2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving

Bingyu Zhou*, Wilko Schwarting†, Daniela Rus† and Javier Alonso-Mora*

Abstract— **When driving in urban environments, an autonomous vehicle must account for the interaction with other traffic participants. It must reason about their future behavior, how its actions affect their future behavior, and potentially consider multiple motion hypothesis. In this paper we introduce a method for joint behavior estimation and trajectory planning that models interaction and multi-policy decision-making. The method leverages Partially Observable Markov Decision Processes to estimate the behavior of other traffic participants given the planned trajectory for the ego-vehicle, and Receding-Horizon Control for generating safe trajectories for the ego-vehicle. To achieve safe navigation we introduce chance constraints over multiple motion policies in the receding-horizon planner. These constraints account for uncertainty over the behavior of other traffic participants. The method is capable of running in real-time and we show its performance and good scalability in simulated multi-vehicle intersection scenarios.**

I. Introduction

Motion planning and decision-making are critical for safe navigation of intelligent vehicles in urban environments. Challenges lie in the uncertain motion of other traffic participants and the difficulty of predicting and accounting for their intentions and interactions with the intelligent vehicle. One of the first examples of this problem arose during the DARPA Urban Challenge [1], where a collision between MIT and Cornell's vehicles [2] was caused by the failure to anticipate the other vehicle's future intentions and interaction. In complex urban scenarios several research questions arise. In this work we focus on the following two, a) how to model the interaction between the ego-vehicle and other traffic participants, and b) how to incorporate the uncertainty of motion intentions into the motion planning framework.

In this paper we propose a joint approach for behavior prediction and planning. The approach leverages the strength of online Partially Observable Markov Decision Processes (POMDP) for behavior prediction and nonlinear receding horizon control, or Model Predictive Control (MPC), for trajectory planning. In particular, we utilize the POMDP framework to model the interactions between the ego-vehicle and the obstacles. The estimator predicts the future speed actions of the obstacle vehicles under multiple motion intentions in the form of policies, while taking into account the planned trajectory of the ego-vehicle. Then, we formulate a nonlinear optimization problem to generate a safe trajectory for the ego-vehicle. The planned trajectory takes into account the predicted behaviors of the obstacle vehicles and the uncertainty over multiple motion hypotheses represented as chance constraints in the receding horizon framework.

The main contributions of this work are:

- A joint behavior estimation and trajectory planning method, which utilizes the strengths of model predictive control (MPC) and online POMDPs to achieve intention-aware navigation for autonomous vehicles in uncertain urban environments.
- A chance constrained formulation of receding horizon control that accounts for the uncertainty in the motion intentions of other traffic participants, over multiple motion policies.

Finally, we evaluate our approach in a typical intersection scenario with multiple traffic participants.

II. Related Work

Motion planning approaches for intelligent vehicles, such as Rapidly Exploring Random Trees (RRT) [3] or Receding Horizon Control [4] typically rely on a prediction of the motion of other traffic participants and do not account for the interaction and cooperation in planning among them. Imagine for instance the case of merging into a high-density lane of slow moving traffic. If cooperation is not modeled, it may occur that all forward paths are deemed unsafe by the planner and the vehicle freezes in place, unable to make progress. Yet, a human driver is typically able to pull in by cooperating with other traffic participants. One way to model interaction is by assuming that all robots employ the same algorithm and that they share the avoidance effort, as in the collision avoidance method of [5], which is over simplified for intelligent vehicles.

Several other methods have been proposed to model the interaction between robots navigating in dense crowds. Interactive Gaussian processes (IGP) [6], [7] model the trajectories of each agent as a sample from a Gaussian process and couple the trajectories through an interaction potential function, but do not scale for real-time applications. Online performance can be achieved with maximum entropy inverse reinforcement learning (IRL) [8], [9], where a set of discrete features are employed and their weights are learned by constructing the appropriate reward functions. IRL is also employed in [10] by anticipating that the other vehicles will respond to the ego-vehicle's action. This assumption could be too optimistic for some uncertain obstacle behaviors.

* Bingyu Zhou and Javier Alonso-Mora are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD, Delft, The Netherlands bryan-by@hotmail.com; j.alonsomora@tudelft.nl

† Wilko Schwarting and Daniela Rus are with the Massachusetts Institute of Technology, 32 Vassar St, Cambridge MA, USA wilkos@mit.edu; rus@mit.edu

This work is supported by NWO Veni grant 15916. Accompanying video: https://youtu.be/nC32GaWHJLI

Since the driving road is usually well defined, the motion planner can take advantage of the structure of the road. If a discrete set of actions is considered for each vehicle, Partially Observable Markov Decision Processes (POMDP) can be employed for cooperative motion planning and decision-making [11], [12]. Both works only schedule the speed action of the robot, which limits their practical application to real vehicles. To reduce the computational complexity of POMDPs, [13] assume that all vehicles' behaviors can be selected from a finite set of policies, such as lane-nominal, lane-change-left and lane-change-right and forward simulate the most likely one. In contrast, we do not limit the behavior of the ego-vehicle to a single motion policy and simultaneously take into account multiple motion policies for the obstacle vehicles. Our planner conserves safety by considering all motion hypothesis for other traffic participants, and models the interaction between vehicles in the behavior estimation.

We rely on the state-of-the-art POMDP solver DESPOT [14] to compute predictions over the motion policies of other vehicles and extend the receding horizon planner of [4] to account for multiple policies and their uncertainty in the planning process for an autonomous vehicle.

III. Problem Formulation

Let A denote the set of vehicles interacting in a traffic scenario, including the ego-vehicle A_0 (this is the vehicle that is controlled with our method), and n obstacle vehicles. The state $\mathbf{z}^0$ of the ego-vehicle and $\mathbf{z}^i$ of obstacle vehicle i are

$$\mathbf{z}^0 = \{x, y, \theta, \delta, v\}, \text{ and } \mathbf{z}^i = \{x, y, \theta, v, g\}, \tag{1}$$

where x, y, θ are the position and heading angle of the vehicle, δ is the steering angle of the ego-vehicle, v is the speed of the vehicle and g is the true motion intention of the obstacle vehicle. For example, in the T-junction scenario shown in Fig. 1, the motion intention g can be defined as a binary value in which 0 represents going straight and 1 turning. We denote by $\mathbf{z} = \{\mathbf{z}^0, \mathbf{z}^1, \ldots, \mathbf{z}^n\}$ the joint state and we define the position of a vehicle as $\mathbf{s} = \{x, y\}$. The controlled actions $\mathbf{u} = \{u^\delta, u^a\}$ of the ego-vehicle are the steering speed u^δ and the throttle u^a.

For a given obstacle vehicle, we consider a set of l motion intentions $\mathcal{G} = \{g_1, \ldots, g_l\}$. For motion intention $g_j \in \mathcal{G}$, the trajectory of vehicle i for a time horizon $[0, m]$ is

$$\pi_i(g_j) = \{\mathbf{s}_0^i, \mathbf{s}_1^i, ..., \mathbf{s}_m^i\},$$

from which speed v_k and orientation θ_k can be extracted.

For vehicle $i \in \{1, \ldots, n\}$ we define the set of all possible motion policies (under all motion intentions) by

$$\Pi^i = \{\pi_i(g_1), \ldots, \pi_i(g_l)\},$$

and their relative weights, or likelihood, by $\mathcal{P}^i = [p_1^i, \ldots, p_l^i]$. The behavior estimation accounts for the future motion of the ego-vehicle via its planned trajectory π^0 since we combine the estimation of the policies Π^i and weights $\mathcal{P}^i$ for each obstacle vehicle with the following optimization.

To compute the optimal inputs $\mathbf{u}^*_{0:m-1}$ for the ego-vehicle, we formulate a chance-constrained nonlinear receding horizon optimization problem for n obstacle vehicles and l policies and of planning horizon m,

$$\begin{aligned}
\mathbf{u}^*_{0:m-1} = \min_{u_{0:m-1}} & \sum_{k=0}^{m-1} J(\mathbf{z}_k, \mathbf{u}_k) + J(\mathbf{z}_m) \\
s.t. \quad & \mathbf{z}^0_{k+1} = f(\mathbf{z}^0_k, u_k), && \forall k = \{0, \ldots, m-1\} \\
& \mathbf{z}^0_{min} < \mathbf{z}^0_k < \mathbf{z}^0_{max}, && \forall k = \{0, \ldots, m\} \\
& \mathbf{u}_{min} < \mathbf{u}_k < \mathbf{u}_{max}, && \forall k = \{0, \ldots, m-1\} \\
& \text{Estimate } \Pi^i \text{ and } \mathcal{P}^i \text{ given } \pi^0, && \forall i = \{1, \ldots, n\} \\
& \mathbf{s}^0_k \notin \mathbf{B}^i_k(\mathcal{P}^i, \Pi^i), && \forall i = \{1, \ldots, n\},
\end{aligned} \tag{2}$$

where $J(\mathbf{z}_k, \mathbf{u}_k)$ is the cost in each stage, which usually reflects the requirements of the motion planning problems, such as the comfort of driving and the quality of tracking a reference road, $J(\mathbf{z}_m)$ is the terminal cost, $f(\mathbf{z}^0_k, \mathbf{u}_k)$ is the state transition of the ego-vehicle, $\mathbf{z}^0_{min}$ and $\mathbf{z}^0_{max}$ are the boundary of the states, $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ are the boundary of the actions, and $\mathbf{s}^0_k \notin \mathbf{B}^i_k(\mathcal{P}^i, \Pi^i))$ is the probabilistic collision avoidance constraint for obstacle vehicle i. This constraint considers the estimated motion policies and is described in detail in Section IV-D, where the weighted sum of Gaussian distributions is utilized to compute uncertainty ellipses. For a representative example see Fig. 2.

We make the assumption that the obstacle vehicles follow, for each motion policy, a known reference path. The behavior

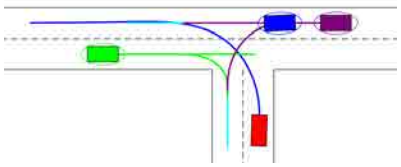


Fig. 1: Example of a typical intersection scenario. The red rectangle is the ego-vehicle. The other three rectangles are the obstacle vehicles. The associated curves for the obstacle vehicles indicate the multiple motion policies.

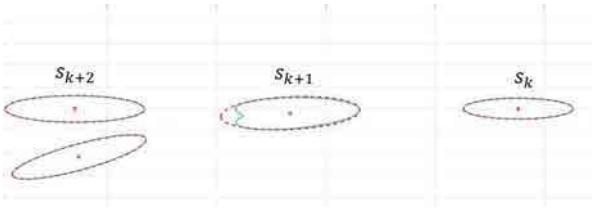


Fig. 2: Uncertainty ellipses for the purple vehicle of Fig.1 under two motion policies. $\mathbf{s}_k$, $\mathbf{s}_{k+1}$, $\mathbf{s}_{k+2}$ are the estimated multi-modal position distributions for a given safety level. If the distributions overlap ($\mathbf{s}_{k+1}$), the minimal area-enclosing ellipse (dash-red) is computed. Otherwise, the level set of each independent Gaussian distribution (solid cyan) is used.

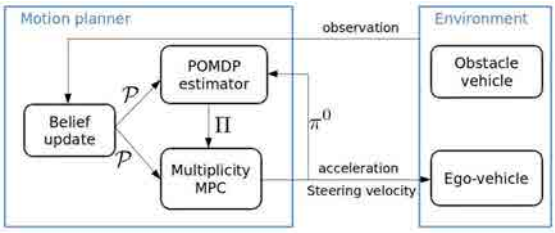


Fig. 3: Schema of the proposed method for joint behavior prediction and planning. $\mathcal{P} = [\mathcal{P}^1, \ldots, \mathcal{P}^n]$ are the belief weights of each obstacle vehicle and motion intention. $\Pi = [\Pi^1, \ldots, \Pi^n]$ are the estimated trajectories for each obstacle vehicle under all possible motion intentions. And π^0 is the planned trajectory for the ego-vehicle.

estimator predicts the speed profiles and likelihood of each policy. The boundaries of the road are also given.

IV. Joint Behavior Estimation & Planning

In this section we describe the proposed method.

A. Overview

An overview of the joint behavior estimation and planning algorithm is given in Fig. 3. The motion planner consists of three components: a POMDP estimator, a chance constrained MPC planner and a belief tracker.

For each motion intention and based on the current observations, the POMDP estimator predicts the optimal speed actions $a_{1:m}$ of each obstacle vehicle at the current belief state $b(\mathbf{z}_t)$. By assuming that the obstacle vehicles closely follow the reference path, the set of estimated trajectories Π_i is computed for each obstacle vehicle.

A receding horizon optimization is then employed to plan the future trajectory π^0 for the ego-vehicle. To guarantee a specified level of safety, it considers the estimated trajectories Π^i under all possible motion intentions, with their associated weights. Then the planned trajectory π^0 is be fed back to the POMDP estimator as the ego-vehicle's driving policy at next time step. Meanwhile, the resulting optimal action is applied by the ego-vehicle.

In Algorithm 1 we summarize the proposed joint behavior estimation and planning method, which is described in the following and consists of three parallel processes (estimation, belief update and planning).

B. Behavior Estimation

We formulate a POMDP to compute the optimal sequence of actions for each motion intention, which is represented as a policy tree Π^i [14] for the obstacle vehicle i. The POMDP prediction model consists of six components, including state space, action space, observation space, transition model and observation model.

1) State space $\mathbf{Z}$: As we only consider the speed actions of the obstacle vehicles, their state space can be reduced to their positions over time. The state of the ego-vehicle consists of its position and orientation. The motion intention $g_i \in \mathcal{G}$ indicates the high level decisions of the obstacle vehicles in a long term, such as going straight, turning left, turning right, stopping.

Algorithm 1 Joint behavior estimation and planning

Π_0^0 given by a constant velocity model and b_0 by a uniform distribution.
while the ego-vehicle does not reach the goal **do**
  // Estimator, Sec. IV-B (Process 1)
  Online POMDP estimator to predict the obstacle vehicles' policy during m time-steps ahead
  Π_k^i = DESPOT (Π_k^0, b_k), $\forall i = \{1, \ldots, n\}$
  // Belief update, Sec. IV-C (Process 2)
  Update the belief b_k from observation $\mathbf{o}_k$ and Eq. (5)
  // Receding-horizon planner, Sec. IV-E (Process 3)
  for $k \in [0, m]$ **do**
    Compute the level set of the multi-policy prediction for each obstacle vehicle and solve Eq. (8) to compute the collision constraints $\mathbf{E}_k^i(\mathcal{P}^i, \Pi^i)$
    Build up equality and inequality constraints Eq. (13)-Eq. (18)
  end for
  Solve Eq. (9) to compute the optimal actions $\mathbf{u}_{0,m-1}^*$ for the ego-vehicle and the planned trajectory Π_k^0
  Execute the first action $\mathbf{u}_0^*$ of Π_k^0 by the ego-vehicle
end while

2) Action space $\mathbf{A}$: Similar to [15] the action space is defined as

$$\mathbf{A} = [Acceleration, Deceleration, Maintain]$$

These discrete actions are sufficient to model the obstacle vehicle's reaction for the ego-vehicle's behavior.

3) Observation space $\mathbf{O}$: The observation space includes the information observed from the sensors. The position and orientation of each vehicle including the ego-vehicle are directly observable. But the motion intention g_i of the obstacle vehicle i is unobservable. The belief of the motion intentions needs to be inferred from the received observations over time.

4) Transition model $T(\mathbf{z}_{t+1}, a_t, \mathbf{z}_t)$: The transition model can be decomposed as,

$$p(\mathbf{z}_{t+1}|a_t, \mathbf{z}_t) = p(\mathbf{z}_{t+1}^0|\mathbf{z}_t^0) p(\mathbf{z}_{t+1}^i|a_t, \mathbf{z}_t^i) \tag{3}$$

where $p(\mathbf{z}_{t+1}^0|\mathbf{z}_t^0)$ is the state transition for the ego-vehicle, which is directly obtained from motion planner, $a_t \in \mathbf{A}$ is the action of the obstacle vehicle. $p(\mathbf{z}_{t+1}^i|a_t, \mathbf{z}_t^i)$ is the state transition for the obstacle vehicle, which can be interpreted for simplicity as a discrete differential driving model:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix} + \begin{bmatrix} (v_t + a\Delta t)\Delta t \cos(\theta_t) \\ (v_t + a\Delta t)\Delta t \sin(\theta_t) \\ \Delta\theta \\ a\Delta t \end{bmatrix} \tag{4}$$

where a is the acceleration applied on the vehicle. And $\Delta\theta$ is computed by closely tracking the reference road path. The reason why we choose the differential drive model instead

of the bicycle model for the obstacle vehicles is that this approximation of the motion can simplify the dynamics computation in the discrete POMDP model. Note that for the ego-vehicle we do consider a bicycle model in the planner.

5) Observation model $O(\mathbf{o}_t, \mathbf{z}_{t+1}, a_t)$: We assume that the poses of the ego-vehicle and the obstacle vehicle are directly observable. The main uncertainty lies on the motion intention g of the obstacle vehicle.

6) Reward $R(\mathbf{z}, a)$: The reward function is chosen based on driving experience and aims to let the obstacle vehicle take maximum progress on the road while avoiding the collision with others. The reward value $R(\mathbf{z}, a)$ is the immediate reward that the agent gets at a state $\mathbf{z} \in \mathbf{Z}$ after applying the action $a \in \mathbf{A}$:

$$R(\mathbf{z}, a) = R_p(\mathbf{z}) + R_c(\mathbf{z}, a) + R_a(a)$$

where

$$R_p = \beta e^{-\frac{d}{2\sigma^2}}$$

β is the scaling parameter to make the progress reward comparable with the other components of reward, d is the distance from the current vehicle position to the goal location, σ controls the variation speed of the progress reward.

$$R_c = \begin{cases} -r_c, & \text{if } \mathbf{s}^0 \in \mathbf{E} \\ 0, & \text{others} \end{cases}$$

where $r_c > 0$ is a tuned penalty for the collision, s^R is the position of the ego-vehicle, $\mathbf{E}$ is the uncertainty ellipses of obstacle vehicle A, which is introduced in IV-E.

$$R_a = \begin{cases} -r_a, & \text{if } a \in [Acceleration, Deceleration] \\ 0, & \text{others} \end{cases}$$

where $r_a > 0$ is a tuned parameter for action penalty.

C. Belief Update

A particle filter is implemented to update the belief of each motion intention g_i for each obstacle vehicle i. Since the set of motion intentions $\mathcal{G}$ is finite, the belief over all intentions forms a discrete probability distribution. The belief update can be represented by a function $b_{t+1} = \tau(b_t, a_t, \mathbf{o}_{t+1})$ based on the current belief b_t, the action a_t and the observation $\mathbf{o}_{t+1} \in \mathbf{O}$ and under the Markov property.

Based on Bayes' rule, in each iteration we observe that the obstacle vehicle moves from position s_t to s_{t+1}, and update the belief of its motion intentions by:

$$b_{t+1} = \eta O(\mathbf{o}_t, \mathbf{z}^i_{t+1}, a_t) \sum_{\mathbf{z}^i \in \mathbf{Z}} T(\mathbf{z}^i_{t+1}, a_t, \mathbf{z}^i_t) b_t \qquad (5)$$

where $\mathbf{z}^i_t$ is the state of the obstacle vehicle at time t, a_t is the action of the obstacle vehicle, $O(\mathbf{o}_t, \mathbf{z}^i_{t+1}, a_t)$ is the probability to observe $\mathbf{o}_t$ at the state $\mathbf{z}^i_{t+1}$ reached by action a_t, $T(\mathbf{z}_{t+1}, a_t, \mathbf{z}_t)$ is the transition probability related to the motion intentions g.

Based on the current belief b_{t+1}, the weights $p_j \in \mathcal{P}$ for each motion intention are computed as:

$$p_j = \frac{\text{number of particles with motion intention } g_j}{\text{overall number of particles}}$$

These weights can also be interpreted as the weights for each trajectory under the associated motion intention.

D. Probabilistic Collision Constraint

Regarding the uncertainty of the obstacle vehicles' motion intentions and the uncertainty of the estimated positions, we utilize the concept of uncertainty ellipses to do probabilistic collision avoidance [16]. The uncertainty of the estimated positions of the obstacle vehicles can be handled by a chance constraint, which indicates that the collision probability is under a certain threshold.

We represent the distribution of future positions of obstacle i by a mixture Gaussian distribution, which handles the uncertainties under multiple policies. For each policy, we have a Gaussian distribution representing the position uncertainty at a certain time step. These position uncertainties are combined in the weighted sum of the Gaussian distributions as $\sum_j p_j \mathcal{N}(\hat{\mathbf{s}}_i(g_j), \sigma_i)$, where p_j is the belief for the motion intention g_j, defined in Section IV-C, $\hat{\mathbf{s}}_i(g_j)$ is the vehicle's position under the motion intention g_j and σ_i is the associated variance. For vehicle i and multiple policies, we denote by $\mathbb{B}^i$ the level set of the mixture Gaussian distribution at level p_ϵ, which is the required maximum probability of a collision, e.g. 0.1%. The level set is then

$$\mathbb{B}^i := \mathbb{B}^i(\mathcal{P}^i, \Pi^i) = \{\mathbf{s} | \sum_{j=1}^{l} p^i_j \mathcal{N}(\hat{\mathbf{s}}_i(g_j), \sigma_i) \geq p_\epsilon\}, \qquad (6)$$

where the variance on the position uncertainties $\sigma^A_{k+1:m}$ are assumed known. The chance constraint is then

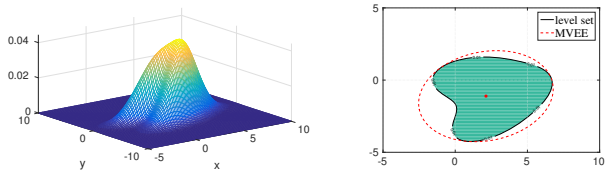
$$\mathbf{s}^0 \notin \mathbf{B}^i(\mathcal{P}^i, \Pi^i) = \mathbb{B}^i \oplus R, \quad \forall i \in [1, n], \qquad (7)$$

where $\mathbf{B}^i(\mathcal{P}^i, \Pi^i) = \mathbb{B}^i \oplus R$ is the Minkowski sum of the level set and the size of the ego-vehicle R. Specifically, the ego-vehicle is enclosed with four discs in order to simplify the collision checking.

The set $\mathbb{B}^i$ is shown as the filled area in the bottom right Fig. 4. It could be possible that the $\mathbb{B}^i$ is a connected and non-convex set, e.g. when two policies are close to each other. To simplify the collision checking, we find the minimal area enclosing ellipse (MAEE) to substitute the non-convex set $\mathbb{B}^i$. It can be computed by minimizing the ellipse area as:

$$\begin{aligned} \min_{A,\mathbf{c}} \quad & \log(|A|) \\ s.t. \quad & \mathbb{B}^i \subset \mathbb{E}^i \\ & \mathbb{E}^i = \{\mathbf{s} | (\mathbf{s} - \mathbf{c})^T A (\mathbf{s} - \mathbf{c}) \leq 1\} \end{aligned} \qquad (8)$$

where $\mathbb{E}^i$ is the MAEE for the connected set $\mathbb{B}^i$, A is a 2x2 matrix encoding the semi-axis and orientation of the ellipse and $\mathbf{c}$ is a vector representing the center of the ellipse. The Minkowski sum of $\mathbb{E}^i$ with the size of the ego-vehicle's disk is denoted as $\mathbf{E}^i := \mathbf{E}^i(\mathcal{P}^i, \Pi^i) = \mathbb{E}^i \oplus R$. Therefore, the



(a) Mixture Gaussian distribution (b) Minimal area enclosing ellipse

Fig. 4: Uncertainty ellipses. (a) shows the position uncertainty as a mixture Gaussian. (b) shows the level set $\mathbb{B}$ at the collision probability p_ϵ and the associated MAEE.

ellipse $\mathbf{E}^i$ is an over-approximation of the set $\mathbf{B}^i$. If the level sets of multiple policies are not connected, then the ellipses given by the level set of each Gaussian distribution are directly employed. With an abuse of notation, $\mathbf{E}^i$ is also denoted as the set of multiple ellipses if the level set has multiple disconnected components. See Figure 2 for an example of the different cases.

E. Receding-horizon Trajectory Planning

To compute the trajectory of the ego-vehicle, we can now rewrite the optimization problem in Eq. 2 by extending the Model Predictive Contour Control (MPCC) formulation of [4] under multiple hypotheses of motion intentions:

$$\begin{aligned}
\min_{\mathbf{u}_{0:m-1}} \quad & \sum_{k=0}^{m-1} J(\mathbf{z}_k^0, \mathbf{u}_k, \lambda_k)\Delta t_k + J(\mathbf{z}_m^0, \lambda_m)\Delta t_k \\
s.t. \quad & \mathbf{z}_{k+1}^0 = f(\mathbf{z}_k^0, \mathbf{u}_k) \\
& \lambda_{k+1} = \lambda_k + v_k \Delta t_k \\
& \mathbf{z}_{min}^0 < \mathbf{z}_k^0 < \mathbf{z}_{max}^0 \\
& \mathbf{u}_{min} < \mathbf{u}_k < \mathbf{u}_{max} \\
& b_l(\lambda_k) + w_{max} \leq d(\mathbf{z}_k, \lambda_k) \leq b_r(\lambda_k) - w_{max} \\
& \mathbf{s}_k^0 \notin \mathbf{E}_k^i(\mathcal{P}^i, \Pi^i) \qquad \forall i = \{1, ..., n\} \\
& \qquad\qquad\qquad\qquad \forall k = \{1, \ldots, m-1\}
\end{aligned} \tag{9}$$

where m is the planning horizon, $f(\mathbf{z}_k, \mathbf{u}_k)$ is the state transition for the ego-vehicle, λ_k is the estimated progress on the reference path, $\mathbf{s}_k^0$ is the position of the ego-vehicle at time step k and $\mathbf{E}^i(\mathcal{P}^i, \Pi^i)$ is the uncertainty ellipse for obstacle vehicle i. Each element of this optimization problem is obtained as follows:

1) Cost function: In MPCC, we define the approximated contour error $\hat{e}^c$ and lag error $\hat{e}^l$ which regulate the reference path tracking performance as:

$$\hat{e}_k^l(x_k, y_k, \lambda_k) = \mathbf{t}_k^T \begin{bmatrix} x_k - x^{ref}(\lambda_k) \\ y_k - y^{ref}(\lambda_k) \end{bmatrix} \tag{10}$$

where $\mathbf{t}_k$ is the longitudinal direction of the vehicle.

$$\hat{e}_k^c(x_k, y_k, \lambda_k) = \mathbf{n}_k^T \begin{bmatrix} x_k - x^{ref}(\lambda_k) \\ y_k - y^{ref}(\lambda_k) \end{bmatrix} \tag{11}$$

where $\mathbf{n}_k$ is the unit normal vector of the reference path at the estimated position.

Thus, the cost function at one time step can be written as:

$$\begin{aligned}
J(\mathbf{z}_k, \mathbf{u}_k, \lambda_k) = w_1 ||\hat{e}_k^c(x_k, y_k, \lambda_k)||^2 + w_2 ||\hat{e}_k^l(x_k, y_k, \lambda_k)||^2 \\
-w_3 v_k \Delta t_k + w_4 ||u_k^a||^2 + w_5 ||u_k^\delta||^2 + w_6 ||\dot{\theta}_k||^2
\end{aligned} \tag{12}$$

where $\mathbf{w} = [w_1, \ldots, w_5]$ are tuning parameters. For smoothness we also penalize the inputs.

2) Dynamics: For the ego-vehicle, the bicycle kinematic model is used to update the state,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \frac{v}{L}\tan(\delta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u^\delta \\ u^a \end{bmatrix} \tag{13}$$

where $[x, y, \theta]$ is the pose of the vehicle, δ is the steering angle, v is the speed and L is the length of the vehicle. The controlled actions of the ego-vehicle are the steering velocity u^δ and acceleration u^a.

3) Progress projected on reference path: The progress is approximated by an integrator with respect to the speed:

$$\lambda_{k+1} = \lambda_k + v_k \Delta t_k \tag{14}$$

where λ_k is the estimated progress on the reference path at time step k.

4) Limits on states and actions: The states and actions of the ego-vehicle are bounded to model the saturation and make the driving behavior more comfortable.

$$\mathbf{z}_{min}^0 < \mathbf{z}_k^0 < \mathbf{z}_{max}^0 \tag{15}$$

$$\mathbf{u}_{min} < \mathbf{u}_k < \mathbf{u}_{max} \tag{16}$$

$$||\dot{\theta}_k|| < \dot{\theta}_{max} \tag{17}$$

5) Road boundary: The road boundary can be used to limit the vehicle's position. The lateral distance $d(\mathbf{z}_k^0, \lambda_k)$ of the ego-vehicle can be computed as the contour error with respect to the reference path, such that $d(\mathbf{z}_k^0, \lambda_k) = \hat{e}^c(\mathbf{z}_k^0, \lambda_k)$. Thus, the lateral distance can be bounded as:

$$b_l(\lambda_k) + w_{max} \leq d(\mathbf{z}_k^0, \lambda_k) \leq b_r(\lambda_k) - w_{max} \tag{18}$$

where b_l is the left road boundary at the progress λ_k, b_r is the right road boundary and w_{max} is the upper bound of the vehicle's outline, which is projected on the reference path's norm direction.

V. Results

We evaluate our method in a typical T-junction traffic situation with five vehicles. For each of the obstacle vehicles, we consider three hypotheses for motion intention

$$\mathcal{G} = \{\text{Going straight, Turning, Stop}\}$$

We recall that for the ego-vehicle this assumption is not made and a trajectory in continuous space will be computed. In our simulation we run in parallel four POMDP estimators, one for each obstacle vehicle, and one MPC planner for the ego-vehicle. We utilize DESPOT [14] and FORCES Pro [17] to solve the POMDP and nonlinear optimization problem

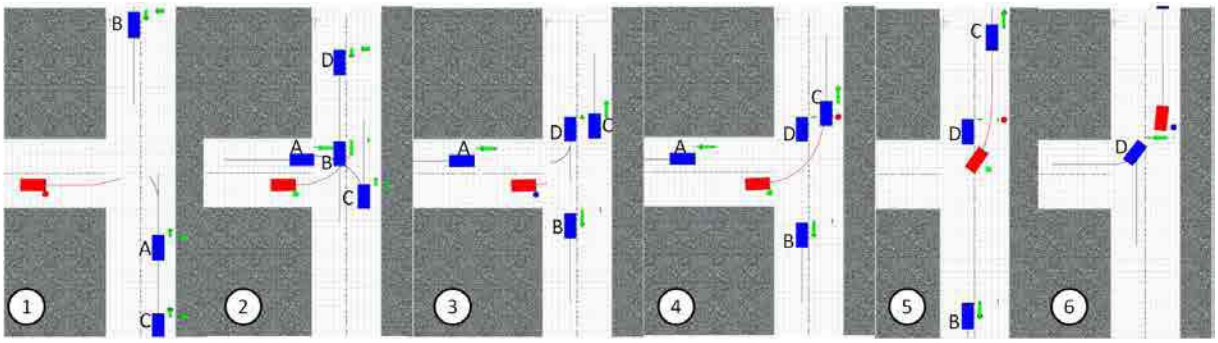
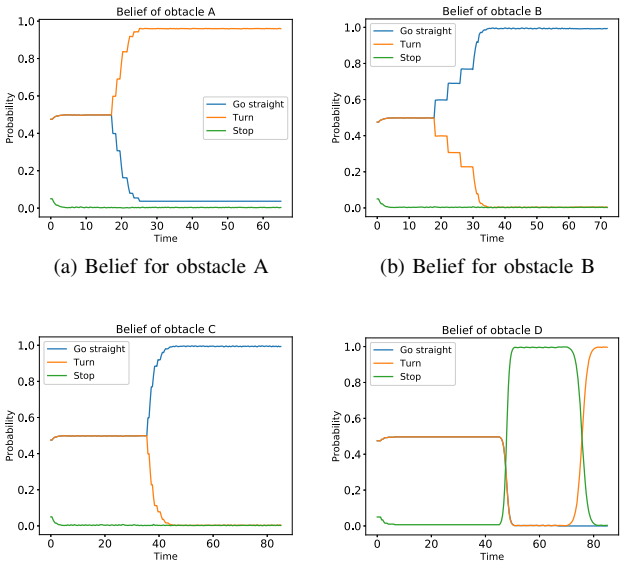


Fig. 5: A sequence of snapshots of a simulation of our method. Red line: planned path. Blue line: predicted paths of the obstacle vehicles under different motion intentions. Dot along with the ego-vehicle: action indicator, red: deceleration, green: acceleration, blue: maintain speed. Green straight arrow: belief of going straight. Green left arrow: belief of turning. The size of the arrows represents the likelihood of the action. Red dot: belief of stopping.

TABLE I: Parameters employed in the simulations

Hyper-parameters MPC planner	Value
Uncertainty level p_ϵ	0.01
Planning horizon m	50
Time interval in planning Δt	0.1 s
Boundary of acceleration u^a (m/s^2)	$-2 < u^a < 2$
Boundary of steering velocity u^δ (rad/s)	$-4 < u^\delta < 4$
Boundary of speed v (m/s)	$0 \leq v \leq 4$
Boundary of steering angle δ (rad)	$-0.4 < \delta < 0.4$
Hyper-parameters POMDP estimator	**Value**
Number of scenarios K	100
Search depth	10
Time interval in prediction (s)	0.1
Discount γ	0.95
Reward for acceleration / deceleration R_{action}	-1
Reward for collision $R_{collision}$	-100
Reward for progress $R_{progress}$	$50e^{-\frac{d}{32}}$
Acceleration Deceleration (m/s^2)	2
Maximum velocity (m/s)	6



(a) Belief for obstacle A (b) Belief for obstacle B

(c) Belief for obstacle C (d) Belief for obstacle D

Fig. 6: Belief of the motion intentions for each obstacle vehicle. Blue: going straight; Orange: turning; Green: stop.

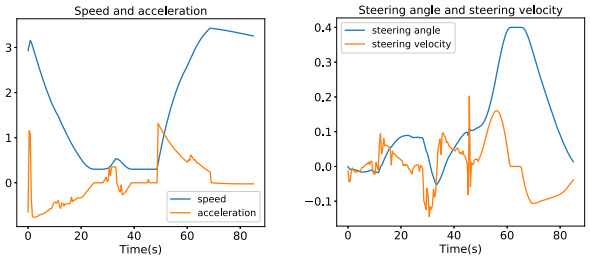
respectively. In Table I we detail the parameters used in our simulation. The policies for the obstacle vehicles are fixed but unknown to the ego-vehicle. We performed multiple simulations with randomized initial conditions and motion policies. All experiments were conducted on an Intel Core i7-4710MQ CPU @ 2.50GHz×8 laptop.

The simulation result is illustrated as a sequence of snapshots in Fig. 5, where the red rectangle represents the ego-vehicle and the blue rectangles represent the obstacle vehicles. The ego-vehicle does not know the motion intention and future behaviors of the obstacle vehicles.

Initially, the ego-vehicle maintains equal beliefs for going straight and turning motion intentions of each obstacle vehicle (Fig. 5 ①). Since the uncertainty of the obstacles' motion are quite high, the ego-vehicle decelerates and waits. When the beliefs of obstacle A and obstacle B are quite certain (Fig. 5 ②), the ego-vehicle tries to move forward with consideration of obstacle C's and obstacle D's estimated behavior. Since the POMDP estimator predicts the obstacle D could possible go straight (Fig. 5 ③), the ego-vehicle maintains its speed to make sure of the obstacle's motion intentions. Suddenly, the obstacle D stops to intentionally give way. Therefore, after the ego-vehicle obtains a high belief that the obstacle D will stop, it accelerates and makes the left turn (Fig. 5 ④, ⑤). Eventually, the ego-vehicle safely merges to the traffic. The belief update for each obstacle vehicle is shown in Fig. 6. And the optimal actions computed from multipolicy MPC are summarized in Fig. 7.

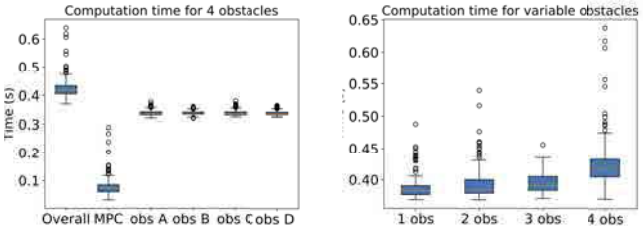
The computation time of our approach is shown in Fig. 8a. For the scenario with four obstacle vehicles, the average time for the joint behavior estimation and planning algorithm was $0.42s$. Nearly 78% computation time was spent on the POMDP estimator since the DESPOT search is expensive. We observe that the multi-policy MPC solver is much more efficient with almost $10Hz$ computational level. Thanks to the parallelization of the POMDP estimator, good scalability is observed, see Fig. 8b.

In Table II we summarize the average and the minimal



(a) Acceleration and speed (b) Steering velocity & steering angle

Fig. 7: Optimal control inputs for the ego-vehicle. (a) acceleration (orange) and speed (blue) computed from the multipolicy MPC. (b) steering velocity (orange) and steering angle (blue) for the ego-vehicle.



(a) 4 obstacles (b) Variable obstacles

Fig. 8: Computation time for our method. (a) Computation time for the scenario of Fig. 5 with four obstacles. The first column shows the overall time of the method. The second column is the time spent on the multi-policy MPC. The third to sixth columns are the time spent by the DESPOT solver to estimate the obstacles' trajectories. (b) Shows the overall computation time with respect to the number of obstacles (the estimation part is parallelized over the obstacle vehicles).

clearance between the ego-vehicle and the obstacle vehicles over multiple simulations. We compare the joint estimation and planning approach with the standalone multi-policy MPC planner without the behavior estimation. In the standalone multi-policy MPC, we assume a constant speed model to predict the future behaviors of the obstacle vehicles for each motion intention. The joint approach keeps a safer distance from the obstacle vehicles, thanks to the consideration of interactions in the POMDP estimator. We also observe that even with a simple constant velocity assumption our multi-policy MPC planner performs well.

VI. CONCLUSION

In this paper we have introduced a joint behavior estimation and trajectory planning algorithm integrating the strengths of POMDP and MPC. As shown in multiple simulations with up to five vehicles, the method accounts for the uncertainty on the motion intentions of other traffic participants and produces safe trajectories for the ego-vehicle. In particular, the chance constrained multi-policy MPC planner is general and can be employed together with other behavior estimators. Furthermore, the method scales well with the number of obstacles thanks to the parallelization of the behavior estimation and the efficiency of code generation for MPC. Future work will seek to improve the performance of the estimator and test the method on autonomous vehicles.

TABLE II: Average clearance and minimal clearance

Algorithm	Avg clearance	Min clearance
Joint behavior estimation & planing	7.11	3.05
Multipolicy MPC	6.97	2.88

REFERENCES

[1] C. Urmson and W. Whittaker, "Self-driving cars and the Urban challenge," *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 66–68, 2008.

[2] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F. R. Kline, "The MIT - Cornell collision and why it happened," *Springer Tracts in Advanced Robotics*, vol. 56, pp. 509–548, 2009.

[3] S. Karaman and E. Frazzoli, "Incremental Sampling-based Algorithms for Optimal Motion Planning," *Robotics: Science and Systems*, p. 20, 2010. [Online]. Available: http://arxiv.org/abs/1005.0416

[4] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1928–1935.

[5] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *IEEE International Conference on Robotics and Automation*, 2012.

[6] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, 2010.

[7] P. Trautman, J. Ma, R. M. R. M. Murray, and A. Krause, "Robot Navigation in Dense Human Crowds: the Case for Cooperation," *Cds.Caltech.Edu*, pp. 2153–2160, 2013.

[8] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," *Proc. - IEEE International Conference on Robotics and Automation*, pp. 4015–4020, 2014.

[9] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably - Cooperative partial motion planning with maximum entropy models." *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[10] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for Autonomous Cars that Leverage Effects on Human Actions," *Proceedings of Robotics: Science and Systems*, 2016.

[11] W. Liu, S. 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, Proceedings*, vol. 2015-August, 2015, pp. 1126–1133.

[12] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 454–460, 2015.

[13] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction," in *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[14] 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.

[15] W. Liu, S.-W. Kim, and A. J. Marcelo H., "Probabilistic Road Context Inference for Autonomous Vehicles," *IEEE International Conference on Robotics and Automation (ICRA2015)*, pp. 1640–47, 2015.

[16] N. E. Du Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," *Proceedings - IEEE International Conference on Robotics and Automation*, 2010.

[17] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH (http://embotech.com/FORCES-Pro), July 2014.