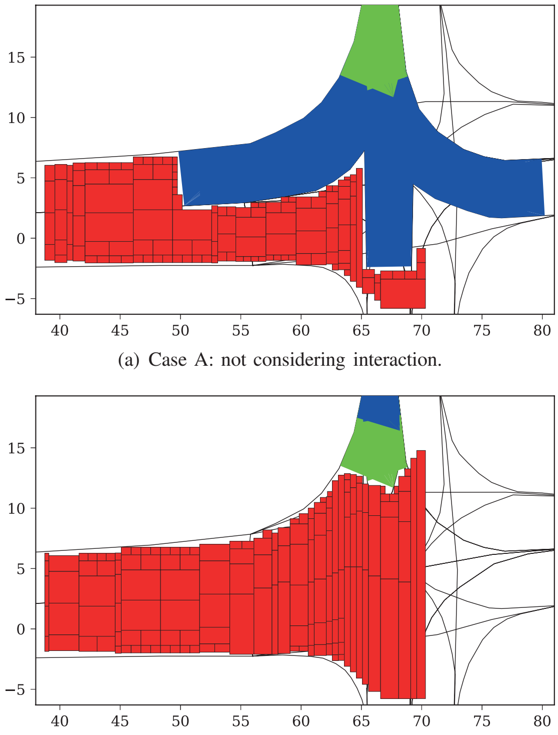


Fig. 11. Initial configuration of Scenario II.



(a) Case A: not considering interaction.

(b) Case B: considering interaction.

Fig. 12. Drivable area of the ego vehicle in Scenario II.

C. Road with Merging Lanes (Scenario III)

In Scenario III, we demonstrate the sorting of vehicles in merging lanes (CommonRoad ID: S=Z_Merge_1a). As shown in Fig. 13, one vehicle is driving in each of the merging lanes. Their occupancy sets are plotted for $t \in [(t_f - \Delta t), t_f]$, where $t_f = 2.3\,\text{s}$. Since the green vehicle v_2 will definitely precede the blue vehicle v_1 (as $t \geq t_{\text{merge}}$), we can remove the unreachable occupancy region of v_1 (see Fig. 13(b)).

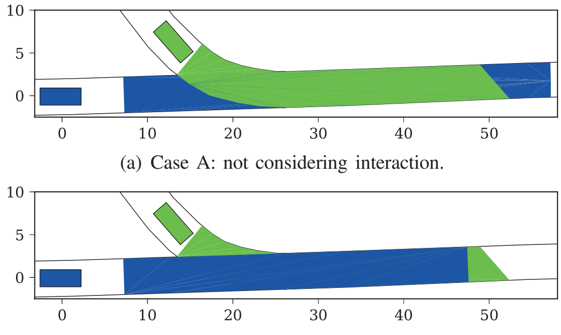
D. Multi-Lane Road (Scenario IV)

As mentioned in Sec. IV, considering interaction is not beneficial in multi-lane roads, which we illustrate with the following example (CommonRoad ID: S=GER_Muc_2b). Fig. 14 shows two lanes with the same driving direction, where two vehicles v_1 and v_2 are driving in the right lane and the ego vehicle in the left lane. It can be seen that considering the vehicles pairwise for determining the interactions is not sufficient, since the following vehicle v_1 might be blocked by two vehicles at once: the preceding vehicle v_2 and the ego vehicle. One might be able to remove small unreachable regions of the following vehicle's occupancies, yet that region is also occupied by the preceding vehicle. Thus, we gain no benefit for the drivable area of the ego vehicle. Moreover, in contrast to two-lane roads, vehicles can easily overtake each other on multi-lane roads and hence almost all occupancy regions are reachable. For this reason, we have not extended our method to multi-lane roads.

VI. Discussion

The numerical examples show that considering interaction between vehicles in the same lane increases the solution space of the ego vehicle. However, removing unreachable occupancies only shows substantial benefit in some cases, e.g. only for bad weather conditions (i.e. low values of $a_{\max}$) or certain configurations of initial states. Thus, we suggest applying our approach selectively. The occupancy sets of all surrounding vehicles should always be predicted independently first. In the remaining computation time during online execution, one can refine the over-approximative prediction by trimming reachable occupancy regions. Due to the anytime property of our algorithm, one can terminate it when computation time is required elsewhere.

We remove unreachable occupancy regions under the assumption that a vehicle does not change to a lane with



(a) Case A: not considering interaction.

(b) Case B: considering interaction.

Fig. 13. Initial configuration and occupancies for $t \in [(t_f - \Delta t), t_f]$ in Scenario III.

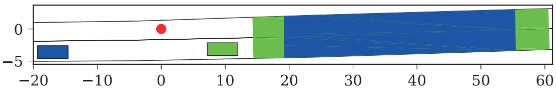


Fig. 14. Initial configuration and occupancies for $t \in [(t_f - \Delta t), t_f]$ in Scenario IV.

the opposite driving direction (constraint C_{lane} of Tab. I). However, as described in the constraint management in [13], we immediately remove C_{lane} if it becomes violated. Thus, the set-based prediction considers the occupancy of the vehicle in its new lane, as well as in front of a preceding vehicle, which might have been removed by our interaction-aware method before the lane change.

VII. Conclusion and Future Work

For the first time, we consider interaction between vehicles in set-based prediction. Our formal approach removes unreachable occupancy regions of vehicles in the same lane by sorting all vehicles and determining unreachable areas. The benefits of our anytime algorithm are demonstrated in numerical experiments based on scenarios from the CommonRoad benchmark repository. Since the drivable area of the ego vehicle is larger in all scenarios when interaction is considered, this work improves the quality of the over-approximative occupancy prediction and increases the safe solution space for the ego vehicle.

Future work contains further experiments on different scenarios. In addition, we wish to include interaction between traffic participants at intersections when considering applicable traffic rules.

Acknowledgment

The authors thank Hannes Bibel for his valuable contribution to developing our approach and Vanessa Bui for creating numerical examples. This work was founded by the BMW Group within the CAR@TUM project.

References

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[2] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 1068–1073.

[3] A. Eidehall, "Multi-target threat assessment for automotive applications," in *Proc. of the 14th International IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 433–438.

[4] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[5] J.-H. Kim and D.-S. Kum, "Threat prediction algorithm based on local path candidates and surrounding vehicle trajectory predictions for automated driving vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2015, pp. 1220–1225.

[6] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 458–464.

[7] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 137–147, 2008.

[8] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[9] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng, "Collision probability assessment for speed control," in *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1043–1048.

[10] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[11] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1237–1247, 2011.

[12] 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.

[13] 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.

[14] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *Proc. of the 19th IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 452–458.

[15] D. Georgiev, P. T. Kabamba, and D. M. Tilbury, "A new model for team optimization: The effects of uncertainty on interaction," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 6, pp. 1234–1247, 2008.

[16] E. Käfer, C. Hermes, C. Woehler, H. Ritter, and F. Kummert, "Recognition of situation classes at road intersections," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2010, pp. 3960–3965.

[17] 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.

[18] M. Althoff, O. Stursberg, and M. Buss, "Safety assessment of driving behavior in multi-lane traffic for autonomous vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2009, pp. 893–900.

[19] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a smartcar," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 7–12.

[20] M. Liebner, C. Ruhhammer, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2012, pp. 1162–1167.

[21] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. of the 13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1625–1631.

[22] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, 2012.

[23] W. Yao, Q. Zeng, Y. Lin, D. Xu, H. Zhao, F. Guillemard, S. Géronimi, and F. Aioun, "On-road vehicle trajectory collection and scene-based lane change analysis: Part II," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 206–220, 2017.

[24] S. Klingelschmitt, F. Damerow, V. Willert, and J. Eggert, "Probabilistic situation assessment framework for multiple, interacting traffic participants in generic traffic scenes," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 1141–1148.

[25] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Evaluating risk at road intersections by detecting conflicting intentions," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4841–4846.

[26] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[27] 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.

[28] United Nations Economic Commission for Europe, "Vienna convention on road traffic," United Nations, 1968.

[29] A. Rizaldi and M. Althoff, "Formalising traffic rules for accountability of autonomous vehicles," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 1658–1665.

[30] 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.

[31] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, [to appear].

[32] C.-G. Wallman and H. Åström, "Friction measurement methods and the correlation between road friction and traffic safety," in *VTI meddelande*. Swedish National Road and Transport Research Institute, 2001.

4 Applications to Safe Motion Planning

In this chapter, we develop methods for achieving safe motions that are enabled by our set-based prediction of the previous Chapter 3. First, we propose a risk assessment approach in Section 4.1, followed by safety verification in Sections 4.2 and 4.3, and ending with safety falsification in Section 4.4.

4.1 IV 2018: Worst-case Analysis of the Time-To-React Using Reachable Sets [66]

Summary Collision mitigation and collision avoidance systems reduce the severity and number of accidents. To determine the latest point in time at which such systems should intervene, we need to perform risk assessment that solves Problem statement 4. Therefore, time-based criticality metrics such as the Time-To-React (TTR) are commonly used. The TTR describes the latest point in time along the current trajectory of the ego vehicle at which an evasive trajectory still exists.

Since it is difficult to find exactly the latest TTR, we over-approximate the TTR at which it is guaranteed that no evasive trajectory exists anymore. Our deterministic upper bound of the TTR can be used to trigger a collision mitigation system with minimal intervention or to find a feasible emergency maneuver that avoids the collision. Such an upper bound of the TTR is obtained by using reachability analysis. We iteratively compute the set of states reachable by the ego vehicle when starting at different states along the current trajectory. As soon as the over-approximative reachable set becomes empty, an evasive trajectory definitely does not exist.

The novelty of our approach is the guaranteed over-approximation of the TTR for arbitrary traffic scenarios and current trajectories. Similar to most risk assessment methods, we also require a prediction of the other traffic participants. Our approach works with any given set-based prediction. For the numerical experiments, we use the set-based prediction of Chapter 3. Thus, all legal behaviors of other traffic participants are anticipated in the risk assessment, which results in a conservative TTR. In contrast, the TTR is often larger if the prediction considers only a single most-likely behavior; yet, the actual risk be might underestimated.

The efficient computation of the over-approximated TTR is demonstrated in different urban and rural traffic scenarios. By comparing our results to an estimated TTR obtained from an optimization-based trajectory planner, we show that our upper bound is a tight over-approximation of the exact TTR. In addition, the computation times of our approach decrease for more critical situations.

Note that the CommonRoad IDs given in the publication are referring to version 2017a of CommonRoad. They differ to the IDs in contemporary versions of CommonRoad, since the construction of IDs has been unified in version 2018a. Thus, we also provide the updated IDs.

The presented scenarios with IDs S=Z_Overtake_1a, S=GER_Ffb_1c, and S=GER_Ffb_2b of version 2017a, are now available under the IDs S=ZAM_Over-1_1:2018a, S=DEU_Ffb-1_2_S-1:2018a, and S=DEU_Ffb-2_2_S-1:2018a, respectively.

Contributions of M. K. M. K. developed the notion of the TTR using reachable sets and the over-approximation of the TTR (both together with S. S.). M. K. developed the search for the minimum, over-approximative TTR (together with S. S.). M. K. designed, conducted, and evaluated the experiments (together with S. S.). M. K. wrote the article (together with S. S.).

Conference paper ©2018 IEEE. Reprinted, with permission, from Sebastian Söngtes, Markus Koschi, and Matthias Althoff, Worst-case Analysis of the Time-To-React Using Reachable Sets, in Proc. of the IEEE Intelligent Vehicles Symposium.

2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Worst-case Analysis of the Time-To-React Using Reachable Sets

Sebastian Söntges*, Markus Koschi*, and Matthias Althoff

Abstract— **Collision mitigation and collision avoidance systems in intelligent vehicles reduce the severity and number of accidents. To determine the optimal point in time at which such systems should intervene, time-based criticality metrics such as the Time-To-React (TTR) are commonly used. The TTR describes the last point in time along the current trajectory at which an evasive trajectory exists. In this paper, we present a novel approach to determine the point in time after which it is guaranteed that no evasive maneuver exists, i.e., by using reachable sets, we over-approximate the TTR. Our deterministic upper bound of the TTR can be used to trigger a collision mitigation system or to find a feasible emergency maneuver which avoids the collision. We demonstrate the efficient computation of the tight over-approximated TTR in different urban and rural traffic scenarios, and compare our results to an estimated TTR using an optimization-based trajectory planner.**

I. Introduction

A. Motivation

Risk assessment is a crucial component of intelligent vehicles to avoid collisions within and beyond the planning horizon [1]. Advanced driver assistant systems (ADAS) have to reliably determine whether the driver is able to avoid potential collisions. If the assumed motion of the vehicle will (most likely) end in a crash, a collision mitigation system can reduce the severity of the impact. Such systems should only intervene if no evasive trajectory exists so that the driver has control of the vehicle as long as possible and to prevent unnecessary interventions (false positives). However, a system also has to detect every unavoidable collision so that no missed intervention occurs (false negatives). Self-driving vehicles, in addition, can use risk assessment to avoid collisions and to obtain optimal trajectories which are the least critical.

B. Related work

We review existing work in the categories *a)* detecting unavoidable collisions, *b)* computing the Time-To-Collision, and *c)* computing the time until the last evasive maneuver.

a) Detecting unavoidable collisions: Collision mitigation systems only intervene at unavoidable collisions, which are often approximately detected by checking a finite set of possible evasive maneuvers [2], [3]. To describe states in which the system eventually collides regardless of what trajectory it follows, the notion of Inevitable Collision States (ICS) was introduced [4]. In order to guarantee that a collision is unavoidable, one has to employ methods which consider the set of all possible trajectories. For this purpose, reachable sets, which are the set of states reachable for a system subject to a set of inputs, are often used. The work in [5] uses backward reachable sets for the example of a lane departure system. The authors of [6] determine all reachable positions while ignoring the velocity domain, which results in overly large reachable regions. In our previous work [7], we compute an over-approximation of the reachable set considering position, velocity, and acceleration constraints. This over-approximation can be used to determine the nonexistence of evasive trajectories [8].

b) Time-To-Collision: For practical employment, one does not only wish to detect whether a collision is unavoidable given the current state, but rather wants to know the time until a collision when continuing the current trajectory. Time-To-Collision (TTC) denotes the time until impact, given a predicted trajectory of the ego vehicle and of each surrounding object [9]. A worst-case analysis of the TTC is described in [10]. To account for uncertainties, one can use stochastic predictions to obtain a probabilistic TTC [11]–[13].

c) Time until last evasive maneuver: The TTC is not sufficient for collision avoidance, since it provides no information about possible evasive maneuvers. For that reason, the Time-To-React (TTR) has been proposed as the remaining time along the current trajectory until which a collision-free and dynamically feasible trajectory still exists [14]. The authors of [14] define the TTR as the maximum of the Time-To-Brake (TTB), Time-To-Steer (TTS), and Time-To-Kickdown (TTK), which correspond to maximum possible braking, steering, and acceleration trajectories, respectively. These time-based metrics are often generalized as Time-To-X (TTX), i.e., the time remaining for an action X to avoid a collision. Since [14] is only designed for restricted traffic situations with one other object, an iterative search strategy using predefined evasive trajectories is proposed in [15] for scenarios with multiple objects. An active safety system for pedestrian avoidance employing the concepts of TTB and TTS is described in [16]. To consider uncertainties when computing the TTR, one can use probabilistic collision detection systems [17]–[19], which are similar to the probabilistic TTC as described above.

It is difficult to exactly determine the TTR, since all possible evasive trajectories have to be evaluated. If only a finite number of evasive trajectories is considered, it cannot be guaranteed that one has found the latest possible

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

All authors are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany. {sebastian.soentges, markus.koschi, matthias.althoff}@tum.de

This work was partially supported by the BMW Group within the CAR@TUM project and by the project "interACT" within the EU Horizon 2020 programme under grant agreement No 723395.

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

trajectory. However, we wish to know when to react at the latest, i.e., the earliest point in time at which an evasive trajectory definitely does not exist.

C. Contribution

We propose an efficient method to over-approximate the TTR. Existing sampling-based methods (e.g., [14]–[16]) under-approximate the TTR, since they determine the time at which they can still obtain a feasible evasive trajectory. In contrast, our novel set-based approach determines an over-approximation of the TTR, since by using reachable sets, we determine the time at which it is guaranteed that no evasive maneuver exists.

Given an assumed motion of the vehicle, our upper bound of the TTR makes it possible for collision mitigation systems to know beforehand when, at the latest, to definitely intervene or warn the driver. Similarly, collision avoidance systems or autonomous vehicles can use the over-approximated TTR as the upper bound when searching for evasive trajectories, since it is guaranteed that no collision-free trajectory exists after that time.

Using our over-approximated TTR, one can now judge the accuracy of existing TTR computations. We show that our upper bound is a tight over-approximation by estimating the TTR using an optimization-based trajectory planner. Note that our method is deterministic, i.e., it always returns the same TTR for the same configuration. Furthermore, our approach is independent of a particular prediction of other objects and can be used with any given set-based traffic prediction.

The remainder of this paper is organized as follows: After defining the problem statement in Sec. II, we present our algorithm to over-approximate the TTR in Sec. III. Sec. IV describes the optimization-based trajectory generation we use for comparison. Examples of traffic scenarios in Sec. V illustrate that we can tightly over-approximate the TTR. We conclude our paper in Sec. VI.

II. Definitions and problem statement

A. Definitions

We model the motion of the vehicle by a dynamical system

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

where $x(t) \in \mathcal{X}$ is the state within the state space $\mathcal{X} \subseteq \mathbb{R}^n$, $u(t) \in \mathcal{U}$ is the input within the set of admissible control inputs $\mathcal{U} \subseteq \mathbb{R}^m$, and t is the time. The solution of (1) for an input trajectory $u(\cdot)$ and an initial state x_0 at time t_0 is denoted by the state trajectory $x\big(t; x_0, u(\cdot)\big)$. We further introduce the planning horizon T and the final time $t_f := t_0 + T$. Since we require that possible trajectories are collision-free, the vehicle must avoid the occupancy of (dynamic) obstacles $\mathcal{O}(t) \subseteq \mathbb{R}^2$. Thus, we define the set of all colliding states by

$$\mathcal{F}(t) := \left\{ x(t) \in \mathcal{X} \,\middle|\, \mathcal{A}\big(x(t)\big) \cap \mathcal{O}(t) \neq \emptyset \right\}, \tag{2}$$

where $\mathcal{A}\big(x(t)\big) \subseteq \mathbb{R}^2$ denotes the occupancy of the vehicle on the road. Using the set of colliding states (obtained from a given prediction), we can assess the risk of the current input trajectory $u_c(\cdot) \in \mathcal{U}$ of the vehicle with initial state $x_0 \notin \mathcal{F}(t_0)$:

Definition 1 (Time-To-Collision) *The Time-To-Collision (TTC) is the maximum time we can continue the current trajectory $u_c(\cdot) \in \mathcal{U}$ before we enter the set of colliding states $\mathcal{F}(\cdot)$:*

$$\mathit{TTC} := \sup_{t_* \in \mathbb{R}} \Big\{ t_* - t_0 \,\Big|\, t_* \in [t_0, t_f], \\ \forall t \in [t_0, t_*] : x\big(t; x_0, u_c(\cdot)\big) \notin \mathcal{F}(t) \Big\}.$$

Definition 2 (Time-To-React [14]) *The Time-To-React (TTR) is the maximum time we can continue the current trajectory $u_c(\cdot) \in \mathcal{U}$ before we have to (and still can) execute an evasive trajectory to avoid entering the set of colliding states $\mathcal{F}(\cdot)$ within the planning horizon T:*

$$\mathit{TTR} := \sup_{t_* \in \mathbb{R}} \Big\{ t_* - t_0 \,\Big|\, t_* \in [t_0, t_f], \exists u(\cdot) \in \mathcal{U}, \\ \forall t \in [t_0, t_*] : x\big(t; x_0, u_c(\cdot)\big) \notin \mathcal{F}(t) \, \wedge \\ \forall t \in [t_*, t_f] : x\big(t; x\big(t_*; x_0, u_c(\cdot)\big), u(\cdot)\big) \notin \mathcal{F}(t) \Big\}.$$

An evasive trajectory in Def. 2 is any trajectory which is collision-free until the end of the planning horizon. To consider all evasive trajectories in the set of admissible inputs, we define the reachable set of (1) given a set of possible initial states $\mathcal{X}_0$:

Definition 3 (Reachable set) *The reachable set is the set of states which are reachable at time t from an initial set $\mathcal{X}_0$ at time t_0 without entering $\mathcal{F}(\cdot)$:*

$$\mathcal{R}(t; \mathcal{X}_0, t_0) := \Big\{ x\big(t; x_0, u(\cdot)\big) \,\Big|\, x_0 \in \mathcal{X}_0, u(\cdot) \in \mathcal{U}, \\ \forall \tau \in [t_0, t] : x\big(\tau; x_0, u(\cdot)\big) \notin \mathcal{F}(\tau) \Big\}.$$

The reachable set is closely related to the existence of an evasive trajectory:

Remark 1 (Existence of collision-free trajectory) *From Def. 3 it immediately follows that a collision-free trajectory exists if and only if the reachable set of the current state x_0 is nonempty at the final time t_f:*

$$\mathcal{R}(t_f; x_0, t_0) \neq \emptyset \Rightarrow \\ \exists u(\cdot) : \forall \tau \in [t_0, t_f] : x\big(\tau; x_0, u(\cdot)\big) \notin \mathcal{F}(\tau).$$

Thus, the TTR can also be expressed in terms of $\mathcal{R}$:

Proposition 1 (Time-To-React using reachable sets) *The TTR is the last point in time along the current trajectory from which the reachable set is nonempty at the end of the planning horizon:*

$$\mathit{TTR} = \sup_{t_* \in \mathbb{R}} \Big\{ t_* - t_0 \,\Big|\, t_* \in [t_0, t_f], \\ \forall t \in [t_0, t_*] : x\big(t; x_0, u_c(\cdot)\big) \notin \mathcal{F}(t) \, \wedge \\ \mathcal{R}\big(t_f; x\big(t_*; x_0, u_c(\cdot)\big), t_*\big) \neq \emptyset \Big\}.$$

Proof: Prop. 1 directly follows from Def. 2 and Def. 3. ■

In order to efficiently search for the TTR, we want to know the time interval in which the TTR is monotonic with respect to time. Using the TTC from Def. 1, we can express the monotonicity:

Proposition 2 (Monotonicity of the TTR) *Given a set of colliding states $\mathcal{F}(\cdot)$ and the current trajectory $u_c(\cdot)$ of the vehicle starting at x_0. If there is no emergency trajectory starting from $u_c(\cdot)$ at $t_1 \geq t_0$, there cannot be any trajectory starting from a later point in time $t_2 \in [t_1, t_0 + TTC]$:*

$$\begin{aligned} t_0 \leq t_1 \leq t_2 \leq t_0 + TTC \leq t_f :& \\ \mathcal{R}\big(t_f; x\big(t_1; x_0, u_c(\cdot)\big), t_1\big) = \emptyset &\Rightarrow \\ \mathcal{R}\big(t_f; x\big(t_2; x_0, u_c(\cdot)\big), t_2\big) = \emptyset&. \end{aligned}$$

Proof: From Def. 3, it follows that $x\big(t_2; x_0, u_c(\cdot)\big) \in \mathcal{R}\big(t_2; x\big(t_1; x_0, u_c(\cdot)\big), t_1\big)$ and thus $\forall t \in [t_2, t_f] : \mathcal{R}\big(t; x\big(t_2; x_0, u_c(\cdot)\big), t_2\big) \subseteq \mathcal{R}\big(t; x\big(t_1; x_0, u_c(\cdot)\big), t_1\big)$. ■

B. Problem statement

In this paper, we want to estimate the Time-To-React according to Prop. 1 for a given obstacle prediction $\mathcal{O}(t)$ by a strict and tight over-approximation $\text{TTR}_{\max} \geq \text{TTR}$, so that we know the maximum time we have to avoid a collision.

To model the motion of the vehicle, we use a velocity- and acceleration-bounded point mass:

$$f(x,u) = \begin{bmatrix} \dot{s_x} \\ \dot{s_y} \\ \dot{v_x} \\ \dot{v_y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_x \\ s_y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{3}$$

$$\begin{aligned} v_{\min,x} \leq v_x \leq v_{\max,x}, \quad & |u_x| \leq a_{\max}, \\ v_{\min,y} \leq v_y \leq v_{\max,y}, \quad & |u_y| \leq a_{\max}, \end{aligned}$$

and assume that the occupancy of the vehicle on the road is a circle with radius r_{ego}:

$$\mathcal{A}\big(x(t)\big) = \Big\{ y \,\Big|\, y \in \mathbb{R}^2, \Big\| \begin{bmatrix} s_x & s_y \end{bmatrix}^T - y \Big\|_2 \leq r_{\text{ego}} \Big\}. \tag{4}$$

The dynamical system (3) is deliberately simple and cannot accurately model a vehicle in emergency situations. However, with the two basic assumptions of bounded velocity and acceleration, it is a valid abstraction of more accurate vehicle models, i.e., the reachable set of the abstract model contains the reachable set of more accurate models, and we may use Prop. 1 to find a valid over-approximation of the TTR.

III. Over-approximation of the TTR

To determine an over-approximative $\text{TTR}_{\max}$ according to Prop. 1, we have to compute the reachable set of (3). Since the computation of the exact reachable set is often computationally not feasible, we resort to a method computing an over-approximation (superset) of the reachable set, i.e., $\forall t \geq 0 : \mathcal{R}^{\supset}(t; \mathcal{X}_0, t_0) \supseteq \mathcal{R}(t; \mathcal{X}_0, t_0)$, as described in Sec. III-A.

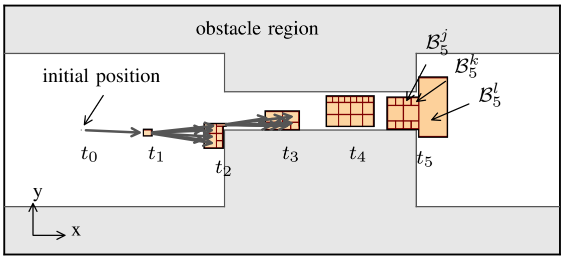


Fig. 1. Reachable set approximation for five time steps. Bold arrows indicate which subsets $\mathcal{B}_{i-1}^q$ contribute to which subsets $\mathcal{B}_i^q$ in the succeeding time step t_i.

Given a method to compute an over-approximation of the reachable set, we can determine an upper bound of the TTR using Prop. 1: If for a candidate t_*, our over-approximation $\mathcal{R}^{\supset}\big(t; x\big(t_*; x_0, u_c(\cdot)\big), t_*\big)$ vanishes at $t = t_f$, there is no evasive trajectory starting from t_*, and $t_* - t_0$ is a valid over-approximative $\text{TTR}_{\max}$. To minimize $\text{TTR}_{\max} \geq \text{TTR}$, we search the set of candidates $t_* \in [t_0, t_f]$ to find the earliest at which the reachable set vanishes, as described in Sec. III-B.

A. Over-approximation of the reachable set

We compute an over-approximation of the reachable set by using [7], which is briefly described in the following. Our method computes $\mathcal{R}_i^{\supset}$ iteratively at discrete points in time t_i. In each iteration, we first propagate $\mathcal{R}_{i-1}^{\supset}$ one time step forward to obtain the set of states $\mathcal{X}_i$:

$$\mathcal{X}_i \supseteq \bigcup_{x_{i-1} \in \mathcal{R}_{i-1}^{\supset}} \ \bigcup_{u(\cdot) \in \mathcal{U}} x\big(t_i; x_{i-1}, u(\cdot)\big). \tag{5}$$

Then, we remove the set of colliding states:

$$\mathcal{R}_i^{\supset} \supseteq \mathcal{X}_i \setminus \mathcal{F}(t_i). \tag{6}$$

Note that this approach uses several approximations which are necessary for an efficient numerical computation. These are due in particular to an efficient set representation (convex polytopes) and required set operations (e.g., set difference and union). For further details, we refer the reader to [7].

The resulting set $\mathcal{R}_i^{\supset}$ is represented by the union of four-dimensional convex polytopes $\mathcal{B}_i^q$ (in the position/velocity domain):

$$\mathcal{R}_i^{\supset} = \bigcup_q \mathcal{B}_i^q. \tag{7}$$

As an example, Fig. 1 shows the computed reachable set at different time steps in the position domain. Each of the polytopes $\mathcal{B}_i^q$ originates from one or more parents $\mathcal{B}_{i-1}^q$. The relationships between each $\mathcal{B}_{i-1}^q$ and its succeeding $\mathcal{B}_i^q$ can be represented as a directed acyclic graph. Each trajectory which visits the set $\mathcal{B}_i^q$ must have visited one of its parents $\mathcal{B}_{i-1}^q$ and must visit one of its children $\mathcal{B}_{i+1}^q$, as is illustrated in Fig. 1 for $i = 1, \ldots, 5$.

B. Search for the minimum TTR$_{\max}$

The reachable set $\mathcal{R}^{\supset}$ allows us to determine whether a candidate t_* yields a valid TTR$_{\max}$. To efficiently find the minimum TTR$_{\max}$ in the set of candidates $t_* \in [t_0, t_f]$, we propose to use binary search. Alg. 1 gives an outline of the discrete-time binary search for TTR$_{\max}$. As the upper bound for the search, we use the TTC, since $u_c(\cdot)$ is only collision-free from t_0 until t_0+TTC (cf. Def. 1 and Prop. 2). The TTC is easily determined using $u_c(\cdot)$ and $\mathcal{F}(\cdot)$. We do not refine the lower bound of the search, since this requires additional computing resources; however, one could use the Time-To-Brake as a lower bound. To compute this time, one can start at the TTC and apply the maximum feasible acceleration backwards along the path of $u_c(\cdot)$ until a state $x\big(t; x_0, u_c(\cdot)\big)$ is reached [20].

Algorithm 1 Discrete-time binary search for TTR$_{\max}$

```
Input: $x_0$ at $t_0$, $\Delta t$, $t_f$, TTC, $\mathcal{F}(t)$, $u_c(\cdot)$
Output: TTR$_{\max}$
low = 0, high = $\lceil \text{TTC}/\Delta t \rceil$
while low < high do
    mid ← $\lfloor (\text{low} + \text{high})/2 \rfloor$
    $t_*$ ← mid $\cdot \Delta t + t_0$
    if $\mathcal{R}^{\supset}\big(t_f; x\big(t_*; x_0, u_c(\cdot)\big), t_*\big)$ is not $\emptyset$ then
        low ← mid + 1
    else
        high ← mid
    end if
end while
return TTR$_{\max}$ ← low $\cdot \Delta t$
```

Usually, the TTR is computed online, i.e., during runtime of the vehicle with regular updates of x_0, $u_c(\cdot)$, and $\mathcal{F}(t)$. Thus, we can use the TTR$_{\max}$ based on the information from the previous planning step to compute the TTR$_{\max}$ based on the current information. By refining the previously obtained TTR$_{\max}$, we can enhance the search and save computation time.

IV. Estimation of the TTR through optimization-based trajectory generation

In order to evaluate the tightness of the over-approximation of our proposed set-based algorithm, we compare the upper bound TTR$_{\max}$ (computed with Alg. 1) with an estimate TTR$_{\approx}$. We determine TTR$_{\approx}$ by searching for the latest point in time from which we can explicitly generate a valid evasive trajectory using the same vehicle model (3). Suppose we have a set of possible TTR$_{\approx}$ candidates. For each candidate, we try to find a feasible trajectory starting from $t_* = t_0 +$ TTR$_{\approx}$. Finally, we choose the longest TTR$_{\approx}$ for which a feasible trajectory can be found. Each trajectory is generated by iteratively solving a convex optimization problem around an initial trajectory guess, as described below.

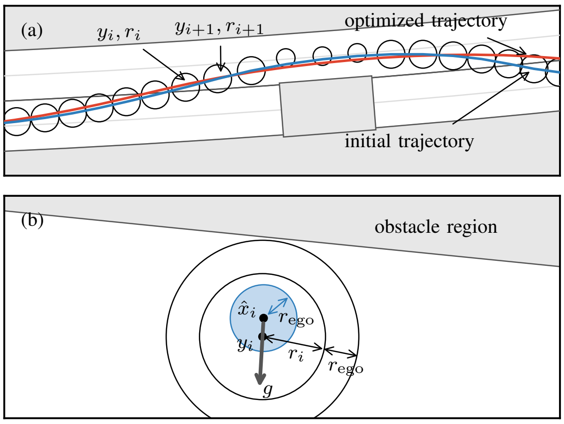


Fig. 2. (a) We obtain the locally optimized trajectory (red) from the initial trajectory (blue). At each time step i, the optimized trajectory may deviate from y_i at most by the radius r_i. (b) The center points y_i and radii r_i are determined from the initial trajectory by searching a circular region which is collision-free and contains the initial trajectory $\hat{x}_i$. The circular region is found by increasing r_i and moving y_i from $\hat{x}_i$ along the direction g (an approximation of the gradient of the distance function to the obstacle region).

To generate a collision-free initial trajectory, we use a depth-first search in the reachable set. The initial trajectory guess $\big[\hat{x}_0, \ldots, \hat{x}_*, \ldots \hat{x}_n\big]^T$ is constructed so that it matches the intended trajectory from time t_0 until time t_* and lies in the reachable set for the remaining time steps. The optimized trajectory is obtained by displacing the states at the points in time $t_{*+1}, \ldots, t_f$. As shown in Fig. 2(a), we constrain the displacement to be smaller than $r_{*+1}, \ldots, r_n$ to ensure that the initial trajectory is only locally optimized, smooth, and collision-free. Instead of directly using the center points $\hat{x}_i$ for the optimization, we introduce the positions y_i, since if $\hat{x}_i$ is close to an obstacle, the allowed displacement r_i would be small and there would only be little space for optimization. We determine y_i and r_j by a local search starting from $\hat{x}_i$, as shown in Fig. 2(b).

The optimization problem is to minimize the absolute maximum acceleration:

$$\begin{aligned} \underset{u_*,\ldots,u_{n-1}}{\text{minimize}} \quad & \Big\| \big[u_*, \ldots, u_{n-1}\big]^T \Big\|_\infty \\ \text{subject to} \quad & x_i = Ax_{i-1} + Bu_{i-1}, \\ & x_* = \hat{x}_*, \\ & \begin{bmatrix} v_{\min,x} \\ v_{\min,y} \end{bmatrix} \le C_1 x_i, \\ & \begin{bmatrix} v_{\max,x} \\ v_{\max,y} \end{bmatrix} \ge C_1 x_i, \\ & \|C_2 x_i - y_i\|_2 \le r_i, \quad i = * + 1, \ldots, n \end{aligned} \tag{8}$$

where

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

and

$$C_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

If we obtain a trajectory with $\|\left[u_*, \dots, u_{n-1}\right]^T\|_\infty \leq a_{\max}$, we accept the trajectory as a valid solution; otherwise, we repeat the optimization with this trajectory as the initial trajectory and with a new set of displacement constraints until the optimization objective cannot be further reduced.

V. Numerical Examples

We demonstrate our computation of the $\mathrm{TTR}_{\max}$ and $\mathrm{TTR}_{\approx}$ in three scenarios, which are included in the CommonRoad benchmarks[1] [21]. Tab. I lists the parameters of the numerical examples. We generate the intended trajectory $u_c(\cdot)$ such that the ego vehicle follows the center of its current lane with constant velocity. To obtain the set of occupied points of other traffic participants $\mathcal{O}(t)$ within the planning horizon, we use our prediction tool SPOT [22], which assumes that other vehicles have limited velocity and acceleration and abide by the traffic rules.

A. Two-lane road (Scenario I)

The first, deliberately simple scenario is a rural two-lane road[2]. Fig. 3(a) illustrates the initial position of the ego vehicle, its current intended trajectory, and a static obstacle in the lane of the ego vehicle.

We obtain $\mathrm{TTR}_{\max} = 0.8\,\mathrm{s}$, since it is the first time along the intended trajectory at which the reachable set becomes empty at t_f. Fig. 3(b) depicts the reachable set which is initialized at $t_* = \mathrm{TTR}_{\max} - \Delta t$. Using the optimization-based trajectory planner, we obtain an evasive trajectory which branches off at $\mathrm{TTR}_{\approx} = 0.7\,\mathrm{s}$, as shown in Fig. 3(c). The maximum acceleration of this trajectory almost requires the maximum allowed acceleration $a_{\max}$. When decreasing the time step size to $\Delta t = 0.01\,\mathrm{s}$, we obtain $\mathrm{TTR}_{\max} = 0.79\,\mathrm{s}$ and $\mathrm{TTR}_{\approx} = 0.72\,\mathrm{s}$, which shows that $\mathrm{TTR}_{\max}$ is a rather tight upper bound.

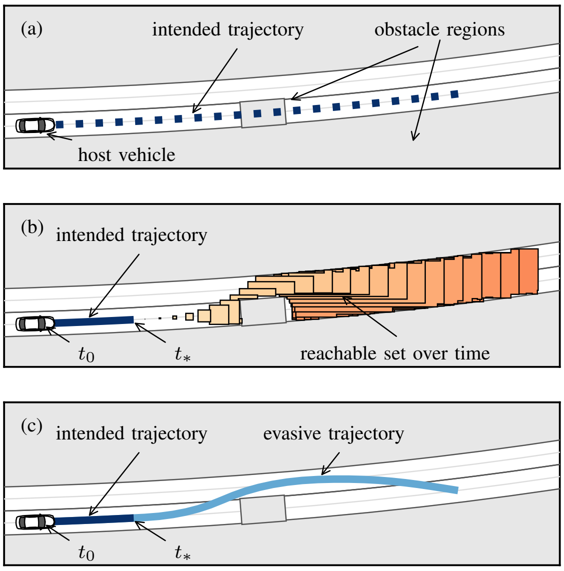


Fig. 3. Results of Scenario I: $\mathrm{TTR}_{\max} = 0.8\,\mathrm{s}$ and $\mathrm{TTR}_{\approx} = 0.7\,\mathrm{s}$. (a) Initial configuration with current trajectory $u_c(\cdot)$ from t_0 until t_f. (b) The reachable set $\mathcal{R}^{\supset}\big(t; x\big(t_*; x_0, u(\cdot)\big), t_*\big)$ starting at $t_* = \mathrm{TTR}_{\max} - \Delta t$ is plotted for all times $t \in [t_*, t_f]$. (c) The latest possible evasive trajectory branches off the intended trajectory at $t_* = \mathrm{TTR}_{\approx}$.

TABLE I

Parameters of the Scenarios (S.) I to III.

Parameter of ego vehicle	**Value**
Initial speed (S. I)	$v_0 = 20.0\,\mathrm{m/s}$
Initial speed (S. II)	$v_0 = 7.0\,\mathrm{m/s}$
Initial speed (S. III)	$v_0 = 14.0\,\mathrm{m/s}$
Minimum velocity (S. I)	$v_{\min,x} = 0.0\,\mathrm{m/s}$
Minimum velocity (S. I)	$v_{\min,y} = -10.0\,\mathrm{m/s}$
Maximum velocity (S. I)	$v_{\max,x} = 25.0\,\mathrm{m/s}$
Maximum velocity (S. I)	$v_{\max,y} = 10.0\,\mathrm{m/s}$
Minimum velocity (S. II, III)	$v_{\min,x/y} = -14.0\,\mathrm{m/s}$
Maximum velocity (S. II, III)	$v_{\max,x/y} = 14.0\,\mathrm{m/s}$
Absolute maximum acceleration	$a_{\max} = 10.0\,\mathrm{m/s^2}$
Radius of vehicle	$r_{\mathrm{ego}} = 0.9\,\mathrm{m}$
Parameter of simulation	**Value**
Initial time	$t_0 = 0\,\mathrm{s}$
Time horizon	$T = 3.0\,\mathrm{s}$
Time step size	$\Delta t = 0.1\,\mathrm{s}$

B. Intersection (Scenario II)

Scenario II features an urban intersection, where the ego vehicle intends a left turn[3]. As shown in Fig. 4(a), an approaching vehicle is predicted to continue straight and another vehicle, whose initial position is located outside of the figure, is predicted to turn right.

Our over-approximation results in $\mathrm{TTR}_{\max} = 1.1\,\mathrm{s}$ and our estimation in $\mathrm{TTR}_{\approx} = 1.0\,\mathrm{s}$. The reachable set and the optimized trajectory starting at state $x\big(t_*; x_0, u_c(\cdot)\big)$ along the intended trajectory are depicted in Fig. 4(b)–(d) for $t_* = \mathrm{TTR}_{\max} - \Delta t$ and different time intervals t. As shown in Fig. 4(b), the reachable set is very small in early time steps, and thus, we do not have much time to react. Once we have evaded the approaching vehicle on the right, we have much space on the road, as shown in Fig. 4(d). Note that we can restrict the reachable set to certain lanes, e.g., lanes with same driving direction, by adding further position constraints.

C. T-Intersection (Scenario III)

Fig. 5(a) illustrates the next urban traffic scenario, where the current trajectory of the ego vehicle continues straight with constant velocity, while three other traffic participants are detected at the T-intersection ahead[4]. Since we are uncertain about the intended maneuver of the other vehicles, the

[1] commonroad.in.tum.de

[2] CommonRoad ID: S=Z_Overtake_1a; based on [23, Fig. 3]

[3] CommonRoad ID: S=GER_Ffb_1c; based on [15, Sec. IV.-A]

[4] CommonRoad ID: S=GER_Ffb_2b; based on [2, Sec. VI.-C]

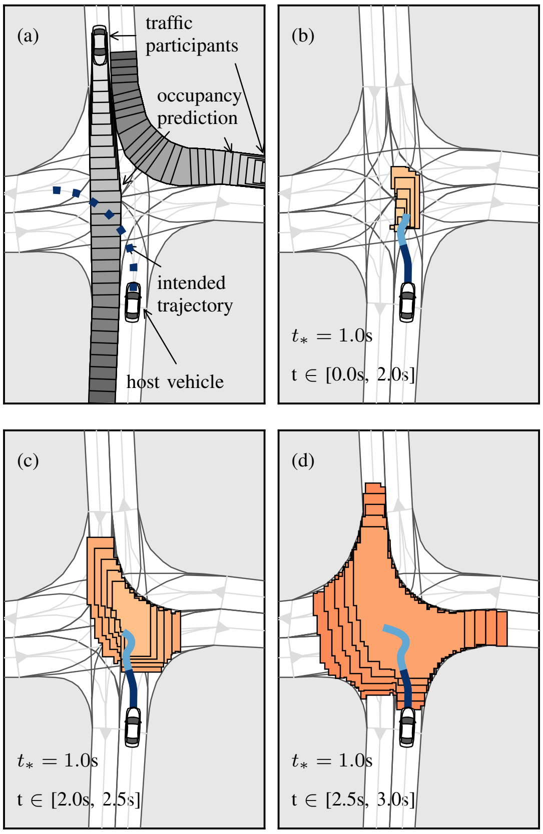


Fig. 4. Results of Scenario II: $\mathrm{TTR}_{\max} = 1.1\,\mathrm{s}$ and $\mathrm{TTR}_{\approx} = 1.0\,\mathrm{s}$. (a) Initial configuration with predicted occupancies $\mathcal{O}(t), t \in [t_0, t_f]$. (b)–(d) The reachable set and the evasive trajectory both starting at $t_* = \mathrm{TTR}_{\approx}$ are shown for different time intervals t.

occupancy prediction includes full acceleration and braking, and, for the vehicle approaching the intersection, turning left and right.

We obtain $\mathrm{TTR}_{\max} = 0.5\,\mathrm{s}$ and $\mathrm{TTR}_{\approx} = 0.3\,\mathrm{s}$. Fig. 5(b)–(d) depicts the reachable set and optimized trajectory starting at different TTR candidates t_*. It can be seen that the reachable set is very small, and thus, only a few evasive maneuvers exist. Note that, as shown in Fig. 5(d), the optimized trajectory starting at $t_* = 0.4\,\mathrm{s}$ leaves the reachable set, and its maximum accleration $\|u\|_\infty = 10.9\,{}^{\mathrm{m}}/{\mathrm{s}^2}$ is larger than $a_{\max}$; thus, this trajectory is not dynamically feasible for our vehicle model, and $\mathrm{TTR}_{\approx} = 0.3\,\mathrm{s}$.

D. Computation times

Next, we examine the computation times required to determine the reachable set for all times from the current candidate t_* until the final time t_f (i.e., line 5 of Alg. 1). Tab. II compares the computation times of all presented scenarios for different starting times t_*. We can see that the computation time of our method drastically decreases for a smaller solution space of the ego vehicle, which is beneficial when trying to efficiently determine the TTR. The computation times have been obtained using a Python/C++

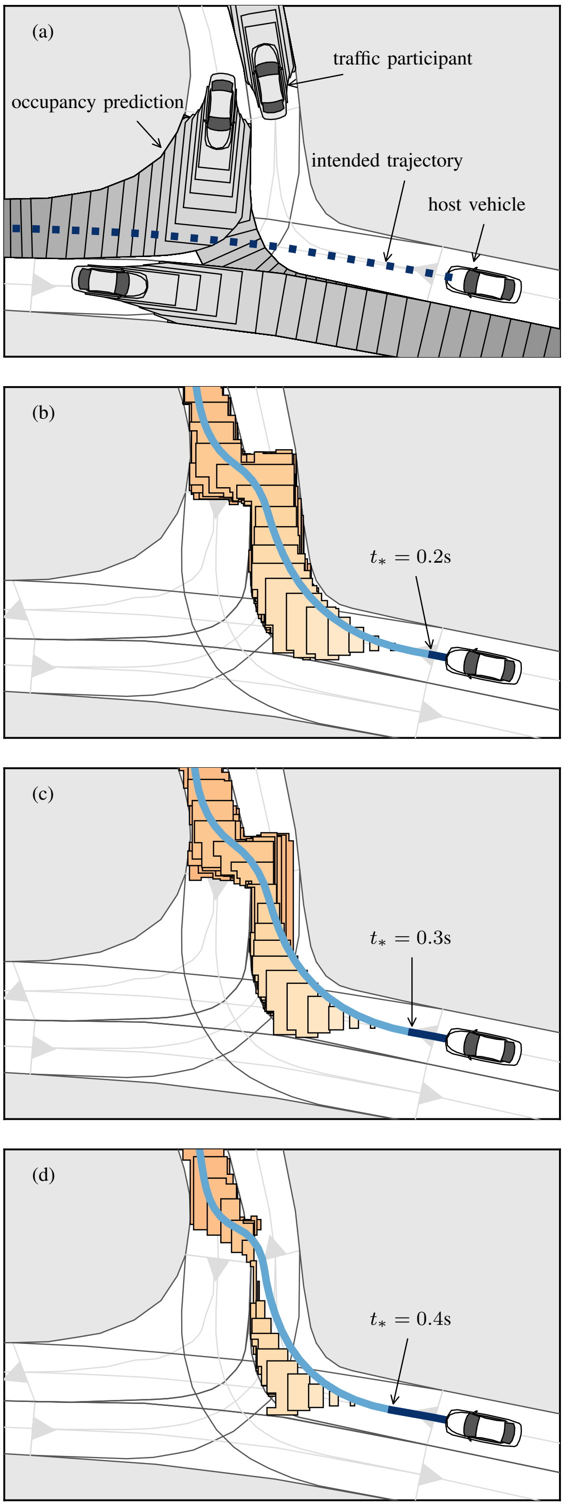


Fig. 5. Results of Scenario III: $\mathrm{TTR}_{\max} = 0.5\,\mathrm{s}$ and $\mathrm{TTR}_{\approx} = 0.3\,\mathrm{s}$. (a) Initial configuration with predicted occupancies $\mathcal{O}(t), t \in [t_0, t_f]$. (b)–(d) Starting at different TTR candidates t_*, the reachable set and the optimized trajectory are plotted for times $t \in [t_*, 2.2\,\mathrm{s}]$.

TABLE II

COMPUTATION TIMES FOR THE REACHABLE SET UNTIL t_f.

Scenario	Initialization time	Computation time
Scenario I	$t_* = 0.7\,\mathrm{s}$	$57\,\mathrm{ms}$
Scenario I	$t_* = 0.8\,\mathrm{s}$	$1\,\mathrm{ms}$
Scenario II	$t_* = 1.0\,\mathrm{s}$	$86\,\mathrm{ms}$
Scenario II	$t_* = 1.1\,\mathrm{s}$	$0.2\,\mathrm{ms}$
Scenario III	$t_* = 0.4\,\mathrm{s}$	$28\,\mathrm{ms}$
Scenario III	$t_* = 0.5\,\mathrm{s}$	$1\,\mathrm{ms}$

implementation on a machine with a $2.6\,\mathrm{GHz}$ Intel Core i7 processor with $20\,\mathrm{GB}$ $1600\,\mathrm{MHz}$ DDR3 memory. (SPOT requires around $30\,\mathrm{ms}$ to compute the occupancy of one traffic participant for the whole planning horizon.)

VI. CONCLUSION AND FUTURE WORK

We present a novel approach to tightly over-approximate the Time-To-React for risk assessment. The proposed method provides an upper bound of the TTR by iteratively computing the set of states reachable by the ego vehicle starting at states along the current trajectory. As soon as the reachable set becomes empty within the planning horizon, an evasive maneuver definitely does not exist. The novelty of our approach is that we obtain a guaranteed over-approximation of the TTR for arbitrary traffic scenarios. Our deterministic approach is independent of the prediction of other objects but can consider uncertainties in their unknown future behavior. Our experiments show that the computation times of our proposed over-approximation of the TTR are very short (below $100\,\mathrm{ms}$) and decrease for more critical situations; thus, our approach is promising for real-time application.

As future work, we wish to define terminal states which can be considered safe for an infinite time horizon so that no finite planning horizon is required.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[2] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[3] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 678–687, 2009.

[4] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2003, pp. 388–393.

[5] 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.

[6] C. Schmidt, F. Oechsle, and W. Branz, "Research on trajectory planning in emergency situations with multiple objects," in *Proc. of the 9th International IEEE Conference on Intelligent Transportation Systems*, 2006, pp. 988–992.

[7] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, 2017, [available online].

[8] ——, "Determining the nonexistence of evasive trajectories for collision avoidance systems," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 956–961.

[9] J. C. Hayward, "Near-miss determination through use of a scale of danger," Pennsylvania Transportation and Traffic Safety Center, Tech. Rep., 1972.

[10] W. Wachenfeld, P. Junietz, R. Wenzel, and H. Winner, "The worst-time-to-collision metric for situation identification," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2016, pp. 729–734.

[11] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, "A novel approach for the probabilistic computation of time-to-collision," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2012, pp. 1173–1178.

[12] J. R. Ward, G. Agamennoni, S. Worrall, A. Bender, and E. Nebot, "Extending time to collision for probabilistic reasoning in general traffic scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 66 – 82, 2015.

[13] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[14] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528–540, 2006.

[15] 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.

[16] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1292–1304, 2011.

[17] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[18] J. Eggert, "Predictive risk estimation for intelligent adas functions," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 711–718.

[19] S. Annell, A. Gratner, and L. Svensson, "Probabilistic collision estimation system for autonomous vehicles," in *Proc. of the 19th IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 473–478.

[20] E. Velenis and P. Tsiotras, "Optimal velocity profile generation for given acceleration limits: theoretical analysis," in *Proc. of the American Control Conference*, 2005, pp. 1478–1483.

[21] 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.

[22] 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.

[23] S. Herrmann, W. Utschick, M. Botsch, and F. Keck, "Supervised learning via optimal control labeling for criticality classification in vehicle active safety," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 2024–2031.

4.2 NMI 2020: Using online verification to prevent autonomous vehicles from causing accidents [71]

Summary In this section, we present the first formal verification technique to guaranteeing legal safety in arbitrary urban traffic situations, which solves Problem statement 3. Our technique serves as a safety layer for existing motion planning frameworks that provide intended trajectories for the ego vehicle and may contain machine learning components. Since such intended trajectories usually cannot ensure legal safety, we verify these trajectories and provide fallback solutions for safety-critical situations to always comply with legal safety. Therefore, we compute all legal behaviors of other traffic participants by using the set-based prediction of Chapter 3. Next, we compute the drivable area of the ego vehicle, which results from its reachable set that is collision-free against the prediction. From this drivable area, we obtain dynamics-aware driving corridors, in which we optimize fail-safe trajectories respecting the safety constraints. As a result, the ego vehicle never causes accidents although other traffic participants are allowed to perform any behavior in accordance with traffic rules.

The benefits of our verification technique are demonstrated in urban scenarios. Intersections, interactions with pedestrians, and lane changes on multi-lane roads are accidents hot spots and are featured by our experiments. Nevertheless, the ego vehicle executes only safe trajectories, even when using an intended trajectory planner that is not aware of other traffic participants. Our results, which are based on real traffic data, indicate that our online verification technique can drastically reduce the number of traffic accidents while the driving behavior of the ego vehicle does not suffer from unreasonable conservativeness.

Contributions of M. K. M. K. developed the verification technique during replanning (together with C. P. and S. M.). M. K. developed the concept and algorithms for the set-based prediction. M. K. designed, conducted, and evaluated the experiments and collected the data (all together with C. P. and S. M.). M. K. wrote the article and the Supplementary Information (both together with C. P. and S. M.).

Journal article

Attachments Additional results, which are taken from the Supplementary Information of this publication, are included in this dissertation directly after presenting the article. The complete Supplementary Information, including the Supplementary Data File and Supplementary Videos, are available at nature.com/articles/s42256-020-0225-y.

ARTICLES
https://doi.org/10.1038/s42256-020-0225-y

nature machine intelligence

Check for updates

Using online verification to prevent autonomous vehicles from causing accidents

Christian Pek[1,2] ✉, Stefanie Manzinger[1,2] ✉, Markus Koschi[1,2] ✉ and Matthias Althoff[1]

Ensuring that autonomous vehicles do not cause accidents remains a challenge. We present a formal verification technique for guaranteeing legal safety in arbitrary urban traffic situations. Legal safety means that autonomous vehicles never cause accidents although other traffic participants are allowed to perform any behaviour in accordance with traffic rules. Our technique serves as a safety layer for existing motion planning frameworks that provide intended trajectories for autonomous vehicles. We verify whether intended trajectories comply with legal safety and provide fallback solutions in safety-critical situations. The benefits of our verification technique are demonstrated in critical urban scenarios, which have been recorded in real traffic. The autonomous vehicle executed only safe trajectories, even when using an intended trajectory planner that was not aware of other traffic participants. Our results indicate that our online verification technique can drastically reduce the number of traffic accidents.

Safety remains a major challenge in the realization of autonomous vehicles. Unsafe decisions by autonomous vehicles can endanger human lives and cause tremendous economic loss in terms of product liability. Although autonomous driving is becoming a reality, recent accidents involving autonomous driving systems have raised major concerns in various institutions[1], and policy makers continue to debate about adequate safety levels for certifying autonomous vehicles[2]. To achieve widespread acceptance, safety concerns must be resolved to the full satisfaction of all road users. So far, automotive safety relies primarily on simulation and testing. However, due to the infinitely many unique real-world scenarios, these techniques cannot ensure strict safety levels[3,4], especially when using machine learning for motion planning[5].

We call for a paradigm shift from accepting residual collision risks to ensuring safety through formal verification. Formal verification describes the process of proving that a system always fulfils a desired formal specification[6]. However, in the context of safe motion planning, specifying all unsafe scenarios and proper reactions of autonomous vehicles is a tedious task[6]. Although it cannot be excluded that autonomous vehicles are involved in accidents, such as when a following car deliberately provokes a rear-end collision, self-inflicted accidents can and should be eliminated. What can we expect from human drivers to avoid self-inflicted accidents? Based on the Vienna Convention on Road Traffic, which serves as a foundation for safe driving in 78 countries, human drivers 'shall avoid any behaviour likely to endanger or obstruct traffic' (article 7 of ref. [7]). Inspired by this general rule, we demand that motions of autonomous vehicles must be collision-free under the premise that other traffic participants are allowed to perform all legal behaviours, that is, all dynamically feasible behaviours that do not violate traffic rules. Following refs. [8,9], we refer to this specification as 'legal safety'.

In contrast to related work, our holistic approach computes all legal behaviours of other traffic participants and collision-free fallback plans for the autonomous vehicle. Our solution serves as a safety layer for existing motion planning frameworks. These frameworks generate intended trajectories but cannot guarantee legal safety. However, in combination with our verification technique, legal safety is ensured. Our technique provides the following three key features:

1. Online situation assessment: The safety of each traffic situation is assessed online during operation of the autonomous vehicle by rigorously predicting all legal future evolutions of the scenario (blue areas, Fig. 1) while accounting for measurement uncertainties. In contrast to classical testing approaches, even previously unseen traffic environments can be handled, that is, scenarios with arbitrary road geometries and number of traffic participants.
2. Fail-safe operation: Our approach ensures that the autonomous vehicle always has a fail-safe trajectory to a standstill in designated safe areas, which serves as a fallback plan in the case where a safety-critical situation occurs (see the fail-safe trajectory in Fig. 1).
3. Correct by construction: Regardless of the provided motion planning framework, which may include machine learning components, our verification technique ensures that the autonomous vehicle operates in compliance with legal safety at all times. Furthermore, our safety guarantees hold even if certain traffic rules are not yet included in our technique, because, from the set of all dynamically feasible behaviours, we only remove the behaviours that are illegal according to the considered traffic rules.

At present, verification is performed during the design process—that is, offline, before systems are deployed[10]. However, offline verification is not suitable for autonomous vehicles, as these vehicles operate in highly uncertain environments in which each scenario is unique. For this reason, online verification approaches have been introduced that verify safety properties during operation of the autonomous vehicles (section II-C of ref. [11]), for example, through logical reasoning[12,13] or avoiding inevitable collision states[14,15]. In the case where a trajectory is classified as unsafe, these approaches usually do not provide an alternative safe plan for the vehicle. In the field of control, popular safety techniques are robust model predictive control approaches[16–18] and correct-by-construction controllers,

[1]Cyber-Physical Systems Group, Department of Informatics, Technical University of Munich, Garching, Germany. [2]These authors contributed equally. Christian Pek, Stefanie Manzinger, Markus Koschi. ✉e-mail: christian.pek@tum.de; stefanie.manzinger@tum.de; markus.koschi@tum.de

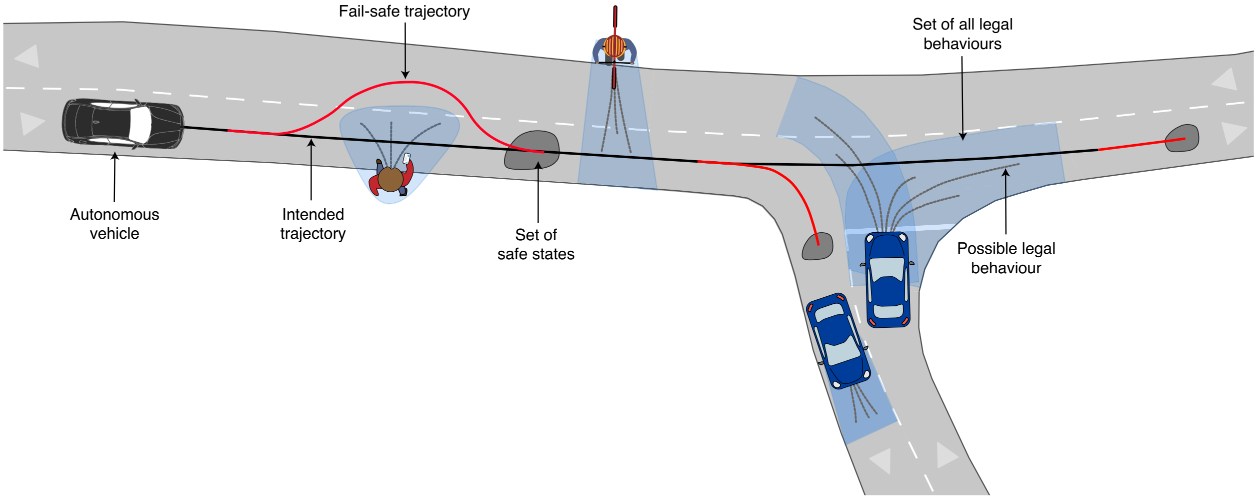


Fig. 1 | Verification of legal safety. Intended trajectories (black line) are usually planned by only considering the most likely behaviours (grey lines) of other traffic participants. Our online verification technique ensures that the autonomous vehicle is safe in accordance with legal safety by maintaining fail-safe trajectories (red lines) at all times. These fail-safe trajectories are collision-free against the set of all legal behaviours (blue areas) of other traffic participants and safeguard the autonomous vehicle along its intended trajectory to safe states (grey areas).

for example, involving barrier certificates[19], Lyapunov functions[20] or automatic controller synthesis[21]. These approaches ensure that the vehicle avoids unsafe states or is kept within an invariant set of safe states[22,23] at all times. Closely related recent approaches incorporate reachability analysis to compute the set of states that a system is able to reach over time. Thus, it can be verified that unsafe states are not reached during operation[9,24–26]. However, these existing approaches are often computationally intractable, do not generalize to arbitrary traffic scenarios or do not provide the required prediction of unsafe sets in dynamic environments.

In the context of autonomous driving, the time-variant unsafe sets are commonly defined as the future occupied positions of other traffic participants, which can be obtained by motion prediction[27]. Existing prediction approaches usually compute a countable set of most likely behaviours by applying probabilistic[28–30] or machine learning methods[31–33]. The safety of autonomous vehicles is guaranteed only if no traffic participant deviates from the few predicted behaviours, but such deviations often occur in real traffic. By incorporating reachability analysis, predictions are able to consider an infinite number of possible future behaviours of dynamic obstacles[9,34–37]. Yet, allowing for all dynamically feasible behaviours of other traffic participants overly limits the manoeuvrability of the autonomous vehicle. Therefore, our reachability-based prediction only considers behaviours that are dynamically feasible in the road network and that do not violate a set of formalized traffic rules (blue areas, Fig. 1).

The motion planner for fail-safe trajectories must cope with small and convoluted solution spaces. Commonly used trajectory planning techniques either discretize the input or state space of the autonomous vehicle[38,39] or apply variational techniques in continuous space[40–42]. The former methods suffer from discretization effects, such that narrow passageways in the solution space may not be found[43] or safe terminal states may not be reached[44]. Although variational-based methods overcome these limitations, the non-convexity of the motion planning problem due to nonlinear vehicle dynamics and collision avoidance poses a major challenge. As a result, variational-based techniques are often computationally complex[45–47] or must be guided through the solution space to work in dense traffic situations[48,49], for example, by specifying driving corridors that represent temporal tactical decisions, such as overtaking an obstacle on the left or right. Approaches for obtaining driving corridors generally do not consider the dynamics of the autonomous vehicle[50–52] and thus may not be able to reason about the drivability of driving corridors. Our approach combines reachability analysis with convex optimization to determine drivable fail-safe trajectories within dynamics-aware driving corridors in arbitrary traffic scenarios (fail-safe trajectories, Fig. 1).

Results

Our verification technique ensures legal safety over consecutive verification cycles. A new verification cycle $c \in \mathbb{N}_+$ begins whenever an intended trajectory I_c is provided by the intended trajectory planner of the existing motion planning framework, where c is incremented by one for each received intended trajectory. The autonomous vehicle can only start executing a new intended trajectory I_c that is starting at t_c if I_c is successfully verified as legally safe. A trajectory is legally safe if it (1) is collision-free against the predicted occupancy sets (that is, occupied positions) that result from all legal behaviours of other traffic participants and (2) leads the autonomous vehicle to a safe terminal state.

Typically, the time horizon T_{I_c} of I_c is several seconds for planning anticipatory motions. However, the predicted occupancy sets of the surrounding traffic participants become increasingly large for longer time horizons due to growing uncertainties regarding their future behaviours. Thus, I_c is often not safe over its entire time horizon T_{I_c}. For the safety verification (Fig. 2a), we therefore do not consider the entire intended trajectory I_c, but only a short part of I_c lasting from t_c until $t_c + \Delta_c^{\text{safe}}$, where $\Delta_c^{\text{safe}} \in \mathbb{R}_+$. We regard this part of I_c as legally safe and refer to it as I_c^{safe} if it is collision-free against the predicted occupancy sets within its entire time duration Δ_c^{safe}. Because I_c^{safe} does not ensure that the autonomous vehicle remains legally safe for $t > t_c + \Delta_c^{\text{safe}}$, we compute a consecutive fail-safe trajectory F_c (the index of F indicates the corresponding intended trajectory I). The fail-safe trajectory F_c needs to smoothly continue I_c^{safe}, be collision-free against the predicted occupancy sets for its entire time horizon T_{F_c}, and transition the autonomous vehicle to a standstill in safe areas. We say that I_c is verified successfully if I_c^{safe} and F_c exist and are computed prior to t_c. The concatenation of I_c^{safe} and F_c represents the verified trajectory and is denoted as $I_c^{\text{safe}} \parallel F_c$.

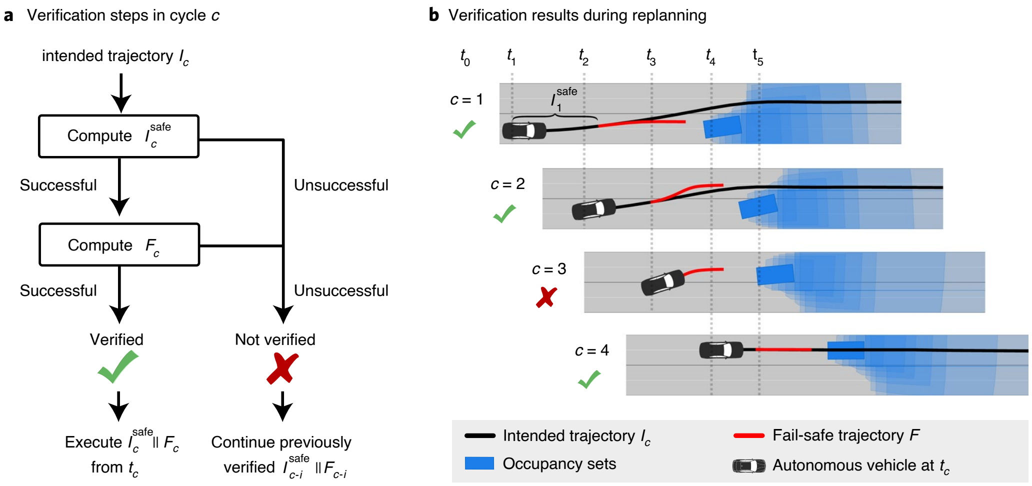

a Verification steps in cycle c
intended trajectory I_c
Compute I_c^{safe}
Successful
Unsuccessful
Compute F_c
Successful
Unsuccessful
Verified
✓
Not verified
✗
Execute $I_c^{\text{safe}} \parallel F_c$ from t_c
Continue previously verified $I_{c-i}^{\text{safe}} \parallel F_{c-i}$
b Verification results during replanning
t_0
t_1
t_2
t_3
t_4
t_5
$c = 1$
I_1^{safe}
✓
$c = 2$
✓
$c = 3$
✗
$c = 4$
✓
Intended trajectory I_c
Fail-safe trajectory F
Occupancy sets
Autonomous vehicle at t_c

Fig. 2 | Verification during replanning. a, In each verification cycle c, the given intended trajectory I_c is verified by computing the safe part I_c^{safe} and the fail-safe trajectory F_c. **b**, If the verification result of cycle c is successful (as in $c \in \{1, 2, 4\}$), the verified trajectory $I_c^{\text{safe}} \parallel F_c$ is executed starting at t_c. If the verification result is unsuccessful (as in $c = 3$), the verified trajectory I_{c-i}^{safe} and F_{c-i} of a previous successful verification cycle $c - i$ is executed until a new intended trajectory is successfully verified again (as in $c = 4$).

Let us explain the verification procedure during replanning using Fig. 2. Initially, at t_0, we assume that the autonomous vehicle is in a safe state (for example, parked). Immediately after the autonomous vehicle successfully verifies a given intended trajectory I_1 in verification cycle $c = 1$ (that is, I_1^{safe} and F_1 are obtained), the vehicle is allowed to engage in the autonomous driving mode at time t_1 and starts executing I_1^{safe} of the verified trajectory $I_1^{\text{safe}} \parallel F_1$ (see the result of $c = 1$ in Fig. 2b). The intended trajectory planner can then provide new intended trajectories I_c, $c > 1$, for verification. If a new trajectory I_c is successfully verified, the autonomous vehicle can transition from the previously verified trajectory to I_c^{safe} of the new verified trajectory $I_c^{\text{safe}} \parallel F_c$ at time t_c (see Fig. 2a and the result of $c \in \{2, 4\}$ in Fig. 2b). If the intended trajectory I_c cannot be verified, the most recently verified trajectory $I_{c-i}^{\text{safe}} \parallel F_{c-i}$ of cycle $c - i$, $i \in \{1, \ldots, c - 1\}$, continues to be executed (see Fig. 2a and the result of $c = 3$ in Fig. 2b). While moving along $I_{c-i}^{\text{safe}} \parallel F_{c-i}$, the fail-safe trajectory F_{c-i} is only executed if no new intended trajectory can be successfully verified before the final time of I_{c-i}^{safe}. This previously verified trajectory $I_{c-i}^{\text{safe}} \parallel F_{c-i}$ remains collision-free as long as other traffic participants do not violate traffic rules, because our set-based prediction has already anticipated all their legal future behaviours. Thus, legal safety is ensured regardless of the verification result.

Experiments on real data. For all verification cycles c in our experiments, the starting time of fail-safe trajectories F_c is equal to the starting time of the next intended trajectory I_{c+1}, that is, $t_c + \Delta_c^{\text{safe}} = t_{c+1}$ (see result for $c = 2$ in Fig. 2b). This is achieved by choosing a constant replanning rate $\Delta t = t_{c+1} - t_c$ (meaning that new intended trajectories should be executed at rate Δt) that is set to the constant duration of I_c^{safe} as $\Delta t = \Delta_c^{\text{safe}}$ for all c. Consequently, when executing a verified trajectory $I_c^{\text{safe}} \parallel F_c$, the transition to the fail-safe trajectory F_c may only occur at t_{c+1}. Thus, in each time interval $[t_c, t_{c+1}]$, the autonomous vehicle either executes I_c^{safe} completely or a part of F_{c-i} of a previously verified $I_{c-i}^{\text{safe}} \parallel F_{c-i}$. In other words, only if the current verification result is not successful do the autonomous vehicles transition from the safe part of an intended trajectory to a fail-safe trajectory.

In urban environments, most accidents occur at intersections and with pedestrians[53]. To demonstrate that our proposed verification technique allows autonomous vehicles to handle these crucial cases, we created two scenarios by recording real traffic with a BMW 7 series vehicle. By post-processing the real-world recordings, as described in the Supplementary Information, and applying our verification technique offline, we obtained the results presented below. For each of the two scenarios we illustrate an overview of the traffic situation using recorded images from the BMW 7 series vehicle and show the verification results of selected verification cycles c (Figs. 3 and 4). In addition, we demonstrate for both scenarios that our method guarantees legal safety for arbitrary intended trajectory planners (Fig. 5). In the Supplementary Information, we further provide a scenario illustrating safe lane changes (where the third most accidents occur[53]), further results including videos, detailed computation times (177 ms on average), all used parameters and software to visualize the verification results for all verification cycles.

Scenario I: left-turn at an urban intersection. In countries where vehicles drive on the right (we apply this throughout this Article), left turns at intersections are among the most hazardous manoeuvres, because the autonomous vehicle must consider the right of way of oncoming vehicles and yield to potential cyclists in their dedicated lane (Fig. 3a). The behaviour of oncoming vehicles or cyclists may change rapidly over time. For example, vehicles may accelerate or decelerate, and cyclists may even stop and dismount, which increases the uncertainty about the future evolution of the traffic scenario. Under all circumstances, the autonomous vehicle must yield to oncoming traffic while not disrupting the traffic flow due to overly conservative behaviour.

Our method accomplishes this challenge by safeguarding the opportunistic intended trajectory with fail-safe trajectories that (1) comply with the right of way and (2) never stop the autonomous vehicle in the intersection area. Because our prediction accounts for all legal behaviours of other traffic participants, our verification technique can decide whether a left turn manoeuvre can be completed before oncoming traffic can enter the intersection. Thus, the autonomous vehicle automatically respects the right of way.

As illustrated in Fig. 3b at $t_1 = 0$ s, the autonomous vehicle first approaches the intersection along its intended trajectory, that is, I_c^{safe}, $c \in \{1, \ldots, 4\}$, is executed. From $t_5 = 2.4$ s to $t_{10} = 5.4$ s, our

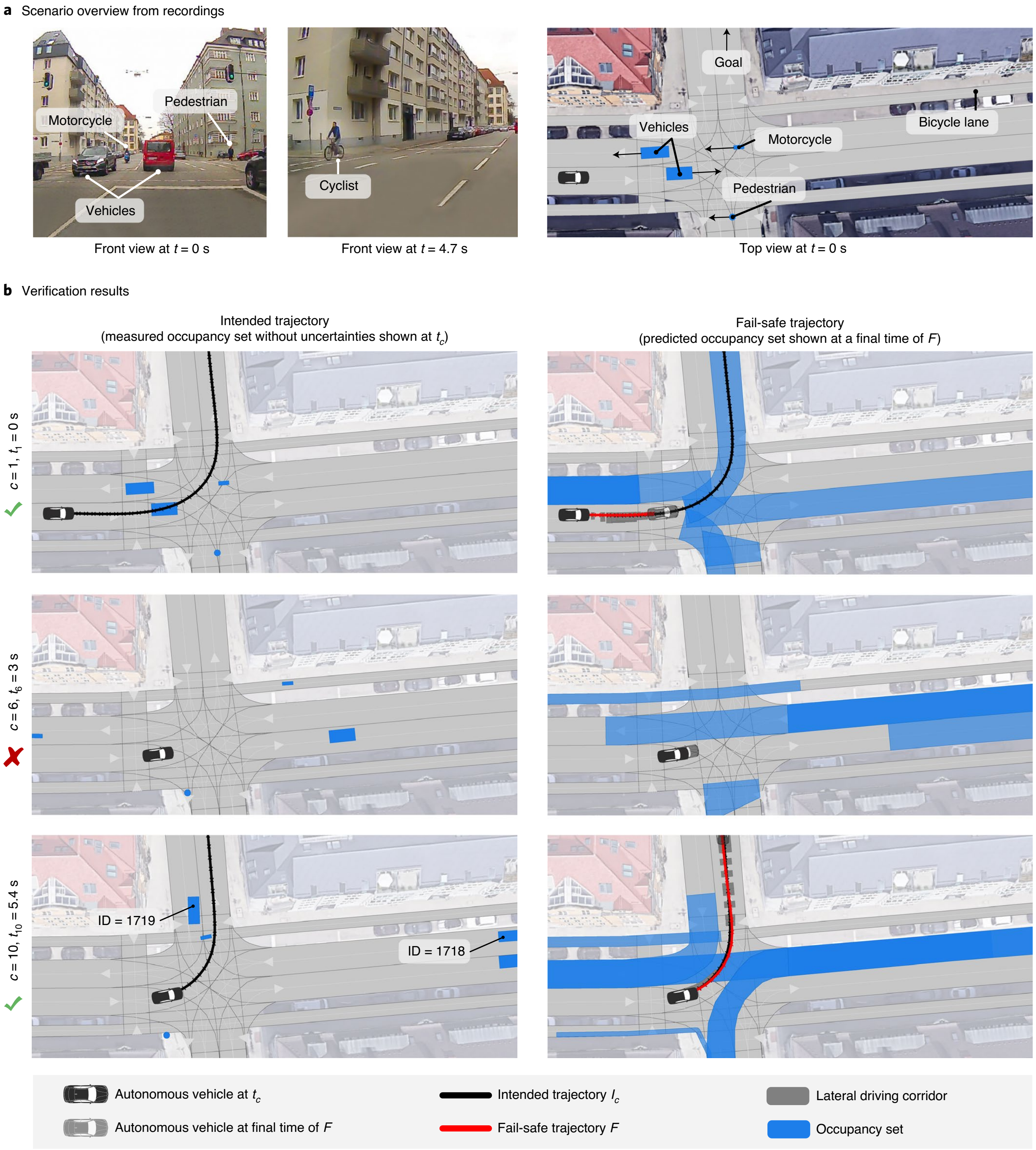


Fig. 3 | Results of Scenario I (urban intersection). a, Camera images and top view of the scenario. **b**, Verification results of selected verification cycles c. The intended trajectory I_c is only shown if it is successfully verified. Credit: Google, GeoBasis-DE/BKG (satellite images).

approach automatically detects that the intended trajectories lead to an unsafe situation in which a collision with the oncoming vehicle within the intersection area cannot be excluded before the cyclist has definitely passed. The fail-safe trajectory thus stops the autonomous vehicle at the intersection (see fail-safe trajectory at $t_6 = 3$ s in Fig. 3b). Immediately after the cyclist has passed, our verification technique successfully verifies an intended trajectory and the autonomous vehicle continues its left turn before oncoming traffic, as shown in Fig. 3b at $t_{10} = 5.4$ s. Note that, in this figure, the fail-safe trajectory overlays the occupancy sets, because the occupancy sets are shown at the final time of the fail-safe trajectory (see Supplementary Fig. 8 for the occupancy sets at intermediate times). Figure 3b also demonstrates that our prediction incorporates traffic rules. Consider the occupancy set of the oncoming vehicle with

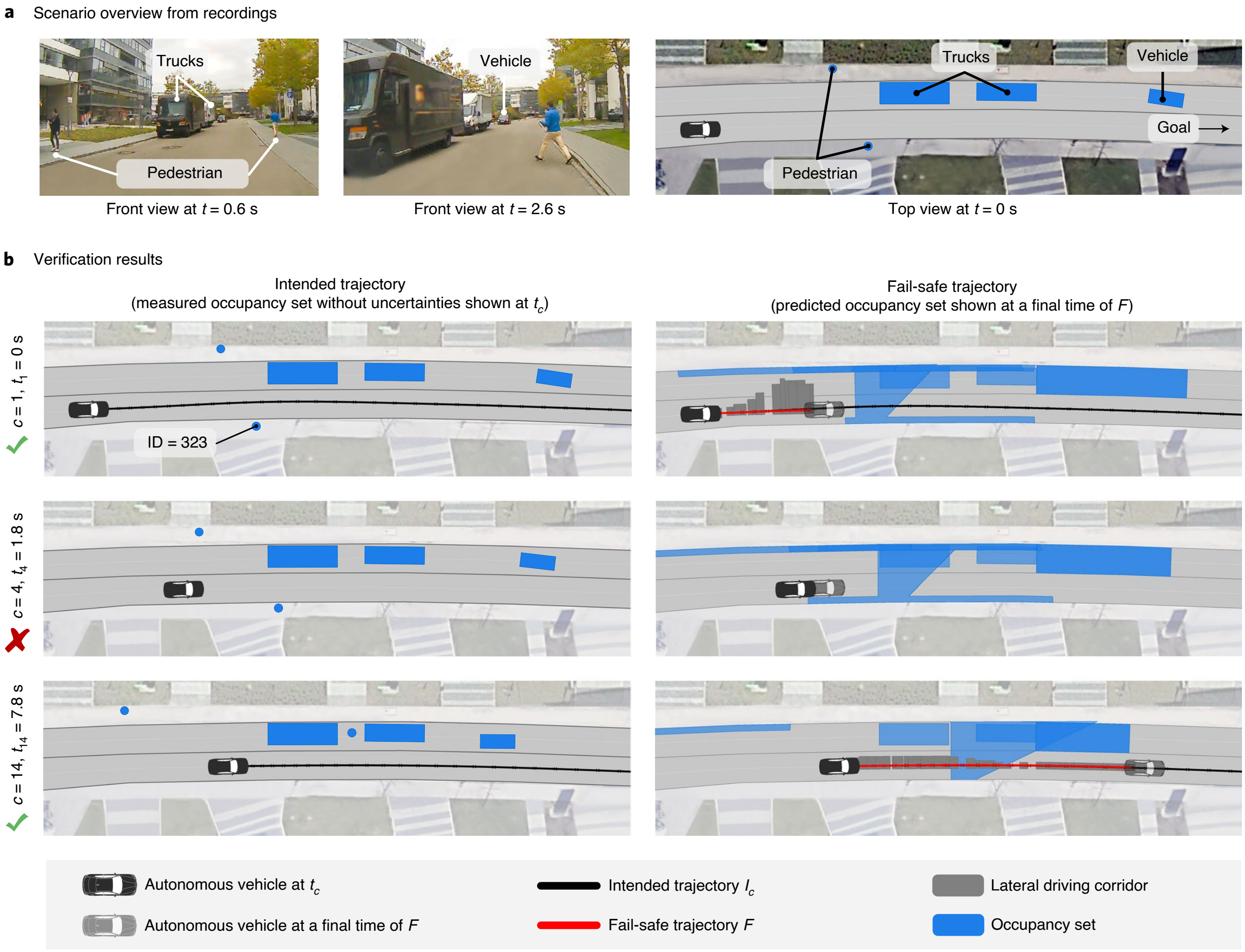

a Scenario overview from recordings
Trucks
Pedestrian
Front view at $t = 0.6$ s
Vehicle
Front view at $t = 2.6$ s
Trucks
Vehicle
Goal
Pedestrian
Top view at $t = 0$ s
b Verification results
Intended trajectory
(measured occupancy set without uncertainties shown at t_c)
Fail-safe trajectory
(predicted occupancy set shown at a final time of F)
$c = 1$, $t_1 = 0$ s
ID = 323
$c = 4$, $t_4 = 1.8$ s
$c = 14$, $t_{14} = 7.8$ s
Autonomous vehicle at t_c
Autonomous vehicle at a final time of F
Intended trajectory I_c
Fail-safe trajectory F
Lateral driving corridor
Occupancy set

Fig. 4 | Results of Scenario II (jaywalking pedestrian). a, Camera images and top view of the scenario. **b**, Verification results of selected verification cycles c. The intended trajectory I_c is only shown if it is successfully verified. Credit: Google, GeoBasis-DE/BKG (satellite images).

ID 1718 at $t_{10} = 5.4$ s. The legal safe distance forbids vehicles to turn after the autonomous vehicle in a way that obstructs the autonomous vehicle. Therefore, the vehicle with ID 1718 is only allowed to continue straight or turn left, but may not yet turn right.

Scenario II: jaywalking pedestrian. Vulnerable road users pose a special challenge to autonomous vehicles, because they often exhibit unexpected changes in behaviour. In particular, pedestrians can quickly change their walking direction, which makes it difficult for autonomous vehicles to react in time. Even though it is illegal for pedestrians to jaywalk, that is, to cross the road in the presence of traffic, pedestrians are occasionally inattentive and cross directly in front of passing vehicles. If the prediction of the autonomous vehicle does not include this behaviour, a fatal accident could occur.

In the first verification cycle $c = 1$ presented in Fig. 4, the pedestrian with ID 323 (in a blue jacket) is walking on the sidewalk and is only looking at his cell phone (Fig. 4a). To anticipate that this inattentive pedestrian may jaywalk, we broaden the set of considered legal behaviours for this pedestrian by relaxing the constraints in its prediction. As a result, the autonomous vehicle computes the future occupancies of this pedestrian for both crossing the road and walking partially on the road parallel to the sidewalk (see occupancy set in Fig. 4b for the fail-safe trajectory at $t_1 = 0$ s; note that occupancy sets of pedestrians are not visualized outside of the road). The resulting fail-safe trajectory F_1 (starting at t_2) ensures that the autonomous vehicle remains behind the pedestrian.

In the next verification cycles $c \in \{2, 3, 4\}$, the autonomous vehicle cannot verify the new intended trajectories. In fact, each intended trajectory collides with the jaywalking pedestrian. Thus, by automatically executing the first computed fail-safe trajectory F_1, the autonomous vehicle slows down to avoid a collision with the pedestrian with ID 323 (see $t_4 = 1.8$ s in Fig. 4b). After the pedestrian crosses, the autonomous vehicle accelerates to the desired velocity, and the fail-safe trajectory implies that the autonomous vehicle is able to pass before the pedestrian may walk back towards the lane of the autonomous vehicle (see $t_{14} = 7.8$ s in Fig. 4b).

As demonstrated in this scenario, our verification technique offers its users, such as mobility providers, the flexibility to define the legal behaviours differently for specific types of traffic participants. For example, when driving past a school, one may wish to anticipate that any child or even any pedestrian may cross the road.

Legal safety for arbitrary intended trajectories. We apply our verification technique to three different intended trajectory planners (for details see Supplementary Information):

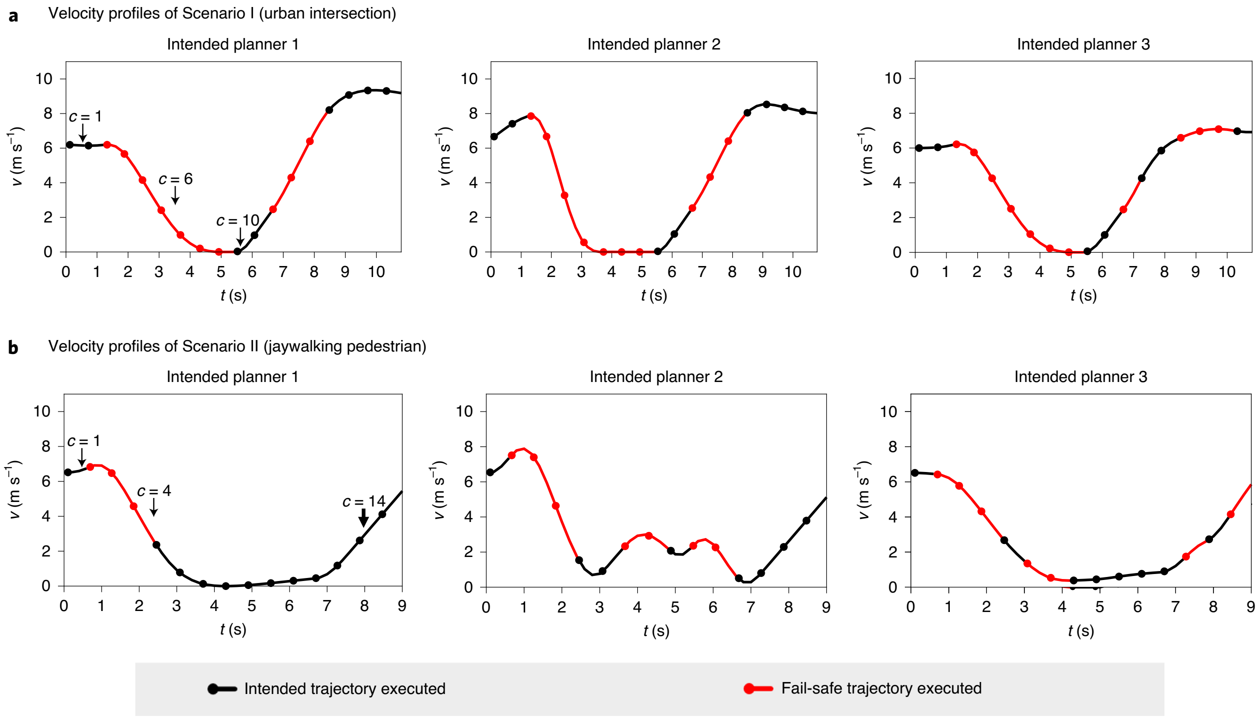


Fig. 5 | Results of the verification technique with different intended planners. **a**, Executed velocity profiles in Scenario I (results of cycles $c\in\{1, 6, 10\}$ are labelled). **b**, Executed velocity profiles in Scenario II (results of cycles $c\in\{1, 4, 14\}$ are labelled).

- Planner 1 uses continuous optimization to plan trajectories that are collision-free with regard to the most likely behaviour of other traffic participants. This planner is also used as an intended trajectory planner for the previous results of Scenarios I and II.
- Planner 2 is based on Planner 1 with the modification that other traffic participants are ignored. With this planner, we mimic a reinforcement learning approach that has not yet learned collision avoidance.
- Planner 3[39] samples in a discrete state space to plan trajectories that are collision-free with regard to the most likely behaviour of other traffic participants.

Figure 5 illustrates the velocity profiles of the autonomous vehicle in Scenarios I and II for each intended trajectory planner. In Scenario I, our verification technique intervenes independently of the applied intended trajectory planner so that the autonomous vehicle stops in front of the intersection (Fig. 5a). Although Planner 2 is not aware of other traffic participants, our verification technique enables the autonomous vehicle to safely turn left. Because Planner 2 tries to reach the desired velocity (8 m s^{-1}) more aggressively than Planners 1 and 3 (see the results of verification cycles $c\in\{1, 2\}$ in Fig. 5a), the subsequently executed fail-safe trajectories cause a rapid deceleration of the autonomous vehicle (peak, −6 m s^{-2}) (see the results of verification cycles $c\in\{3, \ldots, 8\}$ for Intended Planner 2 in Fig. 5a). However, the execution of fail-safe trajectories for Planner 2 causes only a short delay, as the stopping time at the intersection is less than 2 s.

In Scenario II, the intended trajectory planners are not aware of the pedestrian's intention to jaywalk. Therefore, fail-safe trajectories are executed to slow down the autonomous vehicle (see the results of verification cycles $c\in\{2, 3, 4\}$ in Fig. 5b) until Planners 1 and 3 react to the pedestrian. Planner 2 requires permanent guidance to avoid a collision with the pedestrian. Although the type of executed trajectory, that is, I_c^{safe} or F_{c-i}, continuously alternates, the average velocity of the autonomous vehicle with Planner 2 is 5% higher than that with Planner 1 (6.36 m s^{-1} and 6.09 m s^{-1}, respectively).

In summary, we are able to guarantee legal safety for different intended trajectory planners, even when using a planner that ignores other traffic participants. Furthermore, the resulting velocity profiles are smooth and continuous, as fail-safe trajectories are planned with full consideration of the vehicle's dynamics.

Discussion

Certification is the main obstacle to achieving commercial success with the proposed verification technique. Regulatory guidelines have already been prepared for various domains, such as railway systems, industrial robots and aviation systems, but only limited regulations exist for motion planning of autonomous vehicles (for example, ISO 26262 and ISO 21448). We have prepared the ground for certification by formulating legal safety and presenting a verification technique that ensures that this specification is met during operation of the autonomous vehicle. Moreover, the safety guarantees are maintained when adapting our considered set of traffic rules to new requirements. If legal safety becomes a recognized standard for autonomous vehicles, mobility providers can certify our proposed verification technique for usage in their vehicles. As a result, we expect that societal trust in autonomous vehicles will increase and that testing efforts can be significantly reduced, even if motion planning frameworks for generating intended trajectories are changed.

Legal safety is a promising novel safety approach inspired by traffic regulations that is suitable for certification. Related concepts, such as responsibility-sensitive safety[54], not-at-fault driving[26] and compositional and contract-based verification[55], share our premise to avoid (self-inflicted) accidents, but differ substantially to our proposed solution. Responsibility-sensitive safety assumes that other traffic participants act according to common-sense rules

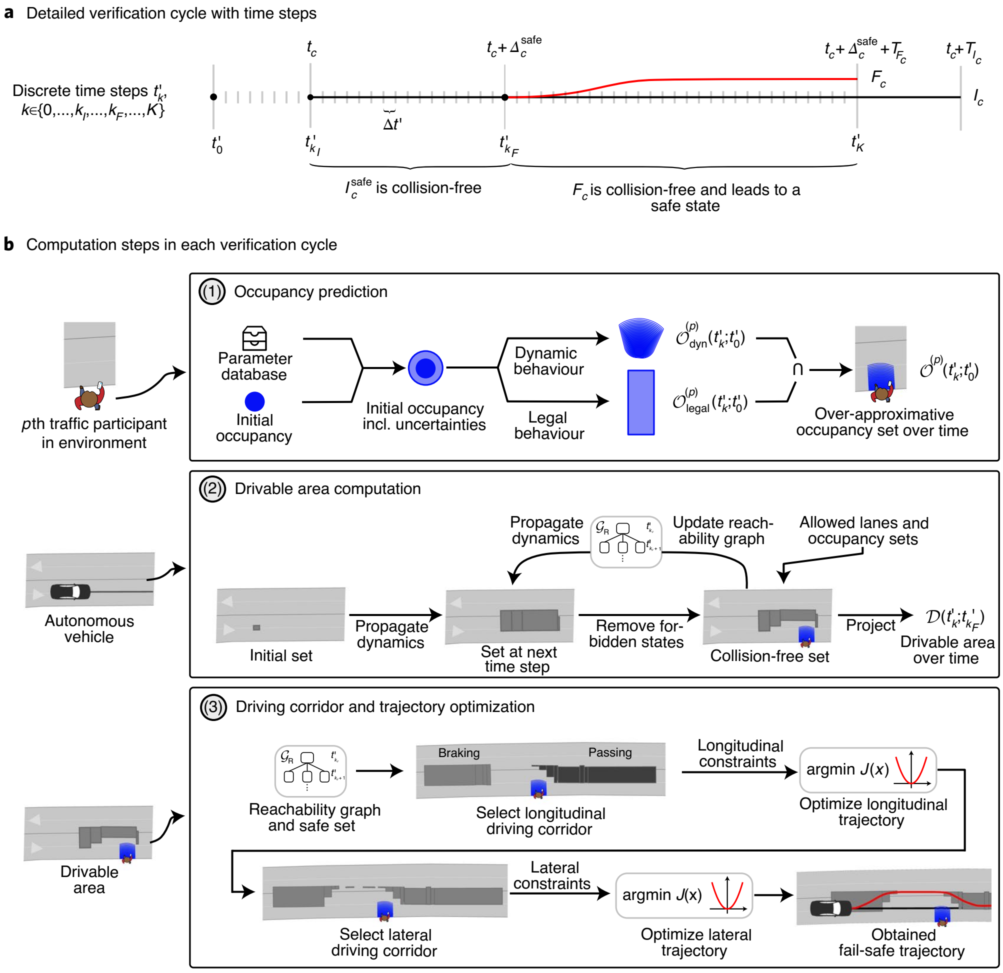

a Detailed verification cycle with time steps
Discrete time steps t'_k, $k\in\{0,\dots,k_I,\dots,k_F,\dots,K\}$
t_c
$t_c+\Delta_c^{\text{safe}}$
$t_c+\Delta_c^{\text{safe}}+T_{F_c}$
$t_c+T_{I_c}$
F_c
I_c
$\Delta t'$
t'_0
t'_{k_I}
t'_{k_F}
t'_K
I_c^{safe} is collision-free
F_c is collision-free and leads to a safe state
b Computation steps in each verification cycle
(1) Occupancy prediction
pth traffic participant in environment
Parameter database
Initial occupancy
Initial occupancy incl. uncertainties
Dynamic behaviour
Legal behaviour
$\mathcal{O}^{(p)}_{\text{dyn}}(t'_k;t'_0)$
$\mathcal{O}^{(p)}_{\text{legal}}(t'_k;t'_0)$
∩
$\mathcal{O}^{(p)}(t'_k;t'_0)$
Over-approximative occupancy set over time
(2) Drivable area computation
Autonomous vehicle
Propagate dynamics
$\mathcal{G}_R$
Update reachability graph
Allowed lanes and occupancy sets
Initial set
Propagate dynamics
Set at next time step
Remove forbidden states
Collision-free set
Project
$\mathcal{D}(t'_k;t'_{k_F})$
Drivable area over time
(3) Driving corridor and trajectory optimization
Drivable area
$\mathcal{G}_R$
Reachability graph and safe set
Braking
Passing
Select longitudinal driving corridor
Longitudinal constraints
argmin $J(x)$
Optimize longitudinal trajectory
Select lateral driving corridor
Lateral constraints
argmin $J(x)$
Optimize lateral trajectory
Obtained fail-safe trajectory

Fig. 6 | Computation steps of the verification technique. **a**, Time discretization t'_k in one verification cycle c. **b**, Overview of the computation steps for verifying an arbitrary intended trajectory. (1) We compute occupancy sets, that is, all legally occupied positions of other traffic participants over time. (2) The drivable area of the autonomous vehicle is computed to determine fail-safe manoeuvres. (3) Longitudinal and lateral driving corridors are selected from the reachability graph, and longitudinal and lateral trajectories are optimized such that a fail-safe trajectory is obtained.

and defines appropriate responses by the autonomous vehicle based on safe distances. However, despite the execution of appropriate responses, self-inflicted accidents cannot be excluded, because other traffic participants may behave differently than expected. Our approach addresses this problem by considering all legal behaviours. Not-at-fault driving computes a single trajectory that is split into moving, braking and stopped phases and is provably collision-free against a given prediction. By contrast, we allow intended trajectories to be planned independently of fail-safe trajectories, for example, using a most likely prediction to optimize comfort. In ref. [55], a finite number of offline-verified, local models are fitted online to the current traffic situation. However, this approach may result in unsafe behaviours if no valid composition of these local models can be found for the current situation. Our verification technique evaluates the safety of situations online and always provides fail-safe trajectories to eliminate self-inflicted accidents. The detailed computation steps of our verification technique are described in the Methods and are visualized in Fig. 6.

Methods

Formal verification is often believed to cause performance drops (for example, lower average velocities resulting in longer travel times) and conservative behaviour in robotic systems[56,57]. However, we believe that autonomous vehicles can offer high performance and ensure legal safety at the same time. This has motivated us to improve on our previous work on set-based predictions[58–60], fail-safe trajectory planning[61] and trajectory planning using reachable sets[62]. Further to our previous work, we present the following innovations:

1. Our proposed verification technique ensures legal safety in complex traffic scenarios and in a computationally efficient way. In particular, by embedding driving corridors[62] into fail-safe trajectory planning[61], we generalize the computation of possible fail-safe manoeuvre options to different traffic situations and can consider multiple safe terminal sets.
2. On various urban scenarios that have been recorded in real traffic including measurement uncertainties, the applicability of the proposed verification technique is demonstrated. In addition, our results indicate that non-conservative driving behaviour can be achieved despite the over-approximative, set-based prediction.
3. The temporal interplay over subsequent verification cycles of our verification technique with the intended trajectory planner of the autonomous vehicle is presented in detail.
4. Further experiments with three different intended trajectory planners validate that our verification technique is able to ensure legal safety for arbitrary intended trajectory planners.

In the following paragraphs, we present the inputs of our verification technique, preliminaries for the reachability analysis, an overview of the algorithmic steps and the safety guarantees for our verification technique. Additional details are provided in the Supplementary Information.

Inputs of the verification technique. Our verification technique is integrated between the motion planning layer and the control layer of the autonomous vehicle (see planning frameworks in refs. [63,64]). In each verification cycle c, our verification technique receives as inputs the intended trajectory I_c and the environment model. The intended trajectories must be kinematically feasible and branch off the previously verified trajectory $I_{c-i}^{\text{safe}} \parallel F_{c-i}$. The environment model must contain the lanes of the road, pedestrian crossings and areas in which the autonomous vehicle is not allowed to stop, which are used to obtain the designated safe areas. For all safety-relevant traffic participants, the environment model must contain their type (that is, vehicle, motorcycle, bicycle or pedestrian) and their current states (that is, a set containing the exact state and bounded measurement uncertainties). If the type of traffic participant is unknown or uncertain, our verification technique can predict the set of future behaviours for all possible types in parallel.

Preliminaries of the verification technique. The motion of the pth traffic participant is governed by the differential equation $\dot{x}^{(p)}(t) = f^{(p)}\left(x^{(p)}(t), u^{(p)}(t)\right)$, where $x^{(p)}$ is the state and $u^{(p)}$ is the input. The admissible states and inputs are bounded by the respective sets $\mathcal{X}^{(p)}(t) \subset \mathbb{R}^{n^{(p)}}$ and $\mathcal{U}^{(p)}(t) \subset \mathbb{R}^{m^{(p)}}$. A possible solution of the differential equation at time t is denoted by $\chi^{(p)}\left(t; x^{(p)}(\tau_0), u^{(p)}(\cdot)\right)$, when starting at state $x^{(p)}(\tau_0) \in \mathcal{X}_0^{(p)}$, where $\mathcal{X}_0^{(p)}$ is the set of states at an initial time τ_0 including measurement uncertainties, and using input trajectory $u^{(p)}(\cdot)$. The reachable set $\mathcal{R}^{\mathrm{e}^{(p)}}(t; \tau_0) \subseteq \mathcal{X}^{(p)}(t)$ describes the set of states that are reachable by the pth traffic participant at a certain point in time $t \geq \tau_0$ when starting in $\mathcal{X}_0^{(p)}$ and applying all admissible inputs $\mathcal{U}^{(p)}(t)$:

$$\begin{aligned}\mathcal{R}^{\mathrm{e}^{(p)}}(t; \tau_0) = \Big\{\chi^{(p)}\left(t; x^{(p)}(\tau_0), u^{(p)}(\cdot)\right) \,\Big|\, & x^{(p)}(\tau_0) \in \mathcal{X}_0^{(p)}, \forall \tilde{\tau} \in [\tau_0, t] : \\ & \chi^{(p)}\left(\tilde{\tau}; x^{(p)}(\tau_0), u^{(p)}(\cdot)\right) \in \mathcal{X}^{(p)}(\tilde{\tau}), u^{(p)}(\tilde{\tau}) \in \mathcal{U}^{(p)}(\tilde{\tau})\Big\}\end{aligned} \tag{1}$$

For brevity, we omit the superscript (p) when referring to the autonomous vehicle. In each verification cycle c, we compute the reachable set of other traffic participants to predict their future movement and that of the autonomous vehicle to obtain its drivable area.

As illustrated in Fig. 6a, we introduce the discrete points in time t'_k for each verification cycle c, where $k \in \{0, \dots, k_I, \dots, k_F, \dots K\} \subseteq \mathbb{N}_0$; for brevity, the notation of t'_k does not reflect its dependency on c. Time t'_0 is the initial time of the prediction, that is, the point in time at which the most recently available environment model has been recorded. Time t'_{k_I} corresponds to the start time of the intended trajectory I_c (that is, $t'_{k_I} = t_c$), t'_{k_F} corresponds to the start time of the fail-safe trajectory F_c (that is, $t'_{k_F} = t_c + \Delta_c^{\text{safe}}$) and t'_K corresponds to the final time of the fail-safe trajectory (that is, $t'_K = t_c + \Delta_c^{\text{safe}} + T_{F_c}$). Without loss of generality, we assume that the times t'_k are multiples of the time step size $\Delta t' \in \mathbb{R}_+$, that is, $t'_k = t'_0 + k\Delta t'$.

Recall that we set Δ_c^{safe} to the replanning rate Δt in our experiments. To minimize the interventions of our verification technique, that is, how often a fail-safe trajectory is executed, the duration Δ_c^{safe} can be dynamically adjusted to optimize the length of I_c^{safe} as described in ref. [65]. To avoid that new intended trajectories cannot be verified solely due to a timeout, intended trajectories I_c should be provided prior to $t_c - \Delta^{\text{verify}}$, where $\Delta^{\text{verify}} \in \mathbb{R}_+$ is the required computation time of our verification method.

Occupancy prediction. The goal in the first step of our verification technique is to over-approximate the area $\mathcal{L}^e(t)$ that exactly encloses the occupied positions of the surrounding traffic participants for all their legal behaviours. Therefore, we first compute all dynamically feasible behaviours and subsequently remove illegal behaviours.

All dynamically feasible behaviours of other traffic participants are obtained using reachability analysis as defined in equation (1). For each pth traffic participant, the environmental model provides the initial states $\mathcal{X}_0^{(p)}$ at t'_0, which are described by a set due to measurement uncertainties (Fig. 6b, step (1)). The dynamics of each traffic participant are abstracted by a second-order integrator model with bounded velocities and accelerations. We compute the reachable set $\mathcal{R}^{(p)}(t; t'_0)$ as a tight over-approximation of the exact reachable set, that is, $\mathcal{R}^{(p)}(t; t'_0) \supseteq \mathcal{R}^{e^{(p)}}(t; t'_0)$, and only for the position domain to allow for an efficient computation. For collision checks with planned trajectories of the autonomous vehicle, we introduce $\mathcal{O}_{\text{dyn}}^{(p)}(t; t'_0)$ as the dynamics-based occupancy set resulting from the over-approximative reachable set $\mathcal{R}^{(p)}(t; t'_0)$ by considering the dimensions of the pth traffic participant (Fig. 6b, step (1)).

Next, we remove behaviours that are not allowed according to traffic rules. Therefore, we formalize a set of traffic rules that is most relevant for motion planning (and which can be easily extended). Let $v^{(p)}$ and $a^{(p)}$ denote the velocity and acceleration of the pth predicted traffic participant, respectively, and $\Diamond^{\text{veh}}$ denotes that the parameter $\Diamond \in \{\overline{v}, \overline{a}\}$ bounding the velocity or acceleration is applicable for vehicles and motorcycles, while $\Diamond^{\text{cyc}}$ is for bicycles and $\Diamond^{\text{ped}}$ is for pedestrians (the values of the parameters are stored in a database generated offline, can be updated online, and are provided in the Supplementary Information). The considered traffic rules for vehicles, motorcycles and bicycles are as follows:

- Maximum velocity is bounded (article 13.2 of ref. [7]): $v^{(p)} \leq v_{\text{limit}} f_S^{(p)}$, where v_{limit} is the legal speed limit of the road and $f_S^{(p)} \geq 1$ is a parameterized speeding factor to consider slight over-speeding. If no speed limit is available, such as for bicycles, $v^{(p)} \leq \overline{v}^{\text{veh/cyc}}$.
- Driving backward is not allowed (article 14.2 of ref. [7]): $v^{(p)} \geq 0$.
- Absolute acceleration is bounded (due to tyre friction): $|a^{(p)}| \leq \overline{a}^{\text{veh/cyc}}$.
- Leaving the road is forbidden (article 14.1 of ref. [7]).
- A safe distance to the autonomous vehicle must be maintained when driving behind it or merging in front of it (articles 13.5 and 11.2d of ref. [7]).
- Changing lanes is only allowed if the new lane has the same driving direction as the previous one (article 11.2c of ref. [7]).

Note that, according to article 11.2c of ref. [7], overtaking in a lane not appropriate to the direction of traffic is only allowed if not endangering or interfering with oncoming traffic. Because such a legal overtaking manoeuvre does not interfere with the motion planning of the autonomous vehicle, we neglect it in our prediction without compromising legal safety.

Although pedestrians are generally not allowed to obstruct vehicular traffic, for example, to jaywalk (article 7.1 of ref. [7]), vehicles are required to take precautions to avoid endangering pedestrians (article 21.1 of ref. [7]). Thus, the considered traffic rules for pedestrians are as follows:

- Absolute velocity is bounded (for example, based on ISO 13855): $|v^{(p)}| \leq \overline{v}^{\text{ped}}$.
- Absolute acceleration is bounded (due to physical capabilities): $|a^{(p)}| \leq \overline{a}^{\text{ped}}$.
- Entering the road is forbidden (articles 7.1 and 20.2 of ref. [7]) except
    - on pedestrian crossings (articles 20.6b and 21.2 of ref. [7])
    - when walking toward the road; then, crossing the road is allowed perpendicularly with a deviation of angle α based on the current heading of the pedestrian (articles 20.6c,d of ref. [7])
    - when walking parallel to the road; then, occupying the strip of the road edge with a width of d_{slack} is allowed, for example, to avoid obstacles on the sidewalk (articles 20.2a, 20.3 and 20.4 of ref. [7]).

In summary, our set of traffic rules either constrains the dynamics of other traffic participants (for example, their maximum velocity), which are considered by $\mathcal{O}_{\text{dyn}}^{(p)}(t; t'_0)$, or constrains the allowed regions in the environment (for example, certain lanes or pedestrian crossings), which are given by the environment model and are denoted by $\mathcal{O}_{\text{legal}}^{(p)}(t; t'_0)$. The resulting over-approximative occupancy set of the pth traffic participant is $\mathcal{O}^{(p)}(t; t'_0) = \mathcal{O}_{\text{dyn}}^{(p)}(t; t'_0) \cap \mathcal{O}_{\text{legal}}^{(p)}(t; t'_0)$ (Fig. 6b, step (1)). To verify that I_c^{safe} and F_c are collision-free, we compute the occupancy sets for consecutive time intervals $[t'_k, t'_{k+1}]$ until the final time of F_c, that is, $\forall\, k \in \{k_I, \dots, K\}$. Note that the time intervals $[t'_k, t'_{k+1}]$ can be of different duration for each k, for example, in case I_c^{safe} and F_c are discretized differently. The predicted occupancy sets of all traffic participants are given by $\mathcal{L}([t'_k, t'_{k+1}]) = \bigcup_p \bigcup_{t \in [t'_k, t'_{k+1}]} \mathcal{O}^{(p)}(t; t'_0)$.

Note that, regardless of how many traffic rules we consider, our prediction always over-approximates the exact set of all legal behaviours, that is, $\mathcal{L}(t) \supseteq \mathcal{L}^e(t)$. The reason is that only behaviours defined as illegal are removed from the over-approximation of all dynamically feasible behaviours. The fewer traffic rules we consider, the more cautiously the autonomous vehicle behaves, because it respects more behaviours than actually allowed according to all traffic rules. However, the autonomous vehicle definitely remains collision-free when other traffic participants adhere to all traffic rules, as prescribed by legal safety. If a collision occurs nonetheless, we can verifiably argue that another traffic participant must have violated traffic rules and that the collision is not self-inflicted by the autonomous vehicle. Nevertheless, we account for humans' tendency to violate traffic rules, such as the speed limit. Therefore, we continuously monitor whether any traffic participant performs a behaviour that is not included in the set of legal behaviours. Whenever violations are detected, this behaviour is automatically added to the prediction result; for example, if another vehicle illegally changes lanes, we no longer exclude this behaviour from our prediction of this vehicle. As a result, our verification technique will attempt to find a new fail-safe trajectory in case the previous one is no longer collision-free. Furthermore, if a traffic participant appears likely to misbehave, such behaviours can be included in our prediction by disabling the corresponding constraint, as demonstrated in Scenario II.

Drivable area computation. To obtain possible sequences of high-level fail-safe manoeuvres (for example, overtaking other vehicles on their left or right), we compute the drivable area of the autonomous vehicle at discrete points in time t'_k with $k \geq k_F$ by projecting its reachable set $\mathcal{R}^e(t'_k; t'_{k_F})$ defined in equation (1) onto the position domain (Fig. 6b, step (2)). As for the prediction of other traffic participants, we abstract the dynamics of the autonomous vehicle using two second-order integrator models in the longitudinal and lateral directions with bounded velocities and accelerations in a road-aligned coordinate system[66]. For computational efficiency, the reachable set is approximated through the union of base sets $\mathcal{B}_k^{(i)}$, $i \in \mathbb{N}_0$, such that $\mathcal{R}^e(t'_k; t'_{k_F}) \approx \bigcup_i \mathcal{B}_k^{(i)}$ holds. The base sets $\mathcal{B}_k^{(i)}$ are the Cartesian products of convex polytopes describing reachable position–velocity pairs in the longitudinal and lateral directions. We use convex polytopes, because they are closed under required set operations such as Minkowski sum, linear mapping and intersection. The projection of base sets $\mathcal{B}_k^{(i)}$ onto the position domain yields axis-aligned rectangles $\mathcal{D}_k^{(i)}$ that represent the drivable area $\mathcal{D}(t'_k; t'_{k_F}) := \bigcup_i \mathcal{D}_k^{(i)}$. The projection of the reachable set onto the position domain can be computed efficiently, because we only need to determine the minimum and maximum position coordinates of the convex polytopes of the base sets $\mathcal{B}_k^{(i)}$.

The state $x(t'_{k_F})$ of the fail-safe trajectory F_c at its start time t'_{k_F} is provided by the final state of I_c^{safe}. We enclose $x(t'_{k_F})$ with a base set such that $x(t'_{k_F}) \in \mathcal{B}_{k_F}^{(0)}$ holds. The reachable set of consecutive points in time t'_{k+1}, $k \geq k_F$, is computed as illustrated in Fig. 6b (step (2)). First, we propagate each base set $\mathcal{B}_k^{(i)}$ of the previous time step forward in time considering all admissible inputs. Second, we remove states outside the set of admissible states $\mathcal{X}(t'_{k+1})$, that is, positions in which the autonomous vehicle collides with the predicted occupancy sets $\mathcal{L}([t'_k, t'_{k+1}])$ or the area $\mathcal{Q}$ outside of the road, to obtain $\mathcal{R}(t'_{k+1}; t'_{k_F}) \approx \bigcup_j \mathcal{B}_{k+1}^{(j)}$ at time t'_{k+1}. Third, we store each base set $\mathcal{B}_{k+1}^{(j)}$ in a directed graph $\mathcal{G}_{\mathcal{R}}$. In $\mathcal{G}_{\mathcal{R}}$, each set $\mathcal{B}_{k+1}^{(j)}$ is associated with exactly one node and an edge indicates that base set $\mathcal{B}_{k+1}^{(j)}$ is reachable from $\mathcal{B}_k^{(i)}$ within one time step. The procedure is repeated until the final time step t'_K is reached.

Driving corridor and trajectory optimization. We generate drivable fail-safe trajectories through continuous optimization. As convex optimization problems can be solved efficiently with global convergence, we convexify the inherently non-convex optimization problem by separating the longitudinal and lateral motion of the autonomous vehicle. However, longitudinal motion planning requires prior knowledge on the lateral motion and vice versa, as both subsystems are dynamically coupled. To overcome this issue, we obtain driving corridors from the drivable area that provide spatio-temporal position constraints for the optimization problems. We refer to the driving corridors for longitudinal and lateral optimization as the longitudinal and lateral driving corridors, respectively. To ensure legal safety for an infinite time horizon, we constrain the driving corridors to end in a safe terminal state based on the designated safe areas, for example, a standstill in the rightmost lane sufficiently far from an intersection. As illustrated in Fig. 6b (step (3)), our motion planner first optimizes the longitudinal trajectory within a longitudinal driving corridor, followed by optimizing the lateral trajectory in a suitable lateral driving corridor. Currently, we constrain fail-safe trajectories to be kinematically feasible, collision-free with respect to road boundaries and the predicted occupancy sets, respect the speed limit and end in a safe state. Further constraints can be imposed to consider additional properties, for example, rules on overtaking or stopping at the boundaries of the field of view of the vehicle.

We represent collision avoidance constraints by a minimum and maximum value on the longitudinal or lateral positions at each point in time. To obtain these limits, we exploit that a connected set in the position domain projected onto either the longitudinal or lateral direction yields an interval. Consequently, we define a longitudinal corridor and a lateral driving corridor for fail-safe motion planning as a temporal sequence of connected sets that are subsets of the drivable area $\mathcal{D}(t'_k; t'_{k_F})$ from time t'_{k_F} to the final time t'_K.

To determine longitudinal driving corridors, we perform a search on the reachability graph $\mathcal{G}_{\mathcal{R}}$ backwards in time starting from the set of safe terminal states (Fig. 6b, step (3)). There may be multiple longitudinal driving corridors, because the drivable area can be disconnected due to surrounding traffic participants. We select the longitudinal driving corridor with the greatest cumulative drivable area from t'_{k_F} to t'_K for trajectory planning (other heuristics can also be applied). For the longitudinal trajectory optimization, we use a fourth-order integrator model with jounce as input and bounded longitudinal velocity, acceleration and jerk. In addition to the collision avoidance constraints from the boundary of the longitudinal driving corridor, the autonomous vehicle must come to a standstill at the final time t'_K. To improve comfort, we choose a quadratic cost function that minimizes acceleration, jerk and jounce as well as deviations from the desired velocity.

The computation and selection of lateral driving corridors are performed similarly to the computation and selection of longitudinal driving corridors with the addition that the connected sets of the lateral driving corridor must provide a unique passing side for each obstacle. The lateral trajectories of the autonomous vehicle are optimized with respect to a linearized kinematic single-track model with limits on the steering actuators. Analogously to planning in the longitudinal direction, the position constraints for collision avoidance are obtained from the boundaries of the lateral driving corridor. We select a quadratic cost function to minimize the lateral distance and orientation deviation from a given reference path and to punish high curvature rates for comfort.

In the case that trajectory optimization is infeasible using the selected lateral or longitudinal driving corridor, we select a driving corridor with the next highest cumulative drivable area for optimization until either a fail-safe trajectory is identified or no further driving corridors remain. In the rare event that no feasible fail-safe trajectory is found, the previously verified trajectory is further executed.

Guarantees of our verification technique. To comply with legal safety, autonomous vehicles must not collide with any legal behaviour of other traffic participants:

$$\forall t \geq t_0 : occ(x(t)) \cap (\mathcal{L}^e(t) \cup \mathcal{Q}) = \emptyset \tag{2}$$

where the operator $occ(x)$ relates the state x of the autonomous vehicle to the set of occupied points in the position domain as $occ(x) : \mathcal{X} \rightarrow Pow(\mathbb{R}^n)$, where $Pow(\mathbb{R}^n)$ is the power set of $\mathbb{R}^n$.

Using the principle of induction, we sketch the proof that our technique ensures legal safety according to equation (2). For the base case ($c = 1$), for $t \geq t_0$, the autonomous vehicle is initially in a safe state in which it can remain. Only if I_c can be successfully verified will the autonomous vehicle start executing $I_c^{\text{safe}} || F_c$ from t_c. This trajectory is collision-free at all discrete time steps $t'_k \in [t_c, t_c + \Delta_c^{\text{safe}} + T_{F_c}]$ against all legal behaviours $\mathcal{L}(t) \supseteq \mathcal{L}^e(t)$ of other traffic participants and the area $\mathcal{Q}$ outside the road. If no new intended trajectory can be successfully verified in a subsequent verification cycle before $t_c + \Delta_c^{\text{safe}} + T_{F_c}$, the fail-safe trajectory F_c transitions the autonomous vehicle to a standstill in a safe terminal state at $t_c + \Delta_c^{\text{safe}} + T_{F_c}$, which is legally safe for all future times. For the inductive step, assuming that the verification result of cycle $c = r$, for any $r \in \mathbb{N}_+$, ensures legal safety, we show that legal safety is also ensured regardless of the verification result of cycle $c + 1$. If the verification is unsuccessful, the autonomous vehicle continues to execute the trajectory $I_{c-i}^{\text{safe}} || F_{c-i}$, $i \in \{0, \ldots, c-1\}$ of the previous cycle c that ensures legal safety by definition. If the verification is successful in cycle $c + 1$, the autonomous vehicle executes $I_{c+1}^{\text{safe}} || F_{c+1}$ from t_{c+1}. In this case, we can apply the same reasoning as in the base case to demonstrate that legal safety is also ensured from t_{c+1} with the verified trajectory $I_{c+1}^{\text{safe}} || F_{c+1}$.

To ensure that the autonomous vehicle is collision-free along I^{safe} and F in continuous time and despite control disturbances and model uncertainties, we refer to the approach in ref. [67].

Data availability

All data gathered and reported in this study are available in the Supplementary data file. This includes the environment model, the intended trajectory and the verification result of each verification cycle for all scenarios.

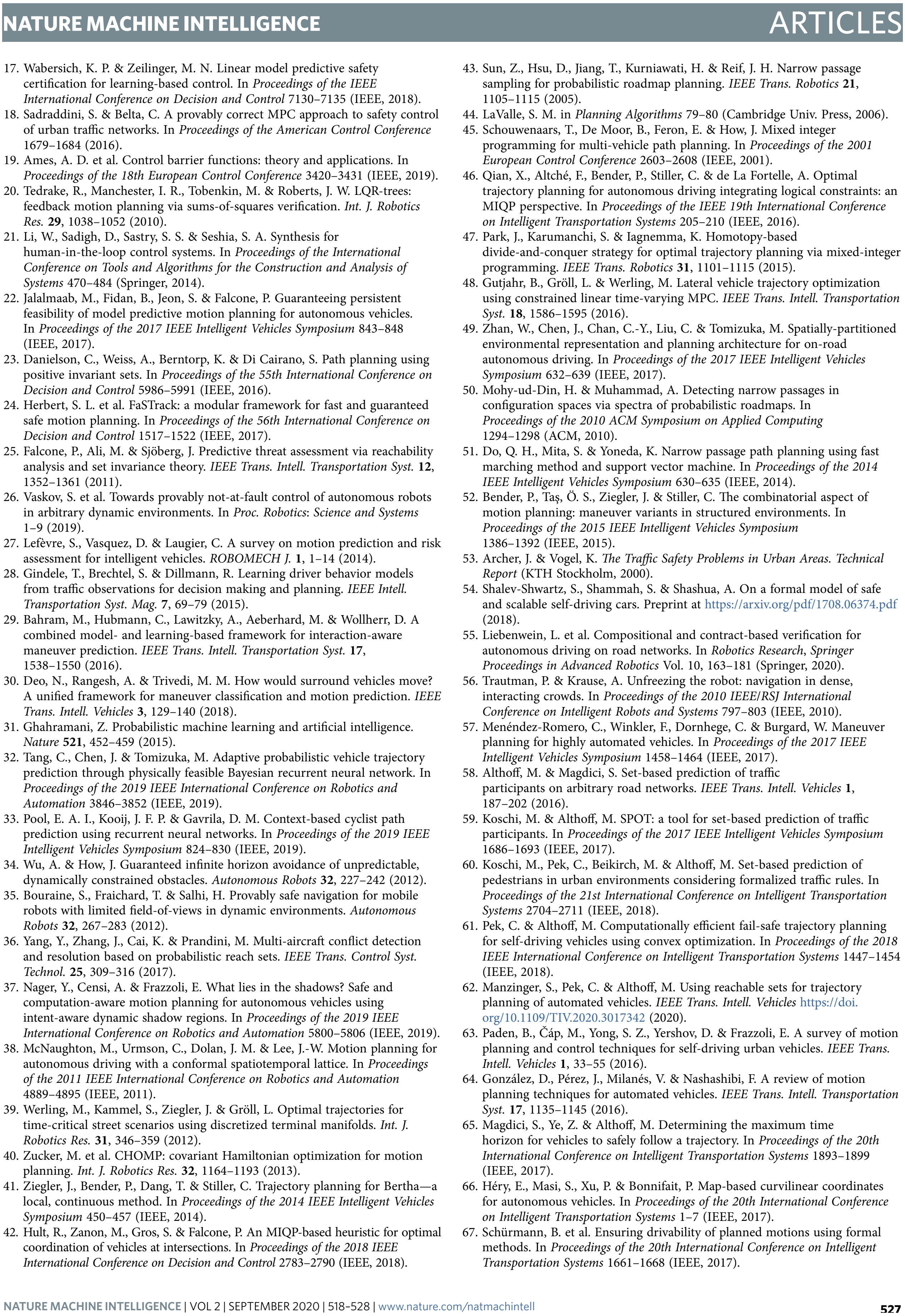
Code availability

The code to visualize and analyse the gathered data and obtained results of this study are included in the Supplementary data file.

Received: 30 October 2019; Accepted: 4 August 2020;
Published online: 14 September 2020

References

1. Favarò, F., Eurich, S. & Nader, N. Autonomous vehicles' disengagements: trends, triggers and regulatory limitations. *Accid. Anal. Prev.* **110**, 136–148 (2018).
2. Anderson, J. M. et al. *Autonomous Vehicle Technology: A Guide for Policymakers* (Rand Corporation, 2016).
3. Koopman, P. & Wagner, M. Autonomous vehicle safety: an interdisciplinary challenge. *IEEE Intell. Transportation Syst. Mag.* **9**, 90–96 (2017).
4. Kalra, N. & Paddock, S. M. Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Res. A Policy Practice* **94**, 182–193 (2016).
5. Seshia, S. A., Sadigh, D. & Sastry, S. S. Towards verified artificial intelligence. Preprint at https://arxiv.org/abs/1606.08514 (2017).
6. Schwarting, W., Alonso-Mora, J. & Rus, D. Planning and decision-making for autonomous vehicles. *Annu. Rev. Control Robot. Autonomous Syst.* **1**, 187–210 (2018).
7. United Nations Economic Commission for Europe. *Convention on Road Traffic. United Nations Conference on Road Traffic* (United Nations, 1968); consolidated version of 2006.
8. Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S. & Mammar, S. Highly automated driving on highways based on legal safety. *IEEE Trans. Intell. Transportation Syst.* **14**, 333–347 (2013).
9. Althoff, M. & Dolan, J. M. Online verification of automated road vehicles using reachability analysis. *IEEE Trans. Robotics* **30**, 903–918 (2014).
10. Koopman, P. & Wagner, M. Challenges in autonomous vehicle testing and validation. *SAE Int. J. Transportation Safety* **4**, 15–24 (2016).
11. Dahl, J., de Campos, G. R., Olsson, C. & Fredriksson, J. Collision avoidance: a literature review on threat-assessment techniques. *IEEE Trans. Intell. Vehicles* **4**, 101–113 (2019).
12. Tumova, J., Hall, G. C., Karaman, S., Frazzoli, E. & Rus, D. Least-violating control strategy synthesis with safety rules. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control* 1–10 (HSCC, 2013).
13. Kress-Gazit, H., Fainekos, G. E. & Pappas, G. J. Temporal-logic-based reactive mission and motion planning. *IEEE Trans. Robotics* **25**, 1370–1381 (2009).
14. Fraichard, T. & Asama, H. Inevitable collision states—a step towards safer robots? In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* 388–393 (IEEE, 2003).
15. Chan, N., Kuffner, J. & Zucker, M. Improved motion planning speed and safety using regions of inevitable collision. In *17th CISM-IFToMM Symposium on Robot Design, Dynamics and Control* 103–114 (Springer, 2008).
16. Koller, T., Berkenkamp, F., Turchetta, M. & Krause, A. Learning-based model predictive control for safe exploration. In *Proceedings of the 2018 IEEE International Conference on Decision and Control* 6059–6066 (IEEE, 2018).



17. Wabersich, K. P. & Zeilinger, M. N. Linear model predictive safety certification for learning-based control. In *Proceedings of the IEEE International Conference on Decision and Control* 7130–7135 (IEEE, 2018).
18. Sadraddini, S. & Belta, C. A provably correct MPC approach to safety control of urban traffic networks. In *Proceedings of the American Control Conference* 1679–1684 (2016).
19. Ames, A. D. et al. Control barrier functions: theory and applications. In *Proceedings of the 18th European Control Conference* 3420–3431 (IEEE, 2019).
20. Tedrake, R., Manchester, I. R., Tobenkin, M. & Roberts, J. W. LQR-trees: feedback motion planning via sums-of-squares verification. *Int. J. Robotics Res.* **29**, 1038–1052 (2010).
21. Li, W., Sadigh, D., Sastry, S. S. & Seshia, S. A. Synthesis for human-in-the-loop control systems. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems* 470–484 (Springer, 2014).
22. Jalalmaab, M., Fidan, B., Jeon, S. & Falcone, P. Guaranteeing persistent feasibility of model predictive motion planning for autonomous vehicles. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium* 843–848 (IEEE, 2017).
23. Danielson, C., Weiss, A., Berntorp, K. & Di Cairano, S. Path planning using positive invariant sets. In *Proceedings of the 55th International Conference on Decision and Control* 5986–5991 (IEEE, 2016).
24. Herbert, S. L. et al. FaSTrack: a modular framework for fast and guaranteed safe motion planning. In *Proceedings of the 56th International Conference on Decision and Control* 1517–1522 (IEEE, 2017).
25. Falcone, P., Ali, M. & Sjöberg, J. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Trans. Intell. Transportation Syst.* **12**, 1352–1361 (2011).
26. Vaskov, S. et al. Towards provably not-at-fault control of autonomous robots in arbitrary dynamic environments. In *Proc. Robotics: Science and Systems* 1–9 (2019).
27. Lefèvre, S., Vasquez, D. & Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.* **1**, 1–14 (2014).
28. Gindele, T., Brechtel, S. & Dillmann, R. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intell. Transportation Syst. Mag.* **7**, 69–79 (2015).
29. Bahram, M., Hubmann, C., Lawitzky, A., Aeberhard, M. & Wollherr, D. A combined model- and learning-based framework for interaction-aware maneuver prediction. *IEEE Trans. Intell. Transportation Syst.* **17**, 1538–1550 (2016).
30. Deo, N., Rangesh, A. & Trivedi, M. M. How would surround vehicles move? A unified framework for maneuver classification and motion prediction. *IEEE Trans. Intell. Vehicles* **3**, 129–140 (2018).
31. Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459 (2015).
32. Tang, C., Chen, J. & Tomizuka, M. Adaptive probabilistic vehicle trajectory prediction through physically feasible Bayesian recurrent neural network. In *Proceedings of the 2019 IEEE International Conference on Robotics and Automation* 3846–3852 (IEEE, 2019).
33. Pool, E. A. I., Kooij, J. F. P. & Gavrila, D. M. Context-based cyclist path prediction using recurrent neural networks. In *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium* 824–830 (IEEE, 2019).
34. Wu, A. & How, J. Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles. *Autonomous Robots* **32**, 227–242 (2012).
35. Bouraine, S., Fraichard, T. & Salhi, H. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. *Autonomous Robots* **32**, 267–283 (2012).
36. Yang, Y., Zhang, J., Cai, K. & Prandini, M. Multi-aircraft conflict detection and resolution based on probabilistic reach sets. *IEEE Trans. Control Syst. Technol.* **25**, 309–316 (2017).
37. Nager, Y., Censi, A. & Frazzoli, E. What lies in the shadows? Safe and computation-aware motion planning for autonomous vehicles using intent-aware dynamic shadow regions. In *Proceedings of the 2019 IEEE International Conference on Robotics and Automation* 5800–5806 (IEEE, 2019).
38. McNaughton, M., Urmson, C., Dolan, J. M. & Lee, J.-W. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation* 4889–4895 (IEEE, 2011).
39. Werling, M., Kammel, S., Ziegler, J. & Gröll, L. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *Int. J. Robotics Res.* **31**, 346–359 (2012).
40. Zucker, M. et al. CHOMP: covariant Hamiltonian optimization for motion planning. *Int. J. Robotics Res.* **32**, 1164–1193 (2013).
41. Ziegler, J., Bender, P., Dang, T. & Stiller, C. Trajectory planning for Bertha—a local, continuous method. In *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium* 450–457 (IEEE, 2014).
42. Hult, R., Zanon, M., Gros, S. & Falcone, P. An MIQP-based heuristic for optimal coordination of vehicles at intersections. In *Proceedings of the 2018 IEEE International Conference on Decision and Control* 2783–2790 (IEEE, 2018).
43. Sun, Z., Hsu, D., Jiang, T., Kurniawati, H. & Reif, J. H. Narrow passage sampling for probabilistic roadmap planning. *IEEE Trans. Robotics* **21**, 1105–1115 (2005).
44. LaValle, S. M. in *Planning Algorithms* 79–80 (Cambridge Univ. Press, 2006).
45. Schouwenaars, T., De Moor, B., Feron, E. & How, J. Mixed integer programming for multi-vehicle path planning. In *Proceedings of the 2001 European Control Conference* 2603–2608 (IEEE, 2001).
46. Qian, X., Altché, F., Bender, P., Stiller, C. & de La Fortelle, A. Optimal trajectory planning for autonomous driving integrating logical constraints: an MIQP perspective. In *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems* 205–210 (IEEE, 2016).
47. Park, J., Karumanchi, S. & Iagnemma, K. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *IEEE Trans. Robotics* **31**, 1101–1115 (2015).
48. Gutjahr, B., Gröll, L. & Werling, M. Lateral vehicle trajectory optimization using constrained linear time-varying MPC. *IEEE Trans. Intell. Transportation Syst.* **18**, 1586–1595 (2016).
49. Zhan, W., Chen, J., Chan, C.-Y., Liu, C. & Tomizuka, M. Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium* 632–639 (IEEE, 2017).
50. Mohy-ud-Din, H. & Muhammad, A. Detecting narrow passages in configuration spaces via spectra of probabilistic roadmaps. In *Proceedings of the 2010 ACM Symposium on Applied Computing* 1294–1298 (ACM, 2010).
51. Do, Q. H., Mita, S. & Yoneda, K. Narrow passage path planning using fast marching method and support vector machine. In *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium* 630–635 (IEEE, 2014).
52. Bender, P., Taş, Ö. S., Ziegler, J. & Stiller, C. The combinatorial aspect of motion planning: maneuver variants in structured environments. In *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium* 1386–1392 (IEEE, 2015).
53. Archer, J. & Vogel, K. *The Traffic Safety Problems in Urban Areas. Technical Report* (KTH Stockholm, 2000).
54. Shalev-Shwartz, S., Shammah, S. & Shashua, A. On a formal model of safe and scalable self-driving cars. Preprint at https://arxiv.org/pdf/1708.06374.pdf (2018).
55. Liebenwein, L. et al. Compositional and contract-based verification for autonomous driving on road networks. In *Robotics Research*, *Springer Proceedings in Advanced Robotics* Vol. 10, 163–181 (Springer, 2020).
56. Trautman, P. & Krause, A. Unfreezing the robot: navigation in dense, interacting crowds. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* 797–803 (IEEE, 2010).
57. Menéndez-Romero, C., Winkler, F., Dornhege, C. & Burgard, W. Maneuver planning for highly automated vehicles. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium* 1458–1464 (IEEE, 2017).
58. Althoff, M. & Magdici, S. Set-based prediction of traffic participants on arbitrary road networks. *IEEE Trans. Intell. Vehicles* **1**, 187–202 (2016).
59. Koschi, M. & Althoff, M. SPOT: a tool for set-based prediction of traffic participants. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium* 1686–1693 (IEEE, 2017).
60. Koschi, M., Pek, C., Beikirch, M. & Althoff, M. Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems* 2704–2711 (IEEE, 2018).
61. Pek, C. & Althoff, M. Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization. In *Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems* 1447–1454 (IEEE, 2018).
62. Manzinger, S., Pek, C. & Althoff, M. Using reachable sets for trajectory planning of automated vehicles. *IEEE Trans. Intell. Vehicles* https://doi.org/10.1109/TIV.2020.3017342 (2020).
63. Paden, B., Čáp, M., Yong, S. Z., Yershov, D. & Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Vehicles* **1**, 33–55 (2016).
64. González, D., Pérez, J., Milanés, V. & Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transportation Syst.* **17**, 1135–1145 (2016).
65. Magdici, S., Ye, Z. & Althoff, M. Determining the maximum time horizon for vehicles to safely follow a trajectory. In *Proceedings of the 20th International Conference on Intelligent Transportation Systems* 1893–1899 (IEEE, 2017).
66. Héry, E., Masi, S., Xu, P. & Bonnifait, P. Map-based curvilinear coordinates for autonomous vehicles. In *Proceedings of the 20th International Conference on Intelligent Transportation Systems* 1–7 (IEEE, 2017).
67. Schürmann, B. et al. Ensuring drivability of planned motions using formal methods. In *Proceedings of the 20th International Conference on Intelligent Transportation Systems* 1661–1668 (IEEE, 2017).

Acknowledgements

We thank S. Kaster for his support in implementing the prediction and S. Steyer for providing the object detection and tracking algorithms. We also thank C. Schürmann for the voice-overs in the Supplementary Videos. This work was partially supported by the BMW Group within the CAR@TUM project, the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF, and the German Research Foundation (DFG) under grants AL 1185/4-2 and AL 1185/3-2.

Author contributions

C.P., S.M. and M.K. developed the verification technique during replanning. M.K. developed the concept and algorithms for the set-based prediction. C.P. and S.M. developed the concept and algorithms for the drivable area computation, driving corridor identification and fail-safe trajectory planning. M.A. developed the main concept of online verification by integrating set-based prediction and fail-safe trajectory generation. He also developed the underlying algorithms for reachability analysis and led the research project. C.P., S.M. and M.K. designed and conducted the experiments and collected the data. The Article and the Supplementary Information were written by C.P., S.M. and M.K.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s42256-020-0225-y.

Correspondence and requests for materials should be addressed to C.P., S.M. or M.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

natureresearch

https://doi.org/10.1038/s42256-020-0225-y

Supplementary information

Using online verification to prevent autonomous vehicles from causing accidents

In the format provided by the authors and unedited

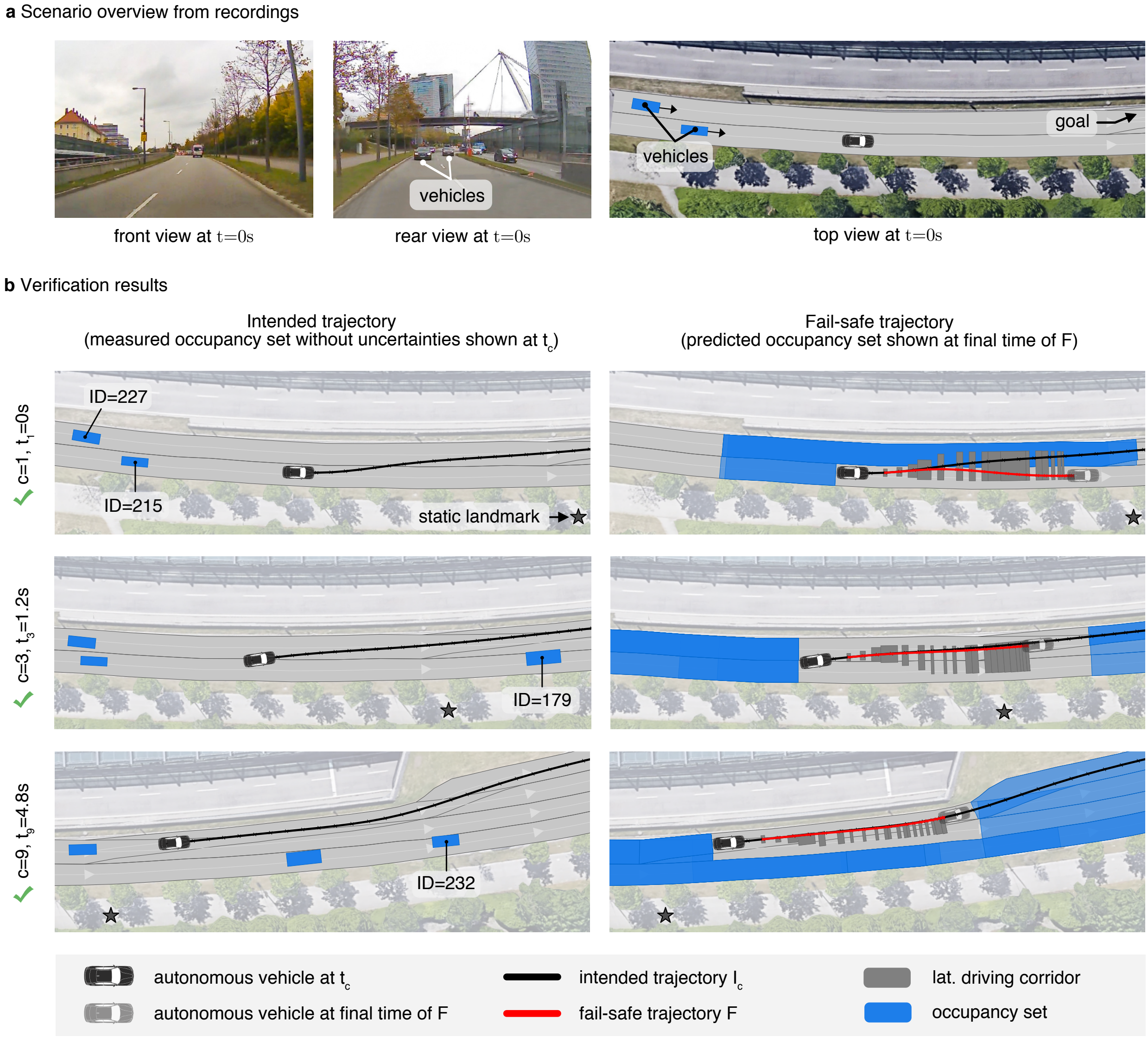
2 Supplementary results

2.1 Scenario III: Ensuring safe lane changes

Dense traffic in urban areas, which is expected to further increase with the rise of autonomous vehicles [89], requires vehicles to manoeuvre in tight spaces. When following a single lane, autonomous vehicles can simply brake if a preceding vehicle performs emergency braking. However, changing lanes is more challenging, as rapidly approaching vehicles from behind must be considered. In addition, if autonomous vehicles are too considerate, they are likely to impede traffic and are not able to find a gap large enough to perform a lane change. Our set-based approach safeguards arbitrary lane changes by planning fail-safe trajectories. As demonstrated in this scenario, autonomous vehicles can perform lane changes using our verification technique without being overly conservative.

In the beginning of Scenario III (see Supplementary Fig. 7a), the autonomous vehicle plans to change lanes and merge in front of the vehicle with ID 227 that approaches from behind (see $t_1 = 0\,\mathrm{s}$ in Supplementary Fig. 7b). However, because this vehicle does not have to maintain a safe distance to the autonomous vehicle, it can accelerate until its velocity reaches the speed limit. A lane change by the autonomous vehicle would then cause a collision. Thus, the fail-safe trajectory swerves back to the initial right lane. This fail-safe trajectory is legally safe, as the vehicle with ID 215 that is currently driving in the same lane behind the autonomous vehicle must maintain a safe distance (i.e., its occupancy set ends just behind the autonomous vehicle for $t_1 = 0\,\mathrm{s}$ in Supplementary Fig. 7b).

In the next verification cycle, the autonomous vehicle determines that the distance to the vehicle with ID 227 in the left lane is sufficiently large, and it can thus safely complete the lane change by executing the intended trajectory (see $t_3 = 1.2\,\mathrm{s}$ in Supplementary Fig. 7b). Thereafter, the autonomous vehicle continues in the left lane, while our fail-safe prediction always



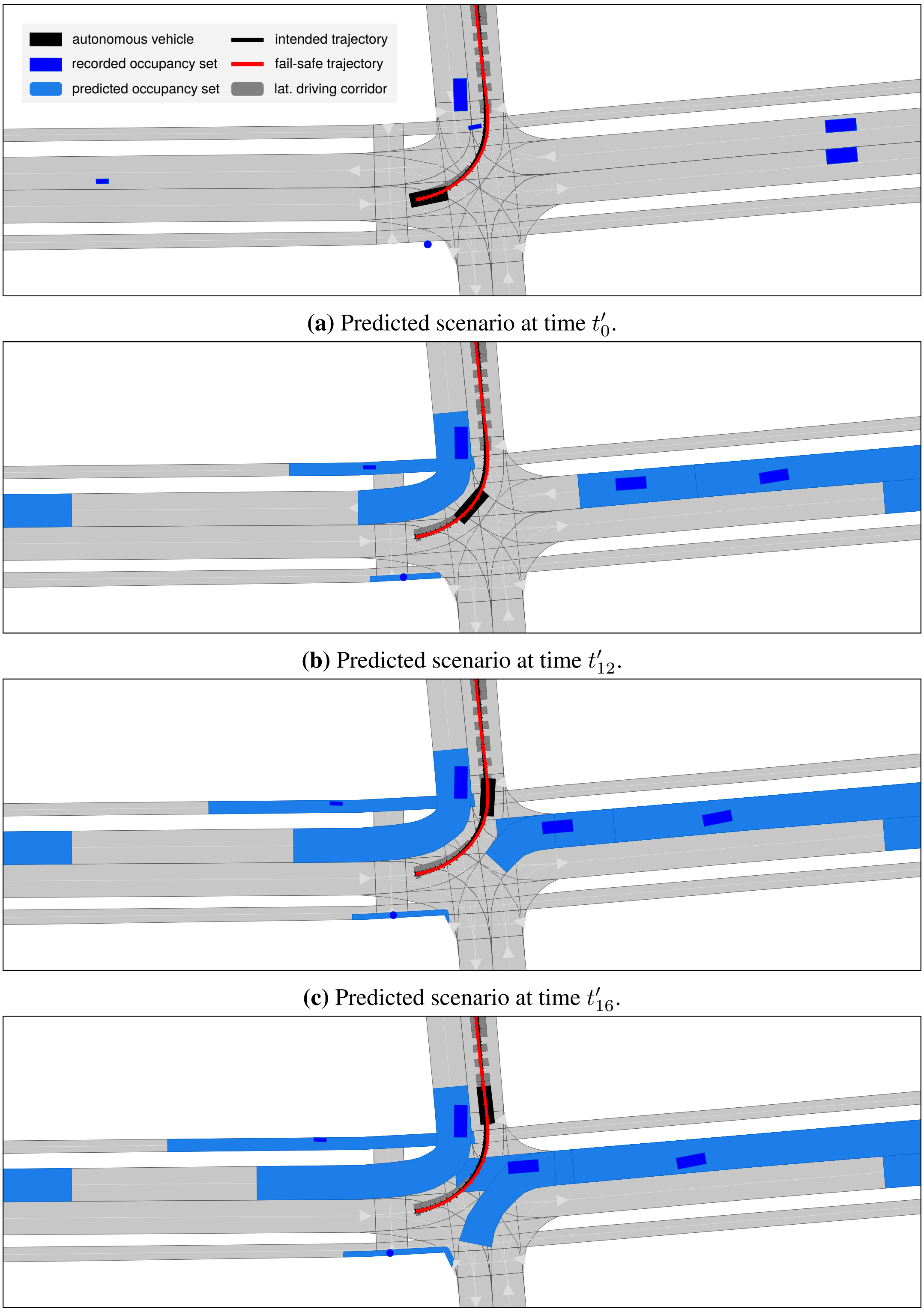
Supplementary Figure 7: Results of Scenario III (lane change) (a) Camera images and top view of scenario. **(b)** Verification results of selected verification cycles c. The intended trajectory I_c is only shown if it is successfully verified. Satellite Images ©Google, GeoBasis-DE/BKG.

anticipates possible lane changes by leading vehicles in the right lane (the occupancies of vehicles with IDs 179 and 232 in front of the autonomous vehicle in Supplementary Fig. 7b).

Note that throughout this scenario, the autonomous vehicle always has a fail-safe trajectory available; however, unlike Scenarios I and II, it never has to execute it, since all intended trajectories are successfully verified. The verification results of this scenario are also illustrated in Supplementary Video 1.

2.2 Detailed results of selected verification cycles

The following Supplementary Figures illustrate the verification results obtained during selected cycles c of Scenarios I–III. Supplementary Fig. 8 shows the intermediate results for verification cycle $c = 10$ of Scenario I, in which the autonomous vehicle is able to turn left at the intersection although the solution space for planning is small (see t'_{16}). In Supplementary Fig. 9, we highlight the occupancy prediction for pedestrians for the different time steps in Scenario II. Supplementary Fig. 10 shows how the obtained fail-safe trajectory in cycle $c = 1$ of Scenario III smoothly aborts the planned lane change if vehicles on the left adjacent lane would accelerate. Each figure shows the predicted occupancy sets of obstacles at different times t'_k. The autonomous vehicle is depicted with reference to I_c^{safe} for $t'_k < t'_{k_F}$ and with reference to F_c for $t'_k \geq t'_{k_F}$. A comprehensive description of Scenarios I and II can be found in the article and of Scenario III in Sec. 2.1 of the Supplementary Information. Other time steps of the verification results can be visualized using our software provided in the Supplementary Data File.



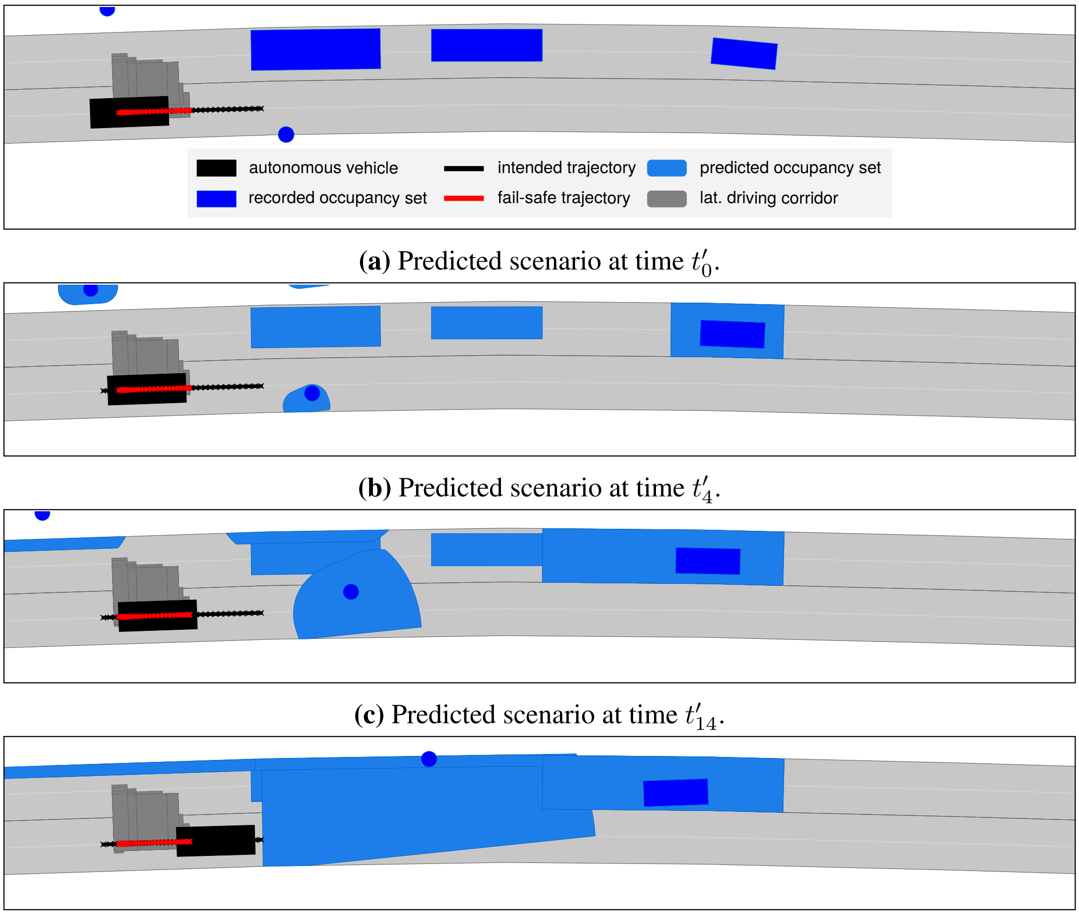
(a) Predicted scenario at time t'_0.

(b) Predicted scenario at time t'_{12}.

(c) Predicted scenario at time t'_{16}.

(d) Predicted scenario at time t'_{18}.

Supplementary Figure 8: Detailed verification results of Scenario I Visualized solution is obtained during verification cycle $c = 10$.



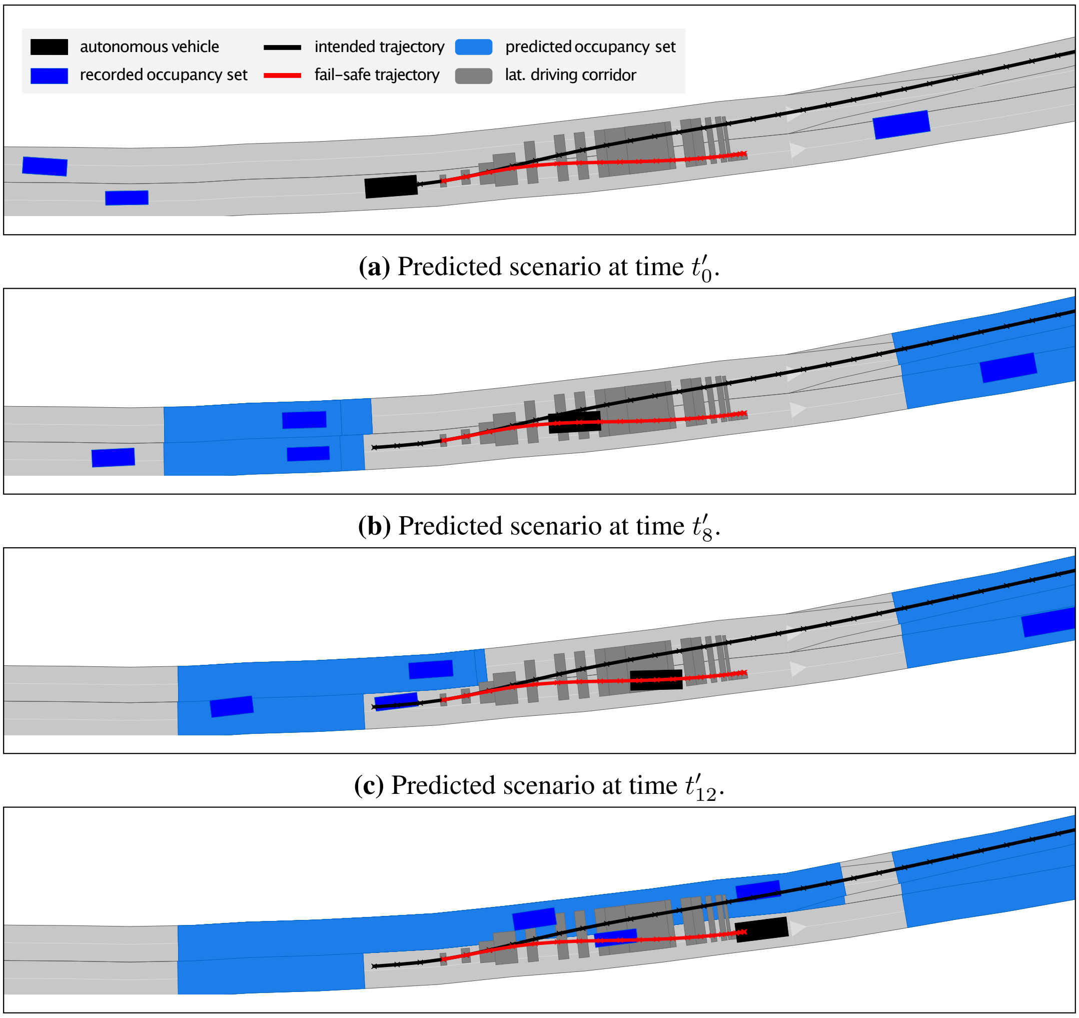
(a) Predicted scenario at time t'_0.

(b) Predicted scenario at time t'_4.

(c) Predicted scenario at time t'_{14}.

(d) Predicted scenario at time t'_{33}.

Supplementary Figure 9: Detailed verification results of Scenario II Visualized solution is obtained during verification cycle $c = 5$.



(a) Predicted scenario at time t'_0.

(b) Predicted scenario at time t'_8.

(c) Predicted scenario at time t'_{12}.

(d) Predicted scenario at time t'_{23}.

Supplementary Figure 10: Detailed verification results of Scenario III Visualized solution is obtained during verification cycle $c = 1$.

4.3 CDC 2017: Enhancing Motion Safety by Identifying Safety-critical Passageways [64]

Summary Formal verification methods, such as using reachability analysis, are capable of guaranteeing safety for a given model and given assumptions, as demonstrated in the previous Section 4.2. However, certain assumptions can be violated by dynamic obstacles during the execution of the verified motion plan, exposing the ego vehicle to potential collisions. Note that the constraint management detailed in Chapter 3 accounts for these violations only in the subsequent planning cycle. Yet, we want to compensate for the invalidated verification in the current planning cycle, i. e., solve Problem statement 2 at t_0 (instead of at $t_1 > t_0$). Therefore, this section 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 inherently safe sections and inherently safety-critical passageways (SCP). For the safe sections, we are able to provide safety guarantees for an infinite time horizon. In contrast, within safety-critical passageways, the ego vehicle is exposed to potential collisions if obstacles violate assumption used for the verification. Thus, we present a method to minimize such safety-critical passageways prior to execution by assigning costs to it and integrating the cost function into the optimization of the trajectory planner. In fact, if we obtain a trajectory that does not contain a safety-critical passageway, we can guarantee safety for an infinite time horizon.

The Point of No Return and the Point of Guaranteed Arrival of a trajectory cannot be determined exactly, but we present an algorithm that computes an under-approximation and an over-approximation of these points. Optionally, the online computation time can be reduced by precomputing both points for different tasks offline.

Numerical examples of overtaking maneuvers highlight the safety benefits of the approach. By employing the information of the safety-critical passageway, trajectories can be chosen that are most robust against violated assumption. In addition, evasive trajectories that react to violated assumptions can be obtained significantly faster.

Contributions of M. K. M. K. developed the notion of the safe sets, of the different assumptions, and of the PNR and PGA (all together with C. P. and M. A.). M. K. developed the computation of the PNR and PGA and their implications for the safety of motion plans (all together with C. P.). M. K. designed, conducted, and evaluated the experiments (together with C. P.). M. K. wrote the article (together with C. P.).

Conference paper ©2017 IEEE. Reprinted, with permission, from Christian Pek, Markus Koschi, Moritz Werling, and Matthias Althoff, Enhancing Motion Safety by Identifying Safety-critical Passageways, in Proc. of the IEEE 56th Annual Conference on Decision and Control.

2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

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 $\mathcal{U}$:

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

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 $\operatorname{occ}(x)$ relates the state vector x to the set of occupied points in Euclidean space as

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

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

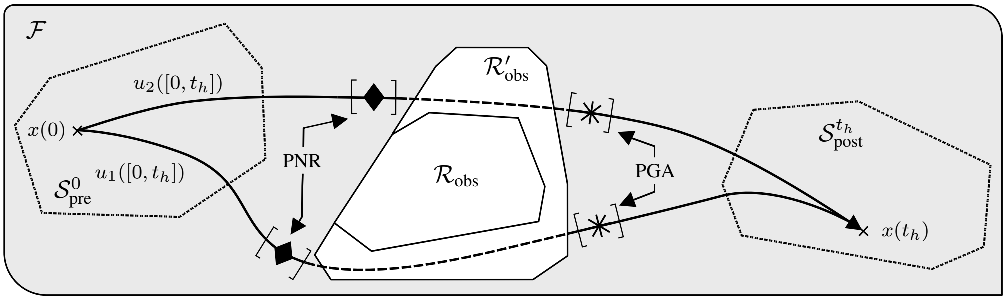

$\mathcal{F}$
$u_2([0,t_h])$
$x(0)$
$\mathcal{S}^0_{\text{pre}}$
$u_1([0,t_h])$
PNR
$\mathcal{R}'_{\text{obs}}$
$\mathcal{R}_{\text{obs}}$
PGA
$\mathcal{S}^{t_h}_{\text{post}}$
$x(t_h)$

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 $u([0,t_h])$:

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

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, $\underline{t}_{\mathrm{PNR}}$ 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*

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

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 $\mathcal{S}_{\mathrm{pre}}$ 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. $\mathcal{R}'_{\mathrm{obs}}$ 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 $\mathcal{S}_{\mathrm{pre}}$ or $\mathcal{S}_{\mathrm{post}}$, 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 $\mathcal{S}_{\mathrm{pre}}$ and $\mathcal{S}_{\mathrm{post}}$ 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 $\mathcal{S}_{\mathrm{pre}} \cap \mathcal{S}_{\mathrm{post}} = \emptyset$, 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 $t_0 = 0\,\mathrm{s}$. 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 $\mathcal{A}_{\mathrm{valid}}$ and initially assuming $\mathcal{A}_{\mathrm{viol}} = \emptyset$, we obtain the

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

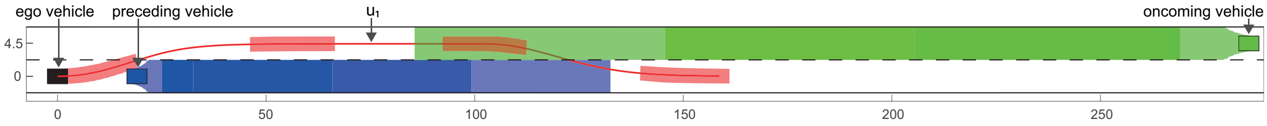


Fig. 2. Since the occupancy of the ego vehicle (red) along its trajectory $u_1([0, t_h])$ does not intersect with the occupancies of other vehicles (blue and green) in any time interval, the motion plan is verified as collision-free. Note that for the sake of clarity, the occupancy sets are only shown 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 $\Delta t = 0.1\,\text{s}$ up to the time horizon $t_h = 10\,\text{s}$ (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 $\overline{t}_{\text{PNR}} = 0.7\,\text{s}$ and $\underline{t}_{\text{PNR}} = 0.6\,\text{s}$ 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	$(x, y, v)^T_{\text{ego}} = (0\,\text{m}, 0\,\text{m}, 16.7\,\text{m/s})^T$
Preceding vehicle	$(x, y, v)^T_{\text{pre}} = (19.0\,\text{m}, 0\,\text{m}, 11.1\,\text{m/s})^T$
Oncoming vehicle	$(x, y, v)^T_{\text{onc}} = (285.6\,\text{m}, 4.5\,\text{m}, 16.7\,\text{m/s})^T$
Speed limit	$v_{\text{lim}} = 16.7\,\text{m/s}$
Maximum velocity	$v_{\max} = 1.2 v_{\text{lim}} = 20\,\text{m/s}$
Switching velocity	$v_S = 5.0\,\text{m/s}$
Maximum acceleration	$\lvert a_{\max} \rvert = 8.0\,\text{m/s}^2$
Lateral distance between the lanes	$\Delta y = 4.5\,\text{m}$
Time horizon	$t_h = 10.0\,\text{s}$
Time step size	$\Delta t = 0.1\,\text{s}$

b) PGA interval: Using SPOT, we obtain $\underline{t}_{\text{PGA}} = 8.4\,\text{s}$ for the lower bound. The sampling method results in $\overline{t}_{\text{PGA}} = 8.5\,\text{s}$ 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 $t = 4.5\,\text{s}$, 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.

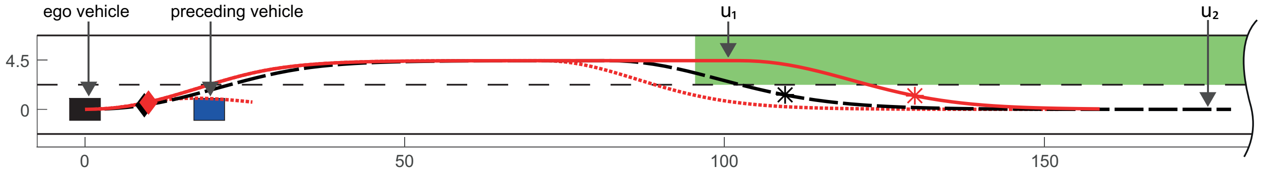


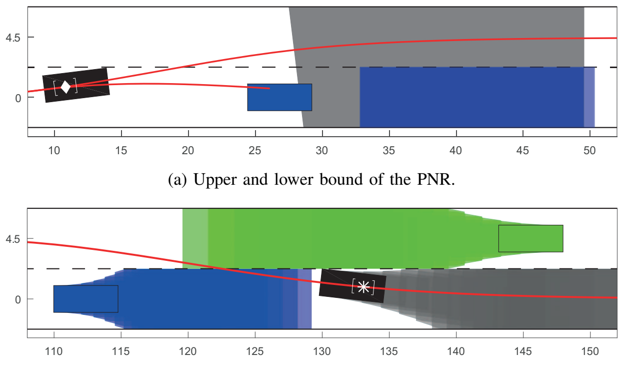
Fig. 3. Trajectory $u_1([0,t_h])$ with max. velocity of $16.7\,\mathrm{m/s}$ is not collision-free if the oncoming vehicle violates $\mathcal{A}_{\mathrm{viol}}$. Two evasive trajectories, denoted by dotted lines, branch off at the PNR and at $t = 4.5\,\mathrm{s}$ within the SCP. An alternative collision-free 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 $c = (\underline{t}_{\mathrm{PGA}} - \bar{t}_{\mathrm{PNR}})$. 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.



(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.

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.

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.

4.4 ITSC 2019: Computationally Efficient Safety Falsification of Adaptive Cruise Control Systems [69]

Summary In the previous sections, we ensured the safety of motion plans using reachability analysis. In this section, we tackle the safety problem from the other side by using extensive testing in terms of falsification. Falsification aims to disprove the safety of systems by providing counter-examples that lead to a violation of safety properties. We present two novel falsification methods that reveal safety gaps in adaptive cruise control (ACC) systems of autonomous vehicles. Our methods use rapidly-exploring random trees to generate motions for a leading vehicle such that the ACC under test causes a rear-end collision, which solves Problem statement 5. To speed up the search, we do not try to directly find the only few possible collision states, but instead define unsafe states that eventually result in a collision and search for those. Since the set of unsafe states is still very small compared to the set of safe states, our second falsification method starts at unsafe states and searches backward in time, which makes finding a counter-example much more likely.

We demonstrate the benefits of our methods by successfully falsifying the safety of contemporary ACC systems and comparing the results to that of existing testing approaches. By integrating unsafe states in the standard forward search approach, we already achieve an improvement in the required computation time of up to 8 times. With this approach, however, we were not able to falsify all ACC systems in a reasonable time period. In contrast, our backward search approach is able to falsify even a sophisticated ACC system with collision avoidance in every test run. By starting the search from a set of unsafe states, our backward search algorithm is able to find counter-examples 300 times faster than standard approaches. The backward search method can also help to reveal implementation errors in ACC systems based on formal methods.

Overall, our proposed methods allow developers to detect safety gaps in their system with minimal effort and already at early stages of their work. The obtained collision scenarios can be directly used to improve the design of their system. While the forward search can be used to generate diverse counter-examples, the backward search aims to quickly find any counter-example.

Contributions of M. K. M. K. developed the definition of the safe and unsafe distance and their corresponding set of states (all together with C. P. and S. M.). M. K. developed the modification of rapidly-exploring random trees, the forward search, and the backward search (all together with C. P. and S. M.). M. K. designed and evaluated the experiments (together with C. P. and S. M.). M. K. wrote the article except Sec. I (Sec. IV together with S. M.).

Attachments The video attachment of this publication is available at go.tum.de/500310.

2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Computationally Efficient Safety Falsification of Adaptive Cruise Control Systems

Markus Koschi*, Christian Pek*, Sebastian Maierhofer*, and Matthias Althoff

Abstract— **Falsification aims to disprove the safety of systems by providing counter-examples that lead to a violation of safety properties. In this work, we present two novel falsification methods to reveal safety flaws in adaptive cruise control (ACC) systems of automated vehicles. Our methods use rapidly-exploring random trees to generate motions for a leading vehicle such that the ACC under test causes a rear-end collision. By considering unsafe states and searching backward in time, we are able to drastically improve computation times and falsify even sophisticated ACC systems. The obtained collision scenarios reveal safety flaws of the ACC under test and can be directly used to improve the system's design. We demonstrate the benefits of our methods by successfully falsifying the safety of state-of-the-art ACC systems and comparing the results to that of existing approaches.**

I. Introduction

Safety is a mandatory requirement for the ever increasing automation of vehicles. However, ensuring safety is a challenging task; even in supposedly simple scenarios like vehicle following (cf. adaptive cruise control (ACC) system in Fig. 1), the variety of stop-and-go behaviors of other vehicles may impose safety-critical situations.

Verifying that motion planning algorithms ensure certain standards is often done by testing the system in a multitude of simulations using large databases of test cases and traffic scenarios [1]–[3]. However, simulations have the significant disadvantage that they may miss testing certain scenarios that inevitably lead to unsafe situations. In contrast, formal verification approaches are able to provide strong safety guarantees by verifying that each action of the vehicle conforms to a formal specification [4]. Nevertheless, safety only holds if the used specification appropriately models the desired safety definition and no implementation mistakes have been made.

To reveal safety-critical flaws in a system, falsification approaches try to disprove the safety instead of proving it [5]. Falsification for motion planning aims to find motions that start in a safe state but eventually enter collision states (cf. Fig. 2). The obtained motions serve as counter-examples and can be used to revise the system's design. Falsification should be an integral part in the development of automated vehicles; however, falsification approaches for automated vehicle functions still have a huge potential in terms of computational efficiency and applicability [6].

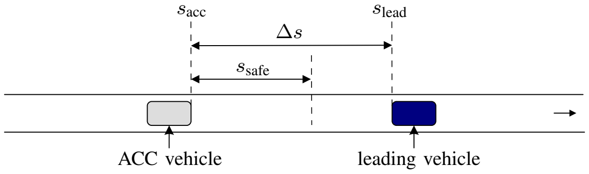


Fig. 1: To follow a leading vehicle, ACC systems adjust the velocity of the ACC-equipped vehicle so that the headway $\Delta s = s_{\text{lead}} - s_{\text{acc}}$ is larger than a safe distance s_{safe}.

A. Related work

In the following paragraphs, we *a)* provide a brief overview of safety mechanisms in state-of-the-art ACC systems and *b)* review existing techniques for generating safety-critical scenarios for automated vehicles.

a) Adaptive cruise control systems: ACC systems automatically adjust the velocity of the controlled vehicle to maintain a certain headway to a leading vehicle [7], [8]. Reviews about major ACC developments can be found in [9], [10]. Proportional integral ACCs (PI-ACCs) use PI controllers to adjust the headway and are still widely used because of their simplicity [11], [12]. Their safety relies mainly on the chosen gains to react to sudden changes in the behavior of the leading vehicle. The Intelligent Driver Model ACC (IDM-ACC) implements a more complex control scheme by switching between different driving modes [13]. As a result, the IDM-ACC can switch between comfortable or rather safe parameterizations. Nevertheless, these ACC systems do not incorporate dedicated collision avoidance mechanisms.

Collision avoidance ACCs (CA-ACCs) explicitly consider collision avoidance by quickly adjusting their response behavior if the desired headway cannot be maintained [14]. Recently proposed ACCs make use of formal methods (FM-ACC), e.g., set invariance theory or formalized traffic rules, to provide safety guarantees during operation [15]–[18].

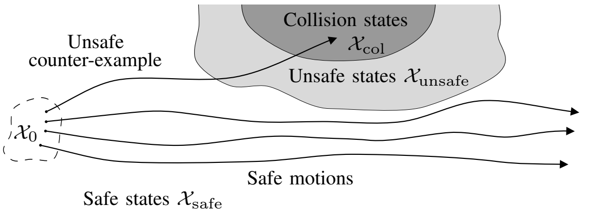

Collision states
$\mathcal{X}_{\text{col}}$
Unsafe counter-example
Unsafe states $\mathcal{X}_{\text{unsafe}}$
$\mathcal{X}_0$
Safe motions
Safe states $\mathcal{X}_{\text{safe}}$

Fig. 2: Falsification of motion planning aims to find counter-examples that transition the system under test from an initial safe state $x_0 \in \mathcal{X}_0 \subset \mathcal{X}_{\text{safe}}$ to a collision state $x \in \mathcal{X}_{\text{col}}$.

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

All authors are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany.
markus.koschi@tum.de, christian.pek@tum.de, sebastian.maierhofer@tum.de, althoff@tum.de

978-1-5386-7023-1/19/$31.00 ©2019 IEEE

b) Generating safety-critical scenarios: To test ACC systems, some approaches synthesize safety-critical scenarios [5], [19]–[22], e.g., by making use of Monte Carlo simulation (MCS) to automatically generate a variety of random scenarios. Although MCS approaches quickly generate various scenarios, they are not designed to specifically find scenarios leading to collisions.

The authors of [23] propose a systematic approach to test collision avoidance systems by primarily simulating scenarios in which leading vehicles suddenly perform emergency braking maneuvers. More sophisticated methods make use of reachability analysis, neural networks, performance metrics, or evolutionary algorithms to automatically generate safety-critical scenarios for fully automated vehicles [24]–[29]. Rapidly-exploring random trees (RRTs) [30] are used in [31]–[33] to falsify the safety of a given system, since RRTs are well suited to efficiently explore large search spaces.

B. Contributions

This paper proposes two approaches based on RRTs to falsify the safety of ACC systems. Since the aforementioned falsification methods are often computationally expensive and do not exploit domain knowledge to efficiently generate counter-examples, our contributions tackle these issues by:

1) drastically improving the performance of forward searches by integrating unsafe states, which eventually lead to collisions and are much easier to reach than collision states (cf. Fig. 2);
2) presenting a novel falsification approach that employs a backward search scheme and can successfully falsify sophisticated ACC systems in significantly less time than that of forward searches; and
3) demonstrating that our approaches successfully falsify state-of-the-art ACC systems from the literature in reasonable time and outperform classical forward search and MCS.

The remainder of this paper is organized as follows. Sec. II introduces necessary mathematical definitions and the problem statement. Next, our falsification algorithms to generate safety-critical situations are presented in Sec. III. In Sec. IV, the proposed approaches are used to falsify the safety of state-of-the-art ACC systems in numerical experiments, and the results are compared to that of Monte Carlo falsification. Conclusions are presented in Sec. V.

II. Definitions and Problem Statement

A. Vehicle configuration

Let us introduce $\mathcal{X} \subset \mathbb{R}^2$ as the set of feasible states x of a vehicle. The state vector $x = [s, v]^T$ consists of the position s and the velocity v, each in the longitudinal direction. Acceleration and jerk are denoted by a and j, respectively. We assume discrete-time systems with a time step size of $\Delta t > 0$. We further introduce $\mathcal{U} \subset \mathbb{R}$ as the set of admissible control inputs $u = a$ of the state transition function f_{motion}, which describes the longitudinal dynamics of a vehicle:

$$\underbrace{\begin{bmatrix} s(t_{k+1}) \\ v(t_{k+1}) \end{bmatrix}}_{x(t_{k+1})} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t_k) \\ v(t_k) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} u,}_{f_{\text{motion}}(x(t_k),u)} \tag{1}$$

with bounded velocity, acceleration, and jerk: $0 \leq v \leq v_{\max}$, $a_{\min} \leq u \leq a_{\max}$, $j_{\min} \leq j \leq j_{\max}$, where $a_{\min}, j_{\min} \in \mathbb{R}_{<0}$ and $j(t_k) = {}^{(u(t_k)-u(t_{k-1}))}/_{\Delta t}$. We adhere to the notation $x\big([t_0, t_n]\big)$ to describe a trajectory of states $x(t_i) \in \mathcal{X}$ for $t_i \in \{t_0, t_1, \ldots, t_n\}$ that satisfy (1) and its constraints, and we use $u\big([t_0, t_n]\big)$ analogously to describe an input trajectory.

As shown in Fig. 1, we consider situations in which an ACC-equipped vehicle is following another vehicle. The variables of the leading and ACC vehicle are denoted by the subscript $\square_{\text{lead}}$ and $\square_{\text{acc}}$, respectively. By defining the reference point for the position of the leading vehicle at its rear end and of the ACC vehicle at its front, the relative distance between both vehicles is $\Delta s := s_{\text{lead}} - s_{\text{acc}}$ (cf. Fig. 1). Their relative velocity is defined as $\Delta v := v_{\text{lead}} - v_{\text{acc}}$.

We treat the ACC control law under test as a black box:

$$u_{\text{acc}} = f_{\text{ACC}}\big(x_{\text{acc}}(t_k), x_{\text{lead}}(t_k), \delta\big), \tag{2}$$

where f_{ACC} is unknown and δ is the reaction delay of the system, e.g., processing time of sensors and actuator delays. For brevity, we combine f_{ACC} and f_{motion} in the function $f_{\text{ACC-motion}}\big(x_{\text{acc}}(t_k), x_{\text{lead}}(t_k), \delta\big)$, which returns $x_{\text{acc}}(t_{k+1})$.

B. Safety definition

To define safe states, we use the established safety definition that the ACC vehicle must remain collision-free at all times [34]. This must hold even if the leading vehicle suddenly performs emergency braking, i.e., $u_{\text{lead}}^{\text{brake}}(t_i) := \max\big(a_{\text{lead}}(t_{i-1}) + j_{\text{min,lead}}\Delta t, a_{\text{min,lead}}\big)$. In response, an ACC vehicle that conforms with [34] will fully brake; during its reaction delay, we allow arbitrary acceleration, which can be over-approximated by full acceleration. Thus, for our safety analysis, we assume that the ACC vehicle applies the control inputs

$$u_{\text{acc}}^{\text{brake}}(t_i) := \begin{cases} \min\big(a_{\text{acc}}(t_{i-1}) + j_{\text{max,acc}}\Delta t, & \\ a_{\text{max,acc}}\big), & t_i - t_k < \delta, \\ \max\big(a_{\text{acc}}(t_{i-1}) + j_{\text{min,acc}}\Delta t, & \\ a_{\text{min,acc}}\big), & \text{otherwise,} \end{cases}$$

where t_k is the point in time at which the leading vehicle starts emergency braking. Let $t_{\text{acc}}^{\text{stop}}$ denote the point in time at which the ACC vehicle is at a standstill.

Definition 1 (Safe distance)
The ACC vehicle can definitely avoid a rear-end collision, if it maintains at least the minimal safe distance s_{safe} to the leading vehicle:

$$s_{safe}(t_k) := \inf\Big(\big\{\Delta s(t_k) \,\big|\, \forall t_i \in \{t_k, t_{k+1}, \ldots, t_{acc}^{stop}\} : \Delta s(t_i) > 0\big\}\Big),$$

where $\Delta s(t_i)$ is obtained by simulating the leading and ACC vehicle according to (1) *from $v(t_k)$ with $u_{lead}^{brake}\big([t_k, t_{acc}^{stop}]\big)$ and $u_{acc}^{brake}\big([t_k, t_{acc}^{stop}]\big)$, respectively.*

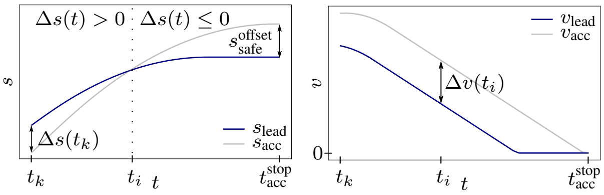
Our definition is based on [34] and [35] with the following extensions: in contrast to [34], we allow for a reaction delay of the ACC vehicle; in contrast to [35], we do not assume that the ACC vehicle maintains constant velocity during the reaction delay, but we allow for arbitrary acceleration; and in contrast to both, we consider limited jerk to allow for more realistic braking profiles.

Definition 2 (Unsafe distance)
The ACC vehicle definitely cannot avoid a rear-end collision with impact velocity of at least v_{col} (which is a parameter ≥ 0), if the leading vehicle performs emergency braking at t_k and if the ACC vehicle maintains less than or equal the maximum unsafe distance s_{unsafe} to the leading vehicle:

$$\begin{aligned} s_{unsafe}(t_k) := \sup\big(\big\{\Delta s(t_k)\,\big|\,\exists t_i \in \{t_{k+1}, t_{k+2}, \dots, t_{acc}^{stop}\} :\\ \Delta s(t_{i-1}) > 0 \wedge \Delta s(t_i) \leq 0 \wedge |\Delta v(t_i)| \geq v_{col}\big\}\big), \end{aligned}$$

where $\Delta s(t_{i-1})$, $\Delta s(t_i)$, and $\Delta v(t_i)$ are obtained by simulation as in Def. 1 except that we set $\delta = 0$ so that the acceleration during the reaction delay is under-approximated.

Algorithmically, Def. 1 and 2 can be evaluated by simulating both vehicles from their given current states at t_k with $u_{\text{lead}}^{\text{brake}}\big([t_k, t_{\text{acc}}^{\text{stop}}]\big)$ and $u_{\text{acc}}^{\text{brake}}\big([t_k, t_{\text{acc}}^{\text{stop}}]\big)$. The safe distance is obtained by adding $\Delta s(t_k)$ to the minimal required offset of the relative position so that both position profiles do not intersect (cf. $s_{\text{safe}}^{\text{offset}}$ in Fig. 3a). The unsafe distance is obtained by adding $\Delta s(t_k)$ to the maximal possible offset of the relative position so that both position profiles intersect and that the absolute value of the relative velocity at this point in time is at least v_{col} (cf. $\Delta v(t_i)$ in Fig. 3b).



(a) Simulated position profiles.

(b) Simulated velocity profiles.

Fig. 3: State plots of the leading and ACC vehicle to determine the safe and unsafe distance.

Definition 3 (Sets of safe, unsafe, and collision states)
Using Def. 1 and 2, we define the set of all safe states of the ACC vehicle at time t_k as

$$\mathcal{X}_{safe}(t_k) := \big\{x_{acc}(t_k) \in \mathcal{X}\,\big|\,\Delta s(t_k) \geq s_{safe}(t_k)\big\},$$

the set of unsafe states as

$$\mathcal{X}_{unsafe}(t_k) := \big\{x_{acc}(t_k) \in \mathcal{X}\,\big|\,\Delta s(t_k) \leq s_{unsafe}(t_k)\big\},$$

and the set of collision states as

$$\mathcal{X}_{col}(t_k) := \big\{x_{acc}(t_k) \in \mathcal{X}\,\big|\,\Delta s(t_k) \leq 0 \wedge |\Delta v(t_k)| \geq v_{col}\big\}.$$

Note that $\mathcal{X} = \mathcal{X}_{\text{safe}}(t_k) \cup \mathcal{X}_{\text{unsafe}}(t_k)$ only holds if $\delta = 0$ and $v_{\text{col}} = 0$.

We use Def. 3 in our approach to detect if we have already generated an unsafe situation for the ACC system. Therefore, we model the search space of our RRT by combining the state spaces of both vehicles. Thus, a node $z(t_k)$ of the search tree $\mathcal{T}$ is defined as state tuple: $z(t_k) := \big(x_{\text{acc}}(t_k), x_{\text{lead}}(t_k)\big)$. We denote a node as safe if $x_{\text{acc}}(t_k) \in \mathcal{X}_{\text{safe}}(t_k)$, as unsafe if $x_{\text{acc}}(t_k) \in \mathcal{X}_{\text{unsafe}}(t_k)$, and as colliding if $x_{\text{acc}}(t_k) \in \mathcal{X}_{\text{col}}(t_k)$.

C. Problem statement

In order to falsify an ACC system (cf. Fig. 2), we aim to find a time series of inputs for the leading vehicle $u_{\text{lead}}(t_i)$, $t_i \in \{t_0, t_1, \dots, t_{\text{col}}\}$, so that when starting in a safe state $x_{\text{acc}}(t_0) \in \mathcal{X}_{\text{safe}}(t_0)$, the ACC vehicle will eventually collide with the leading vehicle at $t_{\text{col}} > t_0 : x_{\text{acc}}(t_{\text{col}}) \in \mathcal{X}_{\text{col}}(t_{\text{col}})$.

III. Falsification Approaches

To quickly find counter-examples bridging the sets $\mathcal{X}_{\text{safe}}(t_k)$ and $\mathcal{X}_{\text{unsafe}}(t_k)$, we efficiently explore the search space using RRTs (see Sec. III-A). We have developed two methods to build the RRT: 1) We search forward in time by creating random behaviors of the leading vehicle and by evaluating the reactions of the ACC vehicle (see Sec. III-B). However, this strategy may require many simulation runs if the set of unsafe states is very small, which is the case for many ACC systems (e.g., an ACC with advanced collision avoidance). 2) Thus, our second approach starts from unsafe states and searches backward in time (see Sec. III-C). As a result, we can find counter-examples in fewer simulation runs.

A. Rapidly-exploring random trees

RRTs are a popular approach for motion planning and have already been used for falsification (cf. Sec. I-A). The standard approach (e.g., [30], [36]) starts at an initial set of nodes and generates new nodes from time step t_k to t_j as follows. After drawing a random sample in the search space, the node $z_{\text{near}}(t_k)$ that is closest to the sample according to a distance measure is selected as its parent. Then, the optimal input u is applied to drive the system from the parent node as close as possible to the sample, resulting in the new node $z_{\text{new}}(t_j)$. This procedure can be repeated such that the same number of nodes, denoted by the parameter z_{num}, is generated for each time step. We have made modifications to this standard approach, which are introduced next and are combined with the standard approach in Alg. 1.

Our sampling process is performed in relative coordinates $\Delta z := [\Delta s, \Delta v]^T$, since only these are relevant for the criticality. To avoid generating behavior that mostly uses the minimal or maximal possible input, we do not sample in the complete search space, but restrict the sampling range depending on the states of already existing nodes. Therefore, we first compute the minimum and maximum differences, $\Delta z_{\min}(t_k)$ and $\Delta z_{\max}(t_k)$, between the states of all nodes at the current time step t_k. To favor an increase of the relative coordinates, we add a bias to obtain the sampling range $\mathcal{Z}(t_k) := [\Delta z_{\min}(t_k) - \Delta z_{\text{add}}^{\min}, \Delta z_{\max}(t_k) + \Delta z_{\text{add}}^{\max}]$ (cf. line 1 of Alg. 1).

Algorithm 1 EXPLORE($\mathcal{T}$, t_k, t_j, $\mathcal{X}_{\text{acc}}(t_j)$)

```
Input: search tree $\mathcal{T}$, current time $t_k$, desired time $t_j$, set of states $\mathcal{X}_{\text{acc}}(t_j)$
Output: generated node $z(t_j)$
1: $\mathcal{Z}(t_k) \leftarrow \mathcal{T}$.GETSAMPLINGRANGE($t_k$)
2: $\Delta z(t_j) \leftarrow$ DRAWSAMPLE($\mathcal{Z}(t_k)$)
3: $z_{\text{near}}(t_k) \leftarrow \mathcal{T}$.GETNEARESTNODE($\Delta z(t_j)$)
4: $x_{\text{acc}}(t_j) \leftarrow$ GETSUCCESSORSTATE($z_{\text{near}}(t_k)$, $\mathcal{X}_{\text{acc}}(t_j)$)
5: $u_{\text{lead}} \leftarrow$ CALCINPUT($z_{\text{near}}(t_k)$, $\Delta z(t_j) + x_{\text{acc}}(t_j)$)
6: $x_{\text{lead}}(t_j) \leftarrow f_{\text{MOTION}}(z_{\text{near}}(t_k), u_{\text{lead}})$
7: return $z_{\text{new}}(t_j) \leftarrow \big(x_{\text{acc}}(t_j), x_{\text{lead}}(t_j)\big)$
```

As a distance measure for the selection of the nearest node $z_{\text{near}}(t_k)$ (cf. line 3), we use the L^2 norm with normalized state values. The normalization is done using the mean and standard deviation of all nodes $z(t_k)$ to avoid the preference of low numerical values in the position and velocity when selecting the closest node.

As an additional input, our algorithm requires the set of states of the ACC vehicle at the next time step, denoted by $\mathcal{X}_{\text{acc}}(t_j)$. From $\mathcal{X}_{\text{acc}}(t_j)$, we select the state $x_{\text{acc}}(t_j)$ that is the successor of the state of the ACC vehicle in $z_{\text{near}}(t_k)$ (cf. line 4). This allows us to compute the input u_{lead} that drives the leading vehicle as close as possible to the sampled configuration in global coordinates $\Delta z(t_j) + x_{\text{acc}}(t_j)$ (cf. line 5).

B. Forward search

First, we present the standard forward search approach known from the literature and then our novel extensions; the complete approach is summarized in Alg. 2.

We initialize the search tree $\mathcal{T}$ with z_{num} randomly selected safe nodes. In each time step t_k, we apply the ACC control law (2) and the motion equation (1) to all state tuples $\big(x_{\text{acc}}(t_k), x_{\text{lead}}(t_k)\big)$ such that we obtain the set of states of the ACC vehicle $\mathcal{X}_{\text{acc}}(t_{k+1})$ at the next time step (cf. line 6 to 8 of Alg. 2). Then, we randomly generate the future behavior of the leading vehicle by exploring the search space using the RRT of Alg. 1 (cf. line 10 of Alg. 2). Before advancing to the next time step, we can optionally remove childless nodes to reduce the memory consumption.

Our extension, illustrated in Fig. 4, allows us to find the state trajectory leading to a collision more quickly. At each time step during the forward search, we check if we have already generated an unsafe node (cf. line 4 of Alg. 2). Since an unsafe state implies that the ACC vehicle will eventually collide if the leading vehicle fully brakes (cf. Def. 2), we let the leading vehicle perform emergency braking (cf. line 14 of Alg. 2) until we have generated a collision node (cf. Fig. 4).

C. Backward search

Searching backward in time is especially difficult, since we cannot compute the inverse of the ACC control law, which would be required to simulate the ACC vehicle backward in time (unless the ACC system is flat [37]). Thus, we generate random inputs for the ACC vehicle to obtain states of the ACC vehicle at previous time steps. However, we need to ensure that the ACC system drives the ACC vehicle into unsafe states again, since we are only interested in generating behaviors that lead to a collision.

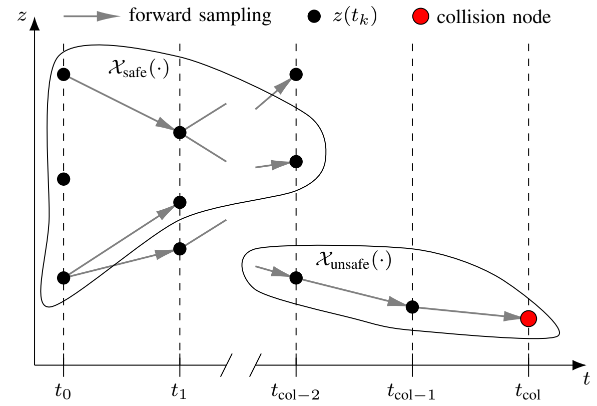


Fig. 4: Falsification by searching forward in time. Starting in safe states at t_0, the RRT is built until an unsafe node is found (i.e., $x_{\text{acc}}(t_k) \in \mathcal{X}_{\text{unsafe}}(t_k)$). Then, the leading vehicle performs emergency braking, which will result in a collision.

Algorithm 2 Falsification by forward search

```
1: $\mathcal{T}$.INITIALIZE($\mathcal{X}_{\text{safe}}(t_0)$, $z_{\text{num}}$)
2: $k \leftarrow 0$
3: while not $\mathcal{T}$.HASCOLLISIONNODE($t_k$) do
4:     $z_{\text{unsafe}}(t_k) \leftarrow \mathcal{T}$.FINDUNSAFENODE($t_k$)
5:     if $z_{\text{unsafe}}(t_k)$ is null then
6:         for all $z(t_k)$ in $\mathcal{T}$ do
7:             $\mathcal{X}_{\text{acc}}(t_{k+1})$.ADD($f_{\text{ACC-MOTION}}(z(t_k), \delta)$)
8:         end for
9:         while $\mathcal{T}$.NUMNODES($t_{k+1}$) $< z_{\text{num}}$ do
10:            $z_{\text{new}}(t_{k+1}) \leftarrow \mathcal{T}$.EXPLORE($t_k$, $t_{k+1}$, $\mathcal{X}_{\text{acc}}(t_{k+1})$)
11:            $\mathcal{T}$.ADDNEWNODE($z_{\text{new}}(t_{k+1})$)
12:        end while
13:    else
14:        $x_{\text{lead}}(t_{k+1}) \leftarrow f_{\text{MOTION}}(z_{\text{unsafe}}(t_k), u^{\text{brake}}_{\text{lead}}(t_k))$
15:        $x_{\text{acc}}(t_{k+1}) \leftarrow f_{\text{ACC-MOTION}}(z_{\text{unsafe}}(t_k), \delta)$
16:        $\mathcal{T}$.ADDNEWNODE($x_{\text{acc}}(t_{k+1}), x_{\text{lead}}(t_{k+1})$)
17:    end if
18:    $k \leftarrow k + 1$
19: end while
20: return $\mathcal{T}$.COLLISIONTRACE
```

Fig. 5 and Alg. 3 illustrate our solution. First, we initialize the search tree $\mathcal{T}$ at an arbitrary time t_h with z_{num} randomly selected unsafe nodes. The falsification successfully terminates if we have generated a safe node (with an optional larger headway distance); otherwise, we continue searching backward in time.

To obtain the states of the ACC vehicle at the next time step t_{k-1} (backward in time), we create random inputs for all $x_{\text{acc}}(t_k)$ in $\mathcal{T}$ (cf. lines 4 to 6 of Alg. 3). The new states for the leading vehicle at t_{k-1} are generated by the RRT of Alg. 1 as in the forward search (cf. line 8 of Alg. 3).

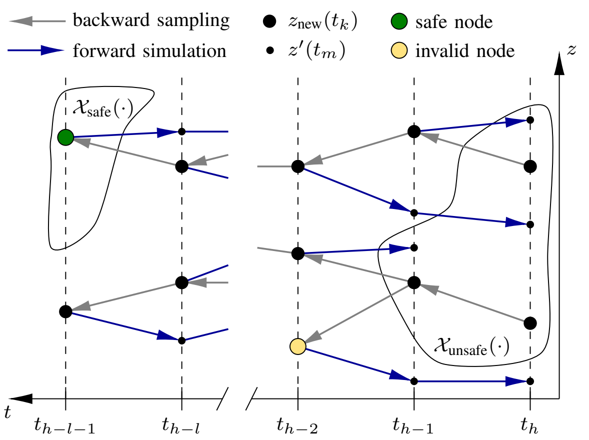


Fig. 5: Falsification by searching backward in time. Starting in unsafe states at t_h, the RRT is built by sampling backward in time until a safe node is found. Yet, a node $z_{\text{new}}(t_k)$ generated by the backward search is discarded as invalid, if applying the ACC control in a forward simulation does not result in a node $z'(t_m)$, $t_k \leq t_m \leq t_h$, in which $x_{\text{acc}}(t_m) \in \mathcal{X}_{\text{unsafe}}(t_m)$.

Algorithm 3 Falsification by backward search

```
1:  $\mathcal{T}$.INITIALIZE($\mathcal{X}_{\text{unsafe}}(t_h)$, $z_{\text{num}}$)
2:  $k \leftarrow h$
3:  while not $\mathcal{T}$.HASSAFENODE($t_k$) do
4:      for all $z(t_k)$ in $\mathcal{T}$ do
5:          $\mathcal{X}_{\text{acc}}(t_{k-1})$.ADD(RANDOMINPUT($z(t_k)$))
6:      end for
7:      while $\mathcal{T}$.NUMNODES($t_{k-1}$) < $z_{\text{num}}$ do
8:          $z_{\text{new}}(t_{k-1}) \leftarrow \mathcal{T}$.EXPLORE($t_k$, $t_{k-1}$, $\mathcal{X}_{\text{acc}}(t_{k-1})$)
9:          for $m \leftarrow k-1$; $m \leq h$; $m$++ do
10:             $z'(t_m) \leftarrow \mathcal{T}$.FORWARDSIM($z_{\text{new}}(t_{k-1})$, $t_m$, $\delta$)
11:             if ISUNSAFENODE($z'(t_m)$) then
12:                 $\mathcal{T}$.ADDNEWNODE($z_{\text{new}}(t_{k-1})$)
13:                 break
14:             end if
15:         end for
16:     end while
17:     $k \leftarrow k-1$
18: end while
19: return $\mathcal{T}$.COLLISIONTRACE
```

Next, we require to check whether the ACC system will still cause a collision when starting at the new generated node $z_{\text{new}}(t_{k-1})$ (cf. lines 9 to 15 of Alg. 3). Therefore, we iteratively simulate forward in time to obtain $z'(t_m)$ for $t_k \leq t_m \leq t_h$ by applying the ACC control law (2), which uses as input the states of the leading vehicle saved in $\mathcal{T}$ (cf. forward simulation in Fig. 5). Note that $z'(t_m)$ is generally different than $z_{\text{new}}(t_m)$. Only if $z_{\text{new}}(t_{k-1})$ itself or any $z'(t_m)$ is an unsafe node, the new node $z_{\text{new}}(t_{k-1})$ is added to the tree; otherwise, we discard the new node as invalid (cf. invalid node in Fig. 5). The final collision trace is obtained by continuing the forward simulation from the unsafe node with emergency braking of the leading vehicle until the guaranteed collision.

IV. Numerical experiments

We evaluate our forward and backward search by falsifying different ACC systems. Our implementation in Python is available at gitlab.lrz.de/tum-cps/safety-falsification-acc. All simulations were executed on a machine with an Intel Xeon Gold 6136 3.00 GHz processor and 128 GB of DDR4 2666 MHz memory. The safety-critical scenarios presented next are included in the CommonRoad benchmark suite[1] [3] and are visualized in the video attachment of this paper. Tab. I lists the parameters of the numerical experiments, where n_{forward} and n_{backward} are the user-defined maximum number of iterations of the forward and backward search, respectively.

A. Introduction to tested ACC systems

The four ACC control laws chosen for falsification vary in their safety integrity, as discussed in Sec. I-A. Their control laws are briefly introduced in this section, and their parameters are listed in Tab. II, where v_{des} denotes the desired velocity of the ACC vehicle, t_{des} the desired time gap between the leading and ACC vehicle, and Δs_{min} the desired minimum relative distance. We assume a reaction delay of 0 s to provide a best case situation for the ACC systems.

1) Proportional integral ACC (PI-ACC): The PI-ACC [11] uses a PI controller:

$$u = k_{\text{p}}\big(\Delta v + k_{\text{q}}\Delta s^{\text{PI}}_{\text{err}}\big) + k_{\text{i}}\frac{1}{\Delta t}\big(\Delta v + k_{\text{q}}\Delta s^{\text{PI}}_{\text{err}}\big), \qquad (3)$$

[1] commonroad.in.tum.de

TABLE I: User-defined parameters of the falsification.

Description	**Parameter with value**
Time step size	$\Delta t = 0.1\,\mathrm{s}$
Max. number of time steps	$n_{\text{forward}} = 12000$, $n_{\text{backward}} = 600$
Number of nodes	$z_{\text{num}} = 250$
Increase of sampling range	$\Delta z^{\min}_{\text{add}} = [0\,\mathrm{m},\ 0\,\mathrm{m/s}]^T$, $\Delta z^{\max}_{\text{add}} = [1.0\,\mathrm{m},\ 0.25\,\mathrm{m/s}]^T$
Minimal impact velocity	$v_{\text{col}} = 0\,\mathrm{m/s}$
Limit on jerk	$j_{\min} = -10.0\,\mathrm{m/s^3}$, $j_{\max} = 10.0\,\mathrm{m/s^3}$
Limit on acceleration	$a_{\min} = -8.0\,\mathrm{m/s^2}$, $a_{\max} = 1.5\,\mathrm{m/s^2}$
Limit on velocity	$v_{\max} = 50.8\,\mathrm{m/s}$

TABLE II: Parameters of the ACC systems. The values for the PI-ACC, IDM-ACC, and CA-ACC are from [11], [13], and the authors of [14], respectively. The parameters Δs_{min}, k_{p}, k_{i}, k_{q}, and b are adapted to increase the collision avoidance performance.

General	**PI-ACC [11]**	**IDM-ACC [13]**	**CA-ACC [14]**
$v_{\text{des}} = 30\,\mathrm{m/s}$	$k_{\text{p}} = 0.2\,\mathrm{1/s}$	$b = 0.02\,\mathrm{m/s^2}$	$K_1 = 0.1\,\mathrm{1/s^2}$
$t_{\text{des}} = 1.5\,\mathrm{s}$	$k_{\text{i}} = 0.1$		$K_2 = 5.4\,\mathrm{1/s^2}$
$\Delta s_{\text{min}} = 3\,\mathrm{m}$	$k_{\text{q}} = 0.1\,\mathrm{1/s}$		$P = 20\,\mathrm{m}$
$\delta = 0\,\mathrm{s}$	$h_0 = 0.1\,\mathrm{s}$		$Q = 1$
	$h_{\text{c}} = 0.2\,\mathrm{s^2/m}$		

where k_{p} is a proportional gain, k_{i} an integral gain, k_{q} a positive constant factor, and $\Delta s_{\mathrm{err}}^{\mathrm{PI}}$ the spacing error $\Delta s_{\mathrm{err}}^{\mathrm{PI}} := \Delta s - \Delta s_{\mathrm{min}} + v_{\mathrm{acc}} h$. The time headway h favors a larger spacing at higher velocities:

$$h = \begin{cases} 1, & h_0 - h_{\mathrm{c}}\Delta v \geq 1,\\ h_0 - h_{\mathrm{c}}\Delta v, & 0 < h_0 - h_{\mathrm{c}}\Delta v < 1,\\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where h_0 is a constant time headway and h_{c} a constant factor.

2) Intelligent Driver Model (IDM-ACC): We use the variant of the IDM for increased safety as proposed in [13]. Its control law is given by

$$u = a_{\mathrm{max}} \left[1 - \left(\frac{v_{\mathrm{acc}}}{v_{\mathrm{des}}} \right)^4 - \left(\frac{\Delta s_{\mathrm{des}}^{\mathrm{IDM}}}{\Delta s} \right)^2 \right], \tag{5}$$

where $\Delta s_{\mathrm{des}}^{\mathrm{IDM}} := \Delta s_{\mathrm{min}} + v_{\mathrm{acc}} t_{\mathrm{des}} + \frac{v_{\mathrm{acc}}\Delta v}{2\sqrt{a_{\mathrm{max}} b}}$.

3) ACC with collision avoidance (CA-ACC): The CA-ACC [14] uses the control law

$$u = K_1 \Delta s_{\mathrm{err}}^{\mathrm{CA}} + K_2 \Delta v R(\Delta s), \tag{6}$$

where K_1 and K_2 are constant gains and $\Delta s_{\mathrm{err}}^{\mathrm{CA}} := \min\big(\Delta s - \Delta s_{\mathrm{min}} - v_{\mathrm{acc}} t_{\mathrm{des}}, (v_{\mathrm{des}} - v_{\mathrm{acc}}) t_{\mathrm{des}}\big)$. The error response function $R(\Delta s)$ is

$$R(\Delta s) = \frac{-1}{1 + P e^{-\frac{\Delta s}{Q}}} + 1, \tag{7}$$

where Q is the aggressiveness coefficient, and P is the perception range coefficient.

4) ACC with safety guarantees (FM-ACC): The FM-ACC [15] is divided into two modes, a nominal control mode and an emergency control mode. The nominal mode uses model predictive control (MPC) and is applied as long as it can satisfy the safe distance. If it cannot find safe input values, the emergency control mode executes a pre-defined emergency deceleration profile so that the ACC vehicle is formally guaranteed to remain collision-free.

B. Forward search

The forward search is able to falsify the PI-ACC and IDM-ACC, but cannot find a counter-example for the CA-ACC and FM-ACC. The state trajectories leading to one of the obtained collisions for the PI-ACC and IDM-ACC are shown in Fig. 6 and Fig. 7, respectively. Tab. III lists the required computation times, the time t_{unsafe} at which the ACC vehicle first entered the set of unsafe states, and the initial and final states of both vehicles. Even though we set $v_{\mathrm{col}} = 0\,\mathrm{m/s}$ (cf. Tab. I), the generated collisions have an impact velocity of $4.6\,\mathrm{m/s}$ and $2.2\,\mathrm{m/s}$ for the PI-ACC and IDM-ACC, respectively. In real traffic, the obtained trajectories of the leading vehicle are likely to occur during stop-and-go traffic on a highway.

C. Backward search

The backward search is able to falsify the CA-ACC, even though the forward search finds no counter-example. Fig. 8 shows the trajectories $x_{\mathrm{lead}}([t_0, t_{\mathrm{col}}])$ and $x_{\mathrm{acc}}([t_0, t_{\mathrm{col}}])$ that are obtained from a backward search; we continued the backward search not only until a safe node is found, but until $s_{\mathrm{safe}}(t_0) \geq 100\,\mathrm{m}$ and $\Delta s(t_0) \geq 235\,\mathrm{m}$, which significantly increased the required computation time (cf. results in Tab. IV without this addition). Tab. III lists the details of the falsification result. The generated counter-example corresponds to a traffic situation in which the leading vehicle is forced to brake due to a traffic jam. Note that the ACC vehicle could have avoided a collision by braking earlier.

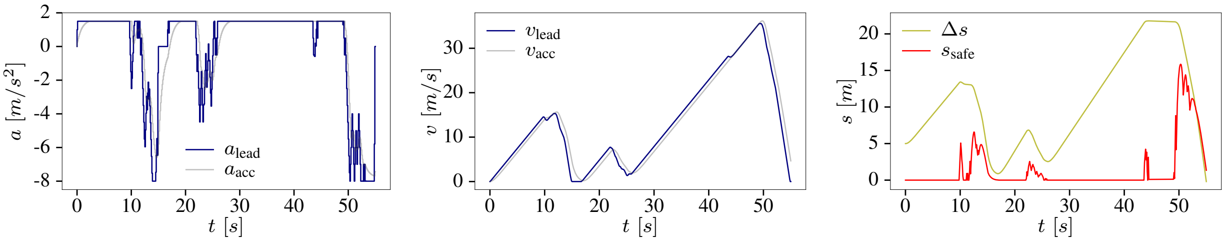
By applying the backward search to the FM-ACC, we were able to identify an error in the code generation of the FM-ACC. After correcting the code generation, the FM-ACC remains collision-free in the backward search, whereas the PI-ACC and IDM-ACC are falsified in every simulation run.

D. Comparison of the computational efficiency

We evaluate the computational efficiency of falsifying an ACC system by comparing the forward search with and without the consideration of unsafe states, backward search, and MCS with each other. To improve the sampling of MCS so that it is more uniform over the input space (since $|a_{\mathrm{min}}| \neq |a_{\mathrm{max}}|$), we bias the sampling with a beta distribution $Beta(\alpha = 14,\ \beta = 2)$. All four approaches attempt to find a collision against the PI-ACC, IDM-ACC, and CA-ACC in 100 simulation runs with an iteration limit of $n = 600$, where a run is aborted as soon as a collision node is generated (the remaining parameters are set as presented in Tab. I and Tab. II). The results of the comparison are given in Tab. IV. The standard forward search (without considering unsafe states) finds 4306 transitions into unsafe states for the PI-ACC and 76 for the IDM-ACC, but it does not exploit these situations to generate a collision. As we can see, the consideration of unsafe states drastically improves

TABLE III: Falsification using the forward search for PI-ACC and IDM-ACC and the backward search for CA-ACC.

Parameter	**PI-ACC**	**IDM-ACC**	**CA-ACC**
Comp. time	3 min	2 min	16 min
t_{unsafe}	53.1 s	53.2 s	5.3 s
$a_{\mathrm{acc}}(t_0)$	$0.0\,\mathrm{m/s^2}$	$0.0\,\mathrm{m/s^2}$	$-5.2\,\mathrm{m/s^2}$
$a_{\mathrm{lead}}(t_0)$	$0.0\,\mathrm{m/s^2}$	$0.0\,\mathrm{m/s^2}$	$-2.1\,\mathrm{m/s^2}$
$v_{\mathrm{acc}}(t_0)$	$0.0\,\mathrm{m/s}$	$9.3\,\mathrm{m/s}$	$42.9\,\mathrm{m/s}$
$v_{\mathrm{lead}}(t_0)$	$0.0\,\mathrm{m/s}$	$19.5\,\mathrm{m/s}$	$18.9\,\mathrm{m/s}$
$\Delta s(t_0)$	5.0 m	8.0 m	237.9 m
$s_{\mathrm{safe}}(t_0)$	0.0 m	0.0 m	100.0 m
$a_{\mathrm{acc}}(t_{\mathrm{col}})$	$-7.4\,\mathrm{m/s^2}$	$-1.0\,\mathrm{m/s^2}$	$-8.0\,\mathrm{m/s^2}$
$a_{\mathrm{lead}}(t_{\mathrm{col}})$	$0.0\,\mathrm{m/s^2}$	$0.0\,\mathrm{m/s^2}$	$0.0\,\mathrm{m/s^2}$
$v_{\mathrm{acc}}(t_{\mathrm{col}})$	$4.7\,\mathrm{m/s}$	$1.4\,\mathrm{m/s}$	$2.7\,\mathrm{m/s}$
$v_{\mathrm{lead}}(t_{\mathrm{col}})$	$0.0\,\mathrm{m/s}$	$0.0\,\mathrm{m/s}$	$0.0\,\mathrm{m/s}$
$s_{\mathrm{safe}}(t_{\mathrm{col}})$	1.4 m	0.5 m	0.4 m

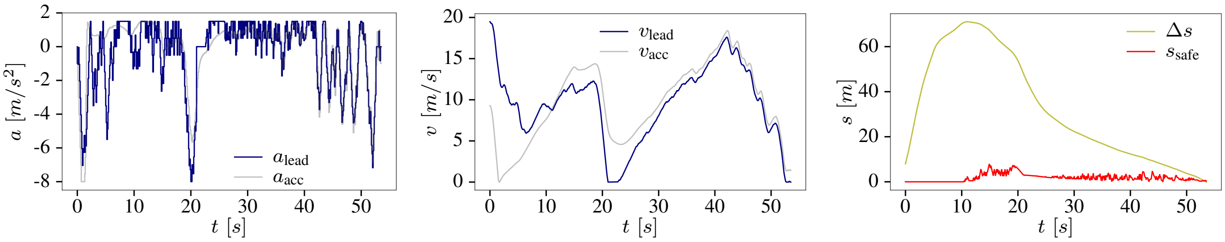


(a) Acceleration profiles.

(b) Velocity profiles.

(c) Relative and safe distance profile.

Fig. 6: The forward search finds a trajectory for the leading vehicle so that the PI-ACC causes a collision (CommonRoad ID: S=ZAM_ACC-1_1_T-1:2018b).

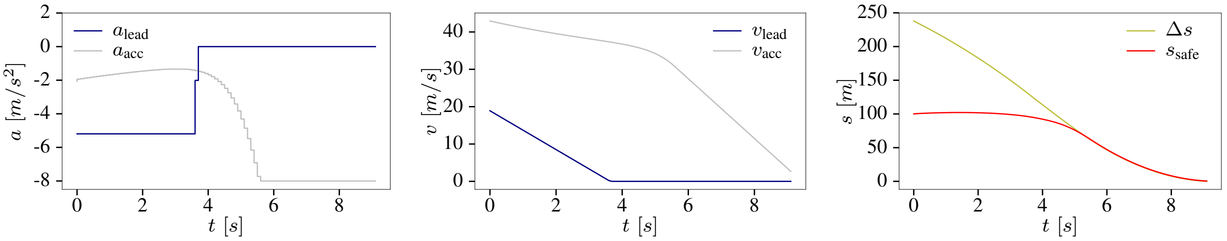


(a) Acceleration profiles.

(b) Velocity profiles.

(c) Relative and safe distance profile.

Fig. 7: The forward search finds a trajectory for the leading vehicle so that the IDM-ACC causes a collision (CommonRoad ID: S=ZAM_ACC-1_2_T-1:2018b).



(a) Acceleration profiles.

(b) Velocity profiles.

(c) Relative and safe distance profile.

Fig. 8: The backward search finds a trajectory for the leading vehicle, which is executed forward in time, so that the CA-ACC causes a collision (CommonRoad ID: S=ZAM_ACC-1_3_T-1:2018b).

the efficiency. The advantages of the forward search are its simple setup and large variance of the generated counter-examples. The advantage of the backward search is that it can falsify more ACC systems as compared to the other methods.

V. Conclusions

This paper presents two novel approaches to efficiently falsify the safety of adaptive cruise control systems. By integrating unsafe states in the standard forward search approach, we already achieve an improvement in the required computation time of up to 8 times. With this approach, however, we were not able to falsify all ACC systems in a reasonable time period. In contrast, our backward search approach is able to falsify even the sophisticated ACC system with collision avoidance in every test run. By starting the search from a set of unsafe states, our backward search algorithm is able to find counter-examples 300 times faster than standard approaches.

Our proposed methods allow developers to detect safety flaws in their system at early stages of the development with minimal effort. While the forward search can be used to generate diverse traffic scenarios, the backward search aims to quickly find an unsafe solution. In the future, we would like to add stress testing of the string stability and extend our method to falsify planning systems that combine longitudinal and lateral motions.

Acknowledgments

The authors thank Freddy Antony Mullakkal-Babu for the valuable discussion of our results and gratefully acknowledge partial financial support by the BMW Group within the CAR@TUM project and by the project *interACT* within the EU Horizon 2020 programme under grant agreement No 723395.

TABLE IV: Comparison of the standard forward search without considering unsafe states (S-FS), our proposed forward search with considering unsafe states (FS), our proposed backward search (BS), and Monte Carlo simulation (MCS) in 100 simulation runs with an iteration limit of $n = 600$.

Results for PI-ACC	**S-FS**	**FS**	**BS**	**MCS**
Number of obtained collisions	0	94	100	1
Avg. number of iterations	600.00	81.14	1.08	594.15
Avg. computation time	84.80 s	10.78 s	0.30 s	17.82 s
Results for IDM-ACC	**S-FS**	**FS**	**BS**	**MCS**
Number of obtained collisions	0	13	100	11
Avg. number of iterations	600.00	553.58	1.00	545.51
Avg. computation time	82.88 s	73.77 s	0.45 s	16.25 s
Results for CA-ACC	**S-FS**	**FS**	**BS**	**MCS**
Number of obtained collisions	0	0	100	0
Avg. number of iterations	600.00	600.00	1.08	600.00
Avg. computation time	84.70 s	81.62 s	0.29 s	8.83 s

References

[1] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, *Model-based testing of reactive systems: Advanced lectures*. Springer, 2005.

[2] L. N. Boyle and J. D. Lee, "Using driving simulators to assess driving safety," *Accident Analysis and Prevention*, vol. 42, no. 3, pp. 785 – 787, 2010.

[3] 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.

[4] 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.

[5] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivačić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 2, pp. 1–30, 2013.

[6] A. Dokhanchi, S. Yaghoubi, B. Hoxha, G. Fainekos, G. Ernst, Z. Zhang, P. Arcaini, I. Hasuo, and S. Sedwards, "ARCH-COMP18 category report: Results on the falsification benchmarks," in *Int. Workshop on Applied Verification of Continuous and Hybrid Systems*, 2018, pp. 104–109.

[7] P. Ioannou, Z. Xu, S. Eckert, D. Clemons, and T. Sieja, "Intelligent cruise control: theory and experiment," in *Proc. of the IEEE Int. Conf. on Decision and Control*, 1993, pp. 1885–1890.

[8] P. A. Ioannou and C. C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 657–672, 1993.

[9] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle System Dynamics*, vol. 48, no. 10, pp. 1167–1192, 2010.

[10] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, 2003.

[11] D. Yanakiev and I. Kanellakopoulos, "Nonlinear spacing policies for automated heavy-duty vehicles," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 4, pp. 1365–1377, 1998.

[12] P. Shakouri, A. Ordys, D. S. Laila, and M. Askari, "Adaptive cruise control system: comparing gain-scheduling PI and LQ controllers," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 964–12 969, 2011.

[13] A. Kesting, M. Treiber, M. Schönhof, and D. Helbing, "Adaptive cruise control design for active congestion avoidance," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 668–683, 2008.

[14] F. A. Mullakkal-Babu, M. Wang, B. van Arem, and R. Happee, "Design and analysis of full range adaptive cruise control with integrated collision avoidance strategy," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2016, pp. 308–315.

[15] 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. 5774–5781.

[16] S. Sadraddini, S. Sivaranjani, V. Gupta, and C. Belta, "Provably safe cruise control of vehicular platoons," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 262–267, 2017.

[17] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Int. Conf. on Decision and Control*, 2014, pp. 6271–6278.

[18] J. Lunze, "Adaptive cruise control with guaranteed collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1897–1907, 2019.

[19] W. Choi and D. Swaroop, "Assessing the safety benefits due to coordination amongst vehicles during an emergency braking maneuver," in *Proc.of the IEEE American Control Conference*, 2001, pp. 2099–2104.

[20] A. Touran, M. A. Brackstone, and M. McDonald, "A collision model for safety evaluation of autonomous intelligent cruise control," *Accident Analysis and Prevention*, vol. 31, no. 5, pp. 567 – 578, 1999.

[21] A. E. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 319–324.

[22] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 137–147, 2008.

[23] Z. Yang, X. Wang, X. Pei, S. Feng, D. Wang, J. Wang, and S. C. Wong, "Longitudinal safety analysis for heterogeneous platoon of automated and human vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3300–3305.

[24] M. Althoff and S. Lutz, "Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1326–1333.

[25] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder, "Accident scenario generation with recurrent neural networks," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3340–3345.

[26] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in *IEEE Intelligent Vehicles Symposium*, 2018, pp. 1555–1562.

[27] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta, "Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles," in *IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 1443–1450.

[28] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 2095 – 2101.

[29] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proc. of the Int. Conf. on Neural Information Processing Systems*, 2018, pp. 9849–9860.

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

[31] A. Bhatia and E. Frazzoli, "Incremental search methods for reachability analysis of continuous and hybrid systems," in *Hybrid Systems: Computation and Control*. Springer, 2004, pp. 142–156.

[32] T. Dreossi, T. Dang, A. Donzé, J. Kapinski, X. Jin, and J. V. Deshmukh, "Efficient guiding strategies for testing of temporal properties of hybrid systems," in *Proc. of the NASA Formal Methods Symposium*, 2015, pp. 127–142.

[33] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," in *Proc. of the NASA Formal Methods Symposium*, 2017, pp. 357–372.

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

[35] M. Althoff and R. Lösch, "Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance?" in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2016, pp. 485–491.

[36] 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.

[37] M. Fliess, J. Lvine, P. Martin, and P. Rouchon, "On differentially flat nonlinear systems," in *Nonlinear Control Systems Design*, 1993, pp. 159 – 163.

5 Conclusions and Future Work

5.1 Conclusions

Novel solutions for safety problems in driver assistance systems and autonomous vehicles have been presented. Our developed set-based prediction is guaranteed to always capture all acceptable behaviors of other traffic participants. Since users can specify which behaviors are considered acceptable, our prediction is useful for various applications. In particular, the prediction can be used for safety verification and can be easily integrated as an additional module in existing motion planning frameworks. Thus, our results enable safe motion planning and facilitate official approval and homologation of autonomous vehicles.

Summary Let us briefly summarize each chapter of this cumulative dissertation, in which we made use of various methods, such as reachability analysis, set invariance theory, formalization of legal text, optimization, and search.

Chapter 1 motivated the need for solving safety problems in motion planning of autonomous vehicles. After discussing related literature and requirements for the prediction of other traffic participants, we highlighted the contributions of this dissertation. Chapter 2 formalized the main problem statements of this dissertation: prediction, motion planning, and falsification. Subsequently, reachability analysis and our environment model were introduced briefly.

Chapter 3 presented three peer-reviewed publications describing the set-based prediction of other traffic participants. This prediction is designed to compute all acceptable behaviors according to a legal specification based on traffic rules. Our nondeterministic models and assumptions were formally defined so that the prediction result always complies with the safety specification. In addition, we are able to consider traffic participants hidden due to occlusions and interactions between detected traffic participants. Our constraint management also handles traffic participants violating traffic rules. Extensive numerical and real-world experiments demonstrated the applicability of our prediction. These experiments also validated our models of other traffic participants and of the environment.

Chapter 4 presented four peer-reviewed publications describing approaches for safe motion planning. Risk assessment, trajectory planning, and decision making were examined for safety flaws, which were then holistically resolved. Above all, a safety layer was developed in Section 4.2 that prevents autonomous vehicles from causing accidents, even in situations that have not been tested before. Various numerical experiments on CommonRoad traffic scenarios, some based on recorded traffic, demonstrated the achieved safety benefits.

Related work Other researchers proposed approaches similar to our set-based prediction during or after the publications of this dissertation have been published. In [47–49], reachable sets of pedestrians are computed, similar to the dynamic-based motion model of Section 3.2, and used to generate synthetic trajectories for surrounding pedestrians and to obtain optimal motion plans for the ego vehicle. In [50], occupancies of pedestrians are computed and are overlaid by a probability distribution. The same authors present a draft on extending their approach to vehicles in [51].

For the benefit of the scientific community, we released our tool SPOT in 2017, which computes the future occupancies of other traffic participants (cf. Section 1.4.2). Until now, several works [52–57] have already used SPOT, e. g., by building upon its theory or for their experiments. In [52], an anytime safety verification is proposed that uses SPOT for prediction and only computes the motion models of SPOT required for a successfully verification. In [53–55], fail-safe trajectories are computed that are collision-free against the predicted result of SPOT. In [56], SPOT has been extended to consider occluded vehicles with right of way. This extended version is used in [57] to obtain safe trajectories for the ego vehicle.

Most safety approaches can only consider a finite time horizon, such as the risk assessment of Section 4.1. In practice, this is often not regarded as an issue due to long planning horizons and high replanning rates. However, the ego vehicle could easily enter a state that inevitable ends in a collision. To remain collision-free for an infinite time horizon, trajectories must end in an state that is safe forever. Such states are denoted as invariably safe states. Based on the prediction of Chapter 3 and the safety concepts of Section 4.3, invariably safe states can be determined as described in the derivative work [54].

Safety first When will autonomous vehicles be safe enough to be employed on the road? This question has been discussed for a long time [28]. In the past, many approaches have focused on increasing the performance of autonomous vehicles, while not focusing on formally ensuring safety and postponing the improvement of safety until after deployment. In fact, it is often believed that when rigorously accounting for safety, the ego vehicle behaves too conservatively. However, we are convinced that safety can and needs to be ensured. If safety is not properly taken care of, the ego vehicle may ignorantly take high risks that could endanger the lives of passengers and other traffic participants. Thus, approaches as presented in this dissertation should be used and further enhanced to quantify risks and to knowingly reduce it. Putting safety first would also simplify achieving legal certifications and strengthen societal trust in autonomous vehicles. The results of Section 4.2 even indicate that nonconservative driving behavior can still be achieved despite rigorously ensuring safety. In addition, if every traffic participant adheres to legal safety, which most traffic participants do, no collisions will occur at all.

Overall, the contributions of this dissertation seem to be feasible for realization in industrial applications. Our developed approaches mostly focus on nominal safety, in particular on the safety of the intended functionality (SOTIF) according to ISO 21448. Other aspects, such as hardware failures for ensuring functional safety, need to be considered according to ISO 26262, for example. In addition, an industrial standard specifying the motion safety of autonomous vehicles does not yet exist [14]. If legal authorities recognize our concept of legal safety as a standard, the presented approaches of this dissertation can be certified for usage in series production of autonomous vehicles.

5.2 Future work

Future work for each section of Chapters 3 and 4 is discussed within each included publication. Here, we highlight important aspects that might be worth improving in the set-based prediction:

- To consider priorities at intersections as additional traffic rules in the prediction, these priorities must be provided by the environment model. Then, another module or an additional part of the set-based prediction can determine which traffic participants have to yield and are not allowed to proceed before others have passed. The difficulty lies in the question until when a traffic participant obligated to yield is still allowed to proceed.

- The interaction between traffic participants has been included in the set-based prediction in Section 3.3. However, we have not yet investigated the interaction between hidden and detected traffic participants. While actual traffic participants can move through artificially created ones (which we introduced as phantoms objects), the opposite is not true. This makes it possible to reduce the over-approximation in some cases, e. g., at roundabouts. In addition, when observing the boundaries of the field of view for multiple time steps, we can conclude that some states are not possible for hidden traffic participants and thus, we can create the phantom objects with a smaller initial set of states.

- The maximum admissible velocity for vehicles is restricted by the curvature of the road due to reaching maximum tire forces. This critical velocity can be computed for a given path based on [58]. In the prediction, however, it is difficult to consider this velocity as a limit while remaining over-approximative due to infinitely many possible paths through the road network. Yet, an stricter velocity constraint can significantly reduce the over-approximation of the prediction, especially in tight curves or when turning.

- The lateral acceleration for vehicles is currently limited only by the maximum absolute acceleration and thus is equal to the maximum braking acceleration. If stricter limits are desired, e. g., from [59], we suggest to use an addition model that constrains the lateral velocity and acceleration and remains over-approximative in longitudinal direction.

- The parameter values for the motion models need to be chosen carefully and in accordance with the desired safety specification. We proposed conservative default values, e. g., based on empirical studies or traffic rules. Yet, especially the acceleration and velocity limits for pedestrians are difficult to argue for. Physiological experiments revealed that peak values are much higher than mean values [60]. Thus, if the time step size of the prediction is large enough, the mean values can be used as stricter acceleration limits.

- When using the set-based prediction solely for online verification of given trajectories (and not to extract constraints for motion planning, such as in Section 4.2), the performance of the prediction can be further optimized. As described by [52], we can reuse previous prediction results as long as the verification remains successful, and we can perform reachability analysis only for selected models until the verification is successful. Both techniques increase the over-approximation of the prediction but reduce the computational cost. In future work, we can select only those traffic participants that are relevant for the ego vehicle, and we can group traffic participants close to each other into a single object, e. g., a group of pedestrians.

- Besides the applications of the set-based prediction in risk assessment, trajectory planning, and decision making, our developed motion models might also be useful in other applications, e. g., in the state estimation for object tracking.

Bibliography

[1] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[2] OECD, “Road accidents (indicator),” 2019. [Online]. Available: https://doi.org/10.1787/8dacf707-en

[3] Federal Highway Administration (FHWA), “Highway statistics 2016,” U.S. Department of Transportation, Federal Highway Administration, Washington D.C., 2017. [Online]. Available: http://www.fhwa.dot.gov/policyinformation/statistics.cfm

[4] P. Koopman and M. Wagner, “Autonomous vehicle safety: An interdisciplinary challenge,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[5] D. Elliott, W. Keen, and L. Miao, “Recent advances in connected and automated vehicles,” *Journal of Traffic and Transportation Engineering*, vol. 6, no. 2, pp. 109–131, 2019.

[6] S. W. Loke, “Cooperative automated vehicles: A review of opportunities and challenges in socially intelligent vehicles beyond networking,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 509–518, 2019.

[7] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, “A survey on cooperative longitudinal motion control of multiple connected and automated vehicles,” *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 1, pp. 4–24, 2020.

[8] K. Maček, D. Vasquez, T. Fraichard, and R. Siegwart, “Towards safe vehicle navigation in dynamic urban scenarios,” *Automatika*, vol. 50, no. 3-4, pp. 184–194, 2009.

[9] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, “Highly automated driving on highways based on legal safety,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 333–347, 2013.

[10] R. Kianfar, P. Falcone, and J. Fredriksson, “Safety verification of automated driving systems,” *IEEE Intelligent Transportation Systems Magazine*, vol. 5, pp. 73–86, 2013.

[11] 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.

[12] S. Mitsch, K. Ghorbal, D. Vogelbacher, and A. Platzer, “Formal verification of obstacle avoidance and navigation of ground robots,” *Int. Journal of Robotics Research*, vol. 36, no. 12, pp. 1312–1340, 2017.

[13] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and decision-making for autonomous vehicles,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018.

[14] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a formal model of safe and scalable self-driving cars,” *arXiv:1708.06374v6 [cs.RO]*, pp. 1–37, 2018.

[15] S. Vaskov, H. Larson, S. Kousik, M. Johnson-Roberson, and R. Vasudevan, “Not-at-fault driving in traffic: A reachability-based approach,” in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2785–2790.

[16] C. Schmidt, F. Oechsle, and W. Branz, “Research on trajectory planning in emergency situations with multiple objects,” in *Proc. of the 9th IEEE Int. Conf. on Intelligent Transportation Systems*, 2006, pp. 988–992.

[17] A. Wu and J. How, “Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles,” *Autonomous Robots*, vol. 32, pp. 227–242, 2012.

[18] S. Bouraine, T. Fraichard, and H. Salhi, “Provably safe navigation for mobile robots with limited field-of-views in dynamic environments,” *Autonomous Robots*, vol. 32, pp. 267–283, 2012.

[19] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[20] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, “Human motion trajectory prediction: A survey,” *Int. Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[21] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, G. Markkula, A. Schieben, F. Tango, N. Merat, and C. Fox, “Pedestrian models for autonomous driving part II: High level models of human behaviour,” *IEEE Transactions on Intelligent Transportation Systems*, 2020, [available online].

[22] A. Rasouli and J. K. Tsotsos, “Autonomous vehicles that interact with pedestrians: A survey of theory and practice,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 900–918, 2019.

[23] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, “Collision avoidance: A literature review on threat-assessment techniques,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 101–113, 2019.

[24] 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, 2019.

[25] D. González, J. Pérez, V. Milanés, 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.

[26] 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.

[27] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, 2018.

[28] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.

[29] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivačić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 2s, pp. 1–30, 2013.

[30] A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, "A survey of algorithms for black-box safety validation," *arXiv:2005.02979 [cs.LG]*, 2020.

[31] 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.

[32] O. Maler, "Computing reachable sets: An introduction," French National Center of Scientific Research, Tech. Rep., 2008.

[33] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Dissertation, Technische Universität München, 2010, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[34] G. Frehse, "An introduction to hybrid automata, numerical simulation and reachability analysis," in *Formal Modeling and Verification of Cyber-Physical Systems*, R. Drechsler, U. Kühne, Ed. Springer Vieweg, Wiesbaden, 2015, pp. 50–81.

[35] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, G. Markkula, A. Schieben, F. Tango, N. Merat, and C. Fox, "Pedestrian models for autonomous driving part I: Low-level models, from sensing to tracking," *IEEE Transactions on Intelligent Transportation Systems*, 2020, [available online].

[36] E. Martí, M. Á. de Miguel, F. García, and J. Pérez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[37] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[38] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[39] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr, “Grid-based object tracking with nonlinear dynamic state and shape estimation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2874–2893, 2019.

[40] Y. Emzivat, J. Ibanez-Guzman, H. Illy, P. Martinet, and O. H. Roux, “A formal approach for the design of a dependable perception system for autonomous vehicles,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2452–2459.

[41] D. Feng, L. Rosenbaum, and K. Dietmayer, “Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3266–3273.

[42] M. T. Le, F. Diehl, T. Brunner, and A. Knoll, “Uncertainty estimation for deep neural object detectors in safety-critical applications,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 3873–3878.

[43] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, “A probabilistic framework for joint pedestrian head and body orientation estimation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1872–1882, 2015.

[44] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram, “Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1717–1726, 2012.

[45] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, “Consistent outdoor vehicle localization by bounded-error state estimation,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1211–1216.

[46] A. Gning and P. Bonnifait, “Constraints propagation techniques on intervals for a guaranteed localization using redundant data,” *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.

[47] M. Hartmann, A. Ferrara, and D. Watzenig, “Data-based reachability analysis for movement prediction of pedestrians and motion planning,” in *Proc. of the IEEE Int. Conf. on Vehicular Electronics and Safety*, 2018, pp. 1–7.

[48] M. Hartmann and D. Watzenig, “Pedestrians walking on reachable sets and manifolds,” in *Proc. of the IEEE Int. Conf. on Mechatronics*, 2019, pp. 562–569.

[49] ——, “Optimal motion planning with reachable sets of vulnerable road users,” in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 891–898.

[50] P. Zechel, R. Streiter, K. Bogenberger, and U. Göhner, “Pedestrian occupancy prediction for autonomous vehicles,” in *Proc. of the 3rd IEEE Int. Conf. on Robotic Computing*, 2019, pp. 230–235.

[51] ——, “Probabilistic interaction-aware occupancy prediction for vehicles in arbitrary road scenes,” in *Proc. of the 3rd IEEE Int. Conf. on Robotic Computing*, 2019, pp. 423–424.

[52] F. Gruber and M. Althoff, “Anytime safety verification of autonomous vehicles,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1708–1714.

[53] C. Pek and M. Althoff, “Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1447–1454.

[54] ——, “Efficient computation of invariably safe states for motion planning of self-driving vehicles,” in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 3523–3530.

[55] S. Manzinger, C. Pek, and M. Althoff, “Using reachable sets for trajectory planning of automated vehicles,” *IEEE Transactions on Intelligent Vehicles*, 2020, [available online].

[56] P. F. Orzechowski, A. Meyer, and M. Lauer, “Tackling occlusions & limited sensor range with set-based safety verification,” in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1729–1736.

[57] 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.

[58] E. Velenis and P. Tsiotras, “Optimal velocity profile generation for given acceleration limits: theoretical analysis,” in *Proc. of the American Control Conference*, 2005, pp. 1478–1483.

[59] P. Zechel, R. Streiter, K. Bogenberger, and U. Göhner, “Assumptions of lateral acceleration behavior limits for prediction tasks in autonomous vehicles,” in *Proc. of the 7th Int. Conf. on Mechatronics Engineering*, 2019, pp. 1–6.

[60] N. Tiemann, “Ein Beitrag zur Situationsanalyse im vorausschauenden Fußgängerschutz,” Dissertation, Universität Duisburg-Essen, 2012, https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-31259/Tiemann_Diss.pdf.

Own Publications

[61] **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, (available at go.tum.de/580663).

[62] 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, (available at go.tum.de/625898).

[63] **M. Koschi** and M. Althoff, "Interaction-aware occupancy prediction of road vehicles," in *Proc. of the 20th IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1885–1892, (available at go.tum.de/280014).

[64] C. Pek*, **M. Koschi***, M. Werling, and M. Althoff, "Enhancing motion safety by identifying safety-critical passageways," in *Proc. of the 56th IEEE Conf. on Decision and Control*, 2017, pp. 320 – 326, (available at go.tum.de/406006).

[65] M. Althoff, S. Urban, and **M. Koschi**, "Automatic conversion of road networks from OpenDRIVE to lanelets," in *Proc. of the IEEE International Conference on Service Operations and Logistics, and Informatics*, 2018, pp. 157–162, (available at go.tum.de/202956).

[66] S. Söntges*, **M. Koschi***, and M. Althoff, "Worst-case analysis of the time-to-react using reachable sets," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1891–1897, (available at go.tum.de/050160).

[67] **M. Koschi**, C. Pek, M. Beikirch, and M. Althoff, "Set-based prediction of pedestrians in urban environments considering formalized traffic rules," in *Proc. of the 21st IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2704–2711, (available at go.tum.de/663155, video attachment available at go.tum.de/074008).

[68] C. Pek, **M. Koschi**, and M. Althoff, "An online verification framework for motion planning of self-driving vehicles with safety guarantees," in *AAET - Automatisiertes und vernetztes Fahren*, 2019, pp. 260–274, (available at go.tum.de/082314).

[69] **M. Koschi***, C. Pek*, S. Maierhofer*, and M. Althoff, "Computationally efficient safety falsification of adaptive cruise control systems," in *Proc. of the 22nd IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2879–2886, (available at go.tum.de/143617, video attachment available at go.tum.de/500310).

[70] **M. Koschi** and M. Althoff, "Set-based prediction of traffic participants considering occlusions and traffic rules," *IEEE Transactions on Intelligent Vehicles*, vol. 6,

no. 2, pp. 249–265, 2020, (available at go.tum.de/081648, video attachment available at go.tum.de/812843).

[71] C. Pek*, S. Manzinger*, **M. Koschi***, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, pp. 518–528, 2020, (available at rdcu.be/b7bC4).

* These authors have equally contributed to the respective work and share the first authorship.

Theses of Supervised Students

[72] V. Bui, “Investigating interaction in set-based traffic prediction,” Bachelor Thesis, Technical University of Munich, 2017.

[73] H. Kirchner, “Randomized generation of road networks and 3d visualization of traffic scenarios in gazebo,” Bachelor Thesis, Technical University of Munich, 2017.

[74] D. Papyan, “Improvements to set-based traffic prediction,” Master Thesis, Technical University of Munich, 2017.

[75] H. Bibel, “Interaction of traffic participants in set-based prediction,” Master Thesis, Technical University of Munich, 2017.

[76] M. Allard, “Formalizing and integrating traffic rules at intersections in set-based prediction,” Bachelor Thesis, Technical University of Munich, 2017.

[77] S. Speth, “Securing functional maturity of automated driving functions,” Bachelor Thesis, Technical University of Munich, 2017.

[78] L. Braunstorfer, “Risk assessment of traffic scenarios by determining the criticality using reachable sets and optimal control,” Bachelor Thesis, Technical University of Munich, 2017.

[79] M. Beikirch, “Collision avoidance in urban environments using set-based prediction,” Master Thesis, Technical University of Munich, 2017.

[80] L. Willinger, “Environmental model with sensor limitations in set-based traffic prediction,” Bachelor thesis, Technical University of Munich, 2017.

[81] S. Urban, “Evaluation of a novel set-based prediction of traffic participants for autonomous driving using real-world measurement data,” Master Thesis, Technical University of Munich, 2018.

[82] L. Streit, “Web-based benchmark for trajectory planning of autonomous vehicles,” Bachelor Thesis, Technical University of Munich, 2018.

[83] A. Gaul, “Over-approximative occupancy prediction of vehicles considering limited acceleration and off-tracking in turns,” Bachelor Thesis, Technical University of Munich, 2018.

[84] F. Schönert, “Online verification of autonomous driving in parking scenarios using set-based prediction,” Master Thesis, Technical University of Munich, 2018.

[85] P. Meyersieck, "Falsification of adaptive cruise control systems (ACC) in automated vehicles," Master Thesis, Technical University of Munich, 2018.

[86] S. Kaster, "Online prediction of vehicles and pedestrians for guaranteed motion safety of autonomous vehicles," Master Thesis, Technical University of Munich, 2019.

[87] C. Baumann, "Risk assessment of traffic scenarios by combining reachability analysis and a user study," Master Thesis, Technical University of Munich, 2019.

[88] M. Althaus, "Consideration of safe distances in online verification for motion planning of autonomous vehicles," Master Thesis, Technical University of Munich, 2019.

[89] J. Kaps, "Set-based prediction of vehicles and pedestrians in urban environments considering traffic rules," Bachelor Thesis, Technical University of Munich, 2019.