Online Verification of Automated Road Vehicles Using Reachability Analysis

Matthias Althoff and John M. Dolan, *Member, IEEE*

***Abstract*—An approach for formally verifying the safety of automated vehicles is proposed. Due to the uniqueness of each traffic situation, we verify safety online, i.e., during the operation of the vehicle. The verification is performed by predicting the set of all possible occupancies of the automated vehicle and other traffic participants on the road. In order to capture all possible future scenarios, we apply reachability analysis to consider all possible behaviors of mathematical models considering uncertain inputs (e.g. sensor noise, disturbances) and partially unknown initial states. Safety is guaranteed with respect to the modeled uncertainties and behaviors if the occupancy of the automated vehicle does not intersect that of other traffic participants for all times. The applicability of the approach is demonstrated by test drives with an automated vehicle of the Robotics Institute at Carnegie Mellon University.**

***Index Terms*—Formal verification, reachability analysis, automated vehicles, autonomous cars, set-based computation.**

I. Introduction

Automated driving will unquestionably provide a variety of benefits. Among them are the reduction of traffic injuries and fatalities, time savings when working in the vehicle, reduction of traffic jams, and mobility for people that previously could not drive. This vision can only be realized if the designers can guarantee that the vehicle will never cause an avoidable crash. In order to meet these high safety requirements, we propose to use formal methods to verify the safety of automated cars. The verification is based on dynamic models that describe possible behaviors of the considered vehicle (ego vehicle) and other surrounding traffic participants. We assume that the uncertainties acting on those models can be chosen large enough to capture all possible behaviors of the real world. If the obtained results are too conservative, one can also provide models that only capture the real behavior up to a user-defined probability. In this work, reachability analysis is used to guarantee the legal safety of planned maneuvers given the aforementioned assumptions, meaning that we guarantee not to cause a collision [1]. Note that it is generally impossible to avoid a collision caused by other traffic participants, e.g., one cannot avoid a collision from behind when one is captured in a traffic jam.

Reachability analysis computes the set of all states reachable when the sets of initial states, sensor measurements, and disturbances are uncertain. Reachable sets of the ego vehicle and other traffic participants make it possible to compute the set of occupied road sections over time. If the occupancy of the ego vehicle does not intersect that of all other relevant traffic participants for all times, safety can be guaranteed. Simulation techniques cannot guarantee safety, since infinitely many possible future scenarios of a traffic scene exist and one can only perform a finite number of simulations.

Simulation techniques can be extended for formal analysis by guaranteeing that simulations starting in a δ-region of the initial state stay in an ϵ-region of the simulation so that a reachable set can be represented by a finite number of simulations, see [2]. All simulation-based approaches have the disadvantage that an exponential number of simulations is required. Considering only the extreme cases requires 2^{n+m+o} simulations, where n is the number of state variables, m is the number of inputs, and o is the number of parameters. Note that time-varying inputs (which may cause resonance) are not even considered by looking only at the extreme cases.

Since every traffic situation is unique, it is necessary that planned maneuvers be constantly verified during the operation of the vehicle, which we call *online verification*. Parts of this computation process can be precomputed and stored in a database, such as time-critical evasive maneuvers. However, it is not possible to store verification results of all possible traffic situations. In order to meet computation time requirements, most previous work in mobile robotics assumes knowledge of the future behavior of other objects in the traffic scene or uses simple models to predict their possible behaviors. The simplest model for unknown holonomic behavior assumes intervals on possible velocities in all directions (e.g. in [3]); more advanced models assume intervals on the acceleration (e.g. in [4]), or both (e.g. in [5]). More complicated non-holonomic models are based on Dubin's car [6], or a tricycle model [7].

All these works assume that the future motion of the ego vehicle is perfectly known, which is a good assumption for slow-moving indoor robots. However, fast maneuvers of automated cars on terrain of varying quality and in varying weather conditions, influence of sensor errors, and the like, require consideration of uncertainties in tracking planned trajectories. It is especially important to consider these uncertainties when systems require certification [8]. Most previous work avoids considering uncertainties in trajectory following due to the inherent challenges in verifying nonlinear dynamic systems with several continuous state variables, as summarized in [9].

Most approaches for nonlinear reachability analysis abstract the nonlinear dynamics to differential inclusions of simpler dynamics, either by simplifying the dynamics within regions of a fixed state space partition [10], [11], resulting in a hybrid (mixed discrete/continuous) system, or by simplification in the vicinity of the reachable set [12]–[14]. The latter approach generally outperforms fixed state space partitions, since it

Matthias Althoff is with the Faculty of Computer Science, Technische Universität München, 85748 Garching, Germany, email: althoff@tum.de

John M. Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, email: jmd@cs.cmu.edu

Manuscript received month day, year; revised month day, year.

does not require the consideration of hybrid dynamics. Approaches which do not use abstraction are mostly based on computationally demanding optimization techniques [15] or on a reformulation of the reachability problem as Hamilton-Jacobi equations, whose solution procedure has exponential complexity in the number of continuous state variables [15]–[17]. When the nonlinear system is monotonic, upper and lower bounds on the reachable set can be easily computed using simulations of corner cases [18], which is used for the model of other traffic participants in the current work. For chemical reaction equations, those upper and lower bounds of the nonlinear system can also be computed efficiently, but a special structure of the dynamics is required [19]. This procedure can still be applied when one can bound the dynamics by monotone systems [20], which is also applied for guaranteed parameter estimation [21].

An alternative to reachability analysis is automated theorem proving, which has been applied to automated cruise control [22]. In that work, it is assumed that all vehicles on the road have to be automated. Additionally, automated theorem proving requires human interaction, see [23, p. 3577], such that it cannot be applied to online verification. The number of required interactions, however, is expected to decrease in coming years.

Constraints for safe vehicle movement, such as avoiding other traffic participants and road boundaries, can also be formulated in a robust model predictive control framework [24]. In model predictive control, an optimal input is computed based on solving an optimal control problem for a finite time horizon, where only the first section of the optimal input trajectory is executed. This procedure is repeated so that the solution adapts to the current situation. In tube-based model predictive control (tube-based MPC), concepts from reachability analysis are mixed with model predictive control. Most of the work on tube-based MPC considers linear systems [25], [26], but concepts for nonlinear systems also exist [27]. However, nonlinear tube-based MPC approaches are computationally too expensive to be used for an online application involving fast dynamics with several state variables, such as the vehicle dynamics of this work.

Another line of work provides formal methods to synthesize trajectories based on temporal logic specifications that are provably correct. In [28] temporal logic specifications are used to specify requirements on missions for unmanned aerial vehicles. Trajectories for automated vehicles in static environments are synthesized in [29] within a discretized environment. A discrete environment is also used in [30] to synthesize plans for teams of robots. Another work synthesizes robotic motion for a point mass (double integrator) by bounding the error to an abstract kinematic model and using the abstraction for the planning task [31].

A completely different paradigm is to analyze planned paths using stochastic methods. Most approaches of this category use Monte-Carlo simulation [32], [33]. A disadvantage of Monte-Carlo simulation is that the computed result differs for the same situation depending on the sampling of possible future scenarios. This is avoided by approaches that compute the stochastic prediction deterministically [34]. Some approaches combine set-based computations as presented in this work with stochastic approaches, such that computationally expensive stochastic dynamics can be restricted to traffic participants for which the occupancy intersects with that of the ego vehicle [35]. However, the set-based computations in [35] are heuristic and thus not applicable to a formal analysis.

The reviewed literature shows that nonlinear continuous systems are usually verified offline due to the complexity of the problem. However, previous work of the authors [36] shows that online verification is theoretically possible when applying the efficient approach first published in [13]. In this work, we present the following innovations compared to [36]:

- The approach is tested on a real vehicle (Cadillac SRX research vehicle of Carnegie Mellon University);
- The vehicle model is validated by real world experiments;
- Instead of only considering the reachable set of the ego vehicle, we also consider the computation of the occupancy of other traffic participants on the road;
- The interaction of the maneuver planner with the verification module is sketched;
- The vehicle controls are modified to fit the interface of the Cadillac SRX;
- The reachability analysis is improved and presented in more detail. Specifically, the computation of the linearization error assumption is now automatically adapted.

The paper is organized as follows: The basic idea of our verification concept is described in Sec. II. Mathematical models of the ego vehicle and other traffic participants are derived in Sec. III. The reachable set computation of the ego vehicle is presented in Sec. IV and the occupancy computation of other traffic participants is described in Sec. V. Results of the test drive are summarized in Sec. VI.

II. Basic Idea and Assumptions

The safety concept presented in this paper is based on the principle that plans are only executed when they are verified for all times. This is achieved by first planning a multidimensional trajectory $\zeta(\cdot)$ the vehicle should follow, where $\zeta(t_f)$ is the reference vector at the final time t_f of the intended plan[1]. Note that the term *trajectory* is used since the reference values are specified over time. In other applications, it is sufficient to follow a set of points, referred to as a *path*. However, paths are not sufficient for many automated maneuvers, such as intersection crossing (one could traverse the intersection arbitrarily slowly), making it necessary to use trajectories [37]. The state of the vehicle $x(t_f)$ might be an inevitable collision state, i.e., a state for which there exists no control action that can possibly avoid a future collision [5], [38], [39]. We prevent inevitable collision states by only accepting intended plans with a subsequent maneuver that brings the vehicle to a stop at a safe location, such that it cannot cause a collision for all future times, see [40, Sec. IV.E]. To focus on the verification aspect, it is assumed that a reference trajectory is already planned by a standard approach (e.g. [40]). Note that any kind of trajectory planner can be combined with the proposed verification scheme.

[1] We use reference trajectory, plan, and planned maneuver interchangeably.

The used trajectory planner should be adapted such that new reference trajectories branch off previous ones at points $x(t_{ver})$ that are reached by the ego vehicle when the verification of the new reference trajectory is completed, as illustrated in Fig. 1. When the verification result is *safe*, the new reference trajectory is chosen, and when it is *unsafe*, the vehicle stays on the previous one. Thus, the braking maneuver leading to the safe stop is only executed if the vehicle repeatedly is not able to find a new safe trajectory. An upper bound of the time for which the new reference trajectory should branch off is easily obtained, since the worst-case verification time is linear in the time required to follow the new reference trajectory t_{exec}, so that $t_{ver} = \nu\, t_{exec}$, where ν is a constant describing the efficiency of the implementation.

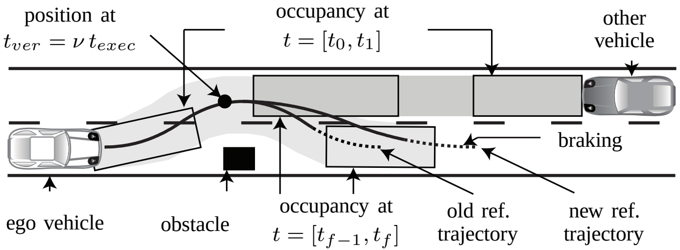


Fig. 1: Verification by checking occupancy intersection.

The verification of each reference trajectory is performed by computing the reachable set of states of the ego vehicle and other traffic participants based on a dynamic model and uncertainties specified by bounded sets. The occupancy of the ego vehicle on the road is determined by considering the size of the vehicle and the projection of reachable sets on position variables and orientation. If, for all times, the occupancy of the ego vehicle does not intersect that of all other traffic participants, and if the drivable area is not exited, the reference trajectory is safe.

An alternative to computing the reachable set of the ego vehicle based on the vehicle dynamics (under consideration of a set of initial states, input trajectories, and a set of parameters), is to simply add a fixed deviation from the reference trajectory. By doing so, one would not distinguish between situations in which a vehicle has to slowly pass through a gap versus those in which a vehicle has to perform an aggressive evasive maneuver. For evasive maneuvers, the deviation from the reference trajectory can easily become more than a meter, as demonstrated in [41]. Even if we increase the occupancy by less than one meter in each direction for the gap scenario in Fig. 2, the safe maneuver will be classified as unsafe, so that the vehicle cannot pass through the gap. Another alternative is to use heuristics to model the dependency of the reachable set on velocity, angular velocity, slip angle, friction coefficient, shape of the reference trajectory, and so on. However, considering all influences is difficult and the result would not be overapproximative and thus not qualify for formal verification and certification.

In order to conclude whether a planned trajectory is safe, several assumptions are made in this work:

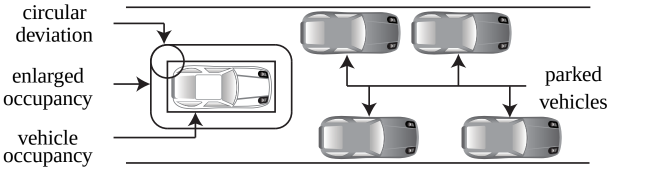


Fig. 2: The ego vehicle intends to pass a narrow gap, which cannot be passed when adding a fixed circular deviation.

1) The vehicle sensors detect all traffic participants relevant for the safety analysis. However, depending on the accuracy of the sensors, one can specify possible uncertainties of measured data.
2) The models that predict the movement of the ego vehicle and other traffic participants enclose all possible real behaviors required to ensure that the ego vehicle does not cause a crash (legal safety [1]). This is achieved by considering bounded, but uncertain, values of sensor noise, disturbances, driver inputs, and uncertain initial states. Note that the time-varying behavior of inputs such as sensor noise and disturbances is arbitrary, as long as the values are within bounded sets.
3) It is assumed that either bounding uncertainties of the sets are chosen large enough to capture all possible values, or that the bounds capture all possible values by a probability bound p_b, e.g. $p_b = 99.999\%$. In the latter case, the verification can only guarantee safety by a certain probability, which depends on the choice of p_b.
4) In order to obtain practical results, we assume that other traffic participants respect traffic rules, as long as no traffic rule violation is detected – corresponding traffic rules are no longer considered once they are violated. Based on this assumption we can guarantee that the ego vehicle does not cause a crash (legal safety [1]).

Given the above assumptions, all possible behaviors are captured by the presented approach, which makes it possible to prove that no collision can occur under the given assumptions. For that reason, we qualify the approach as *formal* to emphasize the rigorousness within the mathematical framework provided by the models.

If reachability analysis or another formal technique was not applied, one would at least require a stability analysis of the trajectory tracker. This, however, is challenging, since the stability analysis depends on the reference trajectory and typically requires finding a Lyapunov function for each reference trajectory (which are infinitely many). Only for special control concepts, such as flatness-based control design, can the dependence on the trajectory be ignored, as long as the model perfectly matches the real behavior. Unfortunately, this is rarely the case due to uncertain parameters (loading of the vehicle, tire-road friction, etc.) and disturbances (road imperfections, wind, slope, etc.) so that the stability analysis of the undisturbed model becomes inconclusive.

III. Mathematical Modeling

This section introduces the dynamic models used for the reachability analysis of the ego vehicle and the occupancy prediction of other traffic participants.

A. Ego Vehicle Model

The vehicle model consists of equations representing the lateral dynamics, the longitudinal dynamics, and the position on the road. All variables of the vehicle are related to the so-called *bicycle model*, which is the standard model for the control design of yaw stabilization systems [42]. The model ignores roll and pitch, such that it suffices to consider only one front and one rear wheel as for a bicycle (see Fig. 3). The authors have shown that effects of high-order models can be captured by the presented low-order model when adding uncertainty [43].

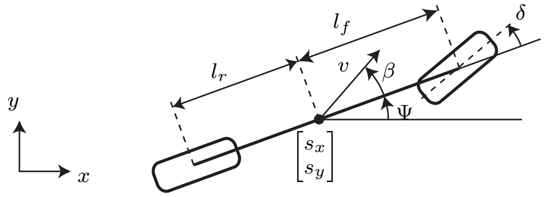


Fig. 3: Bicycle model.

For describing the vehicle dynamics, the cornering stiffnesses C_f, C_r and the distances l_f, l_r form the center of gravity to the axes are introduced, where the indices f and r refer to the front and rear axis. Further, we require the vehicle mass m and the rotational inertia of the yaw axis I_z. The parameter values of the Cadillac SRX are obtained as described in [44] and are listed in Tab. I. The state variables of the bicycle model are the slip angle at the center of mass β, the heading angle Ψ, the yaw rate $\dot{\Psi}$, the velocity v, the x-position s_x, the y-position s_y, and the angle of the front wheel δ, see Fig. 3. Additionally, additive disturbance values y, where the subscript denotes the disturbed variable, are introduced. Those variables model rough roads, wind gusts, and the like. The inputs to the system are the longitudinal acceleration a_x and the rotational speed of the steering angle v_w. The differential equations of the vehicle model are

$$\begin{aligned}
\dot{\beta} &= \Big(\frac{C_r l_r - C_f l_f}{m v^2} - 1\Big)\dot{\Psi} + \frac{C_f}{mv}\delta - \frac{C_f + C_r}{mv}\beta + y_\beta \\
\ddot{\Psi} &= \frac{l_r C_r - l_f C_f}{I_z}\beta - \frac{l_f^2 C_f + l_r^2 C_r}{I_z}\frac{\dot{\Psi}}{v} + \frac{l_f C_f}{I_z}\delta + y_{\dot{\Psi}} \\
\dot{v} &= a_x + y_v \\
\dot{s}_x &= v\cos(\beta + \Psi) + y_{s_x} \\
\dot{s}_y &= v\sin(\beta + \Psi) + y_{s_y} \\
\dot{\delta} &= v_w + y_\delta
\end{aligned} \tag{1}$$

The first two equations describe the lateral dynamics originating from force and moment equilibria due to the lateral tire forces (see [42]). The third equation simply describes the longitudinal dynamics by integrating the commanded longitudinal acceleration to obtain the velocity of the vehicle. Using the kinematics of the vehicle, the derivative of the positions in x- and y-coordinates are obtained by the direction $(\beta + \Psi)$ and absolute value of the velocity v in the fourth and fifth equation. Finally, the front wheel angle is obtained by integration of the commanded steering wheel velocity.

B. Tracking Controller of the Ego Vehicle

The tracking controller in this work provides the commanded steering wheel velocity v_w and the commanded longitudinal acceleration a_x. We use a simple controller with sufficient performance for the driving experiments. The proposed controller is not designed for high performance, but to demonstrate the verification approach. By replacing the equations of the tracking controller, any other control can potentially be considered, as long as the dynamics of the controlled vehicle can be described by ordinary differential equations.

For the tracking controller, we consider a frame that moves along the reference trajectory, such that the x-axis is always tangential and the y-axis is always perpendicular to the reference trajectory, see Fig. 4. Desired values provided by the reference trajectory are denoted by the subscript d. For a concise notation, we introduce the lateral and longitudinal tracking error ϵ_x and ϵ_y:

$$\begin{aligned}
\epsilon_x &= \cos(\Psi_d)(s_{x,d} - s_x) + \sin(\Psi_d)(s_{y,d} - s_y), \\
\epsilon_y &= -\sin(\Psi_d)(s_{x,d} - s_x) + \cos(\Psi_d)(s_{y,d} - s_y).
\end{aligned}$$

A desired front wheel angle is generated by weighting the lateral tracking error and the errors of the yaw angle and rate:

$$\delta_d = \tilde{k}_1\epsilon_y + \tilde{k}_2(\Psi_d - \Psi) + \tilde{k}_3(\dot{\Psi}_d - \dot{\Psi}).$$

The commanded angular velocity of the front wheel is obtained by the proportional control $v_w = k_4(\delta_d - \delta)$. Weighting the longitudinal tracking error and the velocity error results in the commanded longitudinal acceleration:

$$a_x = k_5\epsilon_x + k_6(v_d - v).$$

After introducing the gains $k_i = k_4 \cdot \tilde{k}_i$ for $i \in \{1, 2, 3\}$ and adding sensor noise, which we denote by w and the subscripted disturbed variable, the final control equations are:

$$\begin{aligned}
v_w =& k_1\Big(\cos(\Psi_d)(s_{y,d} - s_y - w_y) - \sin(\Psi_d)(s_{x,d} - s_x - w_x)\Big) \\
&+ k_2(\Psi_d - \Psi - w_\Psi) + k_3(\dot{\Psi}_d - \dot{\Psi} - w_{\dot{\Psi}}) - k_4(\delta - w_\delta), \\
a_x =& k_5\Big(\cos(\Psi_d)(s_{x,d} - s_x - w_x) + \sin(\Psi_d)(s_{y,d} - s_y - w_y)\Big) \\
&+ k_6(v_d - v - w_v).
\end{aligned}$$

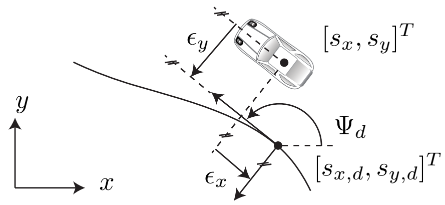


Fig. 4: Moving frame for the used trajectory tracker.

C. Validation of the Ego Vehicle Model

Combining the equations of the vehicle model with those of the tracking controller results in the model of the controlled vehicle. The degree of conformity with real world behavior

TABLE I: Vehicle parameters.

vehicle parameters					
m	I_z	$C_f = C_r$	l_f	l_r	
2273 kg	4423 kg m^2	10.8e4 N/rad	1.292 m	1.515 m	
control parameters					
k_1	k_2	k_3	k_4	k_5	k_6
2	12	4	2	1	10

is shown in Fig. 5 for a double-lane-change maneuver that is formally verified in Sec. VI. It is worth mentioning that double lane change maneuvers are successfully used for validating the lateral dynamics of vehicles, see e.g. [45]–[47]. The plots in Fig. 5 compare the behavior for the yaw angle, the yaw rate, the x- and y-position and the front wheel angle. It can be seen that especially the yaw angle and the position are very well modeled, while the yaw rate and the front wheel angle (which are closely related) show a small deviation due to unmodeled effects such as actuator dynamics and time delay. However, this is no problem for the formal verification, as model mismatches are considered by adding uncertainty.

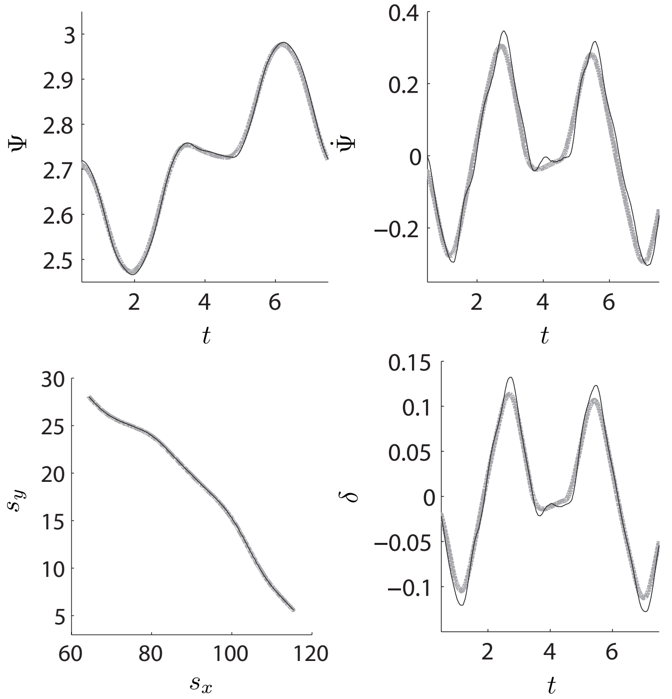


Fig. 5: Comparison of the controlled vehicle model with the data obtained from the double-lane-change driving experiment. The gray line shows the simulation result and the black line the measured data.

D. Model of Other Traffic Participants

The model for other traffic participants is simpler compared to models used for designing trajectory tracking controllers. One reason is that parameters of other traffic participants are typically unknown (unless transmitted via vehicle-to-vehicle communication), so that complicated models requiring identified parameters are useless. The other reason is that the main source of uncertainty is the model input (changing lane, accelerating/decelerating) and not a potential inaccuracy of the dynamic model. We propose a model that satisfies the following constraints:

$C1$: positive longitudinal acceleration is stopped when a parameterized speed $\tilde{v}_{\max}$ is reached ($\tilde{v}_{\max}$ could be set to a certain percentage above the official speed limit).

$C2$: driving backwards in a lane is not allowed.

$C3$: positive longitudinal acceleration is inversely proportional to speed above a parameterized speed v_S (modeling a maximum engine power).

$C4$: maximum absolute acceleration is limited by $\tilde{a}_{\max}$.

$C5$: actions that cause leaving the road/lane/sidewalk/crosswalk boundary are forbidden. Crossing lanes for traffic in the same direction is allowed.

Constraints $C3$ and $C4$ are physical constraints, while the other constraints originate from traffic rules as listed in the Vienna Convention on Road Traffic [48]. The above constraints are considered to be the most important ones describing the uncertain behavior of traffic participants. It should be mentioned that the absence of constraint results in a larger occupancy of other traffic participants and thus only verifies more conservative behaviors of the ego vehicle. Thus, neglecting certain constraints does not result in an unsound verification procedure. This is especially useful since there are many traffic rules and many of them are specific to specific countries. Further rules can be added without requiring changing the basic principles of the presented approach. For other road vehicles, all of the above constraints are potentially active, while e.g. for pedestrians, only constraints $C1$ and $C2$ are enforced and $C5$ is applied to sidewalks and crosswalks instead of road and lane boundaries. When it is sensed that a constraint is violated, it is no longer considered for that particular traffic participant. E.g. when a pedestrian crosses a street where no crosswalk is present, constraint $C5$ is removed and only constraints $C1$ and $C2$ are active. Another example is that when it is sensed that the reversing lights of a vehicle are on, e.g. to start a parallel parking maneuver, constraint $C2$ on driving backwards is removed. The removal of constraints is presented for the considered examples in Sec. VI-B. To describe the system dynamics, we use the same variable symbols as for the ego vehicle, but add a tilde for distinction. The dynamics of other traffic participants are modeled by a point mass:

$$\ddot{\tilde{s}}_x(t) = \tilde{a}_x(t), \quad \ddot{\tilde{s}}_y(t) = \tilde{a}_y(t). \tag{2}$$

In order to restrict $\tilde{a}_x(t)$ and $\tilde{a}_y(t)$ according to the constraints $C1$-$C5$, we introduce unit vectors that point along the longitudinal and lateral directions of the vehicle: $\Phi_{\mathrm{long}}(t) = \frac{1}{\tilde{v}}[\tilde{v}_x(t), \tilde{v}_y(t)]^T$, $\Phi_{\mathrm{lat}}(t) = \frac{1}{\tilde{v}}[-\tilde{v}_y(t), \tilde{v}_x(t)]^T$, where $\tilde{v} = \|[\tilde{v}_x, \tilde{v}_y]^T\|_2$. This makes it possible to formulate $\tilde{a}_x$, $\tilde{a}_y$ by the longitudinal acceleration $\tilde{a}_{\mathrm{long}}(t)$ and the lateral acceleration $\tilde{a}_{\mathrm{lat}}(t)$:

$$\begin{bmatrix}\tilde{a}_x\\ \tilde{a}_y\end{bmatrix} = \Phi_{\mathrm{long}}\tilde{a}_{\mathrm{long}} + \Phi_{\mathrm{lat}}\tilde{a}_{\mathrm{lat}}$$

The lateral acceleration is determined by the maximum absolute acceleration $\tilde{a}_{\max}$ and a normalized steering input u_1, where $u_1 = \pm 1$ represents steering to the left or right using the full tire friction potential:

$$\tilde{a}_{\mathrm{lat}} = \tilde{a}_{\max}\tilde{u}_1.$$

In order to consider constraint $C4$, the remaining acceleration potential in the longitudinal direction is limited to

$$\tilde{a}_{\mathrm{c1,long}} = \sqrt{\tilde{a}_{\max}^2 - \tilde{a}_{\mathrm{lat}}^2}.$$

We further introduce v_S as the speed above which the acceleration is limited by the engine power and no longer by the tire friction so that the acceleration potential becomes inversely proportional to the vehicle speed (see e.g. [33, Sec. II.B.1]). Similarly to the lateral acceleration, we introduce a normalized control input $\tilde{u}_2$ for the longitudinal acceleration, where $\tilde{u}_2 = \pm 1$ represents full braking/acceleration within the acceleration potential. Limited engine power, the restriction to forward driving, and the maximum speed (constraints $C1$-$C3$) are considered by limiting the acceleration to

$$\tilde{a}_{\mathrm{c2,long}} = \begin{cases} \tilde{a}_{\max}\frac{\tilde{v}_S}{\tilde{v}}, & \tilde{v}_S < \tilde{v} < \tilde{v}_{\max} \,\wedge\, \tilde{u}_2 > 0 \\ \tilde{a}_{\max}, & (0 < \tilde{v} \le \tilde{v}_S \,\vee\, (\tilde{v} > \tilde{v}_S \wedge \tilde{u}_2 \le 0)) \\ & \wedge \tilde{v} < \tilde{v}_{\max} \\ 0, & \tilde{v} \le 0 \,\vee\, \tilde{v} \ge \tilde{v}_{\max} \end{cases}$$

Combining $\tilde{a}_{\mathrm{c1,long}}$ and $\tilde{a}_{\mathrm{c2,long}}$ results in the longitudinal acceleration complying with constraints $C1$-$C4$ ($C5$ for road departure is considered later):

$$\tilde{a}_{\mathrm{long}} = \begin{cases} \tilde{a}_{\mathrm{c2,long}}\,\tilde{u}_2, & |\tilde{a}_{\mathrm{c2,long}}\,\tilde{u}_2| \le \tilde{a}_{\mathrm{c1,long}} \\ \tilde{a}_{\mathrm{c1,long}}\,\mathrm{sgn}(\tilde{u}_2), & |\tilde{a}_{\mathrm{c2,long}}\,\tilde{u}_2| > \tilde{a}_{\mathrm{c1,long}}, \end{cases}$$

where sgn() is the sign function. The method for computing the occupancy sets based on this model is presented in Sec. V.

IV. Reachability Analysis of the Ego Vehicle

The behavior of the ego vehicle is uncertain due to sensor noise, disturbances, uncertain initial states and varying parameters. In Fig. 5 it is seen that the model of the controlled vehicle only approximates the real behavior when uncertainties are not considered. To obtain the more general form $\dot{x} = f(x, \zeta, y, w)$ of the controlled vehicle dynamics, the state vector x, the reference vector ζ, the disturbance vector y, and the sensor noise vector w are introduced:

$$\begin{aligned} x &= [\beta,\, \Psi,\, \dot{\Psi},\, v,\, s_x,\, s_y,\, \delta]^T \\ \zeta &= [s_{x,d},\, s_{y,d},\, \Psi_d,\, \dot{\Psi}_d,\, v_d]^T \\ y &= [y_\beta,\, y_\Psi,\, y_{\dot{\Psi}},\, y_v,\, y_{s_x},\, y_{s_y},\, y_\delta]^T \\ w &= [w_x,\, w_y,\, w_\Psi,\, w_{\dot{\Psi}},\, w_v,\, w_\delta]^T \end{aligned}$$

We denote the set of uncertain bounded initial states as $\mathcal{R}(0)$, the set of bounded sensor noise values as $\mathcal{W}$, and the set of uncertain bounded disturbances as $\mathcal{Y}$. The solution to $\dot{x} = f(x, \zeta, y, w)$ for $x(0) = x_0$, $t \in [0, t_f]$, and trajectories $\zeta(\cdot)$, $y(\cdot)$, and $w(\cdot)$ is denoted by $\chi(t, x_0, \zeta(\cdot), y(\cdot), w(\cdot))$. Note that $\zeta(\cdot)$ refers to a trajectory, whereas $\zeta(t)$ refers to the value of the trajectory at time t. The exact reachable set for a reference trajectory $\zeta^*(\cdot)$ and uncertain sets $\mathcal{R}(0)$, $\mathcal{Y}$, and $\mathcal{W}$ is

$$\begin{aligned} \mathcal{R}^e([0, t_f]) = \Big\{ \chi(t, x_0, \zeta(\cdot), y(\cdot), w(\cdot)) \Big| t \in [0, t_f], \\ x_0 \in \mathcal{R}(0), \zeta(t) = \zeta^*(t), y(t) \in \mathcal{Y}, w(t) \in \mathcal{W} \Big\}. \end{aligned}$$

In general, the set of reachable states cannot be computed exactly [49], so that one has to compute overapproximations $\mathcal{R}([0, t_f]) \supseteq \mathcal{R}^e([0, t_f])$. Since reachable sets can be computed efficiently for linear systems (see Sec. IV-B), the given nonlinear equations are linearized in a conservative fashion, i.e., all possible linearization errors are considered, resulting in an overapproximative computation of the reachable set.

A. Conservative Linearization

The reachable set is computed for consecutive time intervals $\tau_k = [t_k, t_{k+1}]$, where $t_{k+1} - t_k = r$ is constant in this work. The advantages of a fixed step size are that the results can be synchronized with the occupancy computations of other vehicles using the same time intervals τ_k and that linearization points can be computed in advance making it possible to parallelize computations as discussed in Sec. IV-D. In order to obtain a concise notation, we introduce the vectors $\tilde{u} = [y^T,\, w^T]^T$ and $z = [x^T,\, \hat{u}^T]^T$ as well as the set $\tilde{\mathcal{U}} = \mathcal{Y} \times \mathcal{W}$. The linearization point is denoted by $z^* = \big[x^*, \tilde{u}^*\big]$, where x^* and $\tilde{u}^*$ are the linearization points of the state and the combined input (disturbance and sensor noise), respectively. Note that the reference trajectory $\zeta(t_k)$ is realized by a zero-order hold such that $\zeta(t_k)$ is constant in the time interval τ_k. For the subsequent derivations, set-based addition and multiplication have to be defined:

$$\begin{aligned} \mathcal{C} \oplus \mathcal{D} :=& \{c + d | c \in \mathcal{C}, d \in \mathcal{D}\}, \\ \mathcal{C} \otimes \mathcal{D} :=& \{c\, d | c \in \mathcal{C}, d \in \mathcal{D}\}, \end{aligned}$$

where c and d are matrices or vectors of proper dimension such that addition and multiplication are defined. Note that the symbol for set-based multiplication is often omitted for simplicity of notation, and that one or both operands can be singletons. In order to avoid parentheses, it is agreed that operations of fixed values have precedence over corresponding set-based operations: $a + b \oplus \mathcal{C} = (a + b) \oplus \mathcal{C}$. This does not apply to different operations, e.g. $a + b \otimes \mathcal{C} \neq (a + b) \otimes \mathcal{C}$.

Using a first-order Taylor expansion around the linearization point z^*, the original differential equation of the i^{th} coordinate is enclosed by the differential inclusion

$$\begin{aligned} &\forall t \in \tau_k : \\ &\dot{x}_i \in f_i(z^*,\, \zeta(t_k)) + \underbrace{\frac{\partial f_i(z, \zeta(t_k))}{\partial z}\Big|_{z=z^*}(z - z^*)}_{=[A(x-x^*)+B(\hat{u}-\hat{u}^*)]_i} \oplus \mathcal{L}_i(\tau_k), \end{aligned} \tag{3}$$

where $\mathcal{L}$ is the set of Lagrange remainders

$$\begin{aligned} \mathcal{L}_i(\tau_k) = \Big\{ \frac{1}{2}(z - z^*)^T \frac{\partial^2 f_i(z, \zeta(t_k))}{\partial z^2}\Big|_{z=\xi}(z - z^*) \\ \Big| \xi \in \mathcal{R}(\tau_k) \times \tilde{\mathcal{U}}, z \in \mathcal{R}(\tau_k) \times \tilde{\mathcal{U}} \Big\} \end{aligned} \tag{4}$$

The Lagrange remainder covers all possible linearization errors when ξ may vary arbitrarily in the set of possible values of x and $\hat{u}$ given by the Cartesian product $\mathcal{R}(\tau_k) \times \tilde{\mathcal{U}}$, see [50]. In this work, we overapproximate $\mathcal{L}(\tau_k)$ in (4) using interval arithmetic [51], which requires the variables z and ξ to be bounded by multidimensional intervals. This is achieved by enclosing the sets $\mathcal{R}(\tau_k)$ and $\tilde{\mathcal{U}}$ by axis-aligned boxes.

The set of linearization errors $\mathcal{L}$ in (4) requires the set of reachable states $\mathcal{R}(\tau_k)$, which in turn requires the set of linearization errors to be computed. This mutual dependence is resolved by assuming a set of linearization errors $\overline{\mathcal{L}}(\tau_k)$ which should be a superset of the exact set of linearization errors $\mathcal{L}(\tau_k)$. In order to obtain tight overapproximations, we use the set of Lagrange remainders $\mathcal{L}(\tau_{k-1})$ of the previously computed time interval τ_{k-1}, which have been obtained by applying interval arithmetic to (4). The new assumption on an overapproximative set of linearization errors of the current time interval τ_k is heuristically obtained by enlarging $\mathcal{L}(\tau_{k-1})$ by a user-defined factor λ:

$$\overline{\mathcal{L}}(\tau_k) = \hat{c} \oplus \lambda(\mathcal{L}(\tau_{k-1}) \oplus (-\hat{c})), \tag{5}$$

where $\hat{c}$ is the volumetric center of $\mathcal{L}(\tau_{k-1})$, which corresponds to the center of mass of a homogeneous solid body. Note that λ is the only ad-hoc assumption that has to be made by the user to obtain the linearization error. In the previous work [36] the assumption $\overline{\mathcal{L}}(\tau_k)$ was fixed for all times, resulting in a larger reachable set. If the assumption does not hold ($\overline{\mathcal{L}}(\tau_k) \not\supseteq \mathcal{L}(\tau_k)$), the verification is aborted and the trajectory is returned as unsafe. Alternatively, one could split the reachable set or enlarge the assumption, but this would result in a non-deterministic time duration of the algorithm, which is in conflict with the proposed framework in Sec. II. When a reference trajectory is returned as unsafe, the previously verified trajectory is executed. Note that the size of $\overline{\mathcal{L}}(\tau_k)$ does not grow constantly, but stabilizes around a certain size since the property $\overline{\mathcal{L}}(\tau_k) \supseteq \mathcal{L}(\tau_k)$ has a shrinking effect and (5) has an enlarging effect. The value of λ can be selected as follows: λ has to be decreased when the reachable set is rapidly overapproximated, and increased when $\overline{\mathcal{L}}(\tau_k) \not\supseteq \mathcal{L}(\tau_k)$.

B. Reachable Set Computation of Linear Systems

Based on the set of linearization errors $\mathcal{L}(\tau_k)$, we compute the reachable set of the linearized system $\dot{x}(t) = Ax(t) + \hat{u}(t)$ in (3), where $\forall t \in \tau_k : \hat{u}(t) \in \hat{\mathcal{U}}(t_k)$ and

$$\begin{aligned}\hat{\mathcal{U}}(t_k) =& f(z^*(t_k), \zeta(t_k)) - Ax^*(t_k)\\ &\oplus B\big(\tilde{\mathcal{U}} \oplus (-\hat{u}^*(t_k))\big) \oplus \overline{\mathcal{L}}(\tau_k).\end{aligned} \tag{6}$$

As a preparation, we split the effect of $\hat{\mathcal{U}}(t_k)$ into its center $\hat{u}_c$ and the translated set $\hat{\mathcal{U}}_\Delta = \hat{\mathcal{U}}(t_k) \oplus (-\hat{u}_c)$. The following algorithm takes advantage of the superposition principle for linear dynamics, see Fig. 6:

1) Starting from $\mathcal{R}(t_k)$, compute the set of all solutions $\mathcal{R}_h(t_{k+1})$ for the affine dynamics $\dot{x} = Ax(t) + \hat{u}_c$ at time t_{k+1}.
2) Obtain the convex hull of $\mathcal{R}(t_k)$ and $\mathcal{R}_h(t_{k+1})$. This encloses all solutions for the current time interval assuming that trajectories from $\mathcal{R}(t_k)$ to $\mathcal{R}_h(t_{k+1})$ are straight lines and that the input is certain ($\hat{\mathcal{U}}_\Delta = 0$).
3) Compute $\mathcal{R}(\tau_k)$ by enlarging the convex hull of 2) to account for the error made by the assumption that trajectories are straight lines and account for the set of uncertain inputs $\hat{\mathcal{U}}_\Delta \neq 0$ (details are explained later).

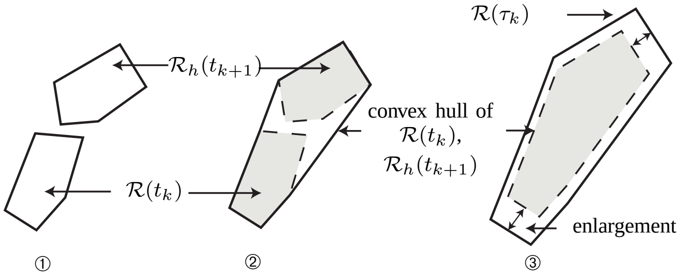


Fig. 6: Steps for the computation of an overapproximation of the reachable set for a linear system.

Using $r = t_{k+1} - t_k$, the solution of $\mathcal{R}_h(t_{k+1})$ is based on the well-known solution of linear time-invariant systems:

$$\mathcal{R}^d_h(t_{k+1}) = e^{Ar}\mathcal{R}(t_k) + \underbrace{\int_0^r e^{A(r-t)}\, dt\, \hat{u}_c}_{=:x_p(r)}.$$

If A is invertible, $x_p(r)$ can be computed as $A^{-1}(e^{Ar} - I)\hat{u}_c$, where I is the identity matrix. However, since A is not always invertible, we compute $x_p(r)$ by integrating the Taylor series of $e^{Ar} = \sum_{i=0}^{\infty}(Ar)^i/(i!)$ for up to η Taylor terms, where η can be set by the user. In order to account for higher order Taylor terms, an interval matrix $\mathcal{E}_p(r) := [-W(r)\,r, W(r)\,r]$ is introduced, whose symmetric bounds $-W(r)\,r$ and $W(r)\,r$ are computed according to [52], so that the particular solution $x_p(r)$ is bounded by

$$x_p(r) \in \underbrace{\Big(\sum_{i=0}^{\eta} \frac{A^i r^{i+1}}{(i+1)!} \oplus \mathcal{E}_p(r)\Big)}_{=:\Gamma(r)} \otimes \hat{u}_c. \tag{7}$$

The reachable set due to the uncertain and convex input $\hat{\mathcal{U}}_\Delta$ is obtained as derived in [52]:

$$\mathcal{R}_p(r) = \bigoplus_{i=0}^{\eta}\Big(\frac{A^i\, r^{i+1}}{(i+1)!} \otimes \hat{\mathcal{U}}_\Delta\Big) \oplus \big([-W(r)\,r, W(r)\,r] \otimes |\hat{\mathcal{U}}_\Delta|\big), \tag{8}$$

where the absolute value of a set of vectors $\mathcal{M}$ is defined elementwise as $|\mathcal{M}|_i := \sup\big\{|m_i| \big| m \in \mathcal{M}\big\}$. The enlargement required to bound all affine solutions within τ_k is denoted by $\mathcal{R}_\epsilon$ and is computed as in [53, Chap. 3.2]. The reachable set for the next point in time and time interval is obtained by combining all previous results and using the operator $\mathrm{co}(\cdot)$ for the convex hull:

$$\begin{aligned}\mathcal{R}(t_{k+1}) =& e^{Ar}\mathcal{R}(t_k) \oplus \Gamma(r)\hat{u}_c \oplus \mathcal{R}_p(r),\\ \mathcal{R}(\tau_k) =& \mathrm{co}\big(\mathcal{R}(t_k), e^{Ar}\mathcal{R}(t_k) \oplus \Gamma(r)\hat{u}_c\big) \oplus \mathcal{R}_\epsilon \oplus \mathcal{R}_p(r)\end{aligned} \tag{9}$$

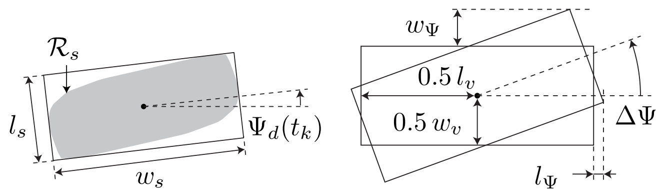
C. Set of Occupied Positions

The reachable set makes it possible to compute the set of positions $\mathcal{OC}(\tau_k)$ occupied by the vehicle on the road for each time interval τ_k. In a first step, the reachable set $\mathcal{R}(\tau_k)$ is projected onto the set of possible positions $\mathcal{R}_s$ (2-dimensional) and orientations $\mathcal{R}_\Psi$ (1-dimensional). Next, we enclose the position set by a rectangle $\mathcal{R}_s$ oriented in the direction of the reference trajectory, i.e., it is rotated by $\Psi_d(t_k)$ from the x-axis (see Fig. 7). The enclosing rectangle has length l_s and width w_s. In a next step, the width and length of the rectangle are enlarged by the dimensions of the vehicle, which has length l_v and width w_v. Finally, the width and length of the rectangle enclosing the occupation have to be enlarged by l_Ψ and w_Ψ due to the uncertain orientation $\mathcal{R}_\Psi$ (see Fig. 7). Using $\Delta\Psi = \max_{\Psi^* \in \mathcal{R}_\Psi}(|\Psi^* - \Psi_d|)$, the enlargement is

$$l_\Psi = 0.5|(1 - \cos(\Delta\Psi))l_v - \sin(\Delta\Psi)w_v|,$$
$$w_\Psi = 0.5|(1 - \cos(\Delta\Psi))w_v - \sin(\Delta\Psi)l_v|.$$

The final dimensions of the enclosing rectangle are:

$$l = l_s + l_v + l_\Psi, \quad w = w_s + w_v + w_\Psi.$$



(a) Uncertain center of mass. (b) Uncertain orientation.

Fig. 7: Enlargement of the vehicle occupation due to uncertain orientation and position.

D. Parallelization

In order to decrease the computation time of the reachability analysis, we attempt to parallelize as many computations as possible. In this work, a fixed step size is chosen to compute the required linearizations in advance. The linearization points are selected along the solution $\chi(t, x_{0,c}, \zeta(\cdot), y_c, w_c)$ starting at the center of the initial set $x_{0,c}$ subject to the constant disturbance y_c and constant sensor noise w_c, which are the centers of the sets $\mathcal{Y}$ and $\mathcal{W}$, respectively. The linearization points are pre-selected for constant time steps as $z^*(t_k) = [\chi^T(t_k, x_{0,c}, \zeta(\cdot), y_c, w_c),\, y_c^T,\, w_c^T]$, which makes it possible to linearize the system dynamics and also pre-compute e^{Ar} ($r = t_{k+1} - t_k$), $\mathcal{E}_p(r)$, $\Gamma(r)$, and $\mathcal{R}_\epsilon$. Thus, for each time interval, only the computations remain that require the reachable set of the previous time step.

An alternative is to use an adaptive step size. The potentially variable step size is chosen such that the error $\mathcal{R}_\epsilon$ in between the linear interpolation of two points in time t_i and t_{i+1} has a constant ratio to the size of the reachable set. In other works, the relative error rather than the absolute error is controlled [54], [55]. Given the computation of $\mathcal{R}_\epsilon$, a variable time increment $r(t_k) = \|A^2(t_k)\|_\infty^{-0.5}$ approximately keeps the ratio of the size of $\mathcal{R}_\epsilon(\tau_k)$ to $\mathcal{R}(\tau_k)$ constant, see e.g. [56]. The time increment for the scenario considered in Sec. VI-A (including the attached braking maneuver) varies from 0.0071 to 0.0189 seconds. The performance gain from this small variation does not exceed that from pre-computing required operations. Another advantage of fixed step size is that the occupancies can be more easily synchronized with other traffic participants when a common time step is used, which is the main argument for choosing a fixed time step.

The required operations for the reachable set computation of the ego vehicle are summarized in Alg. 1, where ParFor loops can be executed in parallel, i.e., each loop can be computed independently. Additionally, the following loops can be computed in separate threads: Loop α (line 5-8), loop β (line 9-17), and loop γ (line 18-20). Note that loop β can only compute the time step k if the preceding loop α has already returned results for this time step or a higher one. The same argument holds for loop γ and β.

Algorithm 1 `occupancyEgo`($\mathcal{R}(0), t_f, ...$)

```
Require: System dynamics $f(x,\zeta,y,w)$, initial set $\mathcal{R}(0)$,
    disturbance set $\mathcal{Y}$, sensor noise set $\mathcal{W}$, reference trajectory
    $\zeta(\cdot)$, time horizon $t_f$, time step $r$, factor $\lambda$
Ensure: $\mathcal{OC}(\tau_k)$
 1: $N = t_f/r$
 2: for $k = 1 \dots N$ do
 3:     compute $z^*(t_k) = [\chi^T(t_k, x_{0,c}, \zeta(\cdot), y_c, w_c),\, y_c^T,\, w_c^T]$
 4: end for
 5: parfor $k = 1 \dots N$ do
 6:     $A_k, B_k, \hat{\mathcal{U}}_k \leftarrow \mathrm{linearize}(f(x,\zeta,y,w), z^*(t_k))$
 7:     compute $e^{A_k r}$, $\mathcal{E}_p(r)$, $\Gamma_k(r)$ (see (7)), $\mathcal{R}_{k,\epsilon}$ (see [53])
 8: end parfor
 9: for $k = 1 \dots N$ do
10:     $\overline{\mathcal{L}}(\tau_k) = \hat{c} \oplus \lambda(\mathcal{L}(\tau_{k-1}) \oplus (-\hat{c}))$ (see (5))
11:     $\hat{\mathcal{U}}(t_k) = f(z^*(t_k), \zeta(t_k)) - A_k x^*(t_k)$
            $\oplus B_k\big(\hat{\mathcal{U}} \oplus (-\hat{u}^*(t_k))\big) \oplus \overline{\mathcal{L}}(\tau_k)$ (see (6))
12:     compute $\mathcal{R}_p(r)$ (see (8))
13:     $\mathcal{R}(t_{k+1}) = e^{A_k r}\mathcal{R}(t_k) \oplus \Gamma_k(r)\hat{u}_c \oplus \mathcal{R}_p(r)$ (see (9))
14:     $\mathcal{R}(\tau_k) = \mathrm{co}\big(\mathcal{R}(t_k), e^{A_k r}\mathcal{R}(t_k) \oplus \Gamma_k(r)\hat{u}_c\big)$
            $\oplus\mathcal{R}_{k,\epsilon} \oplus \mathcal{R}_p(r)$ (see (9))
15:     compute $\mathcal{L}(\tau_k)$ (see (4)); abort if $\mathcal{L}(\tau_k) \nsubseteq \overline{\mathcal{L}}(\tau_k)$
16:     reduce set representation of $\mathcal{R}(t_{k+1})$ (see [2])
17: end for
18: parfor $k = 1 \dots N$ do
19:     compute $\mathcal{OC}(\tau_k)$ based on $\mathcal{R}(\tau_k)$ (see Sec. IV-C)
20: end parfor
```

V. Occupancy of Other Traffic Participants

The occupancy of other traffic participants based on all possible modeled behaviors is computed differently compared to the ego vehicle. In theory, one could also apply reachability analysis and project onto the position and orientation variables to obtain the occupancy. However, the dynamics of the model for other traffic participants is monotone under certain conditions and the occupancy can be exactly computed by constraining only the absolute acceleration. Due to these two properties, we propose a new method that directly computes

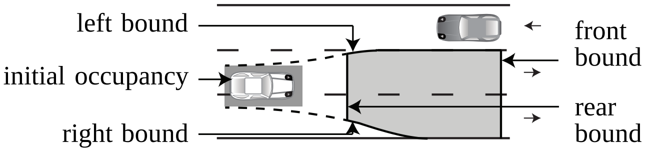


Fig. 8: Initial occupancy and boundaries of the occupancy set for a long time interval.

the occupancy without the need to compute reachable sets of auxiliary variables. The resulting algorithm is much faster than for the ego car, which is important, since one typically has to consider several other traffic participants and only one ego vehicle. The consecutive time intervals τ_k used for the prediction of the ego vehicle are identically used for other traffic participants.

In order to obtain fast and accurate occupancy predictions, we compute different occupancy sets for different abstractions of the dynamic model. We show that intersecting those sets returns an overapproximation of the exact occupancy, which is formalized by introducing the projection operator $\mathrm{proj}()$ of a set and an operator $\mathrm{reach}()$ returning the reachable set of a model M_i.

Proposition V.1 (Overapproximative Occupancy) *Given are models M_i, $i = 1 \dots m$ which are abstractions of model M_0, i.e., $\mathrm{reach}(M_0) \subseteq \mathrm{reach}(M_i)$. The occupancy of the model M_0 can be overapproximated by*

$$\mathrm{proj}\big(\mathrm{reach}(M_0)\big) \subseteq \bigcap_{i=1}^{m} \mathrm{proj}\big(\mathrm{reach}(M_i)\big). \qquad \square$$

Proof: Since $\mathrm{reach}(M_0) \subseteq \mathrm{reach}(M_i)$, we have that

$$\mathrm{reach}(M_0) \subseteq \bigcap_{i=1}^{m} \mathrm{reach}(M_i)$$
$$\rightarrow \mathrm{proj}\big(\mathrm{reach}(M_0)\big) \subseteq \mathrm{proj}\Big(\bigcap_{i=1}^{m} \mathrm{reach}(M_i)\Big)$$

Further, it is shown in [56, Prop. 1] that

$$\mathrm{proj}\Big(\bigcap_{i=1}^{m} \mathrm{reach}(M_i)\Big) \subseteq \bigcap_{i=1}^{m} \mathrm{proj}\big(\mathrm{reach}(M_i)\big).$$

■

We propose two abstractions: The first abstraction allows the vehicle to move arbitrarily in the lateral direction, but considers the longitudinal dynamics along a path (see Sec. V-A). This abstraction provides the rear and front bound of the occupancy set in the driving direction, see Fig. 8. The second abstraction provides the left and right bound in Fig. 8 by considering limited absolute acceleration and by not allowing behavior that results in leaving the drivable area (see Sec. V-B).

A. Occupancy Along Road Boundaries

In this subsection, we use the abstraction that vehicles move along paths while considering constraints $C1$-$C4$, where the lateral positions are arbitrary within lane boundaries (see Fig. 9). The goal of this abstraction is to obtain the rear and front bound, as shown in Fig. 8. The considered paths are initially assumed as centers of lanes, where the position along a path is specified by a function $[\tilde{s}_x,\, \tilde{s}_y]^T = p(\tilde{s}_{lon})$ of the path coordinate $\tilde{s}_{lon}$. The effect of cutting corners is not yet considered in this work. Because of the restricted movement along a path, the normalized steering input $\tilde{u}_1$ is no longer a control input to the vehicle. The state vector for the movement along a path reduces to $\tilde{x} = [\tilde{s}_{lon},\, \tilde{v}]^T$. Due to this abstraction, the longitudinal dynamics are monotone:

Definition V.1 (Monotone dynamics; see [57]) *A system is monotone with respect to the initial state $x(0) \in \mathcal{R}(0)$ and inputs $u(t) \in \mathcal{U}$ when the following property holds for the solution $\chi(t, x(0), u(\cdot))$:*

$$\text{if } \forall i, j, t \geq 0 : x_i(0) \leq \bar{x}_i(0), u_j(t) \leq \bar{u}_j(t) \text{ then}$$
$$\forall i, t \geq 0 : \chi_i(t, x(0), u(\cdot)) \leq \chi_i(t, \bar{x}(0), \bar{u}(\cdot)). \qquad \square$$

A constructive method to prove monotonicity is presented in [57], which returns monotonicity with respect to $\tilde{x}$ and $\tilde{u}_2$ ($\tilde{u}_1$ is no longer an input). Thus, the front bound on the path coordinate can be computed as follows: Start at the maximum initial position and velocity (within the set of possible initial states) and apply full possible acceleration. Obtaining the front bound on the acceleration along a curved path considering C4 requires solving an optimization problem for which a fast semi-analytical method exists [58]. The optimal solution is a bang-bang control, i.e., $\tilde{u}_2$ takes only the values -1 or 1 since the input is already normalized. The rear bound is obtained by applying the full deceleration potential $\sqrt{\tilde{a}_{\max}^2 - \tilde{a}_{\mathrm{lat}}^2}$.

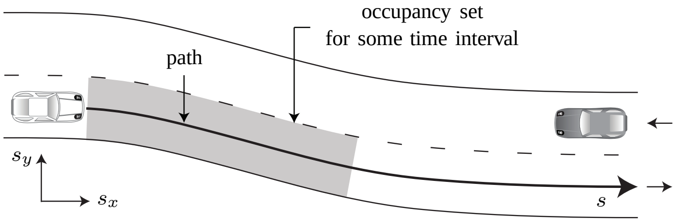


Fig. 9: Occupancy along road boundaries.

B. Occupancy Towards Road Boundaries

Computing the occupancy when the movement is not restricted along a path is much more challenging, since there does not exist a single trajectory that defines the boundary for all times. This is demonstrated in Fig. 10, where simulations for different orientations of the vehicle-fixed acceleration vector are plotted while the absolute value is always $\tilde{a}_{\max}$. The angle $\phi = 90°$ corresponds to a left turn without longitudinal acceleration, whereas $\phi = 180°$ corresponds to full braking without steering. It can be seen that for different times, solutions of different acceleration orientations ($\phi \in \{90°, 110°, 130°\}$) define the border of these 3 solutions. Note that even the union of positions for all acceleration directions is only a subset of the occupancy set, because the acceleration direction is allowed to be time-varying.

To simplify the analysis for the movement on the plane, we restrict ourselves to constraints $C4$ and $C5$ in this setting. This

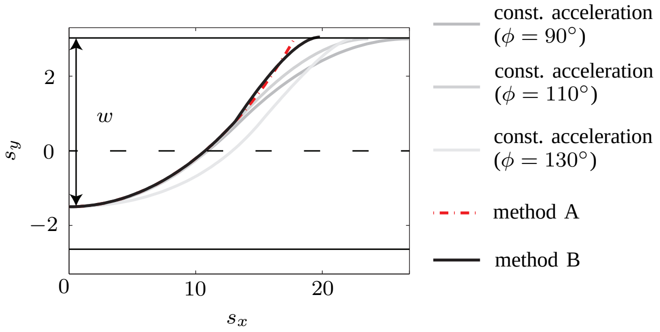


Fig. 10: Occupancy boundary for changing to the left lane.

makes it possible to apply the road-fixed acceleration inputs $\tilde{a}_x(t)$ and $\tilde{a}_y(t)$ as shown in (2), resulting in monotone dynamics, whereas the dynamics of the vehicle-fixed acceleration inputs $\tilde{a}_{\mathrm{lat}}(t)$ and $\tilde{a}_{\mathrm{long}}(t)$ are not monotone. We first consider straight roads with uncertain initial states, where each state variable is bounded by an interval and the x-axis is aligned with the road direction. Due to monotonicity of (2), the left (l) and right (r) occupancy boundary is obviously obtained by starting at

$$\tilde{x}_l(0) = \left[\underline{\tilde{s}}_x(0), \quad \overline{\tilde{s}}_y(0), \quad \underline{\tilde{v}}_x(0), \quad \overline{\tilde{v}}_y(0)\right]^T,$$
$$\tilde{x}_r(0) = \left[\underline{\tilde{s}}_x(0), \quad \underline{\tilde{s}}_y(0), \quad \underline{\tilde{v}}_x(0), \quad \underline{\tilde{v}}_y(0)\right]^T,$$

where under- and overlines represent respectively the lower and upper limits of initial states. This is indicated for uncertain initial positions in Fig. 8. Based on the worst-case initial states, we first compute the occupancy when only the absolute acceleration is limited (constraint $C4$), which we refer to as method A. Note that other constraints, which e.g. forbid leaving the road/lane boundary, are not yet considered. Further constraints are separately considered according to Prop. V.1 so that e.g. in a post-processing step the occupancies beyond road/lane boundaries are cut off. In this setting, the occupancy of the vehicle can be described by circles with center $\tilde{c}(t)$ and radius $\tilde{r}(t)$ when the initial position and velocity are known [4]:

$$\tilde{c}(t) = \begin{bmatrix}\tilde{s}_x(0)\\ \tilde{s}_y(0)\end{bmatrix} + \begin{bmatrix}\tilde{v}_x(0)\\ \tilde{v}_y(0)\end{bmatrix} t, \quad \tilde{r}(t) = \frac{1}{2}\tilde{a}_{\max} t^2.$$

For computing the occupancy we are interested in the boundary that encloses all possible circles:

Proposition V.2 (Boundary of Occupancy) *Without loss of generality, we choose $\tilde{s}_x(0) = 0$, $\tilde{s}_y(0) = 0$, $\tilde{v}_x(0) = v_0$, and $\tilde{v}_y(0) = 0$. The x- and y-coordinate of the boundary are:*

$$\tilde{b}_x(t) = v_0 t - \frac{\tilde{a}_{\max}^2 t^3}{2v_0}, \quad \tilde{b}_y(t) = \sqrt{\frac{1}{4}\tilde{a}_{\max}^2 t^4 - \left(\frac{\tilde{a}_{\max}^2 t^3}{2v_0}\right)^2}.$$

□

Proof: To simplify the proof we introduce the new variable $\hat{b}_x(t) = \tilde{b}_x(t) - v_0\, t$. The possible x- and y-positions of the two circles with radius $\tilde{r}(\cdot)$ at time t and $t + \Delta t$ are:

$$\hat{b}_x^2 + \tilde{b}_y^2 = \tilde{r}^2(t), \qquad (10)$$
$$(\hat{b}_x - v_0\Delta t)^2 + \tilde{b}_y^2 = \tilde{r}^2(t + \Delta t). \qquad (11)$$

Inserting $\tilde{b}_y^2 = \tilde{r}^2(t) - \hat{b}_x^2$ from (10) into (11) and some rewriting results in the x-coordinate of their intersection:

$$\hat{b}_x = \frac{\tilde{r}^2(t) - \tilde{r}^2(t + \Delta t)}{2v_0\Delta t} + \frac{1}{2}v_0\Delta t. \qquad (12)$$

Using $\tilde{r}(t) = \frac{1}{2}\tilde{a}_{\max}t^2$, we obtain after some calculations

$$\tilde{r}^2(t) - \tilde{r}^2(t + \Delta t) = (-\tilde{a}_{\max}^2 t^3 + \mathcal{O}(\Delta t))\Delta t,$$

where $\mathcal{O}(\Delta t)$ includes linear and higher-order terms of Δt. Inserting the above result into (12) and computing the limit for $\Delta t \to 0$ results in $\hat{b}_x(t)$ and thus in $\tilde{b}_y(t)$ using (10). ■

The occupancy $\mathcal{OC}(t_k)$ for specific points in time t_k as well as the left and right boundary are plotted in Fig. 11 for the initial velocity $v_0 = 20$ m/s and $a_{\max} = 10$ m/s^2. It is obvious that the result allows behaviors that result in driving backwards, which is resolved by setting $b_x(t) = b_x(t^*)$ after time $t^* = v_0/a_{\max}$, for which it is no longer ensured that the vehicle has not come to a stop.

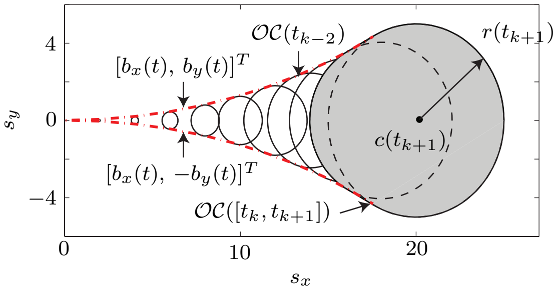


Fig. 11: Occupancy sets according to method A.

It is not yet considered that the lateral acceleration of other vehicles has to change direction when approaching the road boundary to avoid crossing it (constraint $C5$), which results in an unnecessary overapproximation; see the dash-dotted line in Fig. 10. A simple solution that exactly considers constraint $C4$ and $C5$ is yet unknown, but we can propose a solution that overapproximatively considers $C4$ and $C5$, which we refer to as method B. When one neglects the longitudinal dynamics, one obtains the time for switching the acceleration direction to avoid a straight road boundary from a lateral distance w (see Fig. 10) and a lateral initial velocity $v_{0,lat}$ as

$$t_s = \frac{\sqrt{\tilde{a}_{\max} w + \frac{1}{2}v_{0,lat}^2} - v_{0,lat}}{\tilde{a}_{\max}} \qquad (13)$$

for $\sqrt{\tilde{a}_{\max} w + \frac{1}{2}v_{0,lat}^2} - v_{0,lat} \geq 0$, otherwise leaving the road cannot be prevented. The proof is straightforward and omitted due to space limitations. For curved roads the time t_s is obtained by constructing an artificial straight road boundary close to where the occupancy boundary hits the road boundary (see Fig. 12) between the points P_1 and P_2. Prop. V.2 is used to obtain P_1 and P_2 is obtained by computing a feasible solution that touches the road boundary, thus the overapproximation has to lie in between the overapproximation P_1 and the underapproximation P_2. The artificial straight road boundary is obtained by connecting P_1 and P_2 and pushing the line outside using binary search.

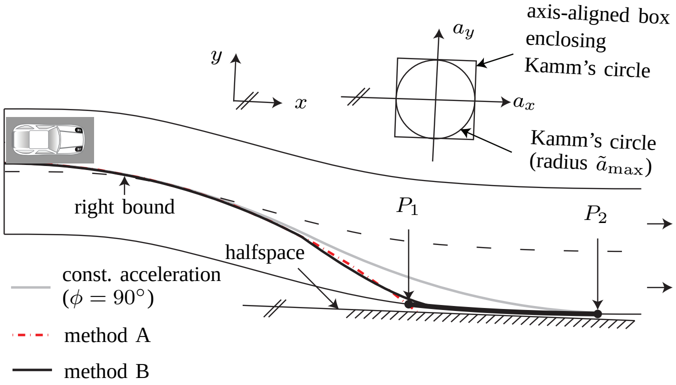

axis-aligned box enclosing Kamm's circle
a_y
a_x
y
x
Kamm's circle (radius $\tilde{a}_{\max}$)
right bound
P_1
P_2
halfspace
const. acceleration ($\phi = 90°$)
method A
method B

Fig. 12: Right boundary of the occupancy on a curved road.

The first part of the occupancy boundary of method B until time t_s is computed as for method A (see Prop. V.2). For the second part of method B, we use a coordinate system where the x-axis is aligned with the artificial straight road boundary. In addition we abstract the original model such that accelerations can take values within a box aligned by the new coordinate system, which encloses all accelerations bounded by the absolute acceleration $\tilde{a}_{\max}$, see Fig. 12. This makes it possible to obtain a worst-case initial state $\hat{x}(t_s)$ for the second phase:

$$\hat{x}(t_s) = \left[\tilde{b}_x(t_s), \quad \tilde{b}_y(t_s), \quad v_{0,x} - \tilde{a}_{\max} t_s, \quad v_{0,y} \pm \tilde{a}_{\max} t_s\right]^T.$$

The initial position is obviously the final position of the first phase and the initial velocity is based on monotonicity: $v_{0,x} - \tilde{a}_{\max} t_s$ is the lowest possible velocity in the x-direction and $v_{0,y} \pm \tilde{a}_{\max} t_s$ is the highest/lowest possible velocity in the y-direction for the left/right bound. Note that the initial state $\hat{x}(t_s)$ is not reachable, but provides a worst-case initial state for the second phase of method B. Once the road/lane boundary is reached, the left/right occupancy bound coincides with the corresponding road/lane boundary. The results of method A and B are compared in Fig. 10, where method B performs better close to reaching the road boundary.

The overall algorithm as described in Alg. 2 works as depicted in Fig. 8. First, the left and right bounds are computed independently using method A or B for the entire time horizon. Note that the bounds continue along the road/lane boundaries once those boundaries are reached. Next, the set in between the left and right bound, denoted by $\mathcal{OC}_{compl}$, is chopped for each time interval τ_k to extract the occupancies for the current time interval, see Fig. 8. The chopping operation is performed separately for longitudinal, left, and right direction. The operation in longitudinal direction is denoted by $\mathrm{chop}_{\mathrm{long}}(\mathcal{OC}_{compl}, \underline{\tilde{s}}_{lon}(t_k), \overline{\tilde{s}}_{lon}(t_k), \mathrm{path})$, where $\underline{\tilde{s}}_{lon}(t_k), \overline{\tilde{s}}_{lon}(t_k)$ are the combinations of lower and upper bounds along the path denoted by path. The chopping for the left border is denoted by $\mathrm{chop}_{\mathrm{lat}}(\mathcal{OC}_{compl}, \tilde{s}_l(t_k), \mathcal{H}_l)$, where $\tilde{s}_l(t_k)$ is the orthogonal distances to the halfspace $\mathcal{H}_l = \{x | n_l^T x \leq d_l\}$, where n_l is the normal and d_l the distance to the origin. The operation moves $\mathcal{H}_l$ in normal direction by the distance $\tilde{s}_l(t_k)$ and computes a set difference with $\mathcal{OC}_{compl}$. The chopping operation for the right side is analogous. Note that the chopping in the lateral direction is not required in Fig. 8 since the lane boundaries are already reached. The distances $\tilde{s}_l(t_k)$, $\tilde{s}_r(t_k)$ are obtained by computing the analytical solution of the double integrator model (see (2)). Note that as an overaproximation for the lateral chopping it is assumed that the vehicle can accelerate in the direction of the corresponding halfspace normal with $\tilde{a}_{\max}$ regardless of the acceleration in perpendicular direction as illustrated by the boxed acceleration circle in Fig. 12.

Algorithm 2 `occupancyOther`$(\mathcal{R}(0), t_f, ...)$

```
Require: Initial set $\mathcal{R}(0)$, time horizon $t_f$, time step $r$,
    parameters $\tilde{a}_{\max}$, $\tilde{v}_{\max}$, $v_S$, halfspaces $\mathcal{H}_l$, $\mathcal{H}_r$, path
Ensure: $\mathcal{OC}_{other}(\tau_k)$
$N = t_f / r$
Set $\underline{\tilde{s}}_{lon}(t_0)$, $\overline{\tilde{s}}_{lon}(t_0)$, $\tilde{s}_l(t_k)$, $\tilde{s}_r(t_k)$ from $\mathcal{R}(0)$
Compute left/right halfspaces $\mathcal{H}_l, \mathcal{H}_r$ (see Fig. 12)
Compute $t_{s,l}, t_{s,r}$ for left/right bound (see (13))
Compute $\mathcal{OC}_{compl}$ using method $A$ or $B$
for $k = 1 \dots N$ do
    Compute $\underline{\tilde{s}}_{lon}(t_k)$, $\overline{\tilde{s}}_{lon}(t_k)$ according to [58]
    $\mathcal{OC}_{other}(\tau_k) = \mathrm{chop}_{\mathrm{long}}(\mathcal{OC}_{compl}, \underline{\tilde{s}}_{lon}(t_k), \overline{\tilde{s}}_{lon}(t_k))$
    if $\tilde{s}_l(t_k) < 0$ then
        $\tilde{u}_r = -1$ for $t < t_{s,r}$, $\tilde{u}_r = 1$ otherwise
        $\underline{\tilde{v}}_{lat}(t_k) = \underline{\tilde{v}}_{lat}(t_{k-1}) + \tilde{a}_{\max} \tilde{u}_r$
        $\underline{\tilde{s}}_{lat}(t_k) = \underline{\tilde{s}}_{lat}(t_{k-1}) + \underline{\tilde{v}}_{lat}(t_{k-1}) r + \tilde{a}_{\max} \tilde{u}_r \frac{r^2}{2}$
        $\mathcal{OC}_{other}(\tau_k) = \mathrm{chop}_{\mathrm{lat}}(\mathcal{OC}_{other}(\tau_k), \tilde{s}_l(t_k), \mathcal{H}_l)$
    end if
    if $\tilde{s}_r(t_k) < 0$ then
        $\tilde{u}_l = 1$ for $t < t_{s,l}$, $\tilde{u}_l = -1$ otherwise
        $\overline{\tilde{v}}_{lat}(t_k) = \overline{\tilde{v}}_{lat}(t_{k-1}) + \tilde{a}_{\max} \tilde{u}_l$
        $\overline{\tilde{s}}_{lat}(t_k) = \overline{\tilde{s}}_{lat}(t_{k-1}) + \overline{\tilde{v}}_{lat}(t_{k-1}) r + \tilde{a}_{\max} \tilde{u}_l \frac{r^2}{2}$
        $\mathcal{OC}_{other}(\tau_k) = \mathrm{chop}_{\mathrm{lat}}(\mathcal{OC}_{other}(\tau_k), \tilde{s}_r(t_k), \mathcal{H}_r)$
    end if
end for
```

VI. Experimental Results

The approach for formally verifying the safety of automated vehicles is applied to a Cadillac SRX which has been modified by the Robotics Institute at Carnegie Mellon University for automated driving. The vehicle is the successor of *Boss*, the vehicle that won the DARPA Urban Challenge in 2007. The hardware design of the new vehicle differs from the previous one by hiding sensors and computing devices, which results in a more production-ready vehicle, see [59] and Fig. 13.

Fig. 13: Cadillac SRX performing lane change maneuver B.

The sensing of the position $\begin{bmatrix} s_x & s_y \end{bmatrix}$, velocity v, and yaw angle Ψ is performed by the Applanix POS LV platform, which uses the GPS signal and an inertial measurement unit. The yaw rate $\dot{\Psi}$ is measured by the built-in sensor required for the yaw-stabilization of the vehicle. We use the rotary position sensor in the steering wheel motor to obtain the angle δ of the front wheel, which is given by a constant ratio.

Interfaces to the actuation of the vehicle are the steering wheel velocity, which corresponds to $\dot{\delta}$ by the steering-wheel-to-front-wheel ratio, and the desired velocity, which is realized by the built-in automatic cruise control (ACC) system. In the future, it will be possible to command the acceleration as proposed by the used mathematical model (1). Since this is not yet possible and we do not have access to the internal ACC system for modeling its dynamics, we restrict ourselves in the experiment to driving with constant velocity.

In order to repeatedly test a maneuver, we performed the test drives in *Robot City*, which is a former steel production site in Pittsburgh that is now dedicated to testing field robots. The driven maneuver is a double-lane-change maneuver, as shown in Fig. 14. This maneuver is integrated into a closed-loop path, such that after each round, two identical double-lane-change maneuvers are performed. Although the reference trajectory is the same for both maneuvers, the result differs, since lane change maneuver A is performed at a spot which has pot holes and loose tarmac, while the road conditions for lane change maneuver B are much better (see Fig. 14). Maneuver A was performed 14 times and maneuver B 13 times. The maneuvers were driven with 7.5 m/s and the maximal lateral acceleration is 2 m/s^2.

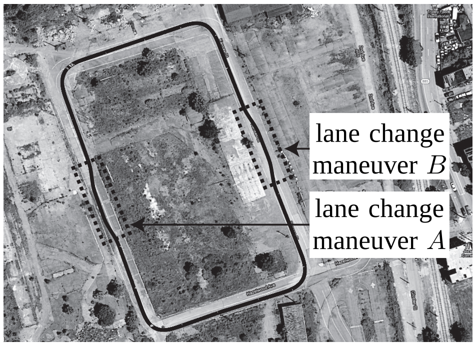


Fig. 14: Test site at Robot City with maneuvers A and B.

The vehicle control is implemented in MATLAB Simulink and runs on a dSPACE AutoBox. The code for computing the reachable sets is implemented in C++ and runs on a separate laptop connected to the software framework for sensing and path planning, which is maintained and extended from the DARPA Challenges (Grand Challenges and Urban Challenge). The C++ code for the reachable set computation can be downloaded from the current website of the first author or by requesting it via email.

A. Reachable Set of the Ego Vehicle

The reachable set of the ego vehicle for the double-lane-change maneuver is plotted in Fig. 15 for selected projections onto the two-dimensional state space. Measurement uncertainties $\mathcal{W}$ are based on the 3σ confidence interval of the specified sensor noise from the manufacturer for w_x, w_y, w_Ψ, which are modeled by a normal distribution. Assuming a normal distribution, the probability that a measurement is captured by the 3σ confidence interval is given by the error function as $\mathrm{erf}(3/\sqrt{2}) = 0.997$. Other measurement uncertainties are estimated from sensor data. The disturbance set $\mathcal{Y}$ is chosen as 0 for all dimensions, except for the dimensions adding uncertainty to $\dot{\beta}$ and $\ddot{\Psi}$, which are altered when the tire contact forces vary due to damaged tarmac. The set values are listed in Tab. II. Other than the measurement uncertainties provided by the manufacturers of those devices, disturbance sets have to be obtained from disturbance observers [60]. To obtain a disturbance set validated by extensive test drives is part of future work. In those test drives, each time a disturbance y_{crit} is estimated that is not within the axis-aligned box of disturbances $\mathcal{Y}$, each interval of $\mathcal{Y}$ has to be enlarged appropriately such that y_{crit} is contained in $\mathcal{Y}$.

TABLE II: Measurement uncertainties and disturbances.

measurement uncertainty, $\rho = [-1, 1]$				
w_x, w_y	w_Ψ	$w_{\dot{\Psi}}$	w_v	w_δ
0.06ρ m	$\frac{0.15\pi}{180}\rho$ rad	$\frac{0.43\pi}{180}\rho$ rad/s	0.06ρ m/s	$\frac{0.02\pi}{180}\rho$ rad
disturbances, $\rho = [-1, 1]$				
y_β	$y_{\dot{\Psi}}$	y_Ψ, y_v, y_{s_x}, y_{s_y}, y_δ		
0.2ρ rad/s	0.2ρ rad/s^2	0		

The initial set is composed by addition of two sets. The first one is the enclosing box $\mathcal{I}$ of all recorded states at the beginning of the maneuver, while states that are not measured have the pseudo-interval $[0, 0]$. The second set $\mathcal{I}_{\mathrm{unc}}$ contains uncertainties not captured by $\mathcal{I}$, and contains measurement uncertainties for measurable states and worst-case assumptions for states that are not measured. Thus, we have that $\mathcal{R}(0) = \mathcal{I} \oplus \mathcal{I}_{\mathrm{unc}}$. The uncertain intervals of $\mathcal{I}_{\mathrm{unc}}$ for s_x, s_y, Ψ, $\dot{\Psi}$, and δ are as for the measurement uncertainty $\mathcal{W}$. Since the slip angle β is not measured, a worst-case interval of $[-1, 1]0.02$ rad is assumed for $\mathcal{I}_{\mathrm{unc}}$.

It can be seen in the plots of the vehicle measurements for $\Psi/\dot{\Psi}$ and $\delta/\dot{\Psi}$ in Fig. 15 that there exist two bundles of recordings, one representing lane change maneuver A (lc A), the other one representing lane change maneuver B (lc B), see Fig. 14. It is expected that we would see more variation in the recorded data, if the same test is performed in more locations and under different weather conditions.

We use a time step size of $r = 0.01$ s and the expansion factor $\lambda = 1.8$ for the reachable set computation. The reachable sets are represented by zonotopes of order 200, where an introduction to zonotopes can be found in [13]. The computation time on a laptop with an Intel i7 Processor with 1.6 GHz and 6 GB memory is 4.2 sec, while the maneuver takes 7.5 sec so that the computation is $\nu = 1.79$ times faster than the maneuver time. It is expected that the computation time will be improved by code optimization and better future processors.

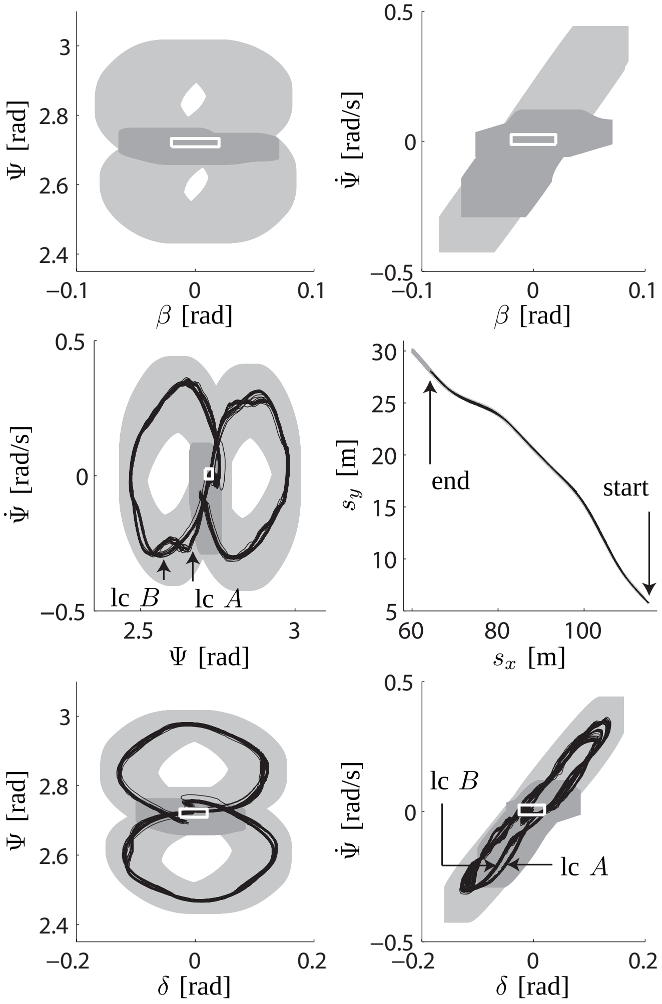


Fig. 15: Reachable set of the double-lane-change for selected projections. The light gray area shows the reachable set for the intended maneuver and the dark gray set the one for the transition to a safe stop. The white box shows the set of initial states. For plots with measurable states, black lines represent the measured values.

B. Safety Verification of the Maneuver

Based on the reachable set computation, the following tasks are verified:

- A standstill in a safe position at the end of the planned maneuver is realized.
- Road boundaries are never violated.
- Collisions with static obstacles are avoided.
- Collisions with other traffic participants are avoided.

We assume that the double-lane-change is in an urban or rural setting with traffic in both directions caused by a static obstacle on the road. Further, there are two vehicles driving in the lane with oncoming traffic, as depicted in Fig. 16 for snapshots of the occupancy sets. Note that the coordinates are rotated so that the road is horizontal in Fig. 17(a) for better use of space instead of diagonal as shown in the s_x/s_y plot of Fig. 15. The static obstacle and the other traffic participants are not present during the test drive, but realized as virtual objects in the traffic scene. Otherwise, a real crash might occur if a failure happens.

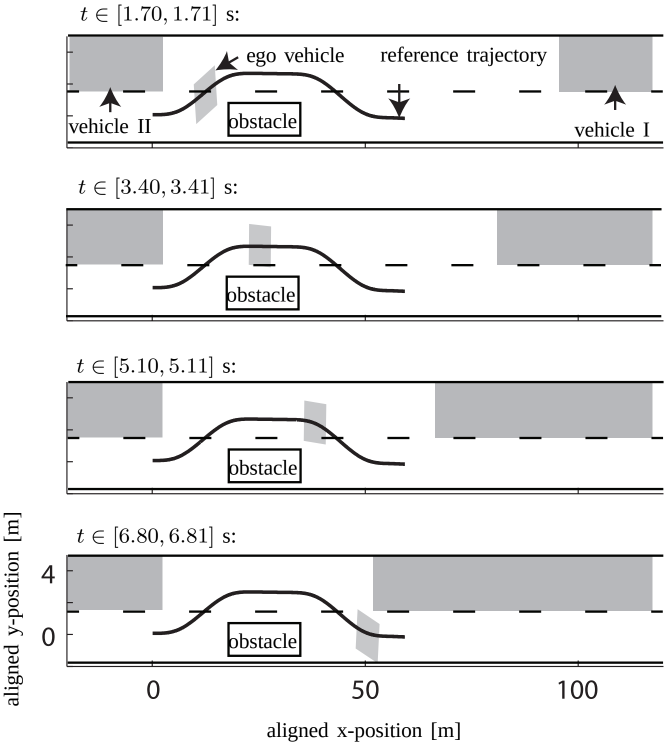
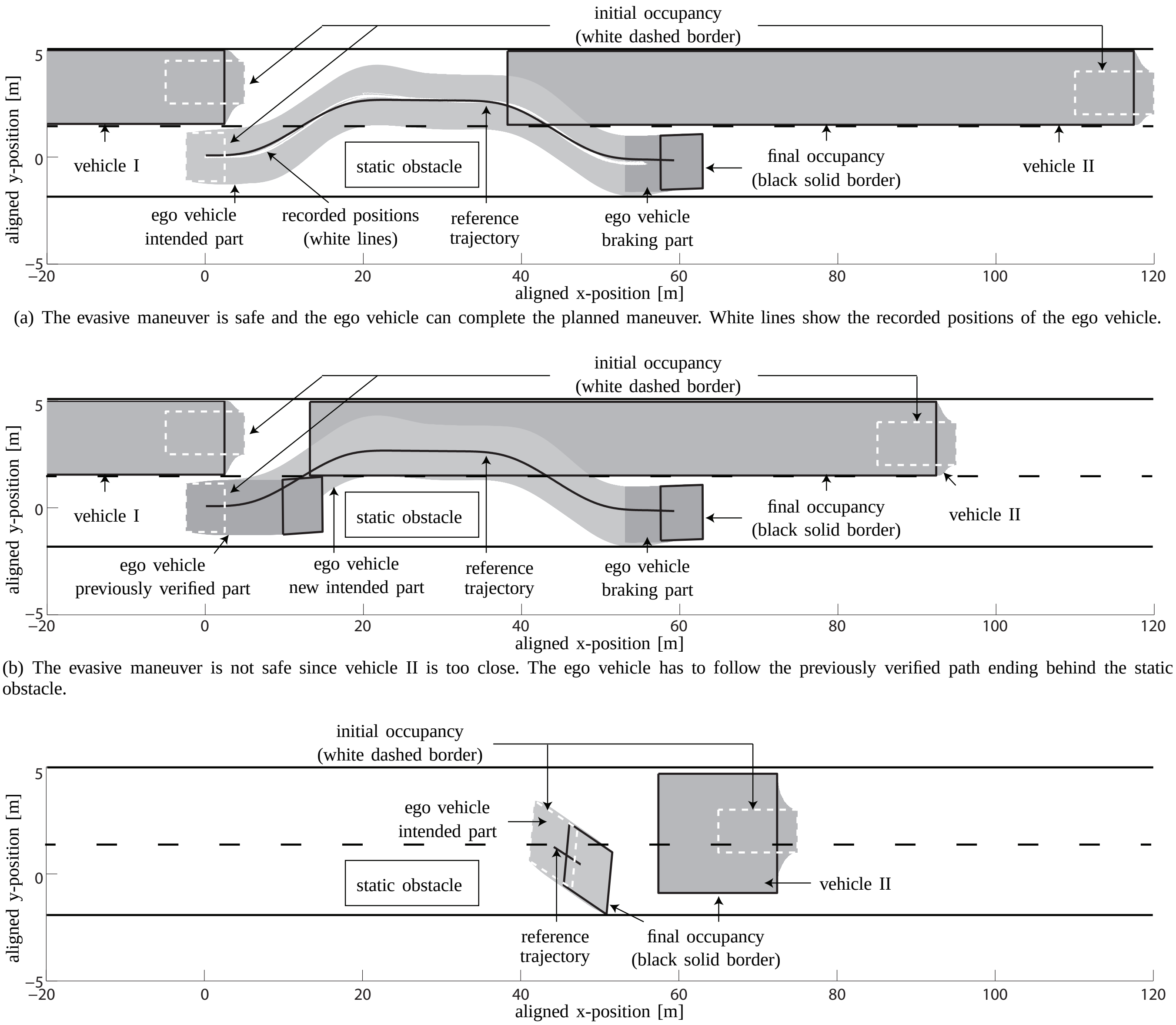


Fig. 16: Snapshots of the occupancy of traffic participants for selected short time intervals.

Three different scenarios are considered: In the first scenario, vehicle II is far enough away that the static obstacle is safely passed by the ego vehicle. Throughout the considered time horizon the other vehicles obey the traffic rules for the considered time horizon, i.e., they restrict their lateral movement to their own lane and respect the virtual speed limit of 7 m/s. However, for the occupancy prediction according to Sec. V we allow a penalization of 20% so that other traffic participants travel with up to $1.2 \cdot 7 = 8.4$ m/s. A violation of traffic rules is presented in the third scenario. In Fig. 16 snapshots of the occupancies of the ego vehicle and other vehicles are plotted every 1.7 s, which illustrates that the maneuver is safe. The unions of occupancy sets for all times are shown in Fig. 17(a). The uncertain initial position and velocity of vehicles I and II with respect to the coordinate system of Fig. 17(a) are $s_{I,x}(0) \in [-5, 5]$ m, $s_{I,y}(0) \in [2.5, 4.5]$ m, $v_{I,x}(0) \in [6, 8]$ m/s, $v_{I,y}(0) \in [-0.2, 0.2]$ m/s, $s_{II,x}(0) \in [110, 120]$ m, $s_{II,y}(0) \in [2, 4]$ m, $v_{II,x}(0) \in [6, 8]$ m/s, $v_{II,y}(0) \in [-0.2, 0.2]$ m/s. The body size of the ego vehicle is $l_v = 4.5$ m , $w_v = 1.8$ m, the maximum absolute acceleration is $a_{\max} = 7$ m/s^2, and the velocity at which the engine power is insufficient to produce forces that exceed the maximal tire force is $v_S = 7.3$ m/s.

The computation time for the other traffic participants (a few hundredths of a second) is negligible compared to that of the ego vehicle. In addition, the occupancy of each traffic participant can be computed in separate processes. Checking the intersection of occupancies is not negligible, but can be started in a separate process, once the reachable set for the first



(a) The evasive maneuver is safe and the ego vehicle can complete the planned maneuver. White lines show the recorded positions of the ego vehicle.

(b) The evasive maneuver is not safe since vehicle II is too close. The ego vehicle has to follow the previously verified path ending behind the static obstacle.

(c) At time $t = 5.9$ s the initially safe situation in Fig. 17(a) becomes unsafe since it is detected that vehicle II has left the road boundary. It is assumed that this vehicle does no longer repect the road boundary, forcing the ego vehicle to brake. The occupancy of the ego vehicle only consists of an intended part since the intention is to stop so that the braking part is not required.

Fig. 17: Occupancy sets for all times in different situations of the double-lane-change maneuver. The light gray regions show the occupancies of the ego vehicle of the intended trajectory section and the dark gray regions of the trajectory to a safe stop. The medium gray tone indicates the occupancy of the other traffic participants.

time interval is obtained. Since the reachable set computation of the ego vehicle takes considerably more time than the collision check, the overall duration is solely determined by the reachable set computation of the ego vehicle when the collision check is performed separately. Thus, with today's computer hardware, we can verify maneuvers $\nu = 1.79$ times faster than the time it takes to execute the maneuver (see Sec. VI-A), which is a prerequisite for the online application. In situations where an immediate danger is sensed, e.g. a child is stepping into the street, verification results have to be obtained within about 0.1 s, which is not yet possible. For this reason, we plan to store verification results for a small number of typical evasive maneuvers in a database, such that results are immediately obtained.

In the second scenario, vehicle II is not far enough away to guarantee that the ego vehicle can safely pass the static obstacle. This scenario has exactly the same setting, except that $s_{II,x}(0) \in [85, 95]$ m. The unions of occupancy sets for all times are shown in Fig. 17(b). The same figure also shows the occupancy of a verified maneuver of the vehicle, which stops the vehicle behind the static obstacle. The verification of that maneuver took place during the movement of the vehicle, but before the verification process of the current reference trajectory, which was returned as unsafe. Since the new maneuver is not safe, the vehicle follows the previously verified plan to stop behind the static obstacle. Once the traffic has cleared, a new plan for passing the static obstacle will become safe.

The third scenario originates from the first scenario, but after some time, vehicle II enters the designated lane of the ego vehicle such that a traffic rule is violated. According to the 4^{th} assumption in Sec. II we assume that other traffic participants respect traffic rules. When a traffic rule violation is detected, the model of the corresponding traffic participant is adapted such that the violated traffic rule is no longer considered. Due to this new situation, the ego vehicle plans a braking maneuver to mitigate a potential impact of vehicle II if it does not return to its own lane. The verification shows that the ego vehicle comes to a safe stop before vehicle II might hit the ego vehicle, as shown in Fig. 17(c). It also shows that the restriction of vehicle II staying in its own lane is no longer considered and that vehicle II can potentially reach large portions of the lane dedicated to the ego vehicle (note the different scaling of the x- and y-position).

VII. Conclusions

To the best knowledge of the authors, we have performed for the first time a non-trivial formal verification during the operation of an automated vehicle. Not only uncertainty in the movement of other traffic participants is considered, but also in the movement of the ego vehicle. Although uncertainties in the movement of the ego vehicle are considerably smaller than the ones of other traffic participants, neglecting uncertainties in the movement of the ego vehicle could cause the vehicle to lose track of the reference trajectory or hit the road boundary, for which a deviation of a few centimeters can be crucial in some situations.

The results show that reachability analysis can be performed efficiently when performing conservative linearization and using zonotopes as a set representation. Specific emergency maneuvers, such as braking or evasion under the use of available tire forces, can be stored in a database, such that in emergency situations a faster verification is possible. Storing results for all other situations is impossible, but the presented approach is already feasible for those (standard) situations. In the future, we plan to propose a general-purpose model for the set-based prediction of other traffic participants considering a wider range of traffic rules.

Acknowledgments

The authors gratefully acknowledge partial financial support by the National Science Foundation (NSF) under grant numbers CNS1035813 and CCF0926181 as well as by the German Research Foundation (DFG) under grant number AL 1185/3-1.

We would also like to thank Kartik Gupta and Alok Sharma for contributing to the C++ implementation of the approach and Daniel Heß for providing Prop. V.2.

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] A. Girard and G. J. Pappas, “Verification using simulation,” in *Hybrid Systems: Computation and Control*, ser. LNCS 3927. Springer, 2006, pp. 272–286.

[3] J. van den Berg, D. Ferguson, and J. Kuffner, “Anytime path planning and replanning in dynamic environments,” in *Proc. of the International Conference on Robotics and Automation*, 2006, pp. 2366–2371.

[4] C. Schmidt, F. Oechsle, and W. Branz, “Research on trajectory planning in emergency situations with multiple objects,” in *Proc. of the IEEE Intelligent Transportation Systems Conference*, 2006, pp. 988–992.

[5] 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.

[6] A. Wu and J. P. How, “Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles,” *Autonomous Robots*, vol. 32, no. 3, pp. 227–242, 2012.

[7] C. F. Chung, T. Furukawa, and A. H. Göktogan, “Coordinated control for capturing a highly maneuverable evader using forward reachable sets,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1336–1341.

[8] H. Täubig, U. Frese, C. Hertzberg, C. Lüth, S. Mohr, E. Vorobev, and D. Walter, “Guaranteeing functional safety: design for provability and computer-aided verification,” *Autonomous Robots*, vol. 32, no. 3, pp. 303–331, 2012.

[9] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler, “Recent progress in continuous and hybrid reachability analysis,” in *Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design*, 2006, pp. 1582–1587.

[10] A. Puri, P. Varaiya, and V. Borkar, “ϵ-approximation of differential inclusions,” in *Proc. of the 34th IEEE Conference on Decision and Control*, 1995, pp. 2892 – 2897.

[11] E. Asarin, T. Dang, and A. Girard, “Reachability analysis of nonlinear systems using conservative approximation,” in *Hybrid Systems: Control and Computation*, 2003, pp. 20–35.

[12] Z. Han and B. H. Krogh, “Reachability analysis of nonlinear systems using trajectory piecewise linearized models,” in *Proc. of the American Control Conference*, 2006, pp. 1505–1510.

[13] M. Althoff, O. Stursberg, and M. Buss, “Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization,” in *Proc. of the 47th IEEE Conference on Decision and Control*, 2008, pp. 4042–4048.

[14] T. Dang, O. Maler, and R. Testylier, “Accurate hybridization of nonlinear systems,” in *Hybrid Systems: Computation and Control*, 2010, pp. 11–19.

[15] A. Chutinan and B. H. Krogh, “Computational techniques for hybrid system verification,” *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 64–75, 2003.

[16] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, “Computational techniques for the verification and control of hybrid systems,” *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.

[17] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, pp. 947–957, 2005.

[18] N. Ramdani, N. Meslem, and Y. Candau, “Computing reachable sets for uncertain nonlinear monotone systems,” *Nonlinear Analysis: Hybrid Systems*, vol. 4, pp. 263–278, 2010.

[19] J. K. Scott and P. I. Barton, “Tight, efficient bounds on the solutions of chemical kinetics models,” *Computers and Chemical Engineering*, vol. 34, no. 5, pp. 717–731, 2010.

[20] ——, “Bounds on the reachable sets of nonlinear control systems,” *Automatica*, vol. 49, no. 1, pp. 93–100, 2013.

[21] M. Kieffer, E. Walter, and I. Simeonov, “Guaranteed nonlinear parameter estimation for continuous-time dynamical models,” in *Proc. of the 14th IFAC Symposium on System Identification*, 2006, pp. 843–848.

[22] S. M. Loos, A. Platzer, and L. Nistor, “Adaptive cruise control: Hybrid, distributed, and now formally verified,” in *Proc. of the 17th International Symposium on Formal Methods*, ser. LNCS 6664. Springer, 2011, pp. 42–56.

[23] N. Aréchiga, S. M. Loos, A. Platzer, and B. H. Krogh, “Using theorem provers to guarantee closed-loop system properties,” in *In Proc. of the American Control Conference*, 2012, pp. 3573–3580.

[24] E. C. Kerrigan and J. M. Maciejowski, “Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution,” *International Journal of Robust and Nonlinear Control*, vol. 14, no. 4, pp. 395–413, 2004.

[25] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *International Journal of Control*, vol. 84, no. 6, pp. 1157–1170, 2011.

[26] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, "Parameterized tube model predictive control," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2746–2761, 2012.

[27] J. M. Bravo, T. Alamo, and E. F. Camacho, "Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets," *Automatica*, vol. 42, pp. 1745–1751, 2006.

[28] S. Karaman and E. Frazzoli, "Linear temporal logic vehicle routing with applications to multi-uav mission planning," *Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1372–1395, 2011.

[29] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2817–2830, 2012.

[30] M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[31] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[32] 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.

[33] 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.

[34] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 299 – 310, 2009.

[35] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, and T. Mikami, "An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 942–953, 2011.

[36] M. Althoff and J. M. Dolan, "Set-based computation of vehicle behaviors for the online verification of autonomous vehicles," in *Proc. of the 14th IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 1162–1167.

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

[38] R. Parthasarathi and T. Fraichard, "An inevitable collision state-checker for a car-like vehicle," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3068–3073.

[39] N. Chan, J. Kuffner, and M. Zucker, "Improved motion planning speed and safety using regions of inevitable collision," in *Proc. of the 17th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control*, 2008.

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

[41] D. Heß, M. Althoff, and T. Sattel, "Comparison of trajectory tracking controllers for emergency situations," in *Proc. of the 16th IEEE Conference on Intelligent Transportation Systems*, 2013.

[42] R. Rajamani, *Vehicle Dynamics and Control*, F. F. Ling and E. F. Gloyna, Eds. Springer, 2006.

[43] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *Proc. of the American Control Conference*, 2012, pp. 3559–3566.

[44] J. Snider, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-09-08, 2009.

[45] E. Kutluay and H. Winner, "Assessment methodology for validation of vehicle dynamics simulations using double lane change maneuver," in *Proc. of the Winter Simulation Conference*, 2012, pp. 1–12.

[46] J. D. Setiawan, M. Safarudin, and A. Singh, "Modeling, simulation and validation of 14 dof full vehicle model," in *Proc. of the International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering.*, 2009, pp. 1–6.

[47] D. Katzourakis, J. C. F. de Winter, S. de Groot, and R. Happee, "Driving simulator parameterization using double-lane change steering metrics as recorded on five modern cars," *Simulation Modelling Practice and Theory*, vol. 26, pp. 96–112, 2012.

[48] L. M. Surhone, M. T. Timpledon, and S. F. Marseken, Eds., *Vienna Convention on Road Traffic*. VDM Publishing, 2010.

[49] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic reachability computation for families of linear vector fields," *Symbolic Computation*, vol. 32, pp. 231–253, 2001.

[50] M. Berz and G. Hoffstätter, "Computation and application of Taylor polynomials with interval remainder bounds," *Reliable Computing*, vol. 4, pp. 83–97, 1998.

[51] L. Jaulin, M. Kieffer, and O. Didrit, *Applied Interval Analysis*. Springer, 2006.

[52] 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.

[53] 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.

[54] P. Prabhakar and M. Viswanathan, "A dynamic algorithm for approximate flow computations," in *Hybrid Systems: Computation and Control*, 2011, pp. 133–142.

[55] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *Proc. of the 23rd International Conference on Computer Aided Verification*, ser. LNCS 6806. Springer, 2011, pp. 379–395.

[56] M. Althoff and B. H. Krogh, "Zonotope bundles for the efficient computation of reachable sets," in *Proc. of the 50th IEEE Conference on Decision and Control*, 2011, pp. 6814–6821.

[57] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 10, pp. 1684–1698, 2003.

[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] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2013, pp. 763–770.

[60] A. Radke and Z. Gao, "A survey of state and disturbance observers for practitioners," in *Proc. of the American Control Conference*, 2006, pp. 5183–5188.

Matthias Althoff is assistant professor in computer science at Technische Universität München, Germany. He received the diploma engineering degree in Mechanical Engineering in 2005, and the 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.

John M. Dolan is a principal systems scientist at Carnegie Mellon Universitys (CMU) Robotics Institute. He received a B.S.E. from Princeton University (1980), and the M.E. (1987) and Ph.D. (1991) degrees from CMU, all in mechanical engineering. During 1980-81 he studied at the Technische Universität München, Germany, on a Fulbright Scholarship, working simultaneously at the German Space Agency (DLR) on the stability and ride quality of magnetically levitated trains. He was the behaviors lead for Carnegie Mellons Tartan Racing team in the 2007 DARPA Urban Challenge. His research interests include autonomous driving, multi-robot cooperation, human-robot interaction, robot reliability, and sensor-based control.