2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Decision Making for Autonomous Driving considering Interaction and Uncertain Prediction of Surrounding Vehicles

Constantin Hubmann[1], Marvin Becker[2], Daniel Althoff[3], David Lenz[4] and Christoph Stiller[5]

***Abstract*— Autonomous 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 intention 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 surroundings allows the autonomous car to incorporate the estimated change in future prediction accuracy in the optimal policy. A compact representation allows a low-dimensional state-space so that the problem can be solved online for varying road layouts and number of other vehicles. This is done with a point-based solver in an anytime fashion on a continuous state-space. We show the results with simulations for the crossing of complex (unsignalized) intersections. Our approach performs nearly as good as with full prior information about the intentions of the other vehicles and clearly outperforms reactive approaches.**

I. Introduction

Recent efforts and developments in the area of Advanced Driver Assistance System (ADAS) show considerable improvements towards the availability of autonomous 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 situations like 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 and create solutions for individual transport in cities. To enable the autonomous car to navigate safely and reliable in such complex real-world environments, various technical challenges from autonomous systems and robotics need to be taken into account.

[1,2,3] are with BMW Group, D-85748 Garching, Germany. (constantin.hubmann@bmw.de, marvin.becker@bmw.de, daniel.althoff@bmw.de)

[4]is with fortiss GmbH, An-Institut Technische Universität München (TUM), Munich, Germany. (david.lenz@fortiss.org)

[5]is with the Institute of Control, Karlsruhe Institut für Technologie (KIT), Karlsruhe, Germany. (stiller@kit.edu)

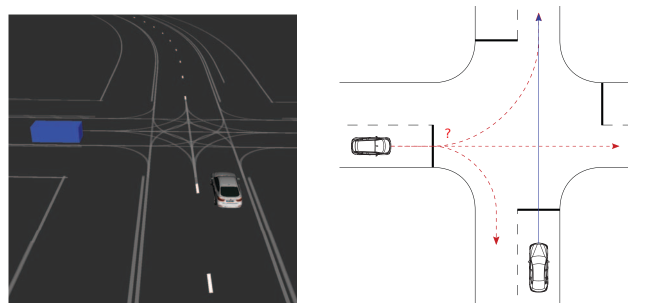

Fig. 1: A typical situation in which the ego vehicle has to decide for an action under uncertainty of the future behavior of the other car (blue box).

Firstly, the perception of the robotic system is uncertain due to the 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 must be predicted. The particular challenge of the prediction is caused by the uncertain information of their current states including hidden states about their goal destination. Thirdly, the motion of the ego vehicle must be collision-free, meet its kinematic and dynamic contraints and follow the traffic rules.

To tackle the complexity of such a robotic system, 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.

Typical scenarios are traversing an unsignalized intersection (see Fig. 1 for an example) or merging on a lane through oncoming traffic at T-Junctions (as evaluated in Sec. V).

The main contribution of this work is the presentation of an online *Partially Observable Markov Decision Process* (POMDP) for autonomous driving. It provides near optimum solutions for the behavior generation on intersections with an arbitrary layout and a variable number of traffic participants with unknown maneuver intentions. Concerning the uncertain prediction of the other road users, this work considers two very interesting aspects. First of all, 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 predicted behavior of the other traffic participants becomes more precise at a certain/estimated point in time and incorporates this in the decision making.

Secondly, a simple interaction model between the autonomous car and the other traffic participants allows for solutions with interactive behavior. 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 Werling *et al.* [4]) resulting in a smoother motion.

The performance of the proposed approach is evaluated in simulation scenarios showing comparable results to scenarios with perfect information about the intentions of the surrounding drivers.

II. Related Work

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 Kammel *et al.* [5] or Montemerlo *et al.* [6], focused on *rule-based systems* like Finite/Hybrid State Machines, that encode the desired behavior of the vehicle in the scenarios encountered. 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 scenario space. 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: Hülsen *et al.* [7] present an ontology that is based on Description Logic to model complex intersection to perform inference on traffic rules and right-of-way. Similarly, Zhao *et al.* [8] uses a combination of a map, control and car ontologies to guide a path planner in unsignalized intersections or narrow roads. Kohlhaas *et al.* [9] 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. Kohlhaas *et al.* [10] extended their planning framework to handle generic intersections and dynamic constraints. Chen *et al.* [11] combine a symbolic task-planning approach that provides subgoals and a kinodynamic trajectory planner to plan rule-compliant and dynamical feasible trajectories. The task and predicates are modeled as logic statements.

All of these knowledge-based approaches have in common, that they assume full knowledge over the states and intentions of the other vehicles and do not regard the interaction between all traffic participants. Uncertainty is only considered by replanning on new sensor information. The same is true for cooperative planning algorithms that mostly rely on V2V communication. For a good overview on this topic, please refer to Chen and Englund [12].

A *planning based approach* for the behavioral layer is presented by Hubmann *et al.* [13]. Although interactions between the autonomous car and the predicted motion of other traffic participants are not considered, it provides good results assuming constant velocity of other agents due its reactive and fast re-planning behavior. Nonetheless, in cases where various different maneuvers of other traffic participants may be assumed, a worst-case assumption of all possible maneuvers may lead to suboptimal behavior or even to standstill, also known as the *frozen robot problem* demonstrated by Trautman and Krause [14].

A method which allows to retrieve an optimal action, given the uncertainty of future behavior, and realize interactive behavior, is a POMDP. Several publications pursue this direction and consider different forms of uncertainty in the planning model. Methods that compute a policy *offline* for a POMDP are powerful and are able to solve very complex situations: Bai *et al.* [15] present an algorithm for solving continuous-state and continuous-observation POMDPs and show simulation results for an intersection scenario. In this example, a simplified intersection model is used and only sensor noise has been considered to generate a policy offline. Brechtel *et al.* [16] show how observation uncertainty and occlusion in intersection scenarios 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' position by considering the exploration capabilities of the robot. Sezer *et al.* [17] solve a T-Crossing with unknown intentions of other vehicles in a discretized state space model and a simple behavioral model as Mixed Observability Markov Decision Process (MOMDP). They show promising results for very natural merging behavior.

The drawback of all those offline methods is, that they are tailored for specific scenarios and the policy cannot be precalculated for all possible scenarios and vehicle configurations.

Online methods on the other hand need to make a trade-off between the state space size, the solution quality and the planning horizon: Ulbrich and Maurer [18] propose a POMDP planning framework for lane change maneuvers with noisy observations. They model the state space on high level features – whether a lane change is beneficial and possible on either side – thus making the problem tractable and online-solvable. Due to the simplified state space the method cannot be easily transferred to intersection planning. An online solver similar to the scenario by Sezer *et al.* [17] is presented in Bai *et al.* [19]: the state space in the intersection area is discretized and the POMDP is solved on this discrete state space. The solution quality and runtime depends strongly on the selected grid. Song *et al.* [20] regard the behaviors of other traffic participants as Hidden Markov Models (HMMs), but the intention estimation is assumed deterministic during planning. Therefore, no uncertainty is considered during planning and the vehicle will not perform information gathering actions. Bai *et al.* [21] extended their prior offline approach based on the DESPOT solver from Somani *et al.* [22] to include the (unknown/hidden) 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 clear. Liu *et al.* [23] on the other hand showed the advantages over a reactive planner at a T-intersection and a roundabout in terms of failure rate and time to traverse the intersection/roundabout in simulated scenarios.

In contrast to the related work presented in this section,

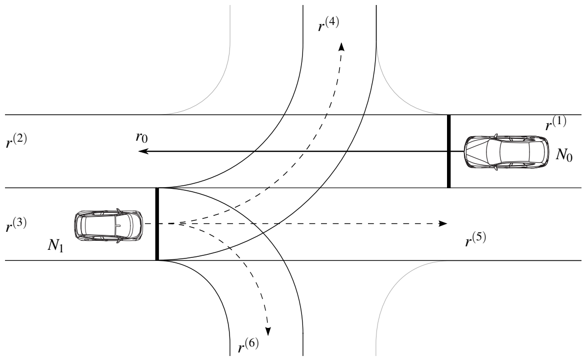

Fig. 2: A typical urban intersection with the autonomous car N_0 driving on path r_0 and one oncoming vehicle N_1 which may intersect with r_0.

we aim to propose a POMDP unified planning framework that combines multiple aspects:

- it is generally applicable for 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 continuous decision making for the ego vehicle, i.e. the generation of a sequence of desired accelerations $a_0 = (a(t=0), a(t=1), a(t=2), \cdots)$, 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 or given by the road geometry of a given map. In a second step, the longitudinal velocity is planned along r_0. This practice is referred to as *path-velocity decomposition* in the literature [24] and reduces the problem to a trajectory planning problem in an only one dimensional workspace.

The environment is populated by a set of agents $\mathscr{N} = \{N_0, \cdots, N_K\}$, 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 $\mathscr{R} = \{r^{(0)}, r^{(1)}, \cdots, r^{(I)}\}$, 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. $r_k : N_k \mapsto r \in \mathscr{R}$ 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 $M^{(i)}$, with $M^{(i)} = succ(r^{(i)})$. An agent traverses from its current to its next path, r'_k, with the unknown probability $P(r'_k | r_k) = \mathscr{G}(r'_k | r_k)$.

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

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

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:

$$a^* := \underset{a(t=0)}{\arg\max}\ \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | a(t=0)\right]. \tag{2}$$

This reward must be inverse proportional to the intersection crossing time, total acceleration (i.e. comfort) and deviation of velocity to a curvature and traffic law-based reference velocity under consideration of limited possible acceleration and collision avoidance.

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 (Sec. II). By preprocessing the map layout, the routes of the vehicles can be encoded in only one dimension of the state-space (see Sec. IV-B). This allows a low-dimensional situation representation with the other vehicles' routes as partially observable variables. Therefore, the problem description from Sec. 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 [25]), 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 vehicle for a set of future scenarios.

Because the utilized Adaptive Belief Tree (ABT) algorithm samples multiple particles 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 firstly presented in the longitudinal planning approach in the authors' previous work [13], 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 $\langle \mathscr{X}, A, T, O, Z, R, b_0, \gamma \rangle$ with the possible states $\mathscr{x} \in \mathscr{X}$ and possible actions $a \in A$ to be executed by the agent. $T(\mathscr{x}', \mathscr{x}, a) = P(\mathscr{x}'|\mathscr{x}, a)$ is the transition probability of ending in state $\mathscr{x}'$ when executing action a in state $\mathscr{x}$. $R(a, \mathscr{x})$ is the reward for selecting action a in state $\mathscr{x}$. 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 $\mathscr{x}$ 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 $Z(\mathscr{x}', a, o) = P(o|\mathscr{x}', a)$ provides the probability to observe a certain observation o after taking action a and ending in the new state $\mathscr{x}'$. Instead of generating a policy $\pi : \mathscr{x} \mapsto a$, 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

$$\pi^* := \arg\max_{\pi} \left(\mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\mathscr{x}_t, \pi(b_t)) | b_0, \pi \right] \right). \quad (3)$$

B. Statespace

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

$$\mathscr{x} = (\mathscr{x}_0, \mathscr{x}_1, \mathscr{x}_2, \cdots, \mathscr{x}_K)^T. \quad (4)$$

$\mathscr{x}_0$ represents the state of the autonomous car and $\mathscr{x}_k \in \{1, \cdots, K\}$ 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 W to the global coordinate system L, ${}^W T_L$, is not bijective, s_k may still be calculated from global coordinates as the particle's route r_k is constant as well as part of the state space.

The autonomous car's state is then defined as

$$\mathscr{x}_0 = \begin{pmatrix} s_0 \\ v_0 \end{pmatrix} \quad (5)$$

and the state of the other vehicles is

$$\mathscr{x}_k = \begin{pmatrix} s_k \\ v_k \\ r_k \end{pmatrix} \quad (6)$$

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.

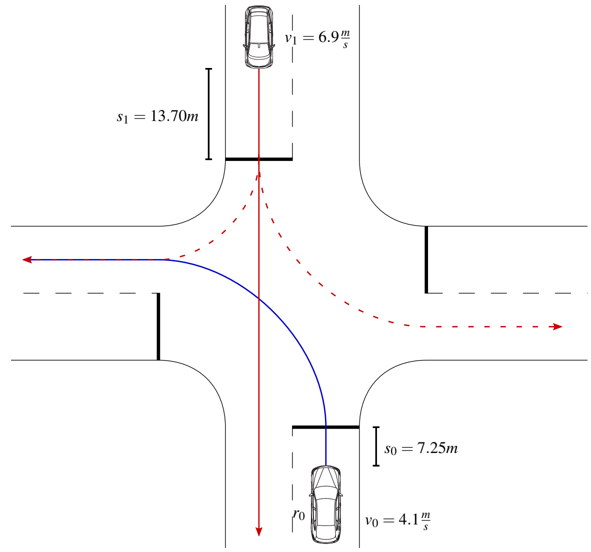

Fig. 3: Visualisation of the various variables of the autonomous and other car's statespace.

C. Actions and Motion Model

The transition model of the other vehicles is defined for discrete time with an interval length of Δt as

$$\begin{pmatrix} s_k' \\ v_k' \\ r_k' \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_k \\ v_k \\ r_k \end{pmatrix} + \begin{pmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \\ 0 \end{pmatrix} a_k, k \in \{1, \cdots, K\}. \quad (7)$$

As mentioned in Sec. 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}:

$$a_{int,k} = \begin{cases} 0, & \text{iff } c(r_k, r_0) = 0, \\ -0.5, & \text{iff } c(r_k, r_0) = 1 \wedge (t_{c,k} - t_{c,0}) \in [0,3]. \end{cases} \quad (8)$$

t_c is the time until the arrival at the conflicting point of the two routes, assuming constant velocity v_k. The acceleration is also constrained by a maximum acceleration a_{max}. The reference velocity is based on a maximum-lateral-acceleration based approach (see [13] for details). As the solver is particle-based, the generated acceleration of the motion model is additionally perturbated by simulated noise (σ^2) to represent various individual driving styles in longitudinal direction. The resulting total acceleration may be written formally as follows:

$$a_k = \min(a_{ref,k} + a_{int,k}, a_{max}) + \mathscr{N}(0, \sigma^2) \quad (9)$$

The transition model of the ego vehicle is defined in the same way as for the other vehicles in (7), 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 $R(\mathscr{X},a)$ is defined as follows:

$$R(\mathscr{X},a) = R_{\text{term}}(\mathscr{X}) + R_{\text{crash}}(\mathscr{X}) + R_v(\mathscr{X}) + R_{\text{acc}}(a). \quad (10)$$

$R_{term}(\mathscr{X})$ is a high positive reward ($R_{\text{term}} = 9000$) when reaching the terminal state (i.e. any position behind the intersection). As the reward is discounted, this reward motivates the vehicle to pass the intersection as fast as possible. Additionally, there are various rewards during exploration: a collision is punished with high negative rewards ($R_{\text{crash}} = -9000$), a deviation to the reference velocity (i.e. a smooth velocity on a road without vehicles [13]) has quadratic negative rewards in the case of a too high velocity (allowed for small deviations, s.t. $R_v = -K_v \cdot (v_0 - v_{ref})^2$ with $K_v = 20$, if $(v_{ref}+2) > v_0 > v_{ref}$ and $K_v = 200$, if $v_0 >= v_{ref}+2$) and linear negative rewards in the case of a too low velocity (allowed but not desired, s.t. $R_v = -25 \cdot |v_0 - v_{ref}|$)). Changing the acceleration has costs $R_{\text{acc}} = -100$ to provide comfort.

E. Observation Space

An observation $o \in O$ is defined as

$$\mathbf{o} = (o_0, o_1, \cdots, o_K)^T \quad (11)$$

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

As perception noise is not considered in this approach and the route of the autonomous car is known, its state is fully observable and may be denoted as

$$o_0 = \begin{pmatrix} s_0 \\ v_0 \end{pmatrix}. \quad (12)$$

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

$$o_k = \begin{pmatrix} v_k \\ x_k \\ y_k \end{pmatrix}. \quad (13)$$

F. Observation Model and Prediction

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

Although the route r_k of another vehicle N_k is unobservable, it is part of its state $\mathscr{X}_k$ 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 uncertainty concerning the chosen, unknown route of another vehicle in future time steps, a simple discriminative classifier is employed. We propose a Naive

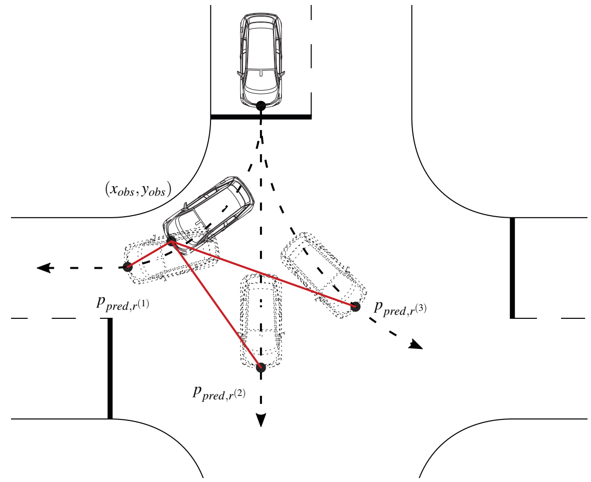

Fig. 4: Demonstration of the distance feature $f_{k,2}$. It is defined as the euclidean distance between the simulated observation's position (x_{obs}, y_{obs}) and the assumed euclidean position $p_{pred,r^{(i)}}$ given a certain route hypothesis $r^{(i)}$.

Bayes classifier with a 2-dimensional feature vector $\mathbf{f}_k$ (velocity and position based, see Fig. 4) for vehicle N_k, that can be generated from the observation space:

$$\mathbf{f}_k = \begin{pmatrix} f_{k,1} \\ f_{k,2} \end{pmatrix} = \begin{pmatrix} |v'_k - v_{ref,r^{(i)}}(s'_k)| \\ \left\| [x'_k \; y'_k]^T - [x_{k,pred,r^{(i)}} \; y_{k,pred,r^{(i)}}]^T \right\|_2 \end{pmatrix}. \quad (14)$$

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

$$P(r_k = r^{(i)}|f_{k,1}, f_{k,2}) = \frac{P(r^{(i)})P(f_{k,1}, f_{k,2}|r^{(i)})}{P(f_{k,1}, f_{k,2})}. \quad (15)$$

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, (15) may be rewritten to:

$$P(r_k = r^{(i)}|f_{k,1}, f_{k,2}) = \frac{P(f_{k,1}|r^{(i)})P(f_{k,2}|r^{(i)})}{\sum_{l=1}^{I} P(f_{k,1}|r^{(l)})P(f_{k,2}|r^{(l)})}. \quad (16)$$

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 route $r^{(i)}$ that is sampled from (16).

G. TAPIR

We use the TAPIR Toolkit [25] for solving the problem online. TAPIR is a point based solver which finds an approximate solution by sampling episodes to construct the belief tree. It is also capable of adapting the belief tree without resampling all epsiodes, when changes in the environment happen.

V. Simulation Results

In the following section, the algorithm is evaluated in a merge scenario on a T-Junction as well as on a more complex

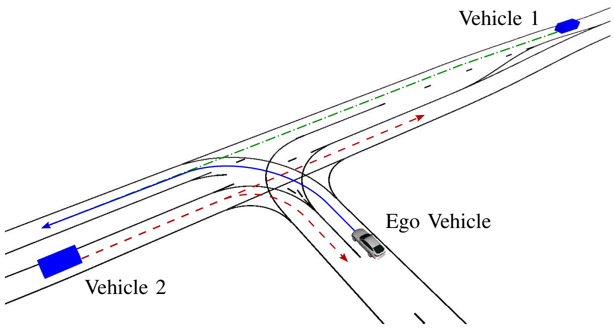

Fig. 5: Top view of the T-junction scenario.

intersection with multiple vehicles. Also, the runtime of the algorithm is examined regarding planning horizon and number of other vehicles. We use a proprietary simulator for the evaluation of the two scenarios. The other vehicles are controlled by the behavior generation module of the simulator. It is an advanced, reactive and rule-based strategy.

A. *Evaluation Settings*

The algorithm was tested with the following parameters: The planning step size Δt is chosen to be 1.0 seconds. The belief tree is searched with 11 000 particles. The algorithm runs with a replanning frequency of 1 Hz. The system (containing the simulation environment and the algorithm) runs on a Intel Core i7-4910MQ CPU with 2.9 GHz.

B. *Merging on a T-Junction*

At first, a merging scenario is presented (see Fig. 5). The ego vehicle approaches a T-Junction at which a left turn is planned. 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 motion model. This uncertainty is incorporated by adding Gaussian noise on the interactive motion model (see (9)). 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 chance 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. Therefore, it is evaluated in the following.

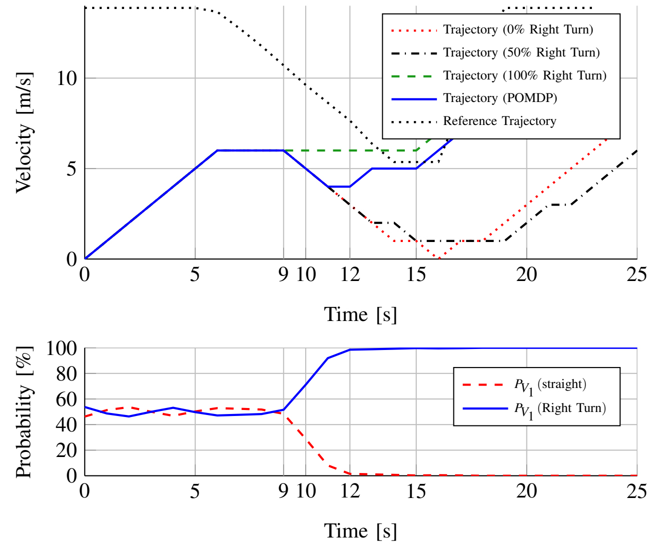

Fig. 6: Evaluation of the T-junction scenario where vehicle 2 is turning right. The upper figure shows the planned velocities over time when the POMDP planning is executed with different fixed probabilities and with the probability from prediction and POMDP sampling. The lower figure shows the probability for a conflict area with vehicle 2 over time in the evaluation without fixed probabilities.

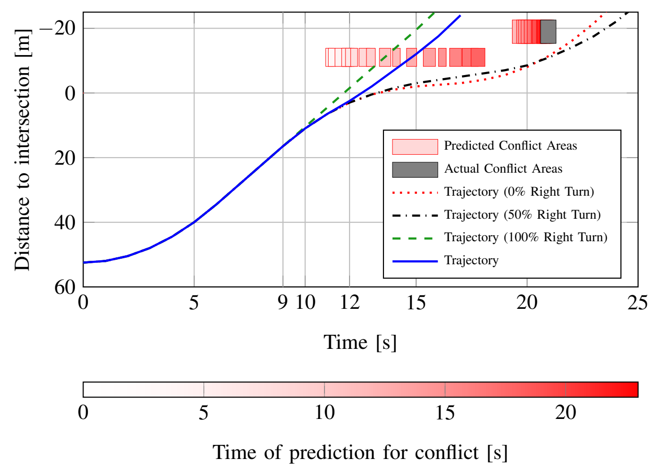

Fig. 7: Evaluation of the T-junction scenario with a right turning vehicle. The figure shows the planned positions over time when the POMDP planning is executed with different fixed probabilities and with the probability from prediction and POMDP sampling.

Fig. 6 shows the driven trajectory of the ego vehicle for different cases as well as the desired reference velocity in the upper figure. The lower Fig. 6 shows the estimated probability for a each maneuver of vehicle 2 over time. It can be seen that the ego vehicle accelerates up to the desired curve velocity (defined in [13]). After 9 seconds, the predicted behavior is still 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. The two possible options, yielding to vehicle 1 or merging immediately, are that way kept open for the ego vehicle. This behavior (also known as *information gathering*) is chosen as the ego vehicle knows,

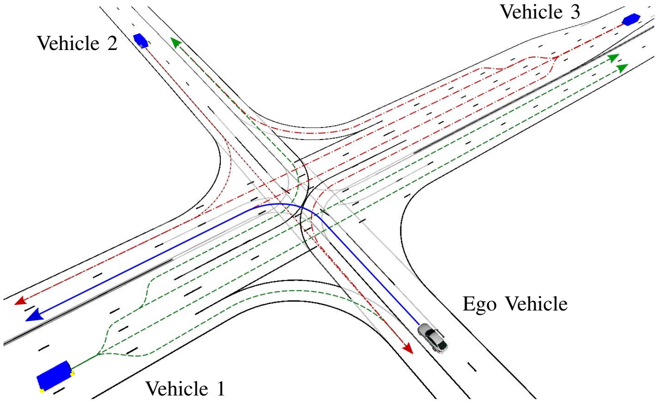

Fig. 8: Top view of the complex intersection scenario with three other vehicles.

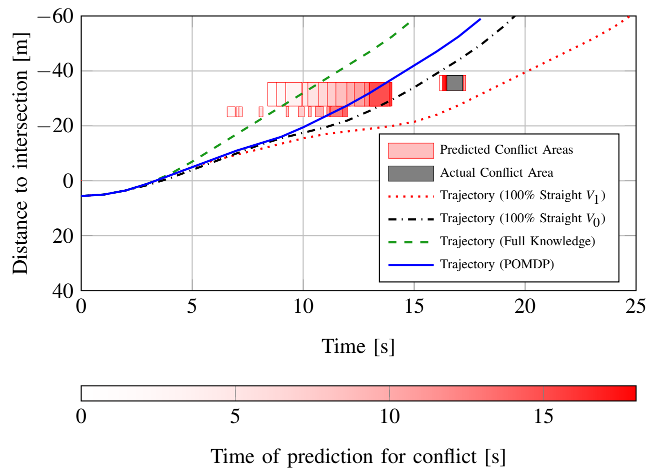

Fig. 9: Evaluation of the complex intersection scenario. The figure shows the planned positions over time for four different evaluations as well as the predicted and actual conflict areas. The lowest rectangles represent the conflict area with vehicle 1, the rectangles in the middle the conflict area with vehicle 2 and the upper rectangles the conflict area with vehicle 3.

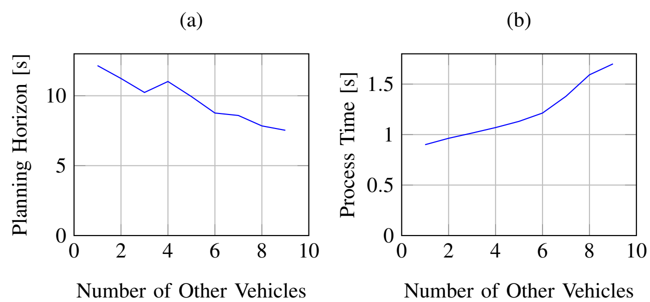

Fig. 10: Evaluation of the performance of the POMDP planner. (a) shows the average planning horizon with respect to the number of other vehicles. (b) shows the average processing time over the number of vehicles.

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 now precise prediction, combined with the earlier tactical behavior, allows the ego vehicle to still cut in before vehicle 1.

For the same scenario, Fig. 7 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.

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 2) as with exact information of the other vehicles' behavior (i.e. V2V communication, resulting in 100 % right turn prediction accuracy). Approaches that do 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. 6 in a very defensive, suboptimal driving trajectory, which does not allow the vehicle to merge in front of the other vehicles.

C. Left turn on a complex, intersection

As mentioned in the beginning, one focus of the approach is to allow to handle various different intersections online. For demonstration purposes, a second, more complex scenario is introduced to demonstrate the generic applicability. Fig. 8 shows a large, unsignalized intersection, with in total 10 different possible routes for the other three vehicles. Our algorithm is able to also solve this complex intersection very similar compared to no uncertainty in the prediction (e.g. use of V2V) as shown in Fig. 9.

D. Quantitative Performance measures

The performance of the algorithm is evaluated concerning the number of considered other vehicles. Therefore, a 4-way intersection is considered with three options for every lane (driving straight, turning left/right) and a varying number of 1 up to 9 vehicles. In Fig. 10a, the planning horizon is plotted over the number of vehicles. More vehicles result in a higher branching factor to represent more potential future scenarios. Therefore it is not solvable in the same time. As the ABT algorithm is anytime, the planning horizon t_{hor} reduces from $t_{hor} = 12\,\mathrm{s}$ (for one vehicle) to $t_{hor} = 7.5\,\mathrm{s}$ (for 9 vehicles) to solve these more complex scenarios.

As shown in Fig. 10a, the solver is able to handle scenarios with multiple vehicles. Despite the anytime behavior of the algorithm, the generation of new particles is not anytime but grows linearly with the number of particles. This is shown in Fig. 10b. The anytime condition for the policy improvement is normally set to 800 ms as the generation of new particles takes up to 200 ms for up to three traffic participants. As can be seen in Fig. 10b, the process time goes up to 1.7s for nine vehicles. The reason for this is, that much more particles have to be generated which takes up to 900 ms.

Therefore, for using the algorithm with more than three vehicles, a more efficient particle generation algorithm has to be implemented.

VI. Conclusion

The main contribution of this work is the presentation of an online capable POMDP framework for autonomous driving under various environment situations. The modeling results in a behavior, which explicitly includes the prediction uncertainty in terms of maneuver uncertainty and longitudinal uncertainty during the performed maneuver. Additionally, not only the current prediction uncertainty is considered but also in what way the prediction accuracy will change in the future. This modeling allows the ego vehicle to optimize its behavior for different future scenarios, to simulate the effect of actions on the movement of other vehicles and to postpone 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.

Additionally, our approach can be easily extended to represent observation uncertainty and to include traffic participants with a higher prediction uncertainty such as pedestrians. However, we expect that more particles are needed by the point based solver to approximate the growing dimensionality of the belief space.

The authors intend to pursue the approach. Possible directions of further research are to increase the planning frequency by adapting the solver, extend the approach to lateral planning and to integrate superior motion and interaction models.

References

[1] SAE International, "Taxonomy and Definitions for Terms Related to On-Road 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 *IEEE Intelligent Vehicles Symposium*, 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, May 2010, pp. 987–993.

[5] S. Kammel, J. Ziegler, B. Pitzer, *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[6] M. Montemerlo, J. Becker, S. Bhat, *et al.*, "Junior: the stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[7] M. Hülsen, J. M. Zöllner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 993–999.

[8] L. Zhao, R. Ichise, T. Yoshikawa, *et al.*, "Ontology-based decision making on uncontrolled intersections and narrow roads," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 83–88.

[9] 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 *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1060–1065.

[10] R. Kohlhaas, D. Hammann, T. Schamm, and J. M. Zollner, "Planning of High-Level Maneuver Sequences on Semantic State Spaces," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 2090–2096.

[11] C. Chen, M. Rickert, and A. Knoll, "Combining Task and Motion Planning for Intersection Assistance Systems," in *IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1242–1247.

[12] L. Chen and C. Englund, "Cooperative Intersection Management: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[13] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *IEEE International Conference on Intelligent Transportation Systems*, 2016.

[14] P. Trautman and A. Krause, "Unfreezing the robot: navigation in dense, interacting crowds," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2010, pp. 797–803.

[15] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1288–1302, Jun. 2014.

[16] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic Decision-Making under Uncertainty for Autonomous Driving using Continuous POMDPs," in *IEEE International Conference on Intelligent Transport Systems (ITSC)*, 2014.

[17] V. Sezer, T. B. Member, D. Rus, *et al.*, "Towards Autonomous Navigation of Unsignalized Intersections under Uncertainty of Human Driver Intent," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3578–3585.

[18] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *IEEE International Conference on Intelligent Transport Systems (ITSC)*, 2013, pp. 2063–2067.

[19] Y. Bai, Z. J. Chong, M. H. Ang, and X. Gao, "An Online Approach for Intersection Navigation of Autonomous Vehicle," in *IEEE International Conference on Robotics and Biomimetics*, IEEE, 2014, pp. 2127–2132.

[20] W. Song, G. Xiong, and H. Chen, "Intention-Aware Autonomous Driving Decision-Making in an Uncontrolled Intersection," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–15, 2016.

[21] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *IEEE International Conference on Robotics and Automation*, 2015.

[22] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT : Online POMDP Planning with Regularization," *Advances in Neural Information Processing Systems*, pp. 1–9, 2013.

[23] 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 *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1126–1133.

[24] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[25] D. Klimenko and J. S. and. H. Kurniawati, "Tapir: a software toolkit for approximating and adapting pomdp solutions online," in *Proc. Australasian Conference on Robotics and Automation*, 2014.