Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction

Constantin Hubmann, Jens Schulz, Marvin Becker, Daniel Althoff, and Christoph Stiller, *Senior Member, IEEE*

***Abstract*—Automated driving requires decision making in dynamic and uncertain environments. The uncertainty from the prediction originates from the noisy sensor data and from the fact that the intention of human drivers cannot be directly measured. This problem is formulated as a partially observable Markov decision process (POMDP) with the intended route of the other vehicles as hidden variables. The solution of the POMDP is a policy determining the optimal acceleration of the ego vehicle along a preplanned path. Therefore, the policy is optimized for the most likely future scenarios resulting from an interactive, probabilistic motion model for the other vehicles. Considering possible future measurements of the surrounding cars allows the autonomous car to incorporate the estimated change in future prediction accuracy in the optimal policy. A compact representation results in a low-dimensional state-space. Thus, the problem can be solved online for varying road layouts and number of vehicles. This is done with a point-based solver in an anytime fashion on a continuous state-space. Our evaluation is threefold: At first, the convergence of the algorithm is evaluated and it is shown how the convergence can be improved with an additional search heuristic. Second, we show various planning scenarios to demonstrate how the introduction of different considered uncertainties results in more conservative planning. At the end, we show online simulations for the crossing of complex (unsignalized) intersections. We can demonstrate that our approach performs nearly as good as with full prior information about the intentions of the other vehicles and clearly outperforms reactive approaches.**

***Index Terms*—Autonomous driving, POMDP, decision making, interaction, motion planning under uncertainty.**

I. Introduction

RECENT efforts and developments in the area of Advanced Driver Assistance Systems (ADAS) show considerable improvements towards the availability of automated driving. Most recent ADAS of various car manufacturers raised the level of autonomy over the last years. Current projects are now aiming at a SAE level of 4 or higher, i.e., a fully autonomous car (see [1] for the definition of levels of autonomy).

This increase of autonomy requires algorithms that are capable of handling complex situations. Especially urban scenarios, such as intersections with multiple pedestrians, traffic lights, cars and bicycles pose a huge challenge and are even difficult to traverse for human drivers. This leads to 21.5% of fatalities and even 40% of all accidents in the US happening at intersections [2].

The goals of autonomous driving are to significantly reduce the number of accidents in traffic as well as to increase comfort, efficiency and create new solutions for individual transport in cities. In order to enable the automated car to navigate safely and reliably in such complex real-world environments, various technical challenges need to be taken into account.

Firstly, the perception of an automated vehicle is uncertain due to noise and range limitations of sensors and occlusions in the environment. Secondly, in order to generate safe trajectories for the ego vehicle, the motion of its surrounding traffic participants need to be predicted. The particular challenge of the prediction is caused by the uncertain information of their current state including hidden variables, such as unknown goal destinations. Additionally, the future behavior of the others is tightly coupled with the future trajectory of the ego vehicle. Thirdly, the motion of the ego vehicle must be collision-free, meet its kinematic and dynamic constraints and follow the traffic rules. To tackle the complexity of such an automated car, the problem is broken down into different tasks, see e.g., [3].

In this work we focus on the behavior generation of the ego vehicle by using motion planning techniques considering the uncertain prediction of the road users. As demonstrated in Fig. 1, we define four key reasons for the uncertain future trajectory of other road users. These are

1) the unknown intention of the other drivers
2) their unknown future longitudinal prediction
3) their probabilistic interaction with the ego vehicle
4) the noisy sensor measurements.

Considering these uncertainties in the planning of the future trajectory of the ego vehicle is required in many scenarios. Typically, these are scenarios in which the ego vehicle drives close to other vehicles, which may potentially cross the ego vehicle's path, whose intentions can only be estimated and whose behaviors are dependent on the behavior of the ego vehicle. This is the

Manuscript received September 1, 2017; revised December 5, 2017; accepted December 10, 2017. Date of publication January 5, 2018; date of current version March 19, 2018. *(Corresponding author: Constantin Hubmann.)*

C. Hubmann, J. Schulz, M. Becker, and D. Althoff are with the BMW Group, Munich 80788, Germany (e-mail: constantin.hubmann@bmw.de; jens.schulz@bmw.de; marvin.becker@bmw.de; daniel.althoff@bmw.de).

C. Stiller is with the Institute of Control, Karlsruhe Institut für Technologie, Karlsruhe 76131, Germany (e-mail: stiller@kit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2017.2788208

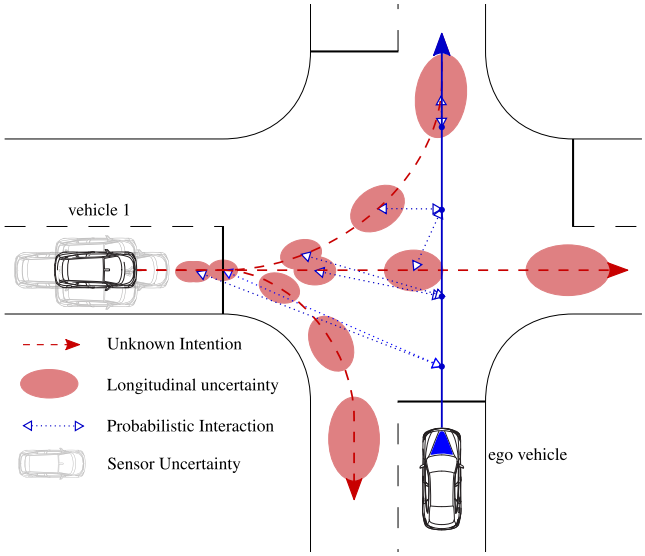

Fig. 1. A typical situation in which the ego vehicle has to decide for an action under various different uncertainties concerning the predicted motion of vehicle 1.

case for merging and crossing scenarios such as at intersections (see Fig. 1 for an example) and roundabouts.

The main contribution of this work is the presentation of an online *Partially Observable Markov Decision Process* (POMDP) algorithm for autonomous driving. It is able to incorporate all the aforementioned uncertainties in the planning problem. It provides optimized solutions for the behavior generation on intersections with an arbitrary layout and a variable number of traffic participants with unknown maneuver intentions. By planning in the belief state, it integrates the prediction and planning problem into a single, combined problem. Furthermore, the approach takes into account that during the execution of a trajectory, the ego vehicle will continuously gather more information about its surrounding. Hence, the algorithm expects that the intention estimation of the other traffic participants becomes more precise at a certain/estimated point in time and incorporates this into the decision making. Secondly, the longitudinal uncertainty of the other vehicle's motion is considered via a probabilistic and interactive motion model.

The result is a sequence of actions, that can be directly used as controller inputs or as goal states for a trajectory planner (e.g., presented by [4]) resulting in a smoother motion.

The performance of the proposed approach is evaluated regarding its convergence to the optimal policy and concerning the influence of various uncertainties in the planned trajectory. Simulated scenarios show that the algorithm is online capable and provide comparable results to scenarios with perfect information about the intentions of the surrounding drivers.

This work is an extension of the authors previous publication [5]. The new main contributions are

- an offline evaluation of the capabilities of the belief state planner for different uncertainties
- an extended action vector
- the introduction of a roll-out heuristic for faster convergence
- convergence analysis of the policy
- complexity analysis of the heuristic and the original problem
- introduction of split batch sampling to avoid dead time delays

II. Related Work

The associated problems of decision making, motion planning and prediction have been widely studied in the area of autonomous driving. However, the existing literature focuses hereby mostly on one of those problems, but does rarely formulate and solve them as a combined problem.

In general, motion planning needs prediction about the future motion of the other agents. Depending on what the prediction is based on and how it influences the own decision making, different categories of planning methods can be distinguished. We introduce the following three main classes: rule-based behavior, reactive behavior, and interactive behavior.

A. Rule-Based Behavior

In cases where the motion planning algorithm does not depend on an explicit trajectory prediction of the surrounding agents, we denote the resulting behavior as *rule-based*. Rule-based methods only incorporate the current state of all agents and define rules on how to act given a specific situation. Implicit assumptions about the behavior of others, as for example for the calculation of time to collision (TTC), may also be included. The behavior of the ego vehicle $\mathcal{T}_0$ can be described as a function of the current vehicle states $\mathcal{X}_{0:K}^{t_0}$:

$$\mathcal{T}_0 = f(\mathcal{X}_{0:K}^{t_0}). \tag{1}$$

During the DARPA Urban Challenge, first approaches for driving strategies in urban traffic have been proposed. Due to the controlled environment most of the teams, for example [6] or [7], focused on rule-based systems like Finite/Hybrid State Machines, that define the desired behavior of the vehicle in the scenarios encountered. [8] demonstrated successfully how decision making on highways can be solved by a rule-based expert system. In order to adapt these approaches to general urban traffic, more and more states and transitions have to be added to account for the huge variety of scenarios. This process is tedious and error-prone and thus does not scale well with increasing complexity of driving situations.

Since then, more general approaches based on *knowledge-bases* or *ontologies* have been developed: [9] used a combination of a map and car ontologies to guide a path planner in unsignalized intersections or narrow roads. [10] proposed the notion of semantic state spaces in which a graph-based planner is able to plan maneuvers according to traffic rules. The composition of these semantic states is based on an ontology.

B. Reactive Behavior

Motion planning algorithms that use an explicit, predicted trajectory of other agents as an input result in *reactive behavior*. Within this category, the prediction of others is assumed to be independent of the own planning and can thus be determined

first. The ego vehicle planning is then able to causally react to the estimated future behavior of its surrounding agents. The behavior of the ego vehicle can be described as a function of the ego vehicle's state and the trajectory prediction $\widetilde{\mathcal{X}}_{1:K}^{t_{0:T}}$ of the other agents:

$$\mathcal{T}_0 = f(\mathcal{X}_0^{t_0}, \widetilde{\mathcal{X}}_{1:K}^{t_{0:T}}). \tag{2}$$

Most motion planning algorithms result in reactive behavior, as they assume a predicted trajectory for the other vehicles. Within this class, interaction may happen implicitly by replanning but is not explicitly planned. The existing literature can be subdivided into two groups: using deterministic prediction and using probabilistic prediction.

1) Deterministic prediction: Reactive motion planning that uses a deterministic trajectory prediction as an input has been widely used. There are many different prediction models that can be utilized within this category, as for example constant velocity and turn rate or lane following models.

A planning based approach that assumes constant velocity of others is presented by [11]. Although interactions between the autonomous car and the predicted motion of other traffic participants are not considered, it provides good results due to its reactive and fast re-planning behavior. Typical optimal trajectory planning approaches, such as shown by [4], also expect an available deterministic motion prediction of the other vehicles. All of these approaches have in common, that they assume full knowledge over the states and intentions of the other vehicles and ignore interaction between traffic participants. Uncertainty is therefore not considered and only treated by replanning on updated prediction.

2) Probabilistic prediction: As human behavior is highly individual and complex, many approaches consider probabilistic prediction as an input for their motion planner. [12] and [13] create a probabilistic cost map considering various motion hypotheses to solve the global planning problem. [14] presents a non-conservative motion planner for cases where the other vehicle has two deterministic maneuver options (passing and yielding) at intersections. The authors present a trajectory planner, that keeps both reactions to possible maneuvers of the other vehicle open as long as possible.

In many approaches, various discrete options of the behavior of other vehicles and their probabilities are determined. However, their prediction is independent of the future motion of the ego vehicle and thus interactive solutions are not possible.

C. Interactive Behavior

As soon as the motion of others and the one of the ego vehicle are modeled as interdependent, the resulting behavior is called *interactive*. This interdependency makes it infeasible to separate the problems of prediction and planning. In cases where various different maneuvers of other traffic participants may be considered, simply respecting all possible trajectories may lead to suboptimal behavior or even to standstill, also known as the *freezing robot problem* demonstrated by [15]. However, they show that considering other vehicles' reactions in planning leads to less conservative results. Within this category, the behavior of the ego vehicle can be described as a function of the ego vehicle's state and the trajectory prediction of the other agents, which in turn depends on the behavior of the ego vehicle:

$$\mathcal{T}_0 = f(\mathcal{X}_0^{t_0}, \widetilde{\mathcal{X}}_{1:K}^{t_{0:T}}) \tag{3}$$

$$\widetilde{\mathcal{X}}_{1:K}^{t_{0:T}} = g(\mathcal{T}_0). \tag{4}$$

Again, two subclasses can be distinguished:

1) Centralized planning: In case that information sharing between agents is possible, the motion planning problem can be formulated for all agents as a multi-agent problem. A global (and even optimal) solution can be determined and distributed to the single agents. For centralized interactive planning, either vehicle-to-vehicle or vehicle-to-infrastructure communication is needed. Centralized planning approaches generally have in common, that they assume full knowledge over the states and intentions of the other vehicles. For a good overview on cooperative planning relying on V2V communication, refer to [16]. It cannot be assumed that perfect information sharing will be available for all existing agents, decentralized planning methods are essential.

2) Decentralized planning: In decentralized interactive planning, each agent has to plan for itself and predict how others will behave given a specific situation. The ego vehicle thus plans its trajectory by also considering the other vehicles. In addition, the other vehicles will adapt to the trajectory of the autonomous car during its execution.

[17] formulate the planning of the ego vehicle and the prediction of others as a combined problem using multi-agent planning. They enumerate discrete solution hypotheses on the basis of homotopy classes and determine continuous trajectories for all agents using a global cost-function. By observing the behavior of others, it is possible to estimate their intentions and choose an appropriate trajectory. [18] define motion models of other agents on highways based on the Intellligent Driver Model (IDM), which depends on the current situation including the ego vehicle's state. Applying Monte-Carlo tree search enables them to estimate how the scene will evolve over time given specific actions of the ego vehicle. [19] model the reactions of other drivers to the autonomous car by approximating the human as an optimal planner with a reward function that was acquired through inverse reinforcement learning. [20] demonstrate a trajectory planning approach that assumes the path of the others to be known but they adapt their velocity profiles to the ego trajectory. [21] employ reinforcement learning using neural networks to learn both low-level as well as maneuver policies. In order to explore future worlds of a complex and interactive environment, they utilize Monte Carlo tree search.

As this work realizes decentralized interactive behavior using belief state planning, the upcoming section gives a more detailed review of other belief state planning methods.

D. Belief State Planning

Belief state planning denotes a planning problem in which only a probabilistic distribution over the current state is available but not the state itself. Herein, a popular problem

formulation is the POMDP. It allows to model interactive behavior and uncertain motion with its probabilistic transition model. The solution to this formulation is an optimal policy, given the various uncertainties. Several publications pursue this direction and consider different forms of uncertainty in the planning model. *Offline* policies are powerful and are able to solve very complex situations: [22] present an algorithm for solving continuous-state and continuous-observation POMDP and show simulation results for an intersection scenario. A simplified intersection model is used and only sensor noise has been considered. [23] show how observation uncertainty and occlusions at intersections can be considered in the decision making. They use an offline planner and a scenario-specific learned discretization of the state space. The resulting policy is able to decrease the uncertainty about other vehicles' positions by considering the exploration capabilities of the robot. [24] show promising results for merging behavior at a T-junction. They formulate the problem as a Mixed Observability Markov Decision Process (MOMDP) with discretized state space using a simple behavior model considering the unknown intentions of other vehicles.

The drawback of all those offline methods is, that they are tailored for specific scenarios. Due to the huge variety of possible scenarios and vehicle configurations, a generally applicable policy cannot be precalculated.

Online methods on the other hand need to make a tradeoff between the solution quality and the complexity of the problem formulation, that depends on the state space size and the planning horizon.

[25] propose a POMDP planning framework for lane change maneuvers with noisy observations. They model the state space on high level features – whether lane changes are beneficial and possible – thus making the problem tractable. Due to the simplified state space, the method cannot be easily transferred to intersection planning. An online solver similar to the scenario by [24] is presented in [26]: the state space in the intersection area is discretized and both solution quality and runtime strongly depend on the selected grid. [27] model the behavior of other traffic participants with Hidden Markov Model (HMM), but the intention estimation is assumed deterministic during planning. Therefore, no uncertainty is considered during planning, hence no information gathering actions are performed. [28] extended their prior offline approach based on the DESPOT solver from [29] to include the unknown intentions of multiple pedestrians. A safe state can be reached very fast due to the low speed of their scenarios. Thus, advantages of long-term POMDP planning over reactive trajectory planning are not very prominent. [30] on the other hand showed the advantages over a reactive planner at a T-junction and a roundabout in terms of failure rate and time to traverse in simulated scenarios. [31] and [32] first estimate the distribution over potential policies and high-likelihood actions that each nearby car might be executing. Through closed-loop forward simulations, they can evaluate the outcomes of the interaction of their own vehicle with other traffic participants.

In contrast to the related work presented in this section, we aim to propose a POMDP unified planning framework that combines multiple aspects:

- it is generally applicable to any intersection geometry and for a variable number of traffic participants
- it considers the current and even predicts the future uncertainty of the intention/motion of other traffic participants and does not rely on (but can profit from) Vehicle-to-Vehicle (V2V) communication
- it is online and anytime capable. Given more time or computational power, the result will improve
- it operates on a continuous state space

III. Problem Statement

This work focuses on the online decision making for the ego vehicle, i.e., the generation of a sequence of desired accelerations $a_0 = (a_0^{t_0}, a_0^{t_1}, a_0^{t_2}, \cdots)$, e.g., for traversing an unsignalized intersection with an arbitrary layout and a variable number of other traffic participants with unknown intentions.

The path of the ego vehicle r_0 is assumed to be collision-free regarding static-obstacles and is either generated by a path planner a priori or simply retrieved from the road geometry of a given map. In a second step, the longitudinal velocity is planned along r_0 with our algorithm. This practice is referred to as *path-velocity decomposition* in the literature [33] and reduces the trajectory planning problem to a one dimensional workspace.

The environment is populated by a set of agents $\mathcal{N} = \{N_0, \cdots, N_K\}$, with $K \in \mathbb{N}_0$ and the ego vehicle N_0. Every other agent N_k, with $k \in \{1, \cdots, K\}$, has a set of future path hypotheses. The path of the ego vehicle, $r^{(0)}$, and all other path hypotheses are defined within the topological map $\mathcal{R} = \{r^{(0)}, r^{(1)}, \cdots, r^{(I)}\}$, with $I \in \mathbb{N}_0$, $r^{(i)} = \{\overrightarrow{q_{i,0}q_{i,1}}, \cdots, \overrightarrow{q_{i,J-1}q_{i,J}}\}$ for $i \in \{0, \cdots, I\}$, $j \in \{0, \cdots, J\}$ and $J \in \mathbb{N}_0$, and $q_{i,j} \in \mathbb{R}^2$ being the position of waypoint j of route i. Every agent N_k is mapped onto a corresponding path, s.t. $r_k : N_k \mapsto r \in \mathcal{R}$ on which it moves with velocity $v_k(t) \in [0, v_{\max}]$ for time $t \in [0, \infty)$. Every path has a set of following path hypotheses $\mathcal{M}^{(i)}$, with $\mathcal{M}^{(i)} = succ(r^{(i)})$. An agent traverses from its current to its next path r'_k with the unknown probability $P(r'_k|r_k)$.

As the various route elements may intersect with each other, an intersection function $c(r_i, r_j)$ is defined as

$$c(r^{(i)}, r^{(j)}) = \begin{cases} 1, & \text{if } r^{(i)} \cap r^{(j)} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j \in \{0, \cdots, I\}, i \neq j. \tag{5}$$

The different paths are retrieved from the road network and are therefore referred to as routes in the following. An example of this route definition can be seen in Fig. 2.

Given the uncertainty about the movement of the other cars, the autonomous vehicle has to continuously choose an optimal acceleration a^* to maximize the expected, cumulative discounted future reward:

$$a_0^* := \arg\max_{a_0} \mathrm{E}\left[\sum_{\tau=0}^{\infty} \gamma^\tau R^{t_{\tau+1}} | a_0^{t_\tau}\right]. \tag{6}$$

The reward should punish collisions, the total acceleration (providing comfort) and the deviation to a traffic-law and curvature based reference velocity.

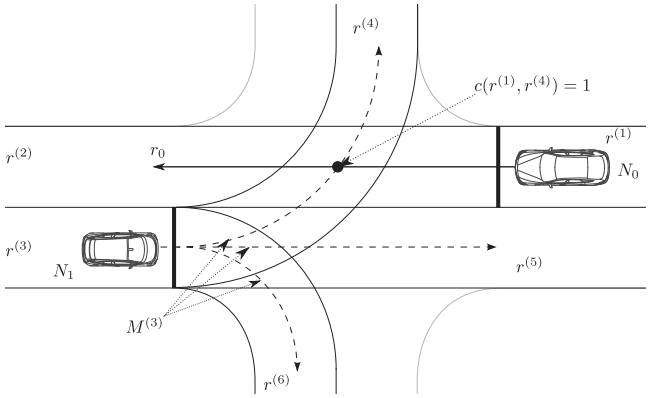

Fig. 2. A typical urban intersection with the autonomous car N_0 driving on path r_0 and one oncoming vehicle N_1 which may intersect with r_0.

IV. Approach

The main focus of this work is on determining the optimal longitudinal motion/acceleration of the ego vehicle while taking into account the uncertain future movements of other vehicles. Furthermore, the approach is independent of the intersection layout and generating a valid prediction of the other vehicles assuming that a valid path for the ego vehicle exists (Section II). By preprocessing the map layout, the routes of the vehicles can be encoded in only one dimension of the state-space (see Section IV-B). This allows a low-dimensional situation representation with the other vehicles' routes as hidden variables. Therefore, the problem description from Section III is formulated as a POMDP. This enables the approach to represent arbitrary situations and, in combination with the state-of-the-art, point-based *Toolkit for approximating and Adapting POMDP solutions in Real time* (TAPIR) (see [34]), to solve the problem online in a continuous state space. A motion and interaction model for the other traffic participants allows the planning of complex and interactive maneuvers. Additionally, we use a stochastic observation model considering potential future measurements of the surrounding vehicles, to optimize the resulting acceleration of the ego vehicle for a set of future scenarios.

Because the utilized Adaptive Belief Tree (ABT) algorithm samples multiple episodes to approximate the solution, the model properties of the POMDP (e.g., probability distributions) do not need to be specified explicitly but as a generative model.

As previously presented in the longitudinal planning approach in the authors' previous work [11], the algorithm provided here solves the motion planning problem in a coarse way on the behavioral layer (see [3] for the definition) and provides the generated, feasible subgoals to the trajectory planning layer for smooth execution.

A. Partially Observable Markov Decision Process

A POMDP is defined by the tuple $\langle \mathcal{X}, A, T, O, Z, R, b_0, \gamma \rangle$ with the possible states $\mathcal{X} \in \mathcal{X}$ and possible actions $a \in A$ to be executed by the agent. $T(\mathcal{X}', \mathcal{X}, a) = P(\mathcal{X}'|\mathcal{X}, a)$ is the transition probability of ending in state $\mathcal{X}'$ when executing action a in state $\mathcal{X}$. $R(a, \mathcal{X})$ is the reward for selecting action a in state $\mathcal{X}$. b_0 is the initial belief over all possible states. Additionally, the discount factor $\gamma \in [0, 1)$ is used to favor immediate rewards over rewards in the future.

The differences to a Markov Decision Process (MDP) are the possible observations $\mathbf{o} \in O$ and the observation function Z. While a MDP assumes that the current state $\mathcal{X}$ is given, i.e., directly observable, a POMDP assumes that the hidden state of the system can be estimated from an observation $o \in O$. The observation function $Z(\mathcal{X}', a, o) = P(o|\mathcal{X}', a)$ provides the probability to observe a certain observation o after taking action a and ending in the new state $\mathcal{X}'$. Instead of generating a policy $\pi : \mathcal{X} \mapsto a$, which maps a state on an action, the policy of a POMDP, $\pi : b \mapsto a$, maps a belief state b on an action. The belief state describes a probability distribution over all possible states. The intial belief is b_0. The solution of a POMDP is the optimal policy, π^*, which maximizes the expected, discounted cumulative reward

$$\pi^* := \arg\max_{\pi} \left(\mathrm{E}\left[\sum_{\tau=0}^{\infty} \gamma^{t_\tau} R(\mathcal{X}^{t_\tau}, \pi(b^{t_\tau})) | b^{t_0}, \pi \right] \right). \quad (7)$$

B. Statespace

To allow the modeling of interactive behavior in the motion model, all the scene's vehicles are represented in the state space. A certain state $\mathcal{X} \in \mathcal{X}$ is defined in continuous space as

$$\mathcal{X} = (\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K)^T. \quad (8)$$

$\mathcal{X}_0$ represents the state of the autonomous car and $\mathcal{X}_k \in \{1, \cdots, K\}$ the states of the surrounding vehicles. The position of the vehicles is described by the Frenet-Serret formulas on the vehicle's route r_k at position s_k.

While the transformation from the lane matched coordinate system L to the global coordinate system W, ${}^W T_L$, is not bijective, s_k may still be calculated from global coordinates as the particle's route r_k is known and part of the state space.

The autonomous car's state is thus defined as

$$\mathcal{X}_0 = \begin{pmatrix} s_0 \\ v_0 \end{pmatrix} \quad (9)$$

and the state of the other vehicles is

$$\mathcal{X}_k = \begin{pmatrix} s_k \\ v_k \\ r_k \end{pmatrix} \quad (10)$$

with the route r_k of vehicle N_k, being the hidden variable which cannot be observed directly. The notation of the state space is illustrated in Fig. 3.

C. Actions and Motion Model

The transition model of the other vehicles is defined for discrete time with a step size of Δt as

$$\begin{pmatrix} s_k' \\ v_k' \\ r_k' \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_k \\ v_k \\ r_k \end{pmatrix} + \begin{pmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \\ 0 \end{pmatrix} a_k, k \in \{1, \cdots, K\}. \quad (11)$$

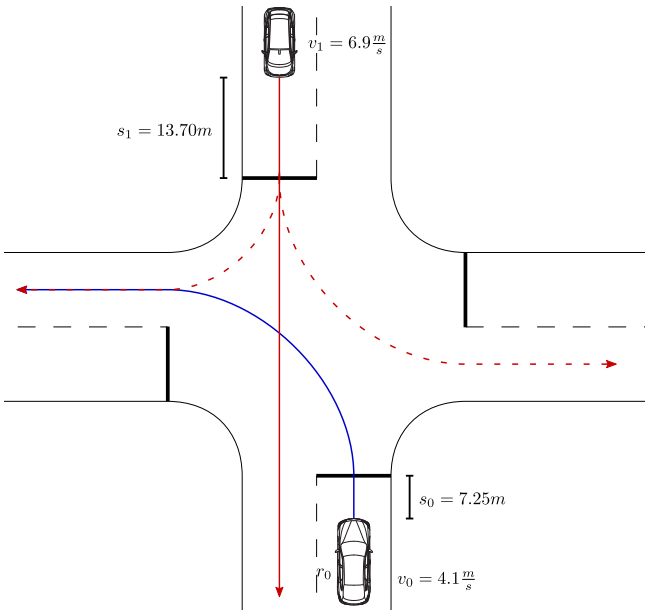

Fig. 3. Visualisation of the various variables of the autonomous and other car's statespace.

As mentioned in Sectio III, the route of any particle representing a vehicle is assumed to be constant, such that $r'_k = r_k$. The acceleration a_k is defined as the sum of an acceleration following the reference velocity, a_{ref}, and an interaction based acceleration a_{int}:

$$a_{int,k} = \begin{cases} 0, & \text{if } c(r_k, r_0) = 0, \\ -1.5, & \text{if } c(r_k, r_0) = 1 \wedge (t_{c,k} - t_{c,0}) \in [1, 5]. \end{cases} \tag{12}$$

t_c is the time until the arrival at the conflicting point of the two routes, assuming constant velocity v_k. The interaction based acceleration is an empirically chosen heuristic value, but could also be learned from training data or be a probabilistic function. The acceleration is also constrained by a maximum acceleration $a_{\max}$. The reference velocity is based on a maximum-lateral-acceleration based approach (see [11] for details). As the solver is particle-based, the generated acceleration of the motion model is additionally perturbed by simulated noise (σ^2). This allows on the one hand to represent various individual driving styles in longitudinal direction. Nonetheless, it is desirable to have an as good as possible motion model (either learned or tuned) to keep σ^2 low. This is the case as a high σ^2 leads to a high degree of uncertainty of the future position/velocity of the other vehicles, which leads to a more conservative policy. This is the case as the distribution over future positions grows, such that there are less potential states without a collision. The resulting total acceleration of the motion model may now be written formally as follows:

$$a_k = \min\left(a_{ref,k} + a_{int,k}, a_{\max}\right) + \mathcal{N}(0, \sigma^2) \tag{13}$$

The transition model of the ego vehicle is defined in the same way as for the other vehicles in (11), except that its route must not need to be incorporated in its state. The ego vehicle's acceleration is determined by the generated policy and the current belief state, s.t. $a_0 = \pi^*(b)$.

D. Reward and Transition Costs

The reward $R(\mathcal{X}, a)$ is defined as follows:

$$R(\mathcal{X}, a) = R_{\text{crash}}(\mathcal{X}) + R_v(\mathcal{X}) + R_{\text{acc}}(a). \tag{14}$$

There are various rewards during exploration: a collision is punished with high negative rewards (R_{crash}). Secondly, the deviation to a reference velocity (defined as a smooth velocity on a road without vehicles, see [11]) creates different costs, depending on the sign of deviation. Therefore, $R_v = -K_{v+}(v_{ref} - v_0)^2$, if $v_0 > v_{ref}$ and $R_v = -K_{v-}(v_{ref} - v_0)$, if $v_0 < v_{ref}$. By quadratically punishing too high velocities, the ego vehicle is more unlikely to clearly overshoot the desired velocity. By only punishing too low velocities in a linear way, the planner tries to drive with desired velocity but allows for slower solutions (e.g., because of a temporarily occupied lane). Changing the acceleration has costs of R_{acc} to provide comfort.

E. Observation Space

An observation $o \in O$ is defined as

$$\mathbf{o} = (o_0, o_1, \cdots, o_K)^T \tag{15}$$

with the observation of the ego vehicle, o_0, and of the other vehicles, $o_k \in \{1, \cdots, K\}$.

As sensor noise is not considered in the algorithm (opposed to the offline scenario generation) and the route of the autonomous car is known, its state is fully observable and is denoted as

$$o_0 = \begin{pmatrix} s_0 \\ v_0 \end{pmatrix}. \tag{16}$$

As the route of the other vehicles is not directly observable, their observations are defined in global coordinates as

$$o_k = \begin{pmatrix} v_k \\ x_k \\ y_k \end{pmatrix}. \tag{17}$$

F. Observation Model and Prediction

The ABT algorithm solves the POMDP by generating the belief tree via sampling of particles with different route hypotheses. Therefore, the observation model $Z(o, \mathcal{X}', a) = P(o|\mathcal{X}', a)$ must not be given explicitly, but rather an potential observation must be sampled given a new state $\mathcal{X}'$ following action a.

Although the route r_k of another vehicle N_k is unobservable, it is part of its state $\mathcal{X}_k$ and therefore $x'_k = (s'_k, v'_k, r'_k)^T$ is generated by the transition model for any particle. With the unambiguous transformation ${}^W T_L$ a corresponding observation of the new state can be created: $(s', v', r')/xmapsto^W T_L(v'_{obs}, x'_{obs}, y'_{obs})$ (see Fig. 2).

To simulate the prediction uncertainty concerning the chosen, unknown route of another vehicle in future time steps, a simple discriminative classifier is employed. We propose a Naive Bayes classifier with a 2-dimensional feature vector $\mathbf{f}_k$ (velocity and

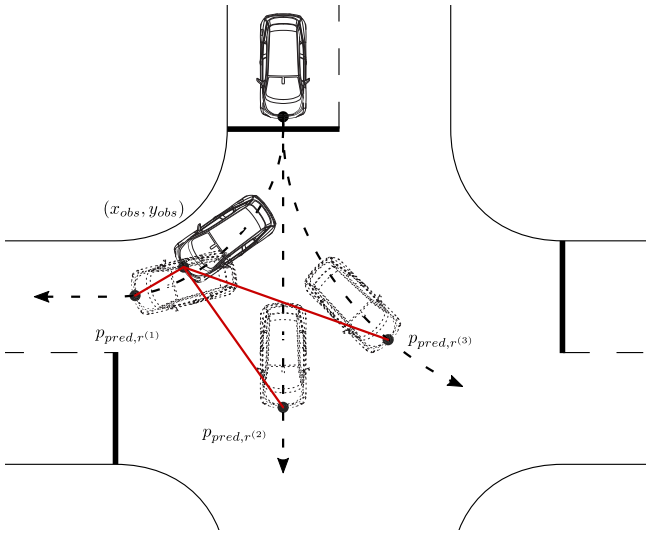

Fig. 4. Demonstration of the distance feature $f_{k,2}$. It is defined as the euclidean distance between the simulated observation's position (x_{obs}, y_{obs}) and the assumed euclidean position $p_{pred,r^{(i)}}$ given a certain route hypothesis $r^{(i)}$.

position based, see (Fig. 4) for vehicle N_k, that can be generated from the observation space:

$$\mathbf{f}_k = \begin{pmatrix} f_{k,1} \\ f_{k,2} \end{pmatrix} = \begin{pmatrix} |v'_k - v_{ref,r^{(i)}}(s'_k)| \\ \left\| [x'_k \; y'_k]^T - [x_{k,pred,r^{(i)}} \; y_{k,pred,r^{(i)}}]^T \right\|_2 \end{pmatrix}. \tag{18}$$

The probability of vehicle N_k being on a certain route $r_k = r^{(i)} \in M^{(i)}$ may then be defined via Bayes rule as

$$P(r_k = r^{(i)} | f_{k,1}, f_{k,2}) = \frac{P(r^{(i)}) P(f_{k,1}, f_{k,2} | r^{(i)})}{P(f_{k,1}, f_{k,2})}. \tag{19}$$

With the assumption that every route has the same a-priori probability $(P(r_k = r^{(1)}) = P(r_k = r^{(2)}) = P(r_k = r^{(3)}) \cdots)$, the law of total probability and the assumption of independent features, (19) may be rewritten to:

$$P(r_k = r^{(i)} | f_{k,1}, f_{k,2}) = \frac{P(f_{k,1} | r^{(i)}) P(f_{k,2} | r^{(i)})}{\sum_{l=1}^{I} P(f_{k,1} | r^{(l)}) P(f_{k,2} | r^{(l)})}. \tag{20}$$

While $P(f_{1/2} | r^{(i)})$ can be learned from sample data, it is simply assumed as normally distributed with $P(f_1 | r^{(i)}) = \mathcal{N}(0, 4.0)$ and $P(f_2 | r^{(i)}) = \mathcal{N}(0, 6.0)$ to simulate the prediction.

The observation o_k is now generated for every particle based on a potential next route $r^{(i)}$ that is sampled from (20).

V. Implementation

A. Theoretic Background on Solving POMDPs

POMDP problems are often considered to be computational infeasible. This is the case as even a discrete, limited number of states $|\mathcal{X}|$ leads to a infinite number of belief states $|\mathcal{B}|$, because the probability distribution over the current belief is continuous. This problem leads to point-based POMDP algorithms which approximate the optimal solution by using a discretized belief space $\mathcal{B}$. For example the well-known Point Based Value Iteration (PBVI) algorithm [35] belongs to this kind of solvers. Another aspect is the reachable belief space. It is a subset $\mathcal{R}(b^{t_0})$ of beliefs which are reachable from the current belief b^{t_0} as outlined in [36]. Searching for the solution in only this subspace reduces the problem to a smaller space, but requires to solve the problem online as the solution is only calculated starting from the current belief.

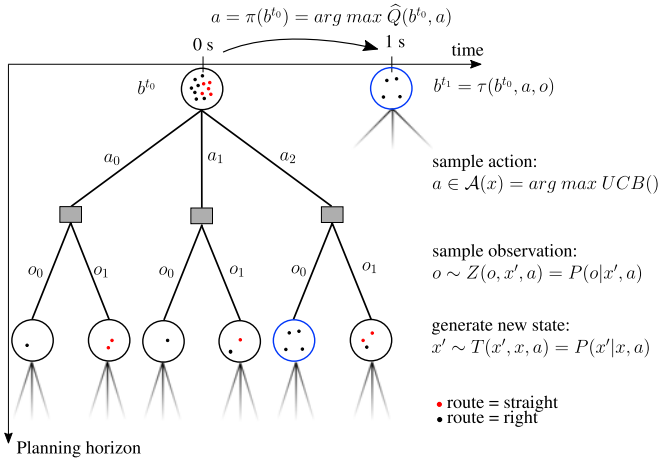

Fig. 5. Approximation of the belief tree by Monte Carlo sampling episodes starting at the current belief.

To approximate the optimal solution online, we use the ABT algorithm [37]. ABT approximates the optimal policy by sampling potential future episodes to construct the belief tree.

B. Tracking of Belief State Via Particle Filter

The current belief state is represented via a set of particles containing the possible state instances (see e.g., Fig. 5). The current belief state is tracked by the solver via an unweighted particle filter using simple rejection sampling. Tracking the belief state, instead of using simple resampling at the actual observation, allows not only to get a better state estimation, but to preserve gained knowledge about the previously approximated value function $\hat{V}(b)$ of a certain state.

C. Monte Carlo Sampling of Episodes

The goal of the ABT algorithm is to determine the optimal policy π^* ((7)). The approximated policy π is retrieved from the approximated Q-function

$$\widehat{Q}(b, a) \approx R(b, a) + \gamma \sum_{o \in O} \tau(b, a, o) V^*(\tau(b, a, o)) \tag{21}$$

such that

$$\pi(b) := \underset{a \in \mathcal{A}}{\arg\max}\ \widehat{Q}(b, a) \tag{22}$$

with $\tau(b, a, o)$ being the transition from one belief state to the next, given an received observation o and executed action a. $\widehat{Q}(b, a)$ is calculated by selecting a particle randomly from the current belief and sampling a potential episode from the particle's state instance up to the planning horizon. $\widehat{Q}(b, a)$ is then

set to the average value of all episodes starting at belief b with action a. The episode sampling is visualized in Fig. 5.

While observations and transitions can be sampled from the corresponding models explained in Section IV, the actions have to be selected. At first the algorithm tries every available action in the current belief node. During further expansion, the most promising actions are chosen. The action a to be expanded is therefore selected via the following formula as proposed in [37]:

$$a := \begin{cases} \sim A', & \text{if } A' \neq \emptyset \\ \underset{a \in \mathcal{A}}{\arg\max}\, \widehat{Q}(b,a) + c\sqrt{\frac{log(|H_b|)}{|H_{(b,a)}|}}, & \text{otherwise} \end{cases} \tag{23}$$

with A' being the uniform distribution over all actions which have not yet been selected in the corresponding belief state, $|H_b|$ the number of episodes starting in belief b and $|H_{(b,a)}|$ the number of episodes starting in b by executing action a. A scalar, proportional exploration coefficient is defined as c and allows to trade off between exploration and exploitation. This type of action selection is called Upper Confidence Bound (UCB). The idea of this UCB algorithm is on the one hand to balance exploration and exploitation and allows the algorithm on the other hand to converge to the optimal Q-function, given enough runtime. As the estimated Q-function $\widehat{Q}(b,a)$ is the mean over all episodes starting at b, it converges only to the optimal Q-function $Q^*(b,a)$ if the actions of the optimal policy are selected more often via the UCB action-selection (see (23)). While this may lead to a non-optimal policy when not enough runtime is given, it can be shown that the algorithm searches first in directions which result in a more conservative policy. This is the case as the branch of the assumed optimal action is most likely expanded and a detected collision leads to a different sampling focus.

D. Deterministic Upper Bound Graph Search Heuristic

If a new belief state is explored, the initial value of the belief state is set to a heuristic estimate to allow the UCB algorithm to converge faster. This heuristic is calculated by performing a full roll-out from the current belief. The roll-out policy shall be an upper bound on the value of the current state [34]. The idea is to use a deterministic roll-out policy which does not respect the uncertainties concerning state and transition model of the other vehicles. Hence, we use a graph search, similar to the one used in [11], but without the Inevitable Collision State (ICS) heuristic. This results in a Dijkstra graph search. While this may be computationally too complex as heuristic, we use an approximation $h(n)$ to the complete exhaustive search result by aborting the graph search after n-steps and using a constant velocity approach until the optimization horizon. We define $\mathcal{C}(h(n))$ as the worst case exploration complexity of nodes of the heuristic, which results in

$$\mathcal{C}(h(n)) = |A|^n \cdot (t_{hor} - n) \tag{24}$$

which is shown in Fig. 6.

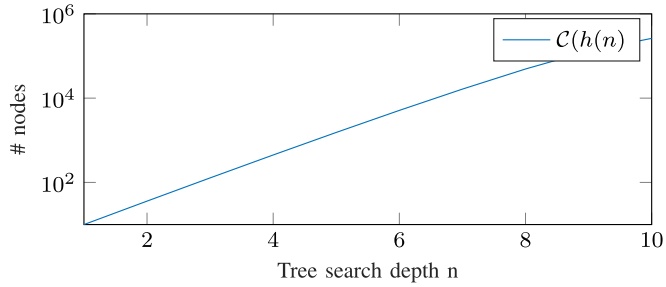

Fig. 6. Time complexity of the roll-out policy h, dependent on a varying search depth n, followed by constant velocity until the planning horizon t_{hor}.

TABLE I
EVALUATION PARAMETERS

c	20000	t_{hor}	8	R_{crash}	10000
f	4 Hz	K_{v+}	-100	K_{v-}	-100
γ	1				

E. Sampling Timing of Episodes

The ABT algorithm is anytime and therefore improves its solution with additional runtime. At the same time, the used particle filter allows not only to track the state but also to keep the already generated episodes which belong to the particle.

While more computing time optimizes the solution, it also introduces a trade-off between approximation quality and replanning frequency. As this work focuses on the behavior generation, we use 200 ms for the optimization of the policy. Nonetheless, we replan with 1 Hz such that the step size of the algorithm is 1 s which allows for larger planning horizons. Therefore, we use the remaining time of 800 ms to generate further episodes in order to already improve the belief tree approximation for the next time step.

VI. RESULTS

In this section, we evaluate the convergence to the optimal policy in offline scenarios and show the planning behavior in simulated online scenarios. We use a proprietary simulator for the evaluation of the two scenarios. The other vehicles are controlled by the agent model developed in [38].

A. Evaluation Settings

While the settings for the demonstrated online scenarios are given in [5], the parameters of all the other scenarios are given in Table I if not mentioned differently. The system (containing the simulation environment and the algorithm) runs on a Intel Core i7-4910MQ CPU with 2.9 GHz.

B. Convergence

Fig. 9 shows the convergence to the optimal policy for varying runtime and depending on the used heuristic. The underlying scenario is shown in Fig. 7. We define the loss function $|\widehat{Q}(b,a^*) - Q^*(b,a^*)|$ as the absolute difference between the optimal action of the ground truth and the approximated

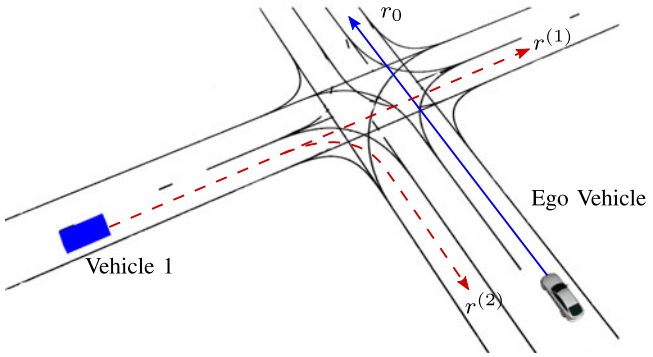

Fig. 7. Intersection scenario for the evaluation of convergence to the optimal policy. The scenario is chosen such that the initial route probabilities are $p(r_1 = r^{(1)}) = 0.05$ and $p(r_1 = r^{(2)}) = 0.95$. The initial velocities are $v_0 = 8.6\frac{m}{s}$ and $v_1 = 8\frac{m}{s}$.

Q-value. The ground-truth is generated by sampling episodes until convergence is reached. The upper plot of Fig. 9 shows the absolute difference to the optimal value is not better approximated than without heuristic. Nonetheless, it is more interesting if the policy, retrieved from the approximated Q-function, delivers the optimal action. The lower plot of Fig. 9 shows the percentage of selecting a non-optimal action. It can be seen, that all heuristics result in a significantly reduced probability for non-optimal action selection for a runtime of up to 500 ms. It can also be seen, that the constant velocity heuristic leads to suboptimal results for longer sampling time as it is a non consistent heuristic (i.e., potentially overestimating costs). The 3-step Dijkstra performs even better than the n-step Dijkstra heuristic. This is the case as the 3-step Dijkstra heuristic needs less calculation time and allows therefore more episodes to be sampled to approximate the policy. This can be seen in the middle plot of Fig. 9.

In general, the problem may become more complex and therefore harder to solve for an increasing number of other vehicles and an increasing number of potential routes. Nonetheless, as the algorithm only searches in the reachable belief space, adding non relevant vehicles (i.e., not directly influencing the ego vehicle's reward) does not make the problem harder to solve. We even noticed, that also problems with more relevant vehicles may lead to faster convergence as the reachable belief space is more constrained by the trajectories of the other vehicles. This may lead then to a smaller exploration space of the algorithm and therefore faster convergence. In summary it may be said, that the convergence of the algorithm does not depend directly on the number of vehicles and routes. It varies very much on the micro traffic situation and is therefore difficult to evaluate.

C. Multi-Trajectory Planning

The approximated policy represents various potential future trajectories. This is the case, as a potential trajectory depends not only on the current but also on future beliefs which may be encountered during the execution of the trajectory. Fig. 8 shows the example policies for the intersection scenario shown in Fig. 7. In this scenario, two possible situations exist: vehicle 1 turns right such that the ego vehicle can cross the intersection directly or vehicle 1 goes straight, such that the ego vehicle has to yield and can only cross the intersection afterwards. The QMDP planner [Fig. 8(a)] has to slow down immediately as it is not able to incorporate future observations in the planning phase. Therefore it directly needs to react to both possible situations. On the contrary, our POMDP planner [see Fig. 8(b)–(e)] for various considered uncertainties) is able to consider both options which results in a policy that delays the decision to the future and decides after receiving further measurements. It can also be seen, that the introduction of further uncertainties (such as motion model uncertainty [Fig. 8(c)], observation uncertainty [Fig. 8(d)] and sensor noise [Fig. 8(f)] results in a more conservative policy. One can also notice, that the introduction of the interaction model [Fig. 8(e)] allows for a less conservative policy (i.e., less braking, faster intersection approach), compared to not incorporating interaction [Fig. 8(d)]. This is the case because the planner takes into consideration during the forward simulation that the other vehicle will react to the planned trajectory of the ego vehicle and e.g., let our autonomous car cross the intersections first if it reaches the intersection first.

D. Merging on a T-Junction

As first online scenario, a merging scenario is presented (see Fig. 10). The ego vehicle approaches a T-Junction where it wants to perform a left turn. While the other vehicles on the main road have the right of way, the ego vehicle must yield if required. The ego vehicle has to decide whether to merge before or after vehicle 1, which is approaching the intersection from the right. While merging before vehicle 1 is absolutely possible, another vehicle (vehicle 2) is approaching the intersection from the left and makes the merging maneuver more complex. Vehicle 2 has two options: driving straight and intersecting the planned path of the ego vehicle or turning right with no effect on the ego vehicle. In addition to the uncertainty of the chosen route of vehicle 2, the ego vehicle also has to consider the uncertain longitudinal prediction of both other vehicles which is realized by the interactive and probabilistic motion model. This uncertainty is incorporated by adding Gaussian noise on the interactive motion model (see (13)).

Given the two options of vehicle 2, two different scenarios may now evolve. In the first one, vehicle 2 drives straight. If that happens, the ego vehicle has to brake down to let the other vehicle pass and has no more chance to merge before vehicle 1. In scenario 2 (vehicle 2 turns right), the ego vehicle has the possibility to merge before vehicle 1, but to enable this, the uncertainty about the behavior of vehicle 2 must be incorporated into the decision making process.

Fig. 11 shows the driven trajectory of the ego vehicle for different cases as well as the desired reference velocity in the upper figure. The lower figure shows the estimated probability for each maneuver of vehicle 2 over time. It can be seen that the ego vehicle accelerates up to the desired curve velocity (defined in [11]). After 9 seconds, the predicted behavior is still

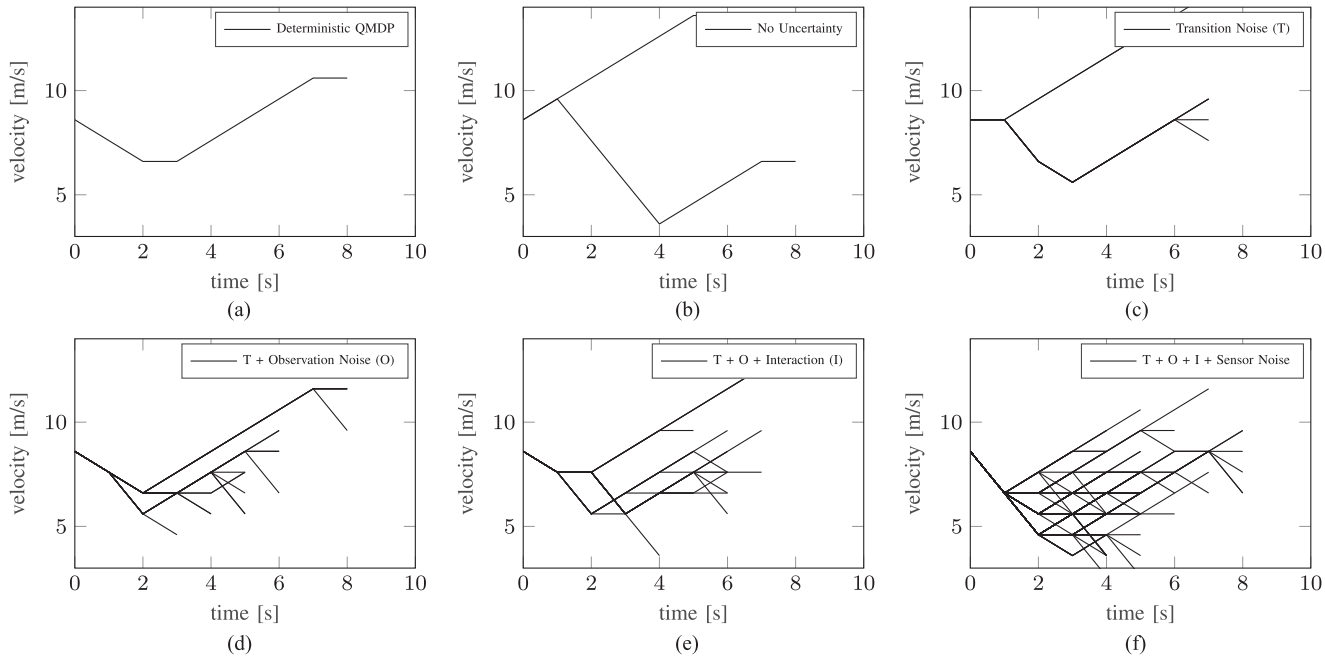

Fig. 8. Comparison of deterministic QMDP and POMDP with different levels of uncertainty. The plots (a)–(f) show the various trajectories which are part of the approximated optimal policy. The policies were recorded with an adapted UCB factor of 200000 to retrieve many trajectories.

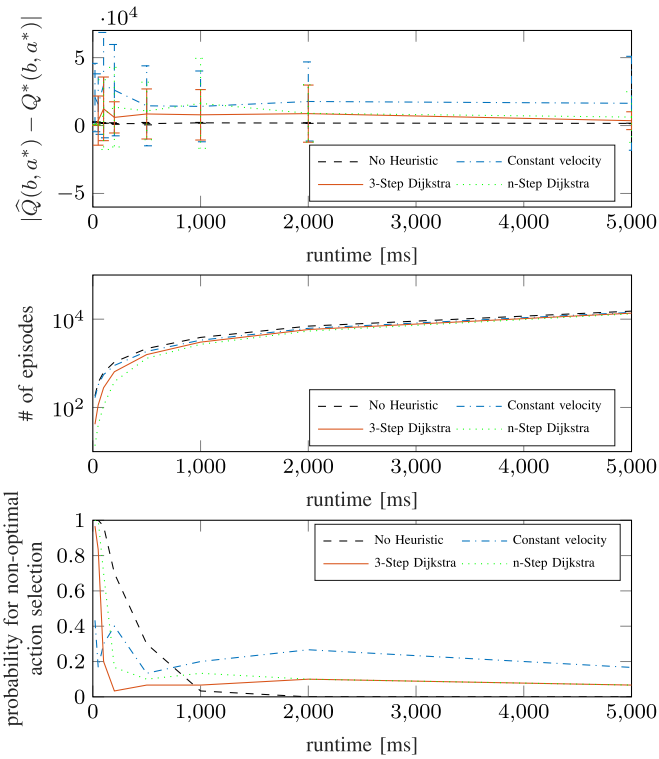

Fig. 9. Comparison of the convergence of different heuristics for the scenario presented in Fig. 7.

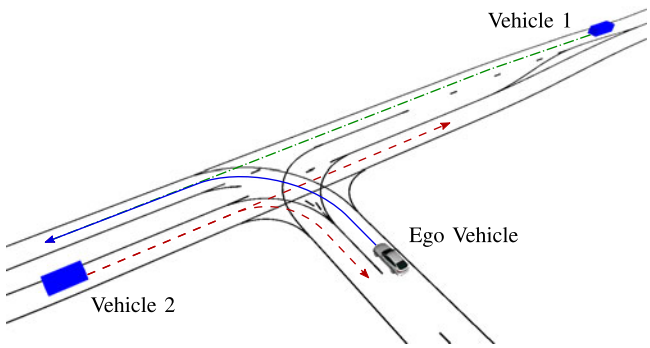

Fig. 10. Top view of the T-junction scenario.

ambiguous, therefore the planner starts to decelerate slightly to reduce the probability of a collision and to have more time to receive new measurements which are expected by the algorithm to lead to a more reliable prediction of vehicle 2. Thus, the two possible options, yielding to vehicle 1 or merging immediately, are kept open for the ego vehicle. This behavior (also known as *information gathering*) is the result of the policy because the observation model has simulated, that the next measurements will lead to a less ambiguous prediction. Because of the observation model, it can even *infer* at what point in time the prediction becomes unambiguous and approach the intersection accordingly. After 12 seconds, the prediction is precise enough such that the ego vehicle can cut in before vehicle 1.

For the same scenario, Fig. 12 plots the position over time and especially the predicted time interval during which the other vehicles occupy the areas that conflict with the path of the ego vehicle. It can be seen that the point of a conflict between the ego vehicle and vehicle 2 is constantly postponed while vehicle 2 breaks upon the intersection, leading to even having no conflict at all when the turning behavior of vehicle 2 becomes apparent.

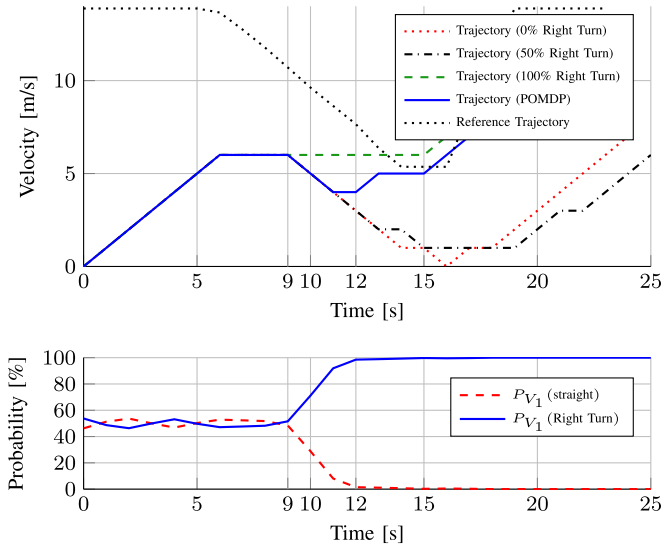

Fig. 11. Evaluation of the T-junction scenario where vehicle 2 is turning right. The upper figure shows the planned velocities over time when the POMDP planning is executed with different fixed probabilities and with the probability from prediction and POMDP sampling. The lower figure shows the probability for a conflict area with vehicle 2 over time in the evaluation without fixed probabilities.

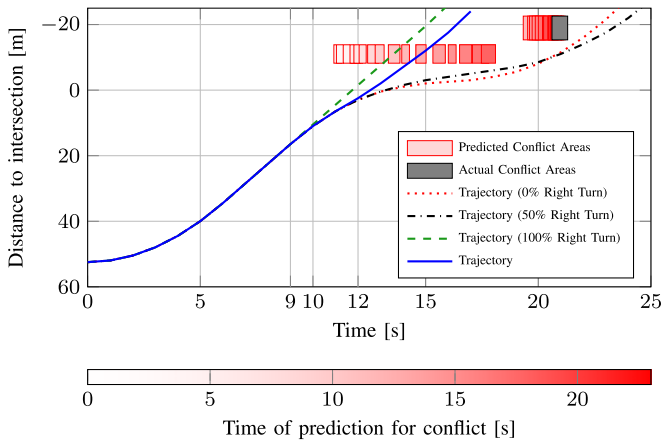

Fig. 12. Evaluation of the T-junction scenario with a right turning vehicle. The figure shows the planned positions over time when the POMDP planning is executed with different fixed probabilities and with the probability from prediction and POMDP sampling.

Our POMDP approach is compared to different prediction models with fixed probabilities in all figures. It can be seen, that our algorithm performs nearly as good (it is able to merge before vehicle 1) as with exact information of the other vehicles' behavior (i.e., V2V communication, resulting in 100% turn prediction accuracy). Approaches that neither consider future observations nor the evolvement of various potential scenarios have to assume that vehicle 2 always goes straight (0% right turn) or to respect both options equally (50% right turn) to ensure safety. This results, as shown in Fig. 11, in a conservative suboptimal trajectory, such that the ego vehicle cannot merge before vehicle 1.

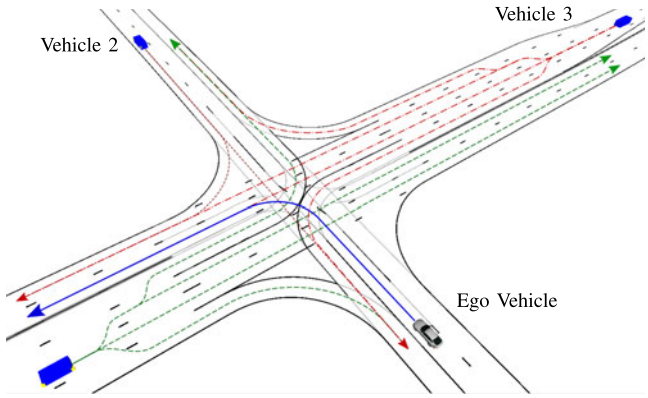

Fig. 13. Top view of the complex intersection scenario with three other vehicles.

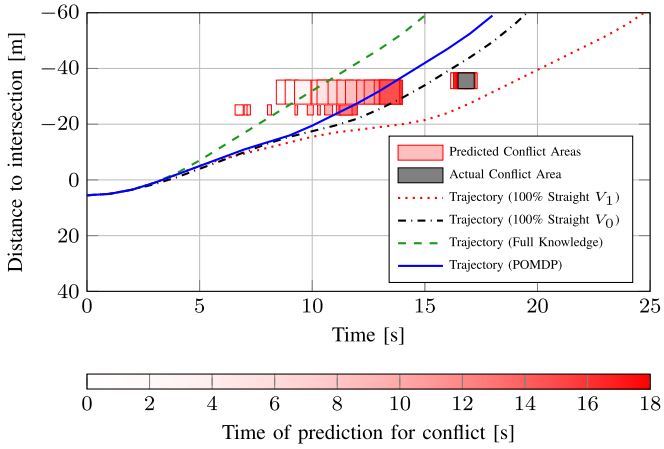

Fig. 14. Evaluation of the complex intersection scenario. The figure shows the planned positions over time for four different evaluations as well as the predicted and actual conflict areas. The lowest rectangles represent the conflict area with vehicle 1, the rectangles in the middle the conflict area with vehicle 2 and the upper rectangles the conflict area with vehicle 3.

E. Left Turn On a Complex, Intersection

As mentioned in the beginning, one focus of the approach is to handle various intersections online. For demonstration purposes, a more complex scenario is introduced to demonstrate the generic applicability. Fig. 13 shows a large unsignalized intersection, with in total 10 different possible routes for the other three vehicles. Our algorithm produces similar results compared to no uncertainty in the prediction (e.g., use of V2V) as shown in Fig. 14.

VII. Conclusion

The main contribution of this work is the presentation of an online capable POMDP framework for autonomous driving in varying situations. We define four key uncertainties which occur during motion planning for automated driving and demonstrate how they can be included in our framework. These are the unknown intentions of other drivers, their probabilistic longitudinal prediction, sensor noise and the interaction between surrounding traffic and the ego vehicle. The framework also allows

to only consider certain arbitrary uncertainties as demonstrated in Fig. 8 which results in faster convergence. Even considering no uncertainties is possible and leads to a deterministic motion planning algorithm which still considers combinatorics and interaction.

By solving the problem with a probabilistic forward simulation for different scenarios, not only the current prediction uncertainty is considered but also in what way the prediction accuracy will change in the future. Additionally, it is also modeled how the other vehicles react to the ego vehicle. This modeling allows the ego vehicle to optimize its behavior for different future scenarios. It results for example in postponing a decision (e.g., to merge or not to merge) by decelerating under the knowledge that more information (i.e., a better prediction) will be present in the future.

It is demonstrated that the approach allows for a behavior which is only possible with full information (V2V) for reactive planners. Convergence analysis considering the optimality of the approximated solution and the benefit of an additional heuristic demonstrate that an approximate solution to the complex POMDP formulation can be found online. Nonetheless, it cannot be guaranteed that the algorithm finds the optimal solution online. This is because we use probabilistic sampling which leads to a varying convergence time and because the needed number of episodes may also scale proportionally with the number of other vehicles, potential routes, planning horizon, considered uncertainties and especially the starting belief state, i.e., the setting of the traffic situation. Nonetheless, it is shown, that the algorithm generates a more conservative policy, e.g., choosing a slower velocity, in the case that it did not converge to the optimal policy.

The framework can be further improved by using more realistic motion/interaction models for the other vehicles. Also, the integration of a more advanced particle filter for better tracking of the current belief and/or further heuristics would decrease the required amount of sampled episodes. The authors intend to pursue the approach, and direct further research to these topics.

REFERENCES

[1] SAE International, "Taxonomy and definitions for terms related to onroad motor vehicle automated driving systems," SAE International, Tech. Rep. SAE J 3016, 2014.

[2] (2010). Crash factors in intersection-related crashes: An on-scene perspective. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/ Public/ViewPublication/811366

[3] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 215–220.

[4] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet Frame," *IEEE Int. Conf. Robot. Automat.*, 2010, pp. 987–993.

[5] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Proc. Intell. Veh. Symp.*, 2017, pp. 1671–1678.

[6] S. Kammel *et al.*, "Team Annieway's autonomous system for the 2007 darpa urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 615–639, 2008.

[7] M. Montemerlo *et al.*, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.

[8] M. Ardelt, C. Coester, and N. Kaempchen, "Highly automated driving on freeways in real traffic using a probabilistic framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1576–1585, Dec. 2012.

[9] L. Zhao *et al.*, "Ontology-based decision making on uncontrolled intersections and narrow roads," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2015, pp. 83–88.

[10] R. Kohlhaas, T. Bittner, T. Schamm, J. M. Zöllner, and J. Z. Marius, "Semantic state space for high-level maneuver planning in structured traffic scenes," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2014, pp. 1060–1065.

[11] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2016, pp. 1010–1016.

[12] F. Damerow and J. Eggert, "Risk-aversive behavior planning under multiple situations with uncertainty," in *Proc.IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 656–663.

[13] M. Ruf *et al.*, "Situation prediction and reaction control (SPARC)," in *Workshop Fahrerassistenzsysteme*, vol. 9, B. Färber *et al.*, Eds., 2014, pp. 55–66.

[14] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proc. 2016 IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 459–464.

[15] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 797–803.

[16] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[17] J. Schulz, K. Hirsenkorn, J. Löchner, M. Werling, and D. Burschka, "Estimation of collective maneuvers through cooperative multiagent planning," in *Proc. 2017 IEEE Intell. Veh. Symp.*, 2017, pp. 624–631.

[18] D. Lenz, T. Kessler, and A. Knoll, "Tactical cooperative planning for autonomous highway driving using monte-carlo tree search," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 447–453.

[19] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverages effects on human actions," in *Proc. Robot., Sci. Syst. Conf.*, 2016, pp. 66–73.

[20] J. R. Ziehn *et al.*, "A tractable interaction model for trajectory planning in automated driving," in *Proc. 2016 IEEE 19th Int. Conf., Proc. Intell. Transp. Syst.*, 2016, pp. 1410–1417.

[21] C. Paxton, V. Raman, G. D. Hager, and M. Kobilarov, "Combining neural networks and tree search for task and motion planning in challenging environments," in *Proc. Int. Conf. Intell. Robots Syst.*, 2017, pp. 6059–6066.

[22] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1288–1302, Jun. 2014.

[23] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decisionmaking under uncertainty for autonomous driving using continuous pomdps," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2014, pp. 392–399.

[24] V. Sezer *et al.*, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 3578–3585.

[25] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proc. IEEE Int. Conf. Intell. Transport Syst.*, 2013, pp. 2063–2067.

[26] Y. Bai, Z. J. Chong, M. H. Ang, and X. Gao, "An online approach for intersection navigation of autonomous vehicle," in *IEEE Int. Conf. Robot. Biomimetics*, 2014, pp. 2127–2132.

[27] W. Song, G. Xiong, and H. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Math. Probl. Eng.*, vol. 2016, pp. 1–15, 2016.

[28] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 454–460.

[29] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online POMDP planning with regularization," *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 1772–1780, 2013.

[30] W. Liu, S.-W. W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online pomdp," in *Proc. IEEE Intell. Veh. Symp.*, 2015, pp. 1126–1133.

[31] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *Proc. 2015 IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1670–1677.

[32] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction," in *Proc. Robot., Sci. Syst. Conf.*, 2015, pp. 2290–2297.
[33] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.
[34] D. Klimenko, J. Song, and H. Kurniawati, "Tapir: A software toolkit for approximating and adapting pomdp solutions online," in *Proc. Australasian Conf. Robot. Automat.*, 2014.
[35] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration an anytime algorithm for POMDPs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1025–1032.
[36] D. Hsu, W. S. Lee, and N. Rong, "What makes some POMDP problems easy to approximate?," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 689–696.
[37] H. Kurniawati and V. Yadav, "An online POMDP solver for uncertainty planning in dynamic environment," in *Proc. Int. Symp. Robot. Res.*, 2013, pp. 611–629.
[38] A. Hochstädter, P. Zahn, and K. Breuer, "A comprehensive driver model with application to traffic simulation and driving simulators," in *Proc. 1st Hum.-Centered Transp. Simul. Conf.*, 2001.

Constantin Hubmann received the Bachelor's and Master's degrees in electrical engineering and information technology from Technische Universität München, Munich, Germany, in 2012 and 2014, respectively. He is currently working toward the Doctoral degree at the Chair of Measurement and Control Engineering, Karlsruher Institut für Technologie, Karlsruhe, Germany, in cooperation with the BMW Group. His research interests include maneuver decisions and motion planning under uncertainty for autonomous driving.

Jens Schulz received the Bachelor's and Master's degree in electrical engineering and information technology from Karlsruhe Institute of Technology, Karlsruhe, Germany in 2012 and 2015, respectively. He is currently working toward the Doctoral degree at the Chair of Robotics and Embedded Systems, Technical University of Munich, Germany, in cooperation with the BMW Group. His research interests include interaction-aware and context-dependent behavior prediction and intention estimation of traffic participants for autonomous vehicles.

Marvin Becker received the Master's degree in mechanical engineering from Technische Universität München, Munich, Germany. He is currently working toward the Doctoral degree at the Institute of Automatic Control, Leibniz Universität Hannover, Hanover, Germany. He wrote a master thesis about the presented algorithm in collaboration with the BMW Group. His research interests include safe human–robot interaction, motion planning, and automatic control.

Daniel Althoff is currently a Research Engineer with the BMW Group, Munich, Germany. He received the diploma engineer degree in electrical engineering in 2008, and the Dr.Ing. degree from Technische Universität München, Munich, Germany, in 2013. From 2013 to 2015, he was a Postdoctoral Fellow with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include motion planning and safety assessment of autonomous systems.

Christoph Stiller (S'93–M'95–SM'99) received the Electrical Engineering degree from Aachen, Germany, and Trondheim, Norway, and the Diploma and Dr. Ing. degrees from Aachen University of Technology, Aachen, Germany, 1988 and 1994, respectively. In 1994, he was a Postdoctoral Scientist with the INRS Telecommunications, Montreal, QC, Canada. In 1995, he joined the Corporate Research and Advanced Development of Robert Bosch GmbH, Hildesheim, Germany. In 2001, he became a Full Professor and the Director of the Institute for Measurement and Control Systems with Karlsruhe Institute of Technology, Karlsruhe, Germany. In 2010, he was appointed as a Distinguished Visiting Scientist for three months at CSIRO, Brisbane, QLD, Australia. In 2015, he was a Guest Scientist for five months with the Bosch RTC and Stanford University, Palo Alto, CA, USA. He served as a President (2012–2013) and several other positions for the IEEE Intelligent Transportation Systems (ITS) Society. He has been an Associate Editor for several IEEE Transactions since 1999, Senior Editor for the IEEE TRANSACTIONS ON INTELLIGENT VEHICLES (since 2016), and Editor-in-Chief for the IEEE ITS MAGAZINE (2009–2011). His Autonomous Vehicle AnnieWAY was a finalist in the Urban Challenge 2007 and the winner and second winner of the Grand Cooperative Driving Challenge 2011 and 2016, respectively. In 2013, he collaborated with Daimler on the automated Bertha Benz Memorial Tour.