

Safety Assessment of Driving Behavior in Multi-Lane Traffic for Autonomous Vehicles

Matthias Althoff, Olaf Stursberg, and Martin Buss

Abstract—An approach for the safety assessment of planned trajectories of autonomous vehicles is presented. Due to the unsafe nature of road traffic, the proposed safety assessment is performed in a probabilistic setting, where the probability distributions of possible future positions of traffic participants are computed based on dynamic models. The underlying algorithms have to be efficient as they need to provide a safety level of the currently planned path while the vehicle is in operation. In order to achieve the necessary efficiency of computation, the dynamics of traffic participants is abstracted into Markov-chains. The approach considers vehicle dynamics, interaction between traffic participants, and lane changes in multi-lane traffic.

I. INTRODUCTION

One of the main challenges for human drivers when navigating through traffic is to cope with the possible actions of other drivers which influence the own driving strategy. In order to smoothly drive through tricky situations, foresighted drivers predict likely actions of other traffic participants such that dangerous situations are avoided. For future autonomous cars, it is desirable to have the capability of foresighted driving which would offer its passengers a safe and stress-free ride. The possible realization of a software module which computes the future positions of other traffic participants allowing to check for safe navigation is presented in this work, as carried out within the collaborative research center *Cognitive Automobiles* [1].

A. Related Work

Short-term traffic prediction is often performed by single simulations of traffic participants what seems to provide satisfying results, see e.g. [2], [3]. However, if one is interested in long-term predictions, one has to compute multiple simulations of a traffic situation in order to thoroughly assess possible future evolutions. For each simulation, the initial situation is varied according to measurement uncertainties and the parameters/inputs are varied according to uncertainties in the behavior of other traffic participants. This approach, which is also referred to as Monte-Carlo simulation, has been investigated in [4], [5], [6], [7], [8]. A general drawback of Monte-Carlo simulation is that, due to the randomized testing, critical situations may be missed. This problem is eliminated when the set of all possibly reachable positions of the traffic participants is computed,

see e.g. [9]. In case the planned path of the autonomous vehicle does not intersect with any set of reachable positions of other traffic participants, the path is guaranteed to be safe. However, road traffic is inherently unsafe if all conceivable worst case behaviors of traffic participants are considered (e.g. violating traffic rules), such that a yes/no-answer about the safety of a planned path is often insufficient. For this reason, the reachable sets of traffic participants are enhanced by a probability distribution of the road occupancy in this and previous work of the authors [10], [11]. This approach is also referred to as stochastic reachability analysis and similar work in the field of traffic safety can only be found for air traffic [12], [13].

B. New Contributions

Compared to the earlier work of the authors [11], the work presented here is extended by the following new main contributions:

- The probability distribution of the acceleration of other traffic participants is not held constant but changes according to an acceleration dynamics.
- The computations of stochastic reachable sets is extended from single-lane to multi-lane traffic for which lane changes have to be considered.

In order to maintain the real time capability of the algorithm, lane changes are considered under certain simplifying assumptions which will be discussed in detail. The new contributions are integrated in the existing framework which also considers interactions between traffic participants. Note that the computations for the interaction have been modified compared to the earlier work in [11] so that the whole approach is more coherent. Interaction is especially important for long-term predictions, where effects due to actions of surrounding traffic participants cannot be neglected.

II. BASIC CONCEPT

In principle, the safety assessment of the autonomous vehicle works like an artificial co-pilot that warns the vehicle (driver) when a dangerous situation is approaching. As a consequence, the planned trajectory of the autonomous vehicle is replanned according to the safety warning and the location of the possible threat. This intervention in the trajectory planning is illustrated in Fig. 1. In order to provide the required information for the safety verification module, the vehicle has to be equipped with sensors perceiving the environment, such that the following information is available after a scene interpretation:

- the geometric description of the relevant road sections,

Matthias Althoff and Martin Buss are with the Institute of Automatic Control Engineering (LSR), Technische Universität München, 80290 München, Germany. Email: {althoff, mb}@tum.de
Olaf Stursberg is with the Institute of Control and System Theory, Dept. of Electrical Eng., University of Kassel, Germany. Email: stursberg@uni-kassel.de

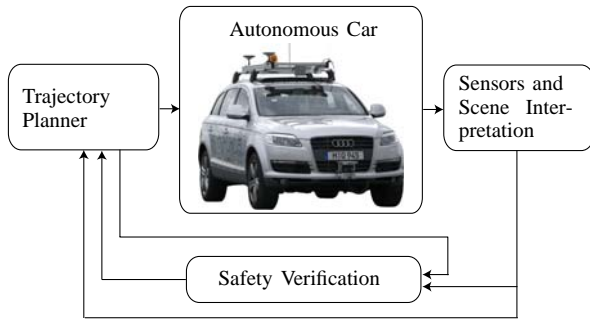


Fig. 1. Conception of the safety assessment.

- the position and geometry of static obstacles,
- the position, velocity and classification of dynamic obstacles.

Additionally, the safety verification module requires the planned trajectory of the autonomous car, which is available within the vehicle anyway. Static obstacles are a special case of dynamic obstacles with zero velocity and thus the further discussion only refers to dynamic obstacles. The classification of dynamic obstacles ($\hat{=}$ other traffic participants) is understood as the grouping into *cars*, *trucks*, *motorbikes*, *bicycles* and *pedestrians*.

Measurements of positions and velocities of other traffic participants are subject to uncertainties which cannot be neglected. Thus, in this work, the initial state (position and velocity) of the traffic participants is modeled by a set of possible values which accounts for the uncertainties in the measurement. Based on a dynamic model of the traffic participants, the future set of positions and velocities is computed, where the acceleration of the traffic participants is allowed to take any value from full braking to full acceleration. The set of possible states is also known as the *reachable set*. Due to the unsafe nature of traffic, the probability distribution within the reachable sets is computed in addition, resulting in *stochastic reachable sets*. The additional probabilistic information allows to compute the probability of a crash and thus to evaluate alternative trajectories of the autonomous car according to their safety. This is exemplarily shown for a simple traffic scene in Fig. 2 where the autonomous vehicle plans to turn into another street when another car is approaching. For the time interval $\tau_1 = [0, t_1]$, no crash can occur, as the stochastic reachable sets do not intersect, whereas a crash may occur in the next time interval $\tau_2 = [t_1, t_2]$.

The traffic scene is predicted for a time horizon of t_f and restarted with updated values of the traffic scene when the prediction is finished. Clearly, the prediction of the traffic scene has to be faster than real time. In order to restart the verification process with a satisfactory frequency, the computations should be several times faster than real time.

III. MODELS OF TRAFFIC PARTICIPANTS

This work focuses on the safety assessment of autonomous cars driving on a road network, i.e. the motion of traffic participants is constrained to designated roads. On that account,

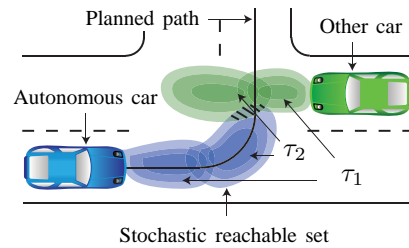


Fig. 2. Stochastic reachable sets of traffic participants.

the possible paths of traffic participants are determined by a finite set of decisions: $\{\textit{left turn}, \textit{right turn}, \textit{go straight}, \textit{left lane change}, \textit{right lane change}\}$. For unstructured environments, such as parking lots, the approach in [14] is suggested which uses the same mathematical principles as introduced herein. The finite set of decisions allows to construct possible paths of a vehicle which are located at the center of the lanes, see Fig. 3. If there are multiple lanes for one driving direction, each lane has a separate path to which the vehicle can change to. As vehicles do not drive in the center of a lane in general, the lateral deviation between the center of the car and the path is modeled as a piecewise constant probability distribution $f(\delta)$, where δ is the deviation from the path (Fig. 3). The deviation probability can be adjusted to different classes of traffic participants: Bicycle drivers are more likely to be found close to the curb, whereas cars and trucks are driving more likely in the center of a lane. The longitudinal probability distribution $f(s)$ in the direction of the paths, is obtained from the dynamical model of the vehicle. As the lateral and longitudinal distribution are assumed to be independent, one can compute the overall distribution as $f(s, \delta) = f(s) \cdot f(\delta)$, see also Fig. 3 for an exemplary probability distribution of a vehicle on path 1.

A. Longitudinal Dynamics

The dynamical model determining the future position s along a path and the velocity v is given by:

$$\dot{s} = v, \quad \dot{v} = \begin{cases} c_1 \cdot (1 - (v/c_2)^2) \cdot u, & \text{for } u > 0 \\ c_1 \cdot u, & \text{for } u \leq 0 \\ 0, & \text{for } v \leq 0, v \geq v^{max} \end{cases} \quad (1)$$

The constant c_1 models the maximum possible acceleration due to tire friction and c_2 the top speed – these constants are chosen according to the specific properties of the different classes of traffic participants¹. The acceleration input u varies from $[-1, 1]$, where -1 represents full braking and 1 represents full acceleration. Backwards driving on a lane is not considered, see (1) ($\dot{v} = 0, v \leq 0$) and the speed limit v^{max} is expected to be met ($\dot{v} = 0, v \geq v^{max}$), where one can specify v^{max} a little higher than the official speed limit. The speed limit v^{max} can also be used to set constraints for

¹Maximum possible acceleration c_1 is chosen as $10 \text{ [m/s}^2\text{]}$ for all types of traffic participants. The top speed c_2 is chosen as 60 [m/s] for cars in this work. Reasonable values for pedestrians, bicycles, trucks and motorbikes would be $4, 8, 25, 60 \text{ [m/s]}$, respectively.

the maximum lateral acceleration of a vehicle when driving through a curve. It is further noted, that the differential equations in (1) can be replaced by any Lipschitz-continuous functions.

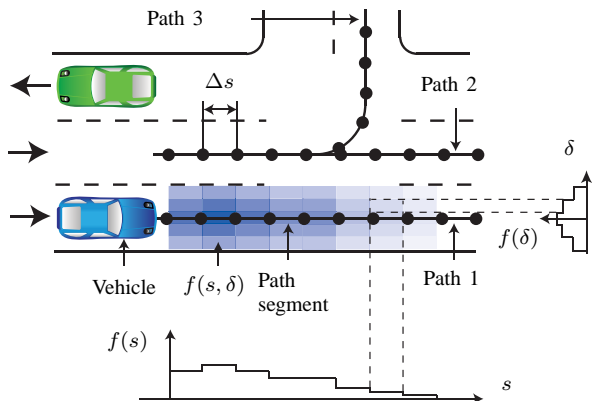


Fig. 3. Probability distribution of the position of a vehicle along a path-aligned coordinate system.

B. Lane Change Restrictions

The consideration of lane changes adds a significant amount of complexity to the computation of stochastic reachable sets, as will be shown later. For this reason, certain restrictions for lane changes are proposed such that their consideration preserves the online capability of this approach. In order to discuss the restrictions, a scenario with 3 lanes for the same driving direction is set up in Fig. 4. Note that vehicle *A* is the autonomous vehicle, while vehicles *B-F* are surrounding traffic participants. The restrictions are the following:

- 1) Lane changes are only considered for vehicles starting within a certain region around the autonomous vehicle: $s^A(0) - \underline{\eta} \leq s \leq s^A(0) + \overline{\eta}$, where $s^A(0)$ is the position of the autonomous vehicle at $t = 0$. Thus, lane changes are only considered for vehicles *D* and *E* in Fig. 4.
- 2) A vehicle does a lane change only once within the prediction horizon, i.e. changing two lanes or changing the lane and returning to the original lane is not considered.
- 3) There is no interaction between vehicles that may change to the same lane. This means, that the probability that vehicle *D* in Fig. 4 changes to the middle-lane is computed independently from the probability that vehicle *E* changes to the middle-lane.

From the second and third restriction follows, that in principle, the lane changes on a road with n lanes can be broken down to lane changes on several virtual roads with two lanes. For this reason, only two lanes with traffic participants *A-D* are considered for lane changes in the following. It is remarked that it has not been checked if the assumptions comply with real traffic, which is part of future work. However, the first assumption is reasonable as human drivers also limit their prediction for possible lane changers to the vicinity of their own vehicle. The second assumption is

justified as a single lane change takes about 6 sec which is similar to the prediction horizon t_f for lane change scenarios [15]. The last assumption does not strongly influence the overall result as the event that two vehicles change to the same lane within the prediction horizon t_f is rare. In the next section, the computation of the longitudinal probability distribution using Markov chains is discussed.

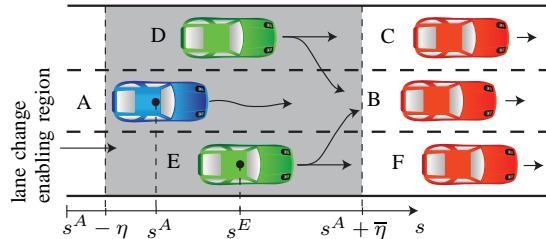


Fig. 4. Considered actions for a three-lane scenario.

IV. COMPUTING STOCHASTIC REACHABLE SETS USING MARKOV CHAINS

Stochastic reachable sets can only be computed exactly for a limited class of systems, such as e.g. linear systems with Gaussian initial states and inputs. Besides this problem, the computation of the stochastic reachable sets has to be faster than real time. A possibility to efficiently compute approximate probability distributions is to abstract the original system (1) by Markov chains. A special property of the proposed abstraction scheme is the following: although the probabilities obtained from the abstracted system are approximate, the reachable set ($\hat{=}$ region with non-zero probability) is over-approximated. This property allows to guarantee safety for the autonomous car if the probability of a crash is zero. Abstractions of continuous or hybrid systems to Markov chains have been reported in e.g. [12], [16], [17], [18]. The specialty of the method proposed in this and previous works of the authors is that the system behavior is over-approximated for time points as well as time intervals, and for bounded uncertain inputs [10], [11]. The over-approximation for consecutive time intervals and possible uncertain inputs allows a complete prediction for which no time point and input is missed.

A. Abstraction of Continuous Dynamics by Markov Chains

A discrete time Markov chain is a stochastic automaton with discrete states $z \in \mathbb{N}^+$. It is not exactly known in which state the system is, such that one computes with the probability $p_i = P(z = i)$ of the state value i where p is the probability vector of the Markov chain. By definition, the probability vector evolves according to a linear map Φ of the previous time step:

$$p(t_{k+1}) = \Phi p(t_k) \quad (2)$$

where $t_k = k \cdot T$ and $k \in \mathbb{N}^+$ is the time step and $T \in \mathbb{R}^+$ is the time step size. The entry $\Phi_{ji} = P(z(t_{k+1}) = j | z(t_k) = i)$ contains the probability that the state transition from state i to state j occurs after one time

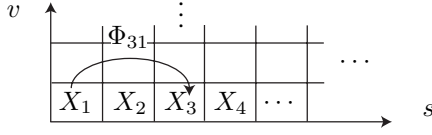


Fig. 5. Discretization of the state space.

step. In order to abstract the continuous vehicle dynamics into a Markov chain, two subtasks have to be dealt with: the discrete states of the Markov chain have to be assigned to value intervals (for position and velocity), and the transition probabilities in Φ are to be determined such that the vehicle dynamics is abstracted in an over-approximative way.

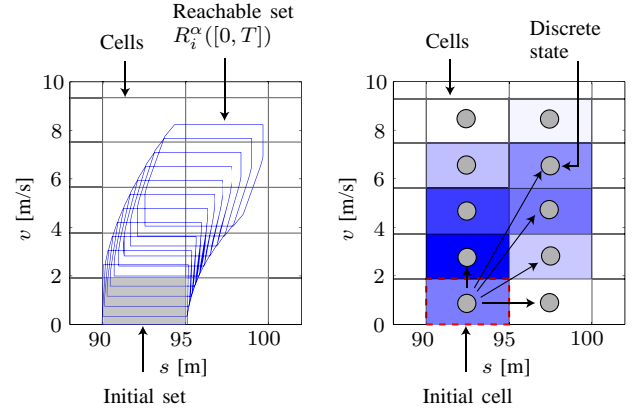
1) *Discretization of the State and Input Space:* The discrete states of the Markov chain are defined as state space regions of a partitioned continuous state space, spanned by the position and velocity of the vehicle, see Fig. 5. In order to obtain a finite number of discrete states, the state space is discretized for a subset $X \subset \mathbb{R}^2$. The subset X is chosen such that the velocity interval ranges from standstill to maximum speed v^{max} and the position interval is chosen as $[0, v^{max} \cdot t_f + s^{detect}]$, where t_f is the prediction horizon and s^{detect} the distance from which other traffic participants can be detected. In this work, the cells of the discretized state space are rectangular and of equal size. Besides the state space, the input interval $u \in [-1, 1]$ is also discretized into equidistant intervals. The cells of the state space are denoted by X_i where the index i refers to the value of the discrete state representing the cell. The input intervals are denoted by U^α where α refers to the value of the discrete input which is denoted by y . In order to distinguish indices of state space cells from indices of input intervals, input indices are written as superscripts and in Greek letters.

2) *Transition Probabilities of the Markov Chains:* Markov chains do not consider inputs, such as the acceleration command u in (1). This shortcoming is circumvented by computing different Markov chains for each input interval U^α . The transition probabilities for different input intervals α are denoted by $\Phi_{ji}^\alpha(T)$ and are defined as $\Phi_{ji}^\alpha(T) = P(z(t_{k+1}) = j | z(t_k) = i, y([t_k, t_{k+1}]) = \alpha)$, where $y([t_k, t_{k+1}])$ denotes the discrete input for the time interval $t \in [t_k, t_{k+1}]$. The transition probability from cell i to another cell j is based on the volume ratio of the reachable set R_i^α starting in cell i under input α that intersects with cell j (see Fig. 6(a)):

$$\Phi_{ji}^\alpha(T) = \frac{V(R_i^\alpha(T) \cap X_j)}{V(R_i^\alpha(T))}, \quad (3)$$

where $V()$ is an operator returning the volume. For further information on the computation of reachable sets, the reader is referred to [19], [20]. As soon as a cell is reached by $R_i^\alpha(T)$, the corresponding transition probability is non-zero, hence the reachable cells (cells with non-zero probability) of the Markov chain over-approximate the corresponding reachable set $R_i^\alpha(T)$. In contrast to the reachable set, the probability values are approximative due to the assumption that the system state within the reachable set is equally distributed for the computation of (3).

In an analogous way, one can compute the transition probabilities $\Phi_{ji}^\alpha([0, T])$ for $t \in [0, T]$ by substituting $R_i^\alpha(T)$ with $R_i^\alpha([0, T])$ which is the reachable set for $t \in [0, T]$. The combined use of transition probabilities for time points $\Phi_{ji}^\alpha(T)$ and time intervals $\Phi_{ji}^\alpha([0, T])$ is clarified in the following subsection. The reachable set of $R_i^\alpha([0, T])$ is exemplarily shown for the vehicle dynamics (1) in Fig. 6(a). The corresponding stochastic reachable set is illustrated in Fig. 6(b) where the circles symbolize the discrete states, which are assigned to the corresponding cells. A transition to a cell is the more likely, the darker the color of the cell is. Note, that the transition probabilities in $\Phi_{ji}^\alpha(T)$ and $\Phi_{ji}^\alpha([0, T])$ are computed offline, such that computationally expensive operations are performed beforehand. The remaining online computations are introduced in the next subsection.



(a) Reachable set for the time interval $t \in [0, T]$. (b) Reachable cells for the time interval $t \in [0, T]$.

Fig. 6. Reachable set of the original system and the corresponding stochastic reachable set of the Markov-chain.

B. Computing Stochastic Reachable Sets using Markov Chains

The computed transition probabilities $\Phi_{ji}^\alpha(T)$ and $\Phi_{ji}^\alpha([0, T])$ allow to compute the probability distribution of the next time step and time interval as shown in (2):

$$\begin{aligned} p(t_{k+1}) &= \Phi^\alpha(T)p(t_k), \\ p([t_k, t_{k+1}]) &= \Phi^\alpha([0, T])p(t_k). \end{aligned} \quad (4)$$

Ultimately, only the probabilities of the time intervals are used in the end for the safety assessment. However, the probability distribution for the time steps has to be computed as an auxiliary distribution. The above computation in (4) can only be applied if the input α is known for each time interval. If this is not the case, one can define the joint probability $p_i^\alpha = P(z = i, y = \alpha)$ and compute the probability distributions as

$$\begin{aligned} p^\alpha(t_{k+1}) &= \Phi^\alpha(T)p^\alpha(t_k), \\ p^\alpha([t_k, t_{k+1}]) &= \Phi^\alpha([0, T])p^\alpha(t_k). \end{aligned}$$

In contrast to (4), the above computation has to be performed for all input values α , as the exact input is unknown. Moreover, one can compute in which way the input probability

evolves over time for each cell i . This aspect and the following consequences are a novelty compared to earlier work in [11], where the input probability distribution was constant. In order to pursue the concept introduced so far, the input probabilities also change according to another Markov chain:

$$p_i^\beta(t_k)' = \Gamma_i^{\beta\alpha} p_i^\alpha(t_k). \quad (5)$$

The prime indicates the instant change of the probability distribution at time t_k and $\Gamma_i^{\beta\alpha}$ is the transition probability that the input value changes from $y = \alpha$ to $y = \beta$ when the state is $z = i$: $\Gamma_i^{\beta\alpha} = P(y' = \beta | y = \alpha, z = i)$. In contrast to the computation of the transition matrices Φ^α for the states, the transition matrices Γ_i for the inputs cannot be computed based on a dynamical model. This is because the input/driving commands of other vehicles are provided by humans or complex computer systems (when vehicles drive autonomously in the future), for what the system dynamics is unknown. As a consequence, the transition matrices Γ_i have to be learned by observation of traffic scenes or set by a combination of simulations and heuristics, where the latter is used in this work. The input transition matrix is composed of an input dynamics matrix Ψ and a priority vector λ , where the prioritization results from many aspects, such as speed limits or interaction with other vehicles. The input dynamics matrix Ψ and the priority vector λ are combined to the following:

$$\begin{aligned} \Gamma_i^{\beta\alpha} &= \text{norm}(\hat{\Gamma}_i^{\beta\alpha}) := \frac{\hat{\Gamma}_i^{\beta\alpha}}{\sum_\beta \hat{\Gamma}_i^{\beta\alpha}}, \\ \hat{\Gamma}_i^{\beta\alpha} &= \text{diag}(\lambda_i^\beta) \Psi^{\beta\alpha}, \quad \forall i: \sum_\beta \lambda_i^\beta = 1, 0 \leq \lambda_i^\beta \leq 1 \end{aligned} \quad (6)$$

The intermediate result $\hat{\Gamma}_i$ is normalized such that the sums of the columns are adding up to 1, in order to ensure that the sum of the probability vector remains 1 after the multiplication with Γ_i , see (5). The state dependence is modeled by the priority vector λ , while the input dynamics matrix Ψ is independent of the state.

In order to discuss the effect of Ψ separately, λ is set to $\lambda_i^\beta = \frac{1}{m}$, $\forall i, \beta$, resulting in $\Gamma_i = \Psi$ where m is the number of inputs. The transition probabilities in Ψ are set according to the heuristics that the bigger the change of the input², the more unlikely is this change. A transition matrix that considers this aspect and further contains the special cases of $\Psi = I$ and $\Psi = \frac{1}{m} O$ where O is the matrix of ones is:

$$\Psi^{\beta\alpha}(\gamma) = \text{norm}(\hat{\Psi}^{\beta\alpha}(\gamma)), \quad \hat{\Psi}^{\beta\alpha}(\gamma) = \frac{1}{(\beta - \alpha)^2 + \gamma}.$$

The parameter γ allows to gradually interpolate the extreme cases $\Psi = I$ and $\Psi = \frac{1}{m} O$, which are represented by the limit $\lim_{\gamma \rightarrow 0} \Psi(\gamma) = I$ and the other limit $\lim_{\gamma \rightarrow \infty} \Psi(\gamma) = \frac{1}{m} O$. Informally speaking, a low value of γ models drivers that do not change their input often, whereas a high value

²As the discrete inputs are numbered in increasing order according to the acceleration intervals, the difference between the input numbers is a measure for the change of the acceleration interval.

models drivers that change their input ($\hat{=}$ acceleration) often. The higher the value of γ , the faster converges the initial input probability distribution from the measured distribution to a steady state distribution. This is illustrated in Fig. 7 for 3 inputs, where the high input numbers represent high positive acceleration, such that the first input $y = 1$ represents full braking and the last input $y = 3$ full acceleration. The initial probabilities are set to $P(y = 1) = 0$, $P(y = 2) = 0.8$, $P(y = 3) = 0.2$ and the probabilities converge to $\frac{1}{3}$ as no prioritization λ is specified.

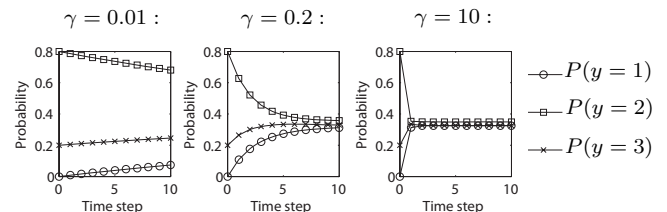


Fig. 7. Input evolution for $\gamma = 0.01, 0.2, 10$.

The priority vector λ is discussed for $\lim_{\gamma \rightarrow \infty} \Psi(\gamma) = \frac{1}{m} O$ such that $\lambda_i^\beta = P(y = \beta | z = i)$ when applying (6) and (5) which is also the setting used in [10], [11]. Two aspects are covered by the priority vector λ_i^β . The first one considers the probability distribution m^β for the input y when the driver is on an open road without any constraints, like e.g. speed limits or other traffic participants. This input probability distribution is independent of the state ($m_i^\beta = m_j^\beta$). The second aspect is that only a subset of inputs meets constraints in certain conditions. Due to the uncertain modeling of traffic situations, the event C of constraint satisfaction is subject to probability and stored in the constraint vector $c_i^\beta := P(C | z = i, y = \beta)$. Since at least the constraints have to be met, the probability vector c_i^β serves as an upper bound for the priority vector λ :

$$\lambda_i^\beta = \begin{cases} m_i^\beta, & \text{if } m_i^\beta \leq c_i^\beta \\ c_i^\beta, & \text{otherwise.} \end{cases} \rightarrow m_i^{\beta-1} := m_i^{\beta-1} + m_i^\beta - c_i^\beta$$

With other words, m_i^β is cut off at c_i^β and the cut-off probability is added to the next lower acceleration interval, see Fig. 8. This is motivated by the fact that, in this work, the vehicle interaction is limited to vehicles driving ahead (see Sec. IV-C), meaning that vehicles have to accelerate less or brake stronger in order to fulfill the safety constraints.

C. Interaction of Traffic Participants

In this subsection, the constraint vector c_i^β for interaction with another vehicle on the same lane is computed. The following vehicle is denoted by A , the leading vehicle is denoted by B , and variables corresponding to one of the vehicles is denoted by an index A or B . For a coarse notation, events such as $z^A = i$ are abbreviated as $z_i^A (\hat{=} z^A = i)$. In order to obtain the constraint vector c_i^α , it is firstly investigated under which configuration of states z_i^A, z_j^B and inputs $y^{A,\alpha}, y^{B,\beta}$ a crash can occur.

Thereo, both vehicles are simulated offline under the

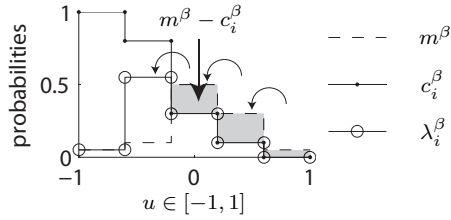


Fig. 8. Combining probability distributions with probabilistic constraints.

following initial conditions for $\Delta t = \kappa \cdot T$ ($\kappa \in \mathbb{N}^+$): $x^A(0) = \text{center}(X_i^A)$, $x^B(0) = \text{center}(X_j^B)$, $u^A(0) = \text{center}(U^{A,\alpha})$, $u^B(0) = \text{center}(U^{B,\beta})$ where $\text{center}()$ returns the volumetric center of a set. After $\Delta t = \kappa \cdot T$, the leading vehicle B suddenly fully brakes and as a reaction to this, the following vehicle A fully brakes, too. The outcome of the simulation determines the conditional probability for satisfying the constraint that a crash occurs with probability less than ϵ which is motivated by driver inattentiveness:

$$P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta}, \Delta t = \kappa \cdot T) = \begin{cases} 1, & \text{no crash simulated} \\ \epsilon, & \text{otherwise} \end{cases}$$

It is remarked that only one possible combination of continuous initial conditions and continuous inputs out of the sets $X_i^A, X_j^B, U^{A,\alpha}$ and $U^{B,\beta}$ has been considered. However, if the cell size of $X_i^A, X_j^B, U^{A,\alpha}$ and $U^{B,\beta}$ is small enough, the introduced simplification has a marginal effect on the final probability distribution of the traffic participants. The conditional probabilities $P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta}, \Delta t = \kappa \cdot T)$ can be obtained for different intervals $\Delta t = \kappa \cdot T$ of constant acceleration y^A and y^B . Long time intervals Δt model the behavior of foresighted drivers, who adjust their acceleration early to changes of other drivers and the other way round for sporty drivers. The discrete probabilities $P(\Delta t = \kappa \cdot T)$ allow to compute

$$P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta}) = \sum_{\kappa} P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta}, \Delta t = \kappa \cdot T) P(\Delta t = \kappa \cdot T).$$

Note that the probability $P(\Delta t = \kappa \cdot T)$ for foresighted or sporty drivers is not obtained online by observation of the driving style, but set according to the general distribution of all drivers. The conditional probabilities $P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta})$ for all configurations are computed offline and stored in an $c \times c$ -array of $d \times d$ -matrices $\Theta_{ji}^{\alpha\beta}$ where c and d is the number of discrete inputs and states. Using the offline computed values in $\Theta_{ji}^{\alpha\beta}$, the constraint vector for vehicle interaction is computed online as:

$$c_i^\alpha = P(C|z_i^A, y^{A,\alpha}) = \sum_{j,\beta} P(C|z_i^A, z_j^B, y^{A,\alpha}, y^{B,\beta}) P(z_j^B, y^{B,\beta}) = \sum_{j,\beta} \Theta_{ji}^{\alpha\beta} p_j^{B,\beta} \quad (7)$$

It is remarked that for the above computation it is assumed that $P(z_i^A, y^{A,\alpha})$ and $P(z_j^B, y^{B,\beta})$ are independent.

D. Lane Change

As a further novelty, lane changes are considered in this work. The prediction of lane change maneuvers has been mostly studied for the own vehicle where the human intention is tried to be predicted, e.g. via Bayesian networks [21]. Bayesian networks have also been investigated for lane change prediction of other vehicles [22], which is however less investigated. While one can use the results from [22] for the lane change probability, an alternative approach is presented here, which uses only the information provided so far for this approach.

1) *Lane Change Probability Approximation:* The following considerations are made for the situation in Fig. 4 but for two lanes with vehicles A – D where vehicle D performs the lane change. This is no loss of generality as lane change maneuvers can be broken down to this case using the assumptions made in Sec. III-B. The probability of a lane change is heuristically obtained from three factors:

- 1) the motivation for driving on the current lane denoted by σ^C ,
- 2) the motivation for driving on the neighboring lane denoted by σ^B ,
- 3) the inconvenience for the vehicle becoming the new following vehicle after the lane change is performed, denoted by σ^A .

The motivation values σ^C and σ^B are obtained from the constraint vector c_i^α of vehicle D following vehicle C or B respectively. Similarly, the inconvenience value σ^A is computed from the constraint vector of vehicle A when following vehicle D after the lane change. As an intermediate step, the probabilities e^α of meeting the constraint event C for a given acceleration input are computed:

$$e^\alpha := P(C|y = \alpha) = \sum_i P(C|z = i, y = \alpha) P(z = i) = \sum_i c_i^\alpha p_i, \quad p_i = \sum_\beta p_i^\beta,$$

where independency of $P(z = i)$ and $P(y = \alpha)$ is assumed as in (7). Next, the distributions $e^{C,\alpha}$, $e^{B,\alpha}$ and $e^{A,\alpha}$ for interaction with the vehicles A, B, and C are weighted by a vector \bar{w} for the vehicles in front and a vector \underline{w} for the new following vehicle (using the scalar product):

$$\sigma^C = \bar{w}^\alpha e^{C,\alpha}, \quad \sigma^B = \bar{w}^\alpha e^{B,\alpha}, \quad \sigma^A = \underline{w}^\alpha e^{A,\alpha}$$

The weights are chosen in a way that acceleration is weighted higher than deceleration because the possibility of high acceleration motivates to drive on a lane. Finally, the probability for a lane change p^{lc} is heuristically computed as:

$$p^{lc} = \frac{2}{\pi} \arctan \left(b \cdot \frac{\sigma^B}{\sigma^C} \frac{\sigma^A}{\sum \underline{w}} \right).$$

The heuristics is chosen such that a motivation ratio $\frac{\sigma^B}{\sigma^C}$ in favor of the neighboring lane as well as a low inconvenience value σ^A motivates a lane change. The arctan function is chosen as it is able to model the saturation ($\mathbb{R}^+ \rightarrow [0, 0.5\pi]$) but can also be replaced by a similar function.

The summation $\sum \underline{w}$ is the maximum possible value for σ^A such that the inconvenience measure is normalized. This is not necessary for σ^B , σ^C as one is only interested in their ratio. In case, there does not exist a vehicle B or C , the corresponding σ values are set to $\sum \bar{w}$ or $\sum \underline{w}$. The parameter b is chosen to $b = \tan(\frac{\pi}{2} \cdot \hat{p}^{lc})$ where \hat{p}^{lc} is the lane change probability if the motivation for both lanes is equal and there is no restriction due to vehicle A .

2) *Longitudinal and Lateral Probability Distributions:* In order to refer to the cases when vehicle D drives on the left or right lane, the notation Dl and Dr is used respectively. The longitudinal probabilities for the left and right lane are computed for a change from left to right as

$$\begin{aligned} \Delta p^\alpha(t_k) &= p^{lc}(t_k) p^{Dl,\alpha}(t_k), \\ p^{Dl,\alpha}(t_{k+1}) &= \Phi^\alpha(T) p^{Dl,\alpha}(t_k) - \Delta p^\alpha(t_k), \\ p_j^{Dr,\alpha}(t_{k+1}) &= \Phi_{ji}^\alpha(T) p_i^{Dr,\alpha}(t_k) + \lambda_j^{Dr,\alpha}(t_k) \sum_\alpha \Delta p_j^\alpha(t_k). \end{aligned}$$

The input probability distribution after the lane change is instantly changed to the distribution of the new lane by computing the added probability as $\lambda_i^{Dr,\alpha}(t_k) \sum_\alpha \Delta p_i^\alpha(t_k)$. Besides the longitudinal probability distribution, the lateral distribution has to be changed as well. Thereto, the lane change is divided into phases, each lasting for a time interval of $[0, T]$ which equals the step size of the Markov chain. From this follows a certain probability distribution \hat{p}_i^{lc} for the i^{th} phase of the lane change as shown in Fig. 11 where t^{lc} is the time passed since the lane change has been initiated. The gray area on the left and right indicates the probability of driving on the initial or the neighboring lane. In the example of Fig. 11, the traffic participant has not finished any lane change yet. In order to account for the uncertainty in the lateral deviation during a lane change, a deviation probability is defined for each phase i of the lane change which is denoted by $f_i(\delta)$. The final deviation probability which is spanned over the initial and neighboring lane is computed as $f(\delta) = \sum_i \hat{p}_i^{lc} \cdot f_i(\delta)$. The lane change probabilities and stochastic reachable sets for a lane change scenario are computed in the next section.

V. NUMERICAL EXAMPLE

The newly introduced Markov-chain for the input dynamics and the possibility to consider lane changes are demonstrated by a numerical example. The considered traffic situation is depicted in Fig. 4 but for two lanes with vehicles A - D , where vehicle A is the autonomous vehicle and a lane change is only considered for vehicle D . The traffic situation can easily be extended to three lanes as discussed in Sec. III-B. In order to motivate a lane change for vehicle D , the initial speed of vehicle C driving ahead is chosen lower than of the vehicles A and B on the adjacent lane. For the considered traffic scenario, the following parameters and settings have been chosen:

The position along the path is discretized in 40 segments of 5 m length each and the lateral deviation of the path is discretized in 7 segments of 0.57 m length each. The velocity is discretized in 10 segments, each representing an

interval of 2.2 m/s and the speed limit on the road is set to 16 [m/s]. Each vehicle is subject to 5 input intervals and the time increment of the Markov chains is chosen to $T = 0.5$ sec. The following values are set exemplarily for the presented numerical example and are not obtained from real world measurements: the γ -value for the input dynamics has been chosen as $\gamma = 0.2$ and the values of the characteristic input probability distribution are chosen as $m = [0.05, 0.1, 0.5, 0.3, 0.05]$. The weighting vectors for the lane change probability are $\bar{w} = [0, 1, 2, 3, 3]$, $\underline{w} = [0, 0, 1, 1, 1]$ and $b = \tan(\frac{\pi}{2} \cdot 0.07)$.

The stochastic reachable sets are displayed in Fig. 9 for 5 selected out of 10 time intervals. Dark regions indicate high probability, while bright regions represent areas of low probability. In order to improve the visualization, the colors are separately normalized for each vehicle. Furthermore, the normalized motivation values σ^B , σ^C , inconvenience value σ^A and lane change probability p^{lc} are plotted in Fig. 10. Additionally, the probability distribution \hat{p}^{lc} at $t = 4$ sec of the lane change phases is shown in Fig. 11. The crash probability of the autonomous vehicle A over time is plotted in Fig. 12. The computation of this scenario has taken 1.27 sec in C++ on a single core desktop PC with 3.7 GHz for a prediction horizon of $t_f = 5$ sec, such that the prediction was 3.9 times faster than real time.

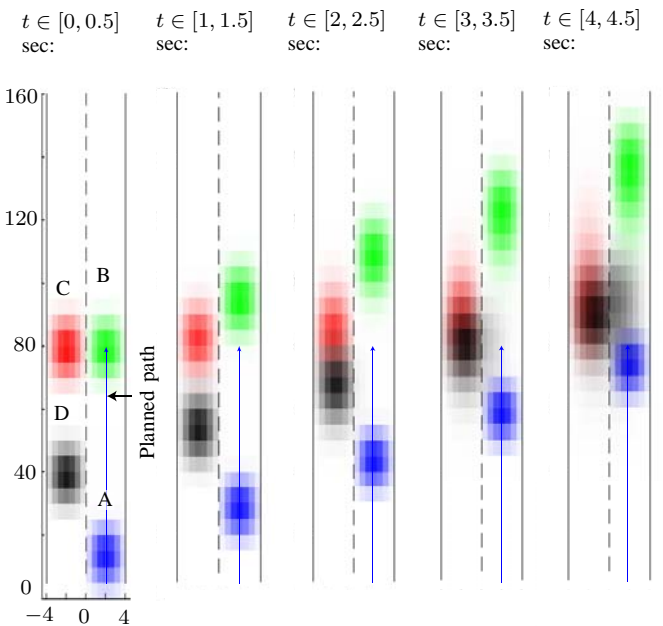


Fig. 9. Stochastic reachable sets for the lane change scenario.

VI. CONCLUSION

The presented approach allows the probabilistic prediction of traffic situations. In order to efficiently compute the prediction, the dynamics of traffic participants is conservatively abstracted to Markov-chains, i.e. the abstraction is complete. Additionally, the input dynamics of traffic participants, their interaction with other traffic participants and the probability of changing the lane is considered. In order to improve

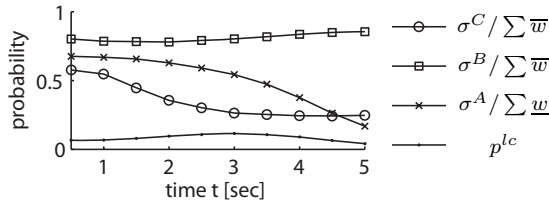


Fig. 10. Evolution of normalized motivation value, inconvenience value and lane change probability.

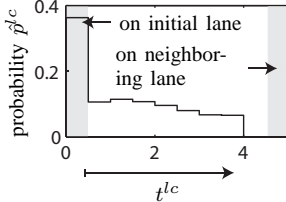


Fig. 11. Probability distribution of lane change phases of vehicle D at $t = 4$ sec.

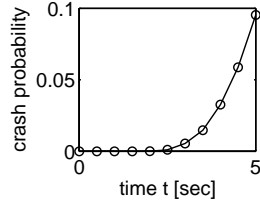


Fig. 12. Probability distribution for a crash of the autonomous car A.

the traffic prediction, it is planned to learn the introduced parameters based on recorded traffic data in the future. The adaptation of the prediction horizon as well as the resolution of the discretized state space to certain situations (e.g. inner-city/highway) is another line of future research. Currently, the C++ code of the traffic prediction is integrated in the software framework of the vehicle shown in Fig. 1 which is explained in detail in [23].

The approach can also be applied for conventional cars by exchanging the trajectory planner module by a trajectory estimator module. Thus, the safety assessment becomes a virtual co-pilot which warns the human driver in critical driving situations. As most of the computations are matrix multiplications, the use of specialized hardware such as FPGAs or graphic cards could reduce the computational burden.

ACKNOWLEDGMENTS

The authors gratefully acknowledge partial support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Centre 28 *Cognitive Automobiles*.

REFERENCES

- [1] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 215–220.
- [2] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 1068–1073.
- [3] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach - its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, pp. 528–540, 2006.
- [4] A. E. Broadhurst, S. Baker, and T. Kanade, "A prediction and planning framework for road safety analysis, obstacle avoidance and driver information," in *Proc. of the 11th World Congress on Intelligent Transportation Systems*, October 2004.

- [5] —, "Monte carlo road safety reasoning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 319–324.
- [6] S. Danielsson, L. Petersson, and A. Eidehall, "Monte carlo based threat assessment: Analysis and improvements," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 233–238.
- [7] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using monte carlo sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 137–147, 2008.
- [8] K. Aso and T. Kindo, "Stochastic decision-making method for autonomous driving system that minimizes collision probability," in *Proc. of the FISITA World Automotive Congress*, 2008.
- [9] 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.
- [10] M. Althoff, O. Stursberg, and M. Buss, "Online verification of cognitive car decisions," in *Proc. of the 2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 728–733.
- [11] —, "Stochastic reachable sets of interacting traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 1086–1092.
- [12] J. Hu, M. Prandini, and S. Sastry, "Aircraft conflict detection in presence of a spatially correlated wind field," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, pp. 326–340, 2005.
- [13] M. Prandini and J. Hu, "Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction," in *Proc. of the 47th IEEE Conference on Decision and Control*, 2008.
- [14] F. Rohrmüller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using markov chains for replanning in dynamic environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2504–2510.
- [15] E. C. B. Olsen, S. E. Lee, W. W. Wierwille, and M. J. Goodman, "Analysis of distribution, frequency, and duration of naturalistic lane changes," in *Proc. of the 46th Annual Meeting of the Human Factors and Ergonomics Society*, 2002, pp. 1789–1793.
- [16] X. Koutsoukos and D. Riley, "Computational methods for reachability analysis of stochastic hybrid systems," in *Hybrid Systems: Computation and Control*, 2006, pp. 377–391.
- [17] J. Lunze and B. Nixdorf, "Representation of hybrid systems by means of stochastic automata," *Mathematical and Computer Modeling of Dynamical Systems*, vol. 7, pp. 383–422, 2001.
- [18] J. Schröder, *Modelling, State Observation and Diagnosis of Quantised Systems*. Springer, 2003.
- [19] M. Althoff, O. Stursberg, and M. Buss, "Verification of uncertain embedded systems by computing reachable sets based on zonotopes," in *Proc. of the 17th IFAC World Congress*, 2008, pp. 5125–5130.
- [20] —, "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.
- [21] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, pp. 431–440, 2007.
- [22] I. Dagli, M. Brost, and G. Breuel, "Action recognition and prediction for driver assistance systems using dynamic belief networks," in *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, ser. LNCS 2592. Springer, 2003, pp. 179–194.
- [23] M. Goebel and G. Färber, "A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 734–740.