MPDM: Multipolicy Decision-Making in Dynamic, Uncertain Environments for Autonomous Driving

Alexander G. Cunningham, Enric Galceran, Ryan M. Eustice, and Edwin Olson

Abstract— **Real-world autonomous driving in city traffic must cope with dynamic environments including other agents with uncertain intentions. This poses a challenging decision-making problem, e.g., deciding when to perform a passing maneuver or how to safely merge into traffic. Previous work in the literature has typically approached the problem using ad-hoc solutions that do not consider the possible future states of other agents, and thus have difficulty scaling to complex traffic scenarios where the actions of participating agents are tightly conditioned on one another. In this paper we present multipolicy decision-making (MPDM), a decision-making algorithm that exploits knowledge from the autonomous driving domain to make decisions online for an autonomous vehicle navigating in traffic. By assuming the controlled vehicle and other traffic participants execute a policy from a set of plausible closed-loop policies at every timestep, the algorithm selects the best available policy for the controlled vehicle to execute. We perform policy election using forward simulation of both the controlled vehicle and other agents, efficiently sampling from the high-likelihood outcomes of their interactions. We then score the resulting outcomes using a user-defined cost function to accommodate different driving preferences, and select the policy with the highest score. We demonstrate the algorithm on a real-world autonomous vehicle performing passing maneuvers and in a simulated merging scenario.**

I. INTRODUCTION

During the last decade, there has been a great deal of work on the development of fully autonomous cars capable of operating in urban traffic. The problem of robust autonomous driving has been investigated through earlier work during the DARPA Grand and Urban challenges, and since then there have been many teams actively developing improved capabilities. Work in areas of path planning, multi-sensor perception and data fusion has allowed vehicles to navigate difficult environments and handle the presence of obstacles and other hazards.

A key challenge for an autonomous vehicle capable of operating robustly in real-world environments is the discrete uncertainty that exists in choosing actions that account for the intent of other agents in the environment. These agents include other drivers on the road, as well as pedestrians at crosswalks and in parking lots. Much of the decision-making made by human drivers is over discrete actions, such as choosing whether to change lanes or whether to pull out into traffic. These decisions need to be informed by both the continuous uncertainty of the state, such as the actual position of vehicles, but also the discrete uncertainty such as whether another driver is making a turn, or is trying to overtake another vehicle.

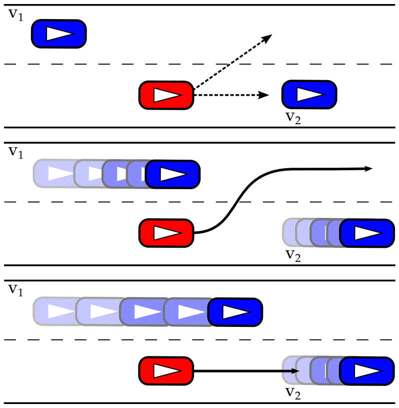


Fig. 1. In the top image, the red car is faced with a discrete choice as to whether change lanes to pass vehicle v_2 in front of vehicle v_1 (middle) or remain behind slow vehicle v_2 (bottom). The MPDM algorithm evaluates these possibilities, while simulating forward the other vehicles, shown in the center and bottom images. By considering the *closed-loop interactions between vehicles*, when the red car pulls in front of v_1, the simulation expects v_1 to slow down to accommodate a safe lane change. Without this consideration, the red car would have assumed v_1 would simply maintain its current speed and declared the lane change infeasible.

Directly managing this discrete uncertainty is significant because accounting for the breadth of behaviors necessary to operate a car within traffic makes traditional techniques, such as hand-tuned finite state machines (FSMs) [1] or trajectory optimization [2] difficult to scale up to full real-world driving. A major drawback of such techniques is that they fail to model interactions between multiple agents, which is key for navigating dynamic traffic environments efficiently and safely.

In contrast, this paper presents multipolicy decision-making (MPDM), a high-level decision process that factors autonomous driving into a set of *policies* that encode closed-loop behaviors and uses online simulation to evaluate the consequences of available policies. The central contributions of this paper are:

- A decision-making algorithm able to handle traffic.
- This technique leverages simulation of *closed-loop interactions* to reason about action consequences.
- An evaluation of MPDM in simulation and on a real-world autonomous vehicle in passing and merging traffic scenarios.

Modeling vehicle behavior as a closed-loop policy for both the car we are controlling *and nearby vehicles*, manages

This work was supported in part by a grant from Ford Motor Company via the Ford-UM Alliance under award N015392 and in part from DARPA under award D13AP00059.

The authors are with the University of Michigan, Ann Arbor, MI 48109, USA.
{alexgc,egalcera,eustice,ebolson}@umich.edu

uncertainty growth by assuming other vehicles will make reasonable, safe decisions. Simulating forward the other vehicles in the environment allows us to account for changes in driver behavior induced by neighboring vehicles, as occurs in scenarios such as merging, where another driver will slow down to make space for a merging vehicle (see Fig. 1).

Policies within this approach are closed-loop deterministic controllers that implement high-level driving behaviors, such as driving along a lane, changing lanes or parking. To select the optimal policy, for each candidate policy we sample a policy outcome from the current world state using forward simulation of both the vehicle state and the future actions of other traffic participants. We can then evaluate a reward function over these sampled outcomes to accommodate user driving preferences, such as reaching goals quickly and ride comfort. We demonstrate the algorithm in a real-world autonomous vehicle performing passing maneuvers and in a simulated lane merging scenario.

Furthermore, we develop our approach in a principled manner deriving from the partially observable Markov decision process (POMDP) model (see [3] for a tutorial), as it provides a powerful framework for simultaneously considering optimality criteria and the inherent uncertainty of dynamic environments. Unfortunately, finding optimal solutions for general POMDPs is intractable [4], [5], especially in continuous state and action domains. However, we will use the POMDP as a formal model to make clear where approximations and assumptions are being made.

II. RELATED WORK

The most notable first early instances of decision-making architectures for autonomous vehicles capable of handling complex traffic situations stem from the 2007 DARPA Urban Challenge [6], an autonomous car race held in a mock urban environment. As mentioned earlier, DARPA participants tackled decision-making using a variety of solutions ranging from FSMs [1] and decision trees [7] to heuristic approaches [8]. However, these approaches were tailored for very specific and simplified situations and were "not robust to a varied world" [8].

A number of investigators have tackled the decision-making problem for autonomous driving through the lens of trajectory optimization. Tran and Diehl proposed a convex optimization method that they apply to an automated car-like vehicle in simulation [9]. However, their approach does not consider dynamic objects. Gu and Dolan used dynamic programming to generate trajectories that do consider dynamic objects in the environment [10]. Although their simulation results are promising, they do not demonstrate their method on a real vehicle. The trajectory optimization method proposed in [2] optimizes a set of costs that seek to maximize path efficiency and comfort, accounting as well for the distance to static and dynamic obstacles. They demonstrate the method on a simple passing maneuver, but their real vehicle results are limited to static obstacles. The key problem with trajectory optimization in traffic scenarios is that they account for where other vehicles are now, but will not be able to account for what the vehicle will do in the future, particularly in response to actions from our vehicle.

POMDPs provide a mathematically rigorous formulation of the decision-making problem in dynamic, uncertain scenarios such as autonomous driving. A variety of general POMDP solvers exist in the literature that seek to approximate the solution, e.g. [11], [12]. Nonetheless, they require computation time on the order of several hours even for problems with very small state, action and observation spaces compared to real-world scenarios. A loose parallelism with our approach in the POMDP literature can be found in [13], where a POMDP solver is proposed that exploits domain knowledge provided as a set of pre-computed initial policies, which the solver then refines and switches on and off over time. However, this approach still requires unreasonable amounts of computation time to be applied to real-world systems. In fact, the idea of assuming finite sets of policies to speed up planning has appeared before, particularly in the POMDP literature [14]–[16]. However, these approaches dedicate significant resources to compute their sets of policies, and as a result they are limited to short planning horizons and relatively small state, observation, and action spaces. In contrast, we propose to exploit domain knowledge from autonomous driving to design a set of policies that are readily available at planning time.

Furda and Vlacic take a more practical approach to the problem by formalizing decision-making for autonomous vehicles using multiple-criteria decision-making (MCDM) theory [17]. However, they do not explicitly consider the potential future intentions of other traffic participants. More recently, Wei et al. presented a two-step decision-making approach that finds suitable velocity profiles by evaluating a set of candidate actions [18]. Nonetheless, their method is targeted at in-lane driving situations, and does not consider complex decisions such as passing or merging. In earlier work, Wei et al. presented modeled interactions between vehicles on highway ramps to perform merging maneuvers [19]. However, their results are limited to simulations. Similarly, Trautman et al. explored modeling interactions of other agents for planning [20], presenting an evaluation with a mobile robot navigating crowded indoor environments.

Overall, there remains a substantial gap in the prior work for principled decision-making that is robust in scenarios with *extensively coupled interactions between agents*. This work addresses this problem directly by modeling the high-level behaviors of all agents in the system.

III. PROBLEM STATEMENT

The problem of decision-making in dynamic, uncertain environments with tight coupling between the actions of multiple agents can be formulated as a POMDP, which provides a mathematical model that connects perception and planning in a principled way. Here, we first formulate the problem as a general multi-agent POMDP. Then, we use this formulation to show where we make approximations in our approach via reasonable domain assumptions, achieving a decision-making system that can control an autonomous vehicle navigating in a multi-agent setting online.

A. General Decision Process Formulation

Let $v \in V$ denote one of N vehicles in the local area, including our controlled vehicle, for which we can define an action $a_t^v \in \mathcal{A}$ that transitions its state $x_t^v \in \mathcal{X}$ at time t to a new state x_{t+1}^v. An action a_t^v is a tuple of the actuated controls on our car for the steering, throttle, brake, shifter, and directionals. Note that to control a vehicle reliably, it is necessary to choose actions of this granularity at a relatively high rate — on the order of 30 to 50 Hz. As a notational convenience, let x_t be the set of state variables for all vehicles, and correspondingly let a_t be the actions of all vehicles.

To model the dynamics and uncertainty in the system, we use a Markovian model to evolve the system forward in time, based on models for dynamics, observation, and driver behavior. A conditional probability function $T(x_t, a_t, x_{t+1}) = p(x_{t+1}|x_t, a_t)$ models the effect of actions on vehicle states. Likewise, we model observation uncertainty as the conditional probability function $Z(x_t, z_t^v) = p(z_t^v|x_t)$ where $z_t \in \mathcal{Z}$ is the combined set of sensor observations at each time t, including observed vehicle states and a map of static hazards in the environment. We further model the behavior of other agents in the environment as a conditional probability distribution $D(x_t, z_t^v, a_t^v) = p(a_t^v|z_t^v, x_t)$ in which the action taken by drivers is conditioned only on the current state and observation.

The core problem we wish to solve in this decision process is to choose an optimal policy π^* for our vehicle, in which a policy for a vehicle is a deterministic mapping $\pi^v : x_t \times z_t^v \to a_t^v$ that yields an action from the current state and observation. The decision process chooses the policy to maximize the reward over a given decision horizon H as follows:

$$\pi^* = \operatorname*{argmax}_{\pi} \sum_{t=0}^{H} \gamma^t \int_{x_t} R(x_t) p(x_t)\, dx_t, \tag{1}$$

where γ^t is a reward discount factor, and $R(x_t)$ is the reward function. We can define the joint density $p(x_t)$ as follows

$$p(x_{t+1}) = \iiint\limits_{x_t\, z_t\, a_t} p(x_{t+1}, x_t, a_t, x_t)\, da_t\, dz_t\, dx_t \tag{2}$$

and decomposing recursively using the state transition, observation, and driver behavior models above yields

$$p(x_{t+1}) = \iiint\limits_{x_t\, z_t\, a_t} p(x_{t+1}|x_t, a_t) p(a_t|z_t, x_t) \\ p(z_t|x_t) p(x_t)\, da_t\, dz_t\, dx_t. \tag{3}$$

Given that this is a multi-vehicle system, we can assume the instantaneous actions for each vehicle will be independent of each other, as the action a_t^v only depends on the current state x_t and the local observation z_t^v. Let the joint density for a single vehicle v be

$$p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v) = p(x_{t+1}^v|x_t^v, a_t^v) p(a_t^v|x_t^v, z_t^v) \\ p(z_t^v|x_t^v) p(x_t^v). \tag{4}$$

Leveraging the independence assumption, we obtain

$$p(x_{t+1}) = \prod_{v \in V} \iiint\limits_{x_t\, z_t\, a_t} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v)\, da_t^v\, dz_t^v\, dx_t^v. \tag{5}$$

To incorporate a deterministic policy π^q for the vehicle $q \in V$ under our control, we can replace the driver behavioral model in the single car joint defined in Eq. 4. With this model constructed, we can estimate the expected reward for a given policy π^q by drawing samples from the full system model in Eq. 5 that we propagate over the entire decision horizon H.

The problem with this formulation is that when sampling from the distribution in Eq. 5, because of the uncertainties at every stage, each sample will have a very small posterior probability due to the large state space of this system. The large state space with many levels of uncertainty results in a combinatorial explosion, particularly as we account for all the possible action sequences other vehicles could undertake.

In the practical case for driving, we want to sample high-likelihood scenarios on which to make decisions. Sampling over the full model will result in many cases of other drivers acting in ways substantially different from how human drivers behave, including swerving off of roads and into other lanes. However, we wish to capture in our model the vast majority of driving, in which all drivers are safe most of the time, so we can anticipate likely actions for other vehicles. The next section applies approximations designed to focus sampling on more likely outcomes.

B. Approximate Decision Process

In this section we will introduce two key approximations that reduce the state space sufficiently to be tractable for real-time use: 1) choosing policies from a finite discrete set of known policies for both our car and other cars and 2) approximating the vehicle dynamics and observation models through deterministic, closed-loop forward simulation of all vehicles with assigned policies. The result of these approximations will be to convert the problem of finding a policy into a discrete decision-making problem over high-level vehicle behaviors.

Let Π be a discrete set of carefully constructed policies, where each policy captures a specific high-level behavior, such as following a lane, or making a lane change. Because we assume other cars on the road follow basic driving rules, we can also select a policy $\pi^v \in \Pi$ to model their behavior. Thus, we can reconstruct the per-vehicle joint from Eq. 4 as

$$p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v) = p(x_{t+1}^v|x_t^v, a_t^v) p(a_t^v|x_t^v, z_t^v, \pi_t^v) \\ p(\pi_t^v|x_t^v) p(z_t^v|x_t^v) p(x_t^v), \tag{6}$$

where we approximate the driver behavior term $p(a_t^v|x_t^v, z_t^v, \pi_t^v)$ as deterministic given π_t^v. The additional term $p(\pi_t^v|x_t^v)$ in comparison to Eq. 4 models the probability of a given policy being selected for this vehicle. In this paper we assume that we can determine the most-likely policy π_t^v for other vehicles given a model of the road network, and focus on the control of our vehicle. We will address accurate computation of $p(\pi_t^v|x_t^v)$ in future work.

Using the formulation for single-vehicle joint distributions of Eq. 6, we finally split out other vehicles $v \in V$ and the vehicle under our control $q \in V$ separately as follows:

$$p(x_{t+1}) \approx \iint\limits_{x^q\, z^q} p^q(x_t^q, x_{t+1}^q, z_t^q, a_t^q, \pi_t^q)\, dz_t^q\, dx_t^q$$
$$\prod_{v \in V | v \neq q} \left[\sum_{\Pi} \iint\limits_{x^v\, z^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v)\, dz_t^v\, dx_t^v \right] . \tag{7}$$

By modeling policies as closed-loop systems, we can reasonably approximate the state transition term $p(x_{t+1}^v|x_t^v, a_t^v)$ in Eq. 6 with a deterministic simulation of the system dynamics. This is a reasonable approximation to make because we assume we have engineered all policies to generate action sequences that are achievable within the safe performance envelope of the vehicle, thereby reducing the impact of uncontrolled vehicle dynamics.

IV. MULTIPOLICY DECISION-MAKING

Our proposed algorithm, MPDM (Algorithm 1), implements the approximate decision process of Sec. III-B using deterministic simulation to approximate the execution of closed-loop policies for both our car and nearby cars. The key is the assumption that agents in the environment execute actions that can be modeled as a set of policies crafted based on knowledge from the autonomous driving domain, efficiently approximating the solution to the problem stated in Sec. III.

MPDM is robust to future uncertainty at the discrete decision level through continuous replanning and at the low-level control level through closed-loop policy models. Similar to model-predictive control techniques, continuous replanning over a longer time horizon H while only executing over a shorter horizon lets us react to changing vehicle behavior. Closed-loop policy models ensure robustness to bounded low-level uncertainty in state estimation and execution, as the policies can adapt to local perturbations.

The algorithm takes as input a set of candidate policies Π, the current most likely estimate over the world state $p(x_0)$, and a decision horizon H. Note that the estimate over the world state includes the most-likely policies currently executed by the other agents, which in this work we determine according to a road network model and the pose of the agents therein. The algorithm then determines a set of applicable policies Π_a given x_0 that are relevant in the current world state. The next step of the algorithm consists in scoring each policy according to a user-defined cost function using forward simulation. In this step, for each applicable policy π, we sample the evolution of the system from state x_0 under π to obtain a sequence of states of the world $\Psi = (x_0, x_1, \ldots, x_H)$, where $x_t = \pi(x_{t-1}, z_{t-1})$ for $0 < t \leq H$. Next, the sequence Ψ is scored using a user-defined cost function. The score obtained, c, is added to the set of scores C. Finally, the optimal policy π^* associated to the highest score in C is returned.

Algorithm 1: MPDM policy election procedure.

```
Input:
  • Set $\Pi$ of available policies for our vehicle and others.
  • Most likely estimate $p(x_0)$ of the state at planning
    time, including the most-likely policies $\pi_0^v \in \Pi$ for
    each of the other vehicles.
  • Planning horizon $H$.
$\Pi_a \leftarrow \emptyset$
foreach $\pi \in \Pi$ do
    if APPLICABLE($\pi, x_0$) then
        $\Pi_a \leftarrow \Pi_a \cup \{\pi\}$
$C \leftarrow \emptyset$
foreach $\pi \in \Pi_a$ do
    $\Psi \leftarrow$ SIMULATEFORWARD($x_0, \pi, H$)
    $c \leftarrow$ COMPUTESCORE($\Psi$)
    $C \leftarrow C \cup \{\langle \pi, c \rangle\}$
$\pi^* \leftarrow$ SELECTBEST($C$)
return $\pi^*$
```

A. Policy Design

Each policy implements a particular closed-loop driving behavior, such as driving along a lane, changing lanes or executing a parking maneuver. At runtime, we execute the currently selected policy in a separate process from the policy election procedure. These policies are individually engineered to account for particular driving behaviors, with varying levels of complexity. For instance, in a simple driving scenario, the policies can be:

- lane-nominal: drive in the current lane and maintain distance to the car directly in front,
- lane-change-left/lane-change-right: a separate policy for a single lane change in each direction,
- park-car: stop the car within a marked parking space.

Contruction of this set of policies is primarily dependent on covering the set of behaviors necessary to navigate the given road network and comply with traffic rules. We can adjust the scope of the behaviors represented to match particular applications, such as limiting the car to highway-only behaviors.

At any given world state x_t, it is likely only a subset of possible vehicle behaviors will be feasible to execute, so we first run an *applicability* check on each available policy. For example, if our car is in the right-most lane of a highway, policies that perform a lane-change to the right would not be applicable.

Note that the policy election procedure detailed in Algorithm 1 does not, in practice, run fast enough to account for either emergency vehicle handling or abrupt changes in system state, so therefore all policies are designed to always yield a safe action. This constraint ensures we can respond to a changing environment at real-time speeds, without being bound by the speed of policy election. This safety criteria is important for managing outlier policies for other vehicles (with relation to $p(\pi_t^v|x_t^v)$) in which we allow our individual policies to manage dangerous cases. In future work, we expand the set of policies for other vehicles to include more readily modeled outlier cases, such as stopping.

B. Multi-vehicle Simulation

By casting the forward simulation as a closed-loop deterministic system, we can capture the necessary interactions between vehicles to make reasonable choices for our vehicle behavior. We choose a likely policy for each other vehicle in the environment, and then step forward via the deterministic state transition model detailed earlier in Eq. 7.

In order to achieve policy election at a real-time rate on the order of 1 Hz or faster, we rely on the closed-loop nature of the low-level control to achieve approximate simulation. While it is possible to perform high-fidelity simulation, in practice we use a simplified simulation model for each vehicle assuming ideal steering control. The key is that the simulation models inter-vehicle interactions sufficiently well to make reasonable decisions about which policy to execute.

C. Policy Election

To select a policy to follow, we need to evaluate the outcomes of the simulations for each policy under consideration using a cost function including a variety of user-defined metrics, and then choose the best policy. The difficulty in this problem is in accounting for the many criteria that appear in real-world driving decisions. An autonomous car must simultaneously reach the destination in a timely manner, but also drive in a way that is comfortable for passengers, while following driving rules and maintaining safety.

We cast these criteria as a set of metrics $m \in \mathcal{M}$, where each metric is a function $m : \{x_t\} \rightarrow \mathbb{R}$ that evaluates the full simulated state and action sequence over the fixed horizon H. Our typical metrics include

- distance to goal: measured from the final pose to the goal waypoint in map,
- lane choice bias: an increasing cost for lanes further away from the right-most lane,
- max yaw rate: the maximum recorded yaw rate during the simulated trajectory, and
- simple policy cost: a hard-coded constant cost for a given policy to break ties.

These metrics capture accomplishment of goals, safety, implementation of "soft" driving rules, and rider comfort. The challenge in combining these metrics is that each one returns a cost in different units with different expected cost distributions. We combine these costs into a single score for each policy by computing a per-policy score using a linear combination of the normalized scores for each metric. For each metric m_j, we compute a corresponding weight w_j that encodes both an empirically tuned importance of the metric depending on user requirements, as well as how informative the metric is within the given set of policies. We downweight uninformative metrics in which there is too little variation among the policies.

V. EVALUATION

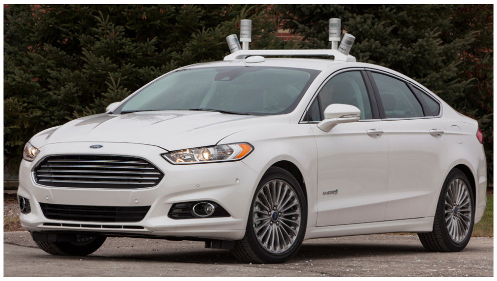


Fig. 2. Our autonomous car platform, a Ford Fusion equipped with four LIDAR units, survey-grade INS, and a single forward-looking camera. All control and perception is performed onboard.

We evaluate MPDM using real-world and simulated experiments in passing and merging scenarios. This evaluation highlights the utility of simulating forward both our car and neighboring cars with closed-loop policies. Passing another vehicle demonstrates switching between policies as better options become available as the vehicle advances. Merging highlights how simulating forward all cars with closed-loop policies allows our system to account for the reactions of other drivers as a consequence of our action.

For the passing scenario, we perform real-world experiments on a closed test track to validate the approach. We demonstrate merging in a simulated environment to allow for a larger number of cars present than is typical in our real-world test environment.

As a scope limitation for this paper, we assume the policy used for other cars is easily inferred from direct observation. The other vehicles in our system perform only a straightforward lane-keeping behavior that will slow down to account for vehicles within their path. This behavior is, however, sufficient to demonstrate the kinds of tightly-coupled vehicle interactions we wish to evaluate.

A. Autonomous Vehicle Platform

For our real-vehicle experiments, we used our autonomous vehicle platform (see Fig. 2). This automated vehicle is a Ford Fusion equipped with a drive-by-wire system, four Velodyne HDL-32E 3D LIDAR scanners, an Applanix POS-LV 420 inertial navigation system (INS), a single forward-looking Point Grey Flea3 camera and several other sensors. An onboard five-node computer cluster performs all planning, control, and perception for the system in realtime.

The vehicle uses prior maps of the area it operates on constructed by a survey vehicle using 3D LIDAR scanners. These prior maps capture information about the environment such as LIDAR reflectivity and road height, and are used for localization and other perceptual tasks. The road network is encoded as a metric-topological map using a derivative

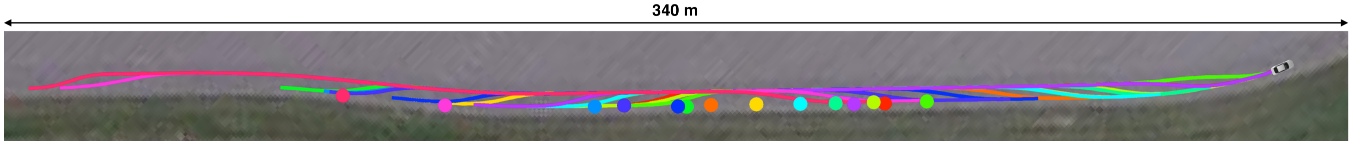
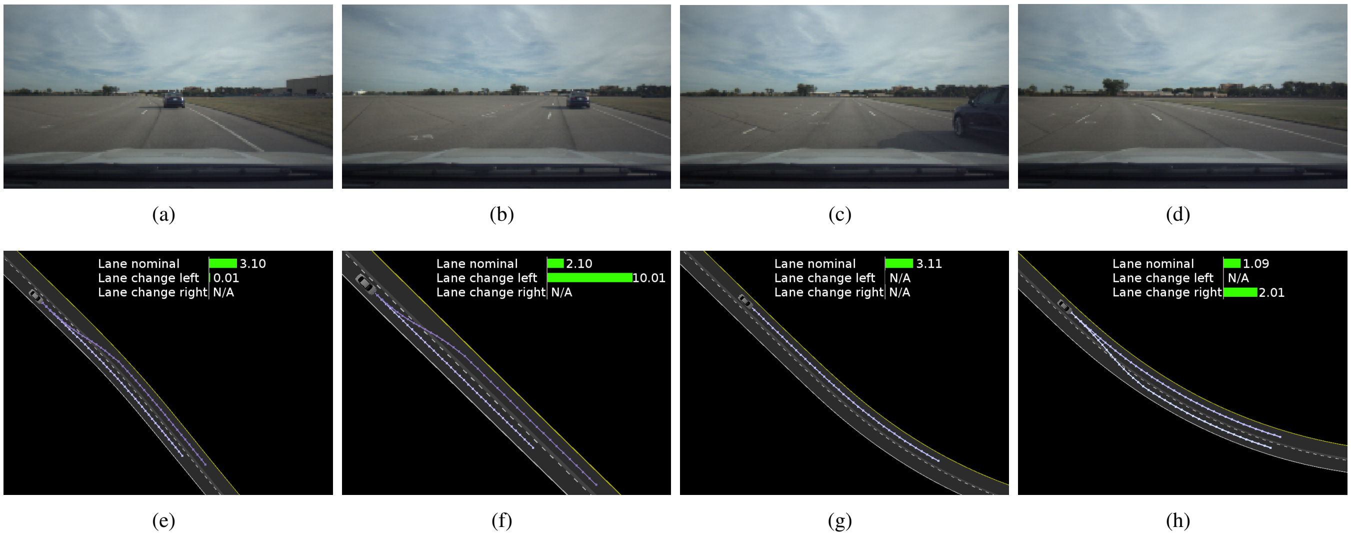


Fig. 3. Trajectories of 14 passing maneuvers executed using MPDM on a test track, overlapped on satellite imagery. Each trajectory is colored with a different color. The circles correspond to the location of the passed vehicle half way through the passing maneuver, where the color of each circle matches that of its associated passing trajectory. A model of our autonomous vehicle platform appears on the far right for scale. Satellite imagery credit: Google.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Execution of a passing maneuver with MPDM. Successive stages of the decision-making procedure are shown in snapshots from an onboard camera on the autonomous vehicle performing the maneuver (top row) and in a visualization of the forward simulations for the available policies and their scores (bottom row). Initially, the vehicle cruises at nominal speed using the lane nominal policy (a), (e). Although the lane change left policy is able to drive the vehicle further according to the forward simulation, a preference to stay in the right lane encoded in the scoring function prevails. As the lane nominal policy encounters a slower vehicle upfront, the lane change left policy achieves the highest score and the vehicle initiates a lane shift (b), (f). After reaching the goal lane and finding a clear course, lane nominal takes over again to speed up and pass the slower vehicle (c), (g). Note that at this point only the lane nominal policy is applicable due to the presence of another vehicle in the adjacent lane. Finally, the vehicle returns to the initial lane, once more, as per the effect of the preference to stay in the right lane when possible encoded in the scoring function (d), (h).

of the route network definition file (RNDF) format [21], providing information about the location and connectivity of road segments and lanes therein.

Estimates over the states of other traffic participants are provided by dynamic object tracking running on the vehicle, which uses LIDAR range measurements. The geometry and location of static obstacles are also inferred onboard using LIDAR measurements.

B. Simulation Environment

For both simulating forward the consequences of policies in MPDM and for evaluating our system, we have developed a multi-agent traffic simulation engine. The engine allows simulation of the system dynamics and perception of the agents involved, and is fully integrated with the real vehicle platform's software architecture via the lightweight communications and marshalling (LCM) framework [22].

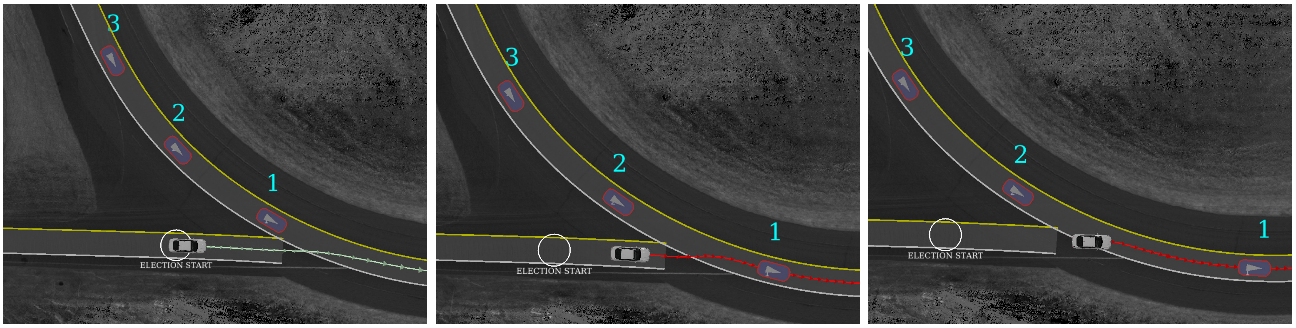
VI. RESULTS

We now report on the results of the evaluation of MPDM both on our real autonomous vehicle platform and in simulation. We first demonstrate MPDM on the real vehicle in a series of passing maneuvers, where our algorithm decides to pass a slower vehicle in the preferred lane of travel. Using these real-world results, we then show the performance of our simulator by comparing its outcomes with the actual trajectories of both our vehicle and other traffic participants. Finally, we demonstrate our algorithm in a simulated merging scenario. All policy elections throughout the experiments use a planning horizon $H = 10$ s discretized at timesteps of $\Delta t = 0.25$ s.

A. Passing Scenario on the Real Vehicle

We evaluated MPDM in a multi-lane setting on a closed test track in a scenario in which we pass a slower human-controlled vehicle. In these experiments, our vehicle starts driving on the right lane and, as it advances, encounters the slower vehicle in its preferred lane of travel that is limiting its progress. At that point, as a consequence of the scoring function used, the vehicle decides to pass the slower vehicle. Fig. 3 shows the trajectory of our controlled vehicle and the location of the passed vehicle halfway through the passing maneuver for the 14 trials we executed in this scenario.

The policies considered for the controlled vehicle in this scenario are lane-nominal, lane-change-left and lane-change-right, while a single lane-nominal policy is considered to simulate forward the future states of the passed vehicle.



(a) (b) (c)

Fig. 5. Simulated merging scenario highlighting the simulation of a policy merging into traffic, even if the gap present is not already large enough, by anticipating the behavior of another car. In the election start point (left), the car evaluates the merging policy to yield its predicted trajectory shown in white. When executing the policy, as in the prediction in simulation, our car drives out in front of car 2, causing car 2 to start slowing (center) as anticipated, and eventually completes the merge into traffic (right). The red trajectory shows the planned path for the merging policy at runtime.

Fig. 4 shows the evolution of the policy scores throughout the passing maneuver.

B. Forward Simulation Evaluation

After running our algorithm in the series of passing maneuvers described above, we are able to compare the outcomes of the forward simulation of policies with the actual real-world trajectories, for both our controlled car and the passed car. For this, we recorded the position of our vehicle (using its pose estimation system) and of the passed vehicle (using the dynamic object tracker) throughout the passing maneuvers.

Fig. 6 shows the prediction error for each policy as the root mean squared (RMS) difference between the simulated consequences of the policies and the actual trajectories of the vehicles involved. Results are shown for the first 5 s of the planning horizon averaged over all policy elections throughout all passing maneuvers, per timestep.

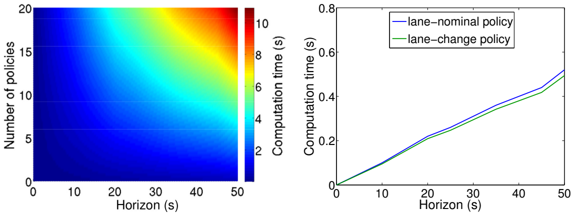
The effect of the delay in computing the prediction can be observed in that the errors are non-zero at $t = 0$ s. That is, the assumed world state has changed from its initial state used in the simulation. A disagreement between the simulated policy for the controlled vehicle and its actual trajectory can be observed, which manifests a difference between the simulated dynamics model and the vehicle's speed controller, particularly in the longitudinal axis. Regarding the prediction errors for the passed vehicle, they are particularly uneven when running the lane-change-right policy. This is the consequence of an impoverished performance of the dynamic object tracker as the passed vehicle is left far behind when completing the passing maneuver.

Despite these inaccuracies, however, our simulation engine's performance was sufficient for the decision-making process in completing the passing maneuvers.

C. Merging Scenario in Simulation

Fig. 5 demonstrates a simulated merging scenario to highlight how simulating forward other car policies allows MPDM to exploit the reactions of other cars. To illustrate the expressive power of our multi-vehicle policy model, we evaluated a policy that will attempt to merge into traffic, even if there is not a sufficient gap present. By anticipating that the reaction of car 2 would be to slow down and match the speed of a car driving into its path, the predicted outcome of the policy is feasible without being overly conservative. A traditional trajectory optimization algorithm (that does not consider the behavior of the other car in response to our vehicle) would likely not be able to execute this maneuver, since it would not be able to predict that a sufficiently large gap would become available. Because this is a simulation, we choose the policy parameters for the other three cars in traffic such that the other cars will avoid colliding with a vehicle in its path, and by using the same policy parameters, we are able to merge between cars 1 and 2 exactly as in the policy election simulation. Note that because all cars execute safe behaviors, including slowing or stopping for vehicles in our path, should car 2 accelerate into our path, any running policy will still avoid a collision. As a comparison, we can disable prediction of the expected reactions of other cars, in which our car waits more conservatively until both cars 2 and 3 pass before pulling into the lane.

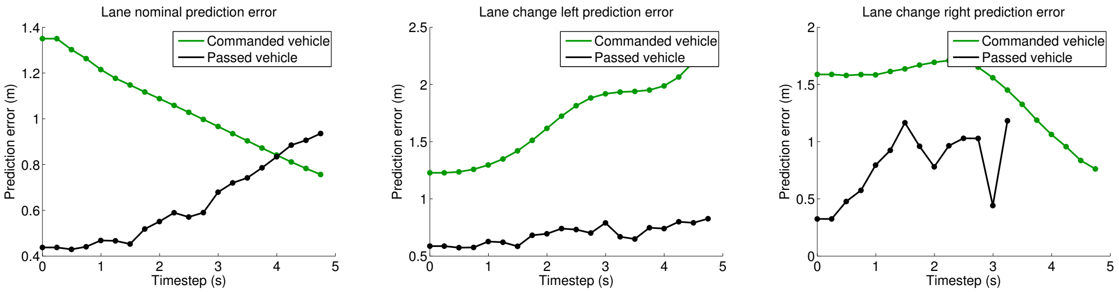
D. Performance



(a) (b)

Fig. 7. Computation time required by the policy election procedure as a function of the planning horizon and the number of policies sampled (a) and as a function of the planning horizon for two of the primary policies used in our experiments (b).

Fig. 7 shows the computation time required to sample policy outcomes and perform a policy election, as evaluated on a development laptop (2.8GHz Intel i7) with similar hardware to the cluster nodes operating on the vehicle. To



(a) (b) (c)

Fig. 6. Prediction errors for the commanded vehicle and the passed vehicle as given by forward simulation of the lane nominal, lane change left and lane change right policies.

obtain the time required as a function of the number of sampled policies, we run MPDM in simulated scenarios with an increasing number of policies (from 1 to 20). As expected, the computation time required by the policy election procedure grows linearly with the number of policies and with the planning horizon, given that the sampled policies demand similar computational requirements.

VII. CONCLUSIONS AND FURTHER WORK

In this paper, we have introduced a principled framework for decision-making within dynamic environments incorporating extensively coupled interactions between agents. By explicitly modeling reasonable behaviors of other vehicles in the environment, in the form of policies, we can make informed high-level behavioral decisions while more accurately accounting for the consequences of our actions. We demonstrated the feasibility of this approach in a passing scenario in a real-world environment, and further demonstrated MPDM can handle traffic cases like merging, in which most other approaches will fail to reasonably account for between-vehicle interactions.

In future work, we will integrate behavioral anticipation into the system to determine what is the most likely policy other cars are following in a principled manner, in order to allow for more complex vehicle interactions.

ACKNOWLEDGMENT

The authors are sincerely grateful to Wayne Williams, Rick Stephenson, and Johannes Strom from Ford Motor Co. for their help in performing the experiments of this work.

REFERENCES

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

[2] W. Xu, J. Wei, J. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. and Automation*, St. Paul, MN, May 2012, pp. 2061–2067.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 12, pp. 99 – 134, 1998.

[4] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[5] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and related stochastic optimization problems," *Artificial Intelligence*, vol. 147, no. 12, pp. 5 – 34, 2003.

[6] DARPA, "Darpa urban challenge," http://archive.darpa.mil/grandchallenge/, 2007.

[7] I. Miller *et al.*, "Team Cornell's Skynet: Robust perception and planning in an urban environment," *J. Field Robot.*, vol. 25, no. 8, pp. 493–527, 2008.

[8] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[9] D. Q. Tran and M. Diehl, "An application of sequential convex programming to time optimal trajectory planning for a car motion," in *Proc. IEEE Conf. Decision and Control*, Shanghai, China, Dec 2009, pp. 4366–4371.

[10] T. Gu and J. Dolan, "On-road motion planning for autonomous vehicles," in *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, C.-Y. Su, S. Rakheja, and H. Liu, Eds. Springer Berlin Heidelberg, 2012, vol. 7508, pp. 588–597.

[11] S. Thrun, "Monte carlo POMDPs," *Advances in Neural Information Processing Systems*, pp. 1064–1070, 2000.

[12] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robot.: Sci. & Syst. Conf.*, Zurich, Switzerland, 2008.

[13] S. Candido, J. Davidson, and S. Hutchinson, "Exploiting domain knowledge in planning for uncertain robot systems modeled as POMDPs," in *Proc. IEEE Int. Conf. Robot. and Automation*, Anchorage, Alaska, May 2010, pp. 3596–3603.

[14] R. He, E. Brunskill, and N. Roy, "Efficient planning under uncertainty with macro-actions," *J. Artificial Intell. Res.*, vol. 40, pp. 523–570, 2011.

[15] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 1772–1780.

[16] T. Bandyopadhyay, K. Won, E. Frazzoli, D. Hsu, W. Lee, and D. Rus, "Intention-aware motion planning," in *Proc. Int. Workshop on the Algorithmic Foundations of Robotics*, ser. Springer Tracts in Advanced Robotics, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds. Springer Berlin Heidelberg, 2013, vol. 86, pp. 475–491.

[17] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making," *IEEE Intell. Transp. Sys. Mag.*, vol. 3, no. 1, pp. 4–17, Spring 2011.

[18] J. Wei, J. Snider, T. Gu, J. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, Michigan, June 2014, pp. 458–464.

[19] J. Wei, J. Dolan, and B. Litkouhi, "Autonomous vehicle social behavior for highway entrance ramp management," in *Proc. IEEE Intell. Vehicles Symp.*, Gold Coast City, Australia, June 2013, pp. 201–207.

[20] P. Trautman, J. Ma, R. Murray, and A. Krause, "Robot navigation in dense human crowds: The case for cooperation," in *Proc. IEEE Int. Conf. Robot. and Automation*, Karlsruhe, Germany, May 2013, pp. 2153–2160.

[21] DARPA, "Route network definition file (rndf) and mission data file (mdf) formats," http://archive.darpa.mil/grandchallenge, 2007.

[22] A. Huang, E. Olson, and D. Moore, "LCM: Lightweight communications and marshalling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Taipei, Taiwan, Oct 2010, pp. 4057–4062.