An Online Verification Framework for Motion Planning of Self-driving Vehicles with Safety Guarantees

Christian Pek[*], Markus Koschi[†], and Matthias Althoff[‡]

Abstract

Self-driving vehicles must be able to safely navigate in any traffic scenario. However, all situations are different; even when clustering them, an impractical amount of scenarios would have to be verified. Thus, we propose a safety framework to verify the safety of each planned trajectory on-the-fly, using formal methods to handle uncertain measurements and future behaviors of traffic participants and disturbances acting on the ego vehicle, among others. Our framework can easily be integrated in existing motion planning architectures and enables fail-safe operation of self-driving vehicles, since we provide a safe plan for any given point in time. We demonstrate the benefits of our framework in different highway and urban scenarios of the CommonRoad benchmark suite.

Keywords: Formal Verification, Motion Planning, Safety, Self-driving Vehicles.

1 Introduction

A recent study has revealed that self-driving vehicles need to be tested for 440 million km to demonstrate that they have a better performance than humans with a 95% confidence level [1]. This translates to 12.5 years of test driving with a fleet of 100 vehicles continuously driving 24 hours a day. Moreover, this intensive testing is required for each major change of the software of the vehicle. Similarly, the number of distinct traffic situations which would need to be tested for a good coverage is estimated to be at least 10^{82}, which is a larger number than the amount of

[*] Christian Pek is with the Technical University of Munich, Department of Computer Science, D-85748 Garching, and BMW Group, D-85716 Unterschleißheim, (e-mail: Christian.Pek@bmw.de).

[†] Markus Koschi is with the Technical University of Munich, Department of Computer Science, D-85748 Garching, (e-mail: Markus.Koschi@tum.de).

[‡] Matthias Althoff is with the Technical University of Munich, Department of Computer Science, D-85748 Garching, (e-mail: Althoff@in.tum.de).

atoms in the universe[1]. Even in allegedly simple scenarios such as vehicle following, behaviors of other traffic participants, e. g., a cut-in, create safety-critical situations. Safety-critical situations have to be resolved by self-driving vehicles in a timely manner in order to not endanger passengers and other traffic participants.

Designing a safety component which is able to deal with all expected situations and is easy to understand for certification authorities is crucial for operational safety as defined in the ISO 26262 standard [2]. Classical verification approaches perform the safety assessment offline before the vehicle is deployed. However, since the environment of the self-driving is usually not known beforehand and highly uncertain, classical verification approaches cannot be applied to guarantee safety. Novel online verification approaches are needed to cope with any traffic situation of the self-driving vehicle during its operation.

1.1 Literature Overview

In the following, we briefly review common planning and verification approaches for the domain of self-driving vehicles.

Many motion planning approaches compute trajectories which are collision-free against the predicted (most-likely) motion of obstacles within the planning horizon [3–5]. In safety-critical situations, [6, 7] plan evasive trajectories to avoid potential collisions. However, when only predicting single behaviors, the safety of planned trajectories only holds if other traffic participants do not deviate from this single predicted motion.

One approach to guarantee safety is logical reasoning. Here, planned motions are checked whether they comply with certain rules, e. g., formulated using higher-order logic as presented in [8]. Multi-lane spatial logic is used to assess the safety of a lane change controller in [9]. The safety of an adaptive cruise control system is verified in [10] by applying quantified differential dynamic logic. Although the application of logical reasoning can guarantee safety, logical expressions for the verification of highly-complex systems are often complex and must be adapted to new scenarios.

Yet in another approach [11–13], motion plans are only executed if they do not end in an *inevitable collision state* (ICS), which are states in which the self-driving vehicle eventually collides, no matter what trajectory it executes. Motion plans are called *passive safe*, if the vehicle is at standstill at the time of collision, which is ensured by pre-computed braking trajectories in [14]. Both ICS and passive safety are computationally expensive, and most works can only handle a single trajectory prediction of traffic participants for online computation.

[1] For every surrounding traffic participant, we require x and y position, velocity, and orientation, i. e., 4 variables per traffic participant. For every lane, we require its width, curvature, and curvature rate, i. e., 3 variables per lane. For the ego vehicle, we require x and y position, velocity, orientation, yaw rate, slip angle, friction coefficient, and mass, i. e., 8 variables. Assuming that we consider 20 possible values for each variable, 10 surrounding traffic participants while neglecting their type, and 5 lanes, we obtain $(20^4)^{10} \cdot (20^3)^5 \cdot 20^8 = 10^{82}$ distinct traffic situations.

Reachability analysis, in contrast, accounts for any feasible future motion of dynamic obstacles [15–17]. 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 self-driving vehicle, one can identify possible future collisions. Safety verification using reachability analysis has been proposed for several domains, e. g., self-driving vehicles [18, 19] and robot manipulators [20]. However, set-based techniques have the disadvantage that unsafe regions may grow rapidly for long planning horizons, eventually blocking the whole drivable area. Nevertheless, the application of reachability analysis to the safety verification of self-driving vehicles is promising.

1.2 Contribution

Online verification verifies systems during their operation. In order to realize online verification, standard offline verification techniques have to be rethought in many ways: (1) one requires anytime algorithms to adjust to the changing environments and timing-constraints of the vehicle, (2) one must consider other surrounding (intelligent) obstacles which appear and disappear on-the-fly, and (3) one must combine planning and control with verification techniques to repair failed verification attempts online. In this work, we propose a novel safety framework for self-driving vehicles using reachability analysis which

1. verifies the safety of motion plans of self-driving vehicles online,
2. considers any feasible (legal) future motions of obstacles in the environment,
3. works with any provided motion planning framework, and
4. allows effortless integration in vehicles and reduces costs for certification.

This paper is structured as follows: In Sec. 2, we explain the structure of our framework and how it can be integrated in existing vehicle architectures. The modules of our framework are described in Sec. 3. Afterwards, we demonstrate the benefits of the proposed framework in a highway and an urban scenario in Sec. 4. We finish with conclusions in Sec. 5.

2 Integration in Existing Vehicle Frameworks

Our online safety verification framework works with any provided motion planning framework. As shown in Fig. 1, it is integrated between the planning and the control layer of the vehicle. Our framework is composed of modules for set-based prediction, fail-safe trajectory generation, and online verification (cf. Sec. 3). The framework only requires the planned trajectories of the vehicle and the detected surrounding obstacles. This design allows components above our safety layer to be changed at any time without any sacrifice of safety, since the safety layer will verify on-the-fly whether the proposed motions are safe or not.

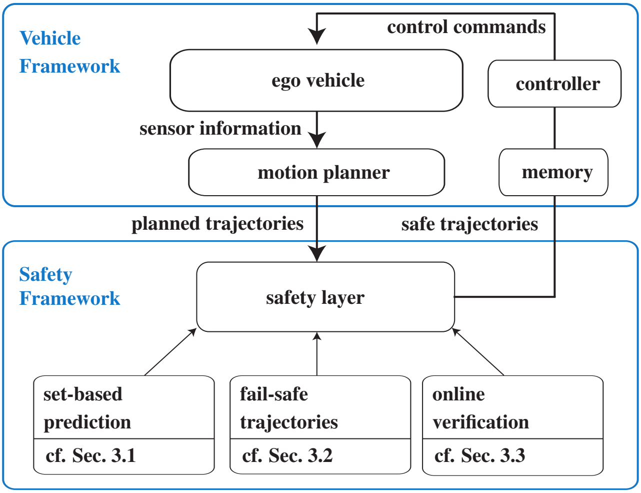


Figure 1: Our proposed safety framework and its possible integration into an existing vehicle framework.

The proposed architecture in Fig. 1 has the big advantage that techniques developed for motion planning do not have to be certified (cf. ISO 26262 [2]), and the code in the motion planner can be easily changed or updated. As a result, our framework can even safely cope with machine learning approaches [21], which are usually hard to certify. Prototypes of our modules consist of only a few thousand lines of code, enabling companies to use model checking tools to prove correctness of the code according to its specification [22]. In case of any malfunction, where new trajectories cannot be obtained during run-time (e. g., due to an error in the computing hardware, including our safety module), the self-driving vehicle remains safe, since it can just execute the previously verified fail-safe trajectory which is stored on a redundant memory (cf. memory module in Fig. 1). By using a redundant memory module, a redundant computing hardware for the safety layer becomes obsolete, reducing the number of hardware components which require certification.

3 Modules of the Safety Framework

The basic idea of our proposed online verification cycle during vehicle operation is presented in Fig. 2. Based on the detected traffic participants, their possible future occupancy (i. e., feasible future positions) is predicted over a specified time horizon (cf. Sec. 3.1). This prediction incorporates uncertainties such as uncertain positions or velocities of surrounding traffic participants. To reasonably restrict the predicted occupancy, we also consider traffic rules; we can explicitly

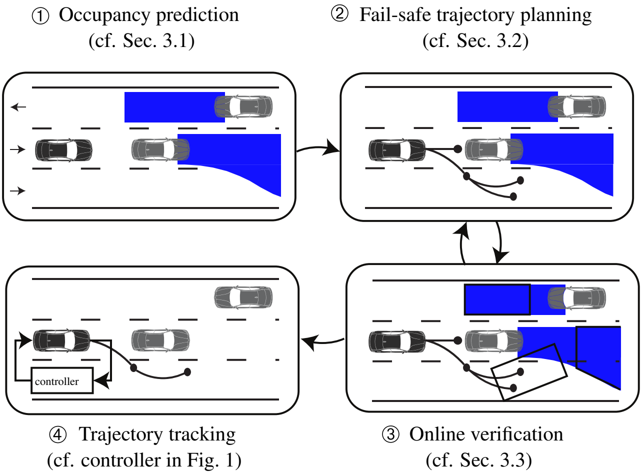

① Occupancy prediction (cf. Sec. 3.1)
② Fail-safe trajectory planning (cf. Sec. 3.2)
③ Online verification (cf. Sec. 3.3)
④ Trajectory tracking (cf. controller in Fig. 1)
controller

Figure 2: Verification cycle for fail-safe operation of self-driving vehicles. (1) The feasible (legal) future motions of other traffic participants are predicted. (2) Based on the computed predictions, fail-safe trajectories are planned. (3) The computed safe trajectories are verified online with respect to the predicted occupancies. Black rectangles illustrate the occupancy of the ego vehicle and obstacles at a certain point in time. (4) The vehicle executes the safe motion.

assume that traffic participants abide by certain rules, but remove this constraint if traffic rules are violated. In step 2 of Fig. 2, we plan safe trajectories, i. e., fail-safe trajectories branching off the intended trajectories given by the motion planner of the vehicle (cf. Sec. 3.2). Subsequently, these safe trajectories are verified as safe with respect to the predicted occupancies of other traffic participants and possible disturbances in the controller of the ego vehicle (cf. Sec. 3.3). Thus, they enable the self-driving vehicle to recover safely if a safety-critical situation occurs. If multiple trajectories are provided by the motion planner of the vehicle, we check the safety of each of them and return the best verified trajectory according to a cost function to the vehicle controller in step 4. The verification cycle of Fig. 2 is repeated in every planning cycle of the vehicle and thus enables fail-safe operation, since a safe plan exists at any given point in time.

3.1 Formal Prediction of Other Traffic Participants

We predict all feasible future motions of surrounding traffic participants to determine possible occupied regions over time [23,24]. Since the exact behavior of surrounding traffic participants is not known, we use reachability analysis to add set-based uncertainty. Let us introduce $\mathcal{X} \subset \mathbb{R}^n$ as the set of possible states x, $\mathcal{X}^0$ as the set of possible initial states considering measurement errors, and $\mathcal{U} \subset \mathbb{R}^m$ as the set of admissible control inputs u of an obstacle $b \in \mathcal{B} \subset \mathbb{N}_+$, whose motion is governed by the differential equation

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

Without loss of generality, we assume that the initial time is $t_0 = 0$. Using an input trajectory $u(\cdot)$, a possible solution of (1) is denoted as $\chi\big(t, x(0), u(\cdot)\big)$.

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

$$\mathcal{R}(r) := \left\{ \chi\big(r, x(0), u(\cdot)\big) \,\middle|\, x(0) \in \mathcal{X}^0, \forall t \in [0,r] : \chi\big(t, x(0), u(\cdot)\big) \in \mathcal{X}, u(t) \in \mathcal{U} \right\}.$$

For collision checking, we introduce a relation from the state space of an obstacle to occupied points in $\mathbb{R}^2$.

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

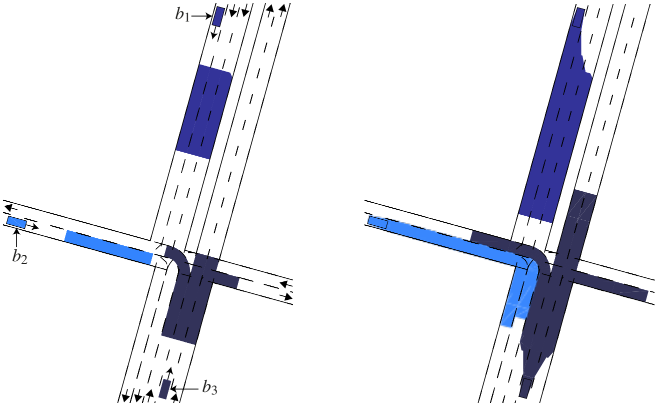
Based on Def. 2, we are able to compute the occupancy of all feasible future behaviors of an obstacle at a point in time t.

Definition 3 (Occupancy Set)**.** *The occupancy set $\mathcal{O}(t) := \mathrm{occ}\big(\mathcal{R}(t)\big)$ describes the set of possibly occupied positions by an obstacle at a given point in time t. For a time interval $[t_1, t_2], t_1 \leq t_2$, we define $\mathcal{O}([t_1, t_2]) = \bigcup_{t_1 \leq t \leq t_2} \mathcal{O}(t)$.*

For each traffic participant (e. g., cars, trucks, motorbikes, bicyclists, and pedestrians), a mathematical model is taken from a database and used to predict all feasible future behaviors in a rigorous and over-approximative way. In order to account for all variations within a class, differential inclusions are used to capture the uncertainty, such as different maximum acceleration and speed.

We consider different layers of assumptions for other traffic participants (cf. Tab. 1) to reduce the size of the occupancy sets by considering only legal behavior, i. e., we initially assume that all traffic participants respect traffic rules. In case an obstacle violates certain assumptions, a less restrictive behavior is assumed individually. Thus, we directly react to misbehavior of other traffic participants.

As an example, we predict three vehicles approaching an uncontrolled intersection. The resulting occupancies $\mathcal{O}(t)$ for different time intervals are shown in Fig. 3. It can be seen that we restrict vehicles from driving backwards (cf. A_{back} in Tab. 1) and from changing to lanes with opposite driving direction (cf. A_{lane} in Tab. 1).



(a) $\mathcal{X}^0$ and $\mathcal{O}([1.5\,\mathrm{s}, 2.0\,\mathrm{s}])$ (b) $\mathcal{O}([0\,\mathrm{s}, 3.0\,\mathrm{s}])$

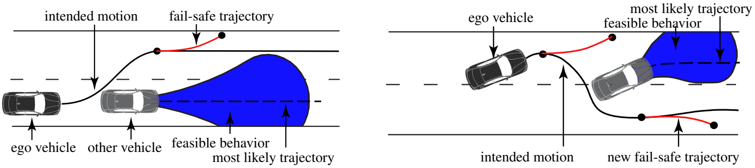
Figure 3: Formal prediction of other traffic participants $\mathcal{B}$. Based on their initial states $\mathcal{X}^0$, we predict all possible occupied regions $\mathcal{O}(t)$ over time.

3.2 Fail-safe Trajectory Planning

Since formal methods consider all eventualities, intended plans might be refused due to a small possibility of failure. However, after initially being unsafe for the entire time horizon, many long-term plans quickly become safe due to the reduced uncertainty about possible behaviors of other traffic participants with new measurements. Thus, we consider two time horizons in parallel as illustrated in Fig. 4a. Intended trajectories provided by the motion planner are used as long-term trajectories, since they typically use less-restrictive assumptions, e. g., other vehicles continue with constant velocity. Please note that those assumptions often do not hold in reality and thus obtained motion plans are not always safe. Therefore, we apply our verification concept only to the first part of the intended long-term reference trajectory of the ego vehicle: We generate safe trajectories by appending fail-safe trajectories (depicted by the red path in Fig. 4a) to the first part of the intended trajectory [26]. The time horizon of this safe trajectory

Table 1: List of motion assumptions based on [25].

Assumption	Description
A_{amax}	Absolute acceleration is limited by $\lvert a_{\max}\rvert$.
A_{back}	Driving backwards in a lane is not allowed, i. e., velocities $v \geq 0$.
A_{vmax}	Positive longitudinal acceleration is stopped when a parameterized speed $v_{\max}$ is reached.
A_{lane}	Changing the lane is only allowed if the new lane has the same driving direction.



(a) Initial Scenario

(b) Future Scenario

Figure 4: We combine the first part of the intended trajectory from the motion planner with our fail-safe trajectory to obtain a safe trajectory which is collision-free with respect to any feasible behavior of obstacles (a). While the ego vehicle moves along its intended trajectory, new fail-safe trajectories are computed (b). If no new valid fail-safe trajectory is found, the ego vehicle must execute the previously computed fail-safe trajectory.

is short, such that the proposed set-based techniques (cf. Sec. 3.1) do not block overly large regions. Fail-safe trajectories ensure that the ego vehicle returns to a safe state in case a safety-critical situation occurs.

Thus, even if other vehicles deviate from the most-likely behavior, as illustrated in Fig. 4b, the ego vehicle remains safe. Remember: Since the previous safe trajectory already considered all possible behaviors of the other traffic participant, it remains safe, even though the situation has changed. In most cases, the planner of the vehicle obtains a new intended trajectory and we are able to generate a new valid safe trajectory so that the previous fail-safe trajectory does not have to be engaged.

In order to obtain our safe trajectories, we have to determine 1) a position along the intended trajectory at which the fail-safe trajectory should start and 2) the optimized fail-safe trajectory itself. For determining the latest possible position to branch off the fail-safe trajectory, we compute simple upper and lower bounds and subsequently use binary search (anytime-properties hold) [27]. To find an optimal fail-safe trajectory, we construct convex optimization problems which separately consider the lateral and longitudinal dynamics of the vehicle [28].

Definition 4 (Convex Trajectory Optimization)**.** *The convex optimization problem for the quadratic cost function $J(x,u):\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ over the time horizon t_h and subject to the set of linear constraints $\mathcal{C}(t)$ on the state and input is defined as*

$$\arg\min_u \int_0^{t_h} J\big((x(t),u(t)\big)dt,\ \ \textit{subject to}\ \forall t:\big(x(t),u(t)\big)\in\mathcal{C}(t).$$

The convex constraint set $\mathcal{C}$ considers the kinematic vehicle model in Fig. 5, which is described with respect to a curvilinear coordinate system aligned to a reference path Γ with orientation θ_Γ. The pose of the ego vehicle is represented by the longitudinal position s, the lateral deviation d, and the orientation θ. Furthermore, $\mathcal{C}$ restricts solutions so that they do not intersect with the predicted occupancy sets of other traffic participants.

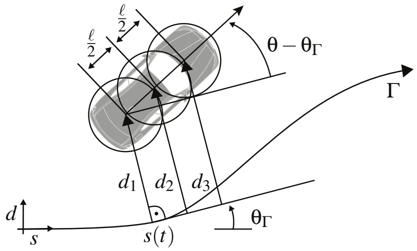


Figure 5: Kinematic vehicle model with respect to a curvilinear coordinate system aligned to the reference path Γ with orientation θ_Γ. The pose of the ego vehicle is described by the longitudinal position s, the lateral deviation d, and the orientation θ.

The usage of convex optimization offers various benefits for the generation of fail-safe trajectories. First of all, we are able to plan trajectories in continuous space. Furthermore, efficient and mature solving techniques for convex optimization problems exist [29]. By punishing high accelerations and jerk in the quadratic cost function J of the longitudinal and lateral optimization problem, we are able to obtain jerk-optimal trajectories. This allows us to provide enhanced comfort for passengers in case the safety-critical situation suddenly resolves (by finding a new fail-safe trajectory along the intended trajectory of the vehicle, as illustrated in Fig. 4b).

3.3 Online Verification of Trajectories

For online verification of the obtained safe trajectory, which corresponds to a combination of the first part of the intended trajectory plus a subsequent fail-safe trajectory (cf. Sec. 3.2), we compute the reachable set of the ego vehicle along this combined trajectory (cf. Fig. 6). In order to determine possible deviations from the planned motion, the reachable sets consider a) the underlying trajectory tracking controller following the motion plan, b) sensor noise and disturbances, and c) model uncertainties [30]. If the ego vehicle deviates from the planned motion, its safety may not be guaranteed anymore. Based on the reachable set of the ego vehicle, we compute the possible occupancies given the dimensions of the ego vehicle.

Definition 5 (Occupancy along Trajectory)**.** *The occupancy $\mathcal{O}_{\text{ego}}(t), t \in [0, t_h]$, describes the occupancy of the ego vehicle along its safe trajectory $x(t), t \in [0, t_h]$, including possible disturbances on the underlying controller.*

Next, we check whether the occupancy of the ego vehicle possibly intersects with the set-based prediction of other traffic participants (cf. Sec. 3.1).

Theorem 1 (Collision-free Safe Trajectory)**.** *The planned safe trajectory $x(t), t \in [0, t_h]$, of the ego vehicle is collision-free with respect to a set of obstacles $b \in \mathcal{B}$, represented by occupancy*

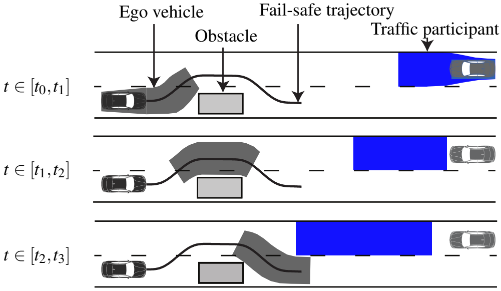


Figure 6: Online verification: the safe trajectory of the ego vehicle is collision-free with respect to other obstacles in the environment if the occupancy set of the ego vehicle does not intersect with the occupancy sets of other obstacles.

sets $O_{\mathcal{B}}(t) := \bigcup_{b\in\mathcal{B}} O_b(t)$ *if*

$$\forall t \in [0, t_h] : O_{\text{ego}}(t) \cap O_{\mathcal{B}}(t) = \emptyset.$$

Proof. The soundness has been shown in [19,30]. □

If there is no intersection with any occupancy of other traffic participants (cf. Fig. 6), the safe trajectory is guaranteed to be collision-free, even if obstacles deviate from the most-likely behavior. Furthermore, the drivability of the safe trajectory is ensured, since we use set-based control techniques to determine possible deviations from the planned motion beforehand [30]. Thus, even if the vehicle deviates from the planned motion due to disturbances, we can still guarantee its safety.

4 Evaluation

Our framework is implemented partly in *C++* and in *Python* and runs on a computer with an Intel i5 1.4GHz processor and 8GB of DDR3 1600MHz memory. We use a highway scenario from the CommonRoad benchmark suite [31] and a recorded urban scenario provided by BMW to demonstrate the benefits of our framework. For both scenarios, we set the maximum absolute acceleration of vehicles to $|a_{\text{max,vehicle}}| = 8\,\text{m}\,\text{s}^{-2}$ and of pedestrians to $|a_{\text{max,pedestrian}}| = 0.6\,\text{m}\,\text{s}^{-2}$.

Fig. 7a shows the initial highway scenario, in which the ego vehicle is following vehicle b_2; used parameters are provided in Tab. 2a. The intended trajectory of the ego vehicle is planned assuming that all other vehicles continue with constant velocity. Fig. 7b shows the predicted occupancies (for improved visibility they are only shown for b_1) and the planned fail-safe trajectory, which lets the ego vehicle swerve to the adjacent shoulder lane to certainly

Table 2: Parameters of the highway and urban scenario.

(a) Highway scenario

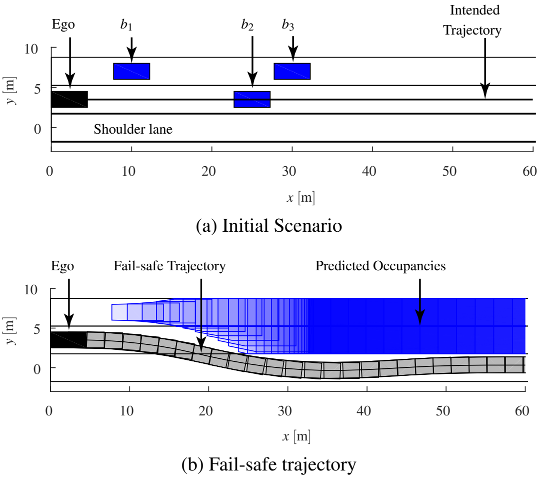
Parameter	Description
Ego vehicle	$(x,y,v)^T_{\text{ego}} = (2.25\,\text{m}, 0\,\text{m}, 23\,\text{m/s})^T$
Vehicle b_1	$(x,y,v)^T_{b_1} = (10\,\text{m}, 7\,\text{m}, 20\,\text{m/s})^T$
Vehicle b_2	$(x,y,v)^T_{b_2} = (25\,\text{m}, 3.5\,\text{m}, 25\,\text{m/s})^T$
Vehicle b_3	$(x,y,v)^T_{b_3} = (30\,\text{m}, 7\,\text{m}, 30\,\text{m/s})^T$
Planning horizon	$t_h = 4.0\,\text{s}, N = 40, \Delta t = 0.1\,\text{s}$

(b) Urban scenario

Parameter	Description
Ego vehicle	$(x,y,v)^T_{\text{ego}} = (6\,\text{m}, -32.5\,\text{m}, 13.8\,\text{m/s})^T$
Pedestrian	$(x,y,v)^T_{b_1} = (8\,\text{m}, -12.5\,\text{m}, 1.4\,\text{m/s})^T$
Planning horizon	$t_h = 5.0\,\text{s}, N = 50, \Delta t = 0.1\,\text{s}$

avoid a collision. The occupancy of the ego vehicle along its safe trajectory is denoted by gray rectangles and does not intersect with the predicted occupancies of other traffic participants at any time. The overall computation time of our framework in this scenario is 16 ms (cf. Tab. 3).

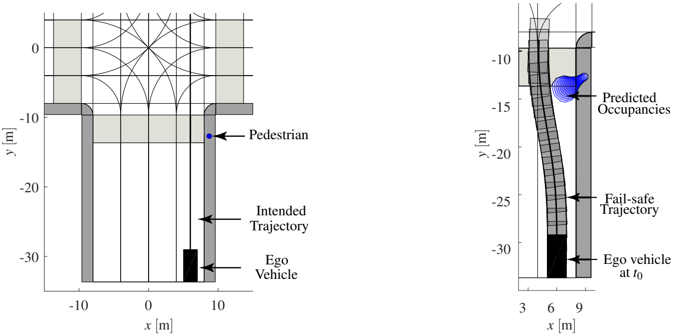
An urban scenario, in which a pedestrian violates traffic rules by suddenly crossing the road during a red light, is shown in Fig. 8a and its parameters are given in Tab 2b. In this scenario, the ego vehicle remains safe by executing our computed fail-safe trajectory (cf. Fig. 8b). Since our occupancies consider any feasible future motion of the pedestrian in an over-approximative way, the ego vehicle is able to avoid a collision. The computation time in this scenario is 14 ms (cf. Tab. 3).



(a) Initial Scenario

(b) Fail-safe trajectory

Figure 7: The ego vehicle follows its leading vehicle b_2 on a highway (a). In case vehicle b_1 suddenly changes to the lane of the ego vehicle, the ego vehicle remains safe by executing the fail-safe trajectory, which initiates a lane change to the adjacent shoulder lane (b).



(a) Initial Scenario

(b) Fail-safe trajectory

Figure 8: The ego vehicle intends to continue straight across the intersection (a). However, if the pedestrian suddenly crosses the lane of the ego vehicle, the ego vehicle remains safe by executing the fail-safe trajectory (b).

5 Conclusions

The proposed framework verifies motion plans online before their execution. Fail-safe plans are provided to withstand component failures or the absence of a timely computed intended motion. Our fail-safe plans are collision-free with respect to any feasible future behavior of other traffic participants and explicitly consider uncertainties originating from sensor noise, disturbances, and modeling uncertainties. We believe that our framework streamlines the development process–motion planning does no longer require certification (e. g., ISO 26262 conformance [2]), this is only required for our proposed safety layer. This significantly reduces the required amount of testing and development costs. Since our framework is independent of the utilized planning framework, it can easily be integrated in existing vehicle frameworks. We have demonstrated the benefits of our framework using a highway and an urban scenario. In a next step, we want to test our framework on a real vehicle.

Table 3: Computation times in the highway and urban scenarios.

Scenario	**Computation times of each component**
Highway	Prediction : 11 ms, Planning : 4 ms, Verification : 1 ms
Urban	Prediction : 5 ms, Planning : 8 ms, Verification : 1 ms

Acknowledgments

This work was partially supported by the BMW Group within the CAR@TUM project and by the German Federal Ministry of Economics and Technology through the research initiative Ko-HAF (`https://www.ko-haf.de/`).

References

[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] International Organization for Stadardization (ISO), "ISO 26262-10:2012 Road vehicles – functional safety," 2012.

[3] 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.

[4] J. Ziegler and M. Werling, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 787–791.

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

[6] C. Ackermann, J. Bechtloff, and R. Isermann, "Collision avoidance with combined braking and steering," *6th Int. Munich Chassis Symposium*, pp. 199–213, 2015.

[7] S. J. Anderson and S. C. Peters, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *Int. Journal of Vehicle Autonomous Systems*, vol. 8, pp. 190–216, 2010.

[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] 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.

[10] 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.

[11] S. Petti and T. Fraichard, “Safe motion planning in dynamic environments,” in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2210—2215.

[12] 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.

[13] D. Althoff, M. Buss, A. Lawitzky, M. Werling, and D. Wollherr, “On-line trajectory generation for safe and optimal vehicle motion planning,” in *Autonomous Mobile Systems*, 2012, pp. 99–107.

[14] 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.

[15] I. M. Mitchell, “Comparing forward and backward reachability as tools for safety analysis,” in *Hybrid systems: computation and control*. Springer, 2007, pp. 428–443.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, “High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning,” in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2156–2162.

[22] E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*. MIT press, 1999.

[23] 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. 1686–1693.

[24] 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 IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2704–2711.

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

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

[27] S. Magdici, Z. Ye, and M. Althoff, “Determining the maximum time horizon for vehicles to safely follow a trajectory,” in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1893–1899.

[28] B. Gutjahr, C. Pek, L. Gröll, and M. Werling, “Efficient trajectory optimization for vehicles using quadratic programming,” *Automatisierungstechnik*, vol. 64, no. 10, pp. 786–794, 2016.

[29] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK; New York: Cambridge University Press, 2004.

[30] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, “Ensuring drivability of planned motions using formal methods,” in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1661–1668.

[31] 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.