MPDM: Multi-policy decision-making from autonomous driving to social robot navigation

Alex G. Cunningham*, Enric Galceran*, Dhanvin Mehta, Gonzalo Ferrer, Ryan M. Eustice and Edwin Olson

Abstract This chapter presents Multi-Policy Decision-Making (MPDM): a novel approach to navigating in dynamic multi-agent environments. Rather than planning the trajectory of the robot explicitly, the planning process selects one of a set of closed-loop behaviors whose utility can be predicted through forward simulation that capture the complex interactions between the actions of these agents. These polices capture different high-level behavior and intentions, such as driving along a lane, turning at an intersection, or following pedestrians. We present two different scenarios where MPDM has been applied successfully: An autonomous driving environment that models vehicle behavior for both our vehicle and nearby vehicles and a social environment, where multiple agents or pedestrians configure a dynamic environment for autonomous robot navigation. We present extensive validation for MPDM on both scenarios, using simulated and real-world experiments.

Alex G. Cunningham
Toyota Research Institute, 2311 Green Rd, Ann Arbor, MI, 48105, USA, e-mail: alex.cunningham@tri.global

Enric Galceran
Autonomous Systems Lab, Institute of Robotics and Intelligent Systems, ETH Zurich, Leonhardstrasse 21, Zurich 8092, Switzerland e-mail: enricg@ethz.ch

Dhanvin Mehta, Gonzalo Ferrer, Edwin Olson
Department of Computer Science and Engineering University of Michigan 2260 Hayward St, Ann Arbor, MI, 48109, USA e-mail: {dhanvinm,gferrerm,ebolson}@umich.edu

Ryan M. Eustice
Department of Naval Architecture and Marine Engineering University of Michigan 2600 Draper Dr, Ann Arbor, MI, 48109, USA e-mail: eustice@umich.edu

*Alex G. Cunningham and Enric Galceran have contributed equally to this work.

1 Introduction

Decision making in dynamic multi-agents environments is challenging due to the uncertainty associated with estimating and predicting future scenarios arising from the complex and tightly-coupled interactions between agents. Sensor noise, action execution uncertainty, tracking data association errors, etc. make this problem harder.

The robot's plan must consider the uncertainty on the continuous state of nearby agents and, especially, over their potential discrete intentions, such as turning at an intersection or changing lanes (Fig. 1). Our goal is to correctly handle uncertainty and prediction, while calculating a solution within a time budget for on-line execution.

Fig. 1 *Left*: Our multi-policy approach factors the actions of the egovehicle and traffic vehicles into a set of policies that capture common behaviors like lane following, lane changing, or turning. *Right*: Multipolicy applied to a different scenario, a social environment.

This chapter describes a planning framework called Multi-Policy Decision Making (MPDM), which poses planning as a discrete-valued decision problem over a library of hand-engineered closed-loop behavioral policies. We treat the underlying behavioral policies, roughly equivalent to control laws as black boxes whose outputs can be predicted using a forward simulation.

For any particular situation, our approach chooses the best policy on-line by simulating the likely future outcomes of each policy over a time horizon. The robot does not compute a nominal trajectory it simply "elects" a particular closed-loop behavior until the planning cycle runs again. Dynamically switching between candidate policies allows the robot to adapt to different situations that are likely to arise.

MPDM can be applied to a variety of domains, ranging from autonomous driving to mobile indoor robots. It is a powerful framework in that it allows relatively complex behaviors to be derived from a relatively small set of underlying policies.

This chapter is structured as follows: in the following section §2 we will present the formulation required to obtain the MPDM. The next two sections describe the authors application of MPDM to a number of real-world systems, including both how the basic formulation of MPDM was adapted from one setting to another and how the underlying policies were developed. In particular, an autonomous driving scenario §3 based on our ICRA [11] and RSS [25] papers, as well as the ongo-

ing journal under review [26]. The social environment §4 borrows from our IROS publication [48].

2 Problem Formulation

The POMDP model provides a mathematically rigorous formalization of the decision-making problem in dynamic, uncertain scenarios. We initially formulate this problem as a full POMDP which we then approximate by exploiting domain knowledge to reformulate the problem as a discrete decision over a small set of high-level policies for the robot.

Let V denote the set of agents near the robot including the robot. At time t, an agent $v \in V$ can take an action $a_t^v \in \mathrm{A}^v$ to transition from state $x_t^v \in \mathrm{X}^v$ to x_{t+1}^v. As a notational convenience, let $x_t \in \mathrm{X}$ include all state variables x_t^v for all agents at time t, and similarly let $a_t \in \mathrm{A}$ be the actions of all agents.

We model the agent dynamics with a conditional probability function capturing the dependence of the dynamics on the states and actions of all the agents in the neighborhood.

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

Similarly, we model observation uncertainty as

$$Z(x_t, z_t^v) = p(z_t^v | x_t), \tag{2}$$

where $z_t^v \in \mathrm{Z}^v$ is the observation made by agent v at time t, and $z_t \in \mathrm{Z}$ is the vector of all sensor observations made by all agents. These observations are provided by the perception module to the robot (see Fig. 2). For the rest of the agents considered during planning, we transform the observations into each agent's coordinate frame, considering the robot's state as an observation.

The robot's goal is to find an optimal policy π^* that maximizes the expected sum of rewards over a given decision horizon H, where a policy is a mapping $\pi : \mathrm{X} \times \mathrm{Z}^v \to \mathrm{A}^v$ that yields an action from the current MAP estimate of the state and an observation:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=t_0}^{H} R(x_t, \pi(x_t, z_t^v))\right], \tag{3}$$

where $R(\cdot)$ is a real-valued reward function $R : \mathrm{X} \times \mathrm{A} \to \mathbb{R}$. The evolution of $p(x_t)$ over time is governed by

$$p(x_{t+1}) = \iiint_{\mathrm{XZA}} p(x_{t+1}|x_t, a_t) p(a_t|x_t, z_t) p(z_t|x_t) p(x_t)\, da_t\, dz_t\, dx_t. \tag{4}$$

However, modeled agents can still react to nearby agents via z_t^v. Thus, the joint density for a single agent v can be written as

$$p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v) = p(x_{t+1}^v | x_t^v, a_t^v) p(a_t^v | x_t^v, z_t^v) p(z_t^v | x_t^v) p(x_t^v), \tag{5}$$

and assuming independent agent actions leads to

$$p(x_{t+1}) = \prod_{v \in V} \iiint_{\mathrm{X}^v \mathrm{Z}^v \mathrm{A}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v)\, da_t^v\, dz_t^v\, dx_t^v. \tag{6}$$

Despite the independence assumption, marginalizing over the large state, observation, and action spaces in Eq. 6 is still too expensive. A possible approximation to speed up the process, commonly used by general POMDP solvers [60, 2] is to solve Eq. 3 by drawing samples from $p(x_t)$. However, sampling over the full probability space with random walks yields a large number of low probability samples, such as those with agents not abiding by traffic rules. Our proposed approach samples more strategically from high likelihood scenarios to ensure computational tractability.

2.1 The Multi-Policy Approximation

The key idea we leverage is that, rather than plan nominal trajectories, we can think of behavior as emerging from choosing closed-loop policies. For instance, in indoor social environments, the robot can plan in terms of following or stopping. In the vast majority of traffic situations, traffic participants behave in a regular, predictable manner, following traffic rules. Typical behaviors that conform to these rules can greatly limit the action space to be considered and provides a natural way to capture closed loop interactions. Thus, we can structure the decision process to reason over a limited space of closed-loop policies for both the robot and other agents.

Closed-loop policies [1] allow approximation of agent dynamics including their interactions and observation models from §2 through deterministic, coupled forward simulation of multiple agents with their assigned policies. Therefore, we can evaluate the consequences of our decisions over available policies (for both the robot and other agents), without needing to evaluate for every control input of every agent.

This assumption does not preclude our system from handling situations where reaction time is key, as we engineer all policies to produce robot behavior that seeks safety at all times.

More formally, let Π be a discrete set of policies π_i, where each policy is a hand-engineered to capture a specific high-level behavior. The internal formulation of a given policy can include a variety of local planning and control algorithms. We will cover different design choices for policies in the sections below. The key requirement for policy execution is that it works under forward simulation, which allows for a very broad class of algorithms. Thus, the per-agent joint density from Eq. 5 can now be approximated in terms of π_t^v:

[1] In this paper, we use the term *closed-loop policies* to mean policies that react to the presence of other agents, in a coupled manner. The same concept applies to the term *closed-loop forward simulation*.

$$p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v) = p(x_t^v)p(z_t^v|x_t^v)p(x_{t+1}^v|x_t^v, a_t^v)p(\pi_t^v|x_t, \mathbf{z_{1:t}})p(a_t^v|x_t^v, z_t^v, \pi_t^v). \tag{7}$$

Finally, since we have full authority over the policy executed by our controlled car $q \in V$, we can separate our agent from the other agents in $p(x_{t+1})$ as follows, using the per-agent distributions of Eq. 7:

$$p(x_{t+1}) \approx \iint_{\mathrm{X}^q \mathrm{Z}^q} p^q(x_t^q, x_{t+1}^q, z_t^q, a_t^q, \pi_t^q)\, dz_t^q\, dx_t^q \prod_{v \in V | v \neq q} \left[\sum_{\Pi} \iint_{\mathrm{X}^v \mathrm{Z}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v)\, dz_t^v\, dx_t^v \right]. \tag{8}$$

We have thus far factored the action space from $p(x_{t+1})$ by assuming actions are given by the available policies.

However, Eq. 8 still requires integration over the state and observation spaces. We address this issue as follows. Given samples from $p(\pi_t^v|x_t, \mathbf{z_{0:t}})$ that assign a policy to each agent, we simulate forward both the robot and the other agents under their assigned policies to obtain sequences of predicted states and observations. These forward roll-outs incorporate interactions. We evaluate the expected sum of rewards using these sample rollouts over the entire decision horizon in a computationally feasible manner.

We simplify the full POMDP solution in our approximate algorithm by reducing the decision to a limited set of policies and performing evaluations with a single set of policy assignments for each sample. The overall algorithm acts as a single-stage Markov Decision Process MDP, which does remove some scenarios from consideration, but for sufficiently high level behaviors is not a major impediment to operation.

In addition, our approach approximates policy outcomes as deterministic functions of state, but because policies internally incorporate closed-loop control, the actual outcomes of policies are well-modeled by deterministic behavior. Even though we assume a deterministic transition model, we can incorporate uncertainty in terms of the state (see §4).

The policies used in this approach are still policies of the same form as in the POMDP literature, but under the constraint that the policy must be one of a predetermined policy set.

Algorithm 1 describes the essence of the MPDM approach, where the functions SIMULATEFORWARD and COMPUTEREWARD have been expressed as a generic call to adapt to domain dependent characteristics. In Fig. 2 is depicted an illustration of the MPDM algorithm in action.

In the remainder of the chapter, we present how we applied MPDM to two very different domains - autonomous driving and mobile indoor robots. We especially elaborate on the agent model we used for each - the handcrafted policies considered. The types of agents in the environment as well as the environmental constraints such as lanes in highways or walls/obstacles in indoor social settings affect the agent motion model used to make future predictions. The *agent motion model* governs the

Algorithm 1: Policy selection.

Input:
- Current MAP estimate of the joint state, $x_0 \in \mathrm{X}$.
- Set of available and applicable policies $\Pi' \subseteq \Pi$.
- Planning time horizon H.

$\mathrm{R} \leftarrow \emptyset$ // Rewards for each forward propagation
foreach $\pi \in \Pi'$ **do**
  $\Psi^{\pi} \leftarrow \textsc{SimulateForward}(x_0, \pi, H)$ // Ψ^{π} captures all agents
  $\mathrm{R} \leftarrow \mathrm{R} \cup \{(\pi, \textsc{ComputeReward}(\Psi^{\pi}))\}$
return $\pi^* \leftarrow \textsc{SelectBest}(\mathrm{R})$ // As described in Eq. 3

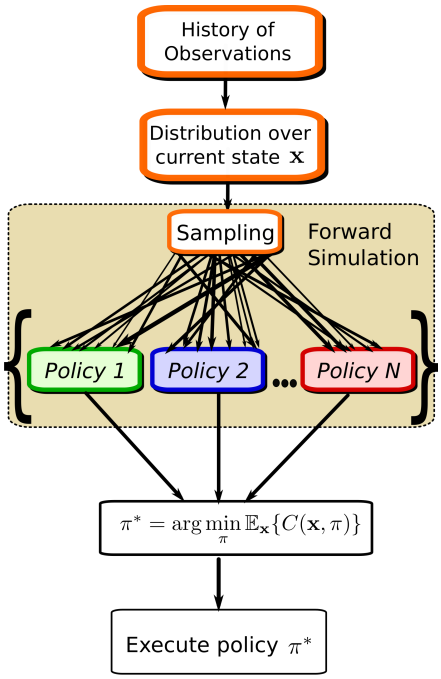


Fig. 2 The robot maintains a distribution over the state of each agent based on past observations. MPDM makes future predictions based on a motion model and the inferred states of all agents under consideration. For each policy π available and applicable to our robot, we simulate forward the system until the decision horizon H, which yields a set of simulated trajectories Ψ^{π}. We then evaluate the reward r_{π} for each rollout Ψ, and finally select the policy π^* maximizing the expected reward. The number of samples is domain-dependent, either drawn over the space of other vehicle policy assignments or over the space of the vehicle intial state (see §3 and §4). The process continuously repeats in a receding horizon manner. After one optimal policy π^* is chosen, then the system executes it.

state that needs to be inferred based on historical observations. For instance, vehicles on a highway, probably given by the structure above mentioned, have a limited set of actions that they typically do, and abnormal behaviors are easily detected. On the other hand, pedestrians on a urban environment present a more complex modeling challenge, since more diverse action are possible. A vehicle moving on an unstructured environment, such as a parking lot, or a non signaled area, result in more complex behaviors, which in turn require a more complex agent model.

3 Case Study 1: Autonomous Driving

The driving scenario is by design structured: in most traffic situations, vehicles behave in a regular, predictable manner, following traffic rules. We assume that such driving rules (lane rules, signals, etc.) determine the behavior of vehicles.

In our system, a state x_t^v is a tuple of the pose, velocity, and acceleration and an action a_t^v is a tuple of controls for steering, throttle, brake, shifter, and turn signals.

In this homogeneous environment, we can use the MPDM approach and assume that all vehicles are following a closed-loop policy at any given time - the car could be keeping to its lane, or changing lanes, or yielding. Thus, the policy takes on the additional role of a latent variable used to predict the future trajectories of the neighboring vehicles. The dynamics of a vehicle is determined by it's present policy as well as the policies followed by all neighboring vehicles. The egovehicle maintains a posterior distribution over the latent variable values (closed-loop policies) based on prior observations.

In our system, we consider V as all vehicles that are tracked by our LIDAR system (typically within 50m). An observation z_t^v, made by vehicle v, is a tuple including the observed poses and velocities of nearby vehicles and an occupancy grid of static obstacles. We consider only a limited field of view and do not account for observations that are far away from the robot.

Typically the position, velocity and acceleration of cars can be reliably tracked, but the uncertainty in the environment stems from the uncertainty about which closed-loop policy the other vehicles are following. We model uncertainty on the behavior of other agents with the following driver model:

$$D(x_t, z_t^v, a_t^v) = p(a_t^v | x_t, z_t^v), \tag{9}$$

where $a_t^v \in \mathrm{A}$ is estimated as a switching sequence policies. The driver model $D(x_t, z_t^v, a_t^v)$ implicitly assumes that the instantaneous actions of each vehicle are independent of each other.

Changepoint detection can be used to efficiently infer when a vehicle changes its policy. Thus, MPDM is used for behavioral anticipation of other agents, inferring policies and then integrating anticipation with policy selection as described in Alg. 1.

To carry out the evaluations we use the autonomous vehicle platform (Fig. 3) for data collection and active autonomous driving. Our vehicle, a drive-by-wire Ford Fusion, is equipped with a sensor suite including four Velodyne HDL-32E 3D LIDAR scanners, an Applanix POS-LV 420 inertial navigation system (INS), and GPS. 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 that capture information about the environment such as LIDAR reflectivity and road height, and are used for localization and tracking of other agents. The road network is encoded as a metric-topological map that provides information about the location and connectivity of road segments, and lanes therein.

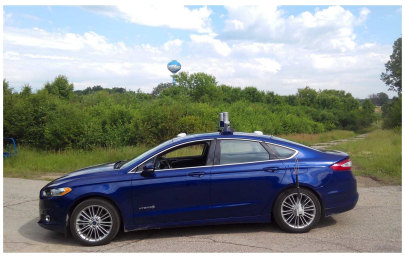


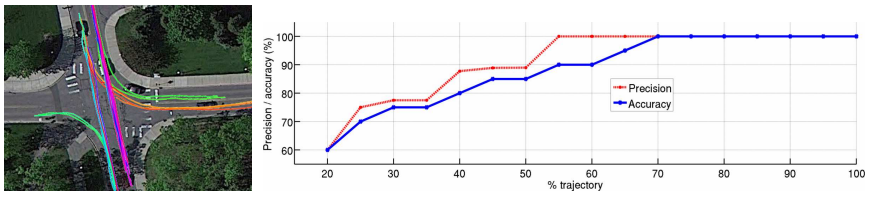
Fig. 3 The autonomous car platform. The vehicle is a Ford Fusion equipped with a sensor suite including four LIDAR units and survey-grade INS. All perception, planning, and control is performed onboard.

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

3.1 Behavior Anticipation

Given the history of the agents in environment, our goal is to estimate a distribution over their current policy assignments so that we can sample over possible other-vehicle policies during decision-making. The results presented in Fig. 4(b) indicate that we solve this in a robust manner.

The traffic-tracking dataset used to evaluate behavior anticipation consists of 67 dynamic object trajectories recorded in an urban area. Of these 67 trajectories, 18 correspond to "follow the lane" maneuvers and 20 to lane change maneuvers, recorded on a divided highway. The remaining 29 trajectories (shown in Fig. 4(a)) correspond to maneuvers observed at a four-way intersection regulated by stop signs. All trajectories were recorded by the dynamic object tracker onboard the vehicle and extracted from approximately 3.5 h of total tracking data.



(a) Collected dataset (b) Precision and accuracy curves

Fig. 4 A total of 29 trajectories in the traffic-tracking dataset used to evaluate our multi-policy framework, overlaid on satellite imagery (a). Precision and accuracy curves of current policy identification via changepoint detection, evaluated at increasing subsequences of the trajectories. Our method provides over 85% accuracy and precision after only 50% of trajectory completion, while the closed-loop nature of our policies produce vehicle behavior that seeks safety in a timely manner regardless of anticipation performance (b).

The performance of behavior anticipation can be observed in Fig. 4(b). In all experiments we use a C implementation of our system running on a single 2.8GHz Intel i7 laptop computer. For a deeper presentation of the changepoint detection method for behavioral anticipation, and experiments please refer to [25, 26], and more on occlusions in [27].

3.2 Results

We tested the full behavioral anticipation and decision-making system in both real-world and simulated highway traffic scenarios to demonstrate feasibility in a real vehicle environment and evaluate the effect of policy sampling strategies on decision results. The two-vehicle scenario we used is illustrated in Fig. 5, showing both our initial simulation of the test scenario and the real-world driving case.

In particular, this scenario highlights a case where identifying the behavior of another vehicle, in this case the second lane change of vehicle 2, causes the system to decide to initiate our lane change as soon as the it is clear the vehicle 2 is going to leave the lane. This extends our previous experimental results from [11], which demonstrated many trials of simple overtaking of a vehicle on a two-lane road assuming a single possible behavior for the passed vehicle.

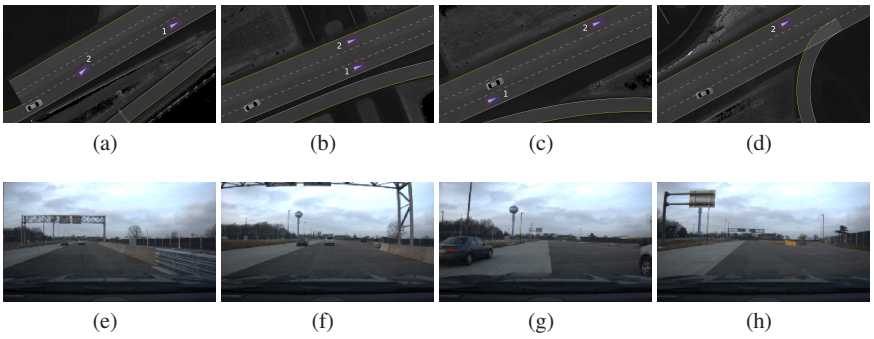


Fig. 5 Two-vehicle passing scenario executed in both simulation (top) and on our test vehicle, shown from the forward-facing camera. Note while the vehicles do not have the same timing in both cases, the structure of the scenario is the same in both. In this scenario, the egovehicle starts behind both traffic vehicles in the right lane of the three-lane road. The traffic vehicle 1 drives in the right lane along the length of the road, while traffic vehicle 2 makes two successive lane changes to the left. We remain in the right lane behind vehicle 1 until vehicle 2 initiates a lane change from the center to left lane, and at that point we make a lane change to the center lane. We pass both vehicles and return to the right lane.

In both real-world and simulated cases, we ran Algorithm 1 using a 0.25 s simulation step with a 10 s rollout horizon, with the same multi-threaded implementation of policy selection. The target execution rate for policy selection is 1 Hz, with a sep-

arate thread for executing the current policy running at 30 Hz. The process uses four threads for sample evaluation, and because the samples are independent, the speedup from multi-threading is roughly linear so long as all threads are kept busy. In this scenario, for both the egovehicle and the traffic vehicles, we used a pool of three policies that are representative of highway environments:

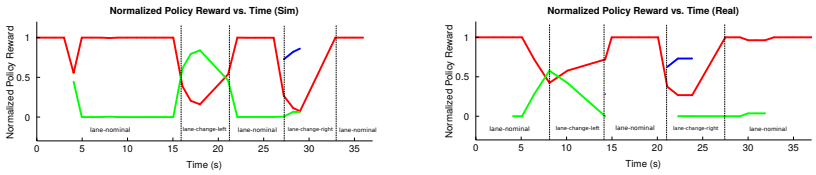
$$\Pi = \{\text{lane-nominal}, \text{lane-change-left}, \text{lane-change-right}\}.$$

In the context of autonomous cars, our typical metrics capture accomplishment of goals, safety, implementation of "soft" driving rules, and rider comfort.

We use a straightforward set of metrics in this scenario to compose the reward function with empirically tuned weights. The metrics used are as follows:

1. *Distance to goal*: scores how close the final rollout pose is to the goal.
2. *Lane bias*: penalizes being far from the right lane.
3. *Maximum yaw rate*: penalizes abrupt steering.
4. *Dead end distance*: penalizes staying in dead-end lanes depending on distance to the end.

These costs are designed to approximate the idealized value function that might come from a classical POMDP solution and to avoid biases due to heuristic cost functions.



(a) (b)

Fig. 6 These time-series plots show rewards for each policy (where policies *lane-nominal*, *lane-change-left* and *lane-change-right* are red, green and blue, respectively) is available to the egovehicle for both the simulated (top) and real-world version of the test scenario, with policy rewards normalized at each timestep. The dashed lines indicate the transitions between currently running policies based on the result of the elections. Discontinuities are due to a policy not being applicable, for reasons such as a vehicle blocking a lane change, or *lane-change-right* not being feasible from the right lane.

As can be seen through the policy reward trends in Fig. 6, there are clear decision points in which we choose to execute a new policy, which results in stable policy selection decisions. Discontinuities, such as the reward for *lane-change-right*, are expected as some policies are applicable less often, and in the middle of a maneuver such as a lane change, it is not possible that no policies can be initiated. In cases where a policy cannot be preempted until completed, such as lane-changes, another policy may have a higher reward but not induce policy switch due to concurrent policy execution and selection, such as in Fig. 6(b) at 10 s, where we continue

a lane-change even though *lane-nominal* has a locally higher reward. The reward in this case is higher because trajectory generation within the lane-change policy expects to start at a lane center, not while between lanes as during the lane change itself.

From the demonstrations in both simulation and real-world experiments, the policy selection process makes qualitatively reasonable decisions as expected given the reward metric weights. Further evaluation of the correctness of decisions made, however, will require larger-scale testing with real-world traffic in order to determine whether decisions made are statistically consistent with the median human driver.

4 Case Study 2: Social Environment

We use MPDM on an indoor and social environment, where our robot navigates among pedestrians. In this environment, outcomes are harder to predict, such as people suddenly stopping or changing directions, whereas in the driving scenario, other vehicles generate less unexpected events, and that allowed us to define a more complex and accurate model of vehicles.

The agent model must capture the dynamics of agents including reaction to other agents. At the same time, agents (humans) can instantaneously stop or change direction without signaling making it very difficult to predict future scenarios. The choice of model trades-off accuracy with computational efficiency. Complex models capturing interactions between groups of pedestrians and crowd dynamics may be more accurate but inferring parameters may be computationally expensive and may require observations that are not feasible under the provided sensor setup.

The key aspects of more unstructured environments is that inference over the agent's state is subject to a great inaccuracy. In order to apply the MPDM approach to its full potential, we propose a computationally light agent model and inference procedure. In consequence, the robot can re-plan frequently, which helps reduce the impact of this uncertainty. We use a simple reactive motion model. Each pedestrian in the vicinity treats all other agents as obstacles and uses a potential field based on the Social Force Model (SFM) [19, 33] to guide it towards its goal with a desired speed. The forward roll-outs capture the interactions between agents. For each pedestrian, the goal is not directly observable. It is assumed to be one of a small set of salient points and is estimated using a naive Bayes Classifier. The significant parameters of this model that typically contribute most to the uncertainty are the inferred goal and the preferred speed of the agent. Hence, MPDM maintains a distribution over the current state of each agent and samples initial configurations Each sample is then forward simulated and contributes towards the expected utility of the policy.

Dynamically switching between the candidate policies allows the robot to adapt to different situations. In the social environment, the set of most representative policies are:

$$\Pi = \{\text{Go-solo}, \text{Follow other agent}, \text{Stop}\}.$$

A robot executing the *Go-Solo* policy treats all other agents as obstacles and uses a potential field based on the Social Force Model (SFM) [19, 33] to guide it towards its goal.

In addition to the *Go-Solo* and the *Stop* policy, the robot can use the *Follow* policy to deal with certain situations. Our intuition is that in a crowd, the robot may choose to *Follow* another person sacrificing speed but delegating the task of finding a path to a human. *Following* could also be more suitable than overtaking a person in a cluttered scenario as it allows the robot to progress towards its goal without inconveniencing other pedestrians.

We show the benefits of switching between multiple policies in terms of navigation performance, quantified by metrics for progress made and inconvenience to fellow agents. We demonstrate the robustness of MPDM to measurement uncertainty (§4.1). Finally, we test the MPDM on a real environment and evaluate the results (§4.1.1).

Evaluating navigation behavior objectively is a challenging task and unfortunately, there are no standard metrics.

We propose three metrics that quantify different aspects of the emergent navigation behavior of the robot.

1. *Progress* (PG) - measures distance made good.
2. *Force* (F) - penalizes close encounters with other agents, calculated at each time step.
3. *Blame* (B) - penalizes velocity at the time of close encounters which is not captured by *Force*.

In order for the robot's emergent behavior to be socially acceptable, each policy's utility is estimated trading-off the distance traveled towards the goal (*Progress*) with the potential disturbance caused to fellow agents (*Force*).

4.1 Simulation

We simulate an indoor domain, freely traversed by a set of agents while the robot tries to reach a goal. We use the Intel i7 processor and 8GB RAM for our simulator and LCM [34] for inter-process communication.

We assume that the position of the robot, agents, the goal point, and obstacles are known in some local coordinate system. However, the accuracy of motion predictions is improved by knowing more about the structure of the building since the position of walls and obstacles can influence the behavior of other agents over the 3 second planning horizon. Our implementation achieves these through a global localization system with a known map, but our approach could be applied more generally.

The hallway domain (Fig. 7) is modeled on a $3m \times 25m$ hallway at the University of Michigan. The maximum permitted acceleration is $3m/s^2$ while the maximum

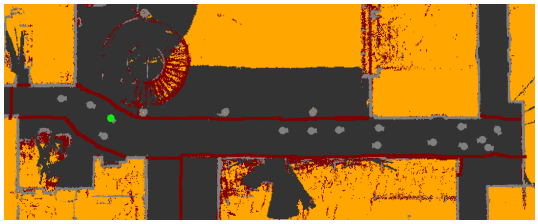


Fig. 7 The simulated indoor domain chosen to study our approach. The hallway domain where 15 agents are let loose with the robot and they patrol the hallway while the robot tries to reach its destination.

speed $|v|_{max}$ is set to $0.8m/s$. MPDM is carried out at 3Hz to match the frequency of the sensing pipeline for state estimation in the real-world experiment. The planning horizon is $3s$ into the future.

4.1.1 Experiments with Noise

For MPDM, the more accurate our model of the dynamic agents, the better is the accuracy of the predicted joint states. Most models of human motion, especially in complicated situations, fail to predict human behavior accurately. This motivates us to extensively test how robust our approach is to noisy environments.

In our simulator, the observations z are modeled using a stationary Gaussian distribution with uncorrelated variables for position, speed and orientation for the agent. We parameterize this uncertainty by a scale factor k_z: $\{\sigma_{p_x}, \sigma_{p_y}, \sigma_{|v|}, \sigma_\theta\} = k_z \times \{2cm, 2cm, 2cm/s, 3^\circ\}$. The corresponding diagonal covariance matrix is denoted by $\text{diag}(\sigma_{p_x}, \sigma_{p_y}, \sigma_{|v|}, \sigma_\theta)$. These uncertainties are propagated in the posterior estate estimation $P(\boldsymbol{x}|\boldsymbol{z})$.

The robot's estimator makes assumptions about the observation noise which may or may not match the noise injected by the simulator. This can lead to over and under-confidence which affects decision-making. In this section, we explore the robustness of the system in the presence of these types of errors. We define the assumed uncertainty by the estimator through a scale factor k_e, exactly as described above.

Varying the environment uncertainty k_z for a fixed level of estimator uncertainty k_e to understand how MPDM performs. We have studied the impact of different levels of environment uncertainty (k_z) at regular intervals of $\text{diag}(4cm, 4cm, 4cm/s, 6^\circ)$. The estimation uncertainty k_e is fixed at $\text{diag}(10cm, 10cm, 10cm/s, 15^\circ)$.

Fig. 8 shows the performance of the robot for the hallway domain. We observe that the *Blame* increases at a lowest rate for MPDM with the complete policy set. If the option of stopping is removed, we notice that the addition of the follow policy allows the robot to maintain comparable *Progress* while reducing the force and *Blame* associated. Given the option of stopping, the robot still benefits from the option of following as it can make more *Progress* while keeping *Blame* and *Force* lower.

We observe that MPDM allows the robot to maintain *Progress* towards the goal while exerting less *Force* and incurring less *Blame*. We also observe that the robot is more robust to noise in terms of *Blame* incurred (lesser rate of increase).

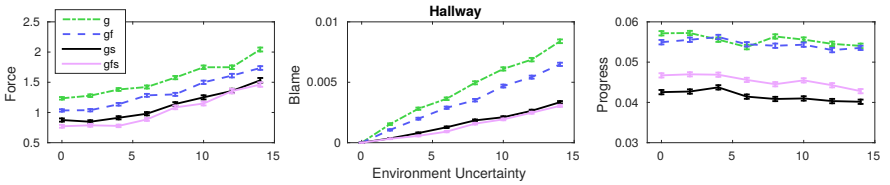
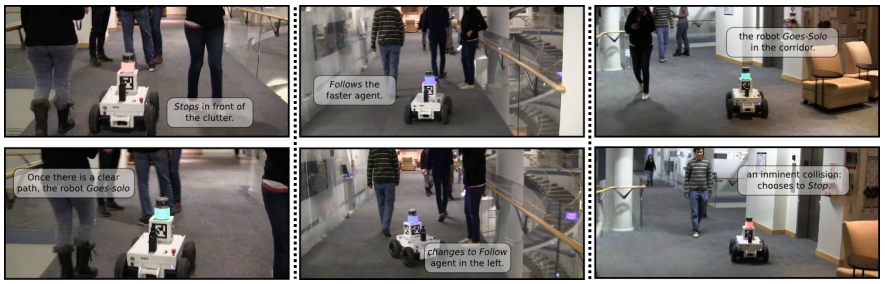


Fig. 8 Simulation results varying uncertainty in the environment (k_z) for a fixed posterior uncertainty (k_e). We show results for 4 combinations of the policies, varying the flexibility of MPDM: *Go-Solo* (g), *Go-Solo* and *Follow* (gf), *Go-Solo* and *Stop* (gs) and the full policy set (gfs). The data is averaged in groups of 10. We show the mean and standard error. *Left:* Increasing the noise in the environment makes the robot more susceptible to disturbing other agents and vice-versa. We can observe that the *Force* when combining all the policies (gfs) is much lower than when using a single policy (g) in the hallway domain. *Center:* A lower *Blame* indicates better behavior as the robot is less often the cause of inconvenience. The robustness of MPDM can be observed in milder slope across both domains. *Right:* Higher *Progress* is better. The *Go-Solo* performs better, however at the price of being much worse in *Force* and *Blame*. With more flexibility, (gfs) is able to achieve greater *Progress* and lower *Force* as compared to (gf).



(a) (b) (c)

Fig. 9 Real situations (a,b and c) illustrating the nature of the MPDM. On the top row is depicted some situations while testing the robot navigation in a real environment. On the bottom row are shown the same configurations, but delayed by a few seconds. The lights on the robot indicate the policy being executed, being green for *Go-Solo*, blue *Follow* and red *Stop*. By dynamically switching between policies, the robot can deal with a variety of situations.

4.2 Real World Experiments

Our real-world experiments have been carried out in the hallway that the simulated hallway domain was modeled on §4.1. We implemented our system on the MAGIC robot [52], a differential drive platform equipped with a Velodyne 16 laser scanner used for tracking and localization. An LED grid mounted on the head of the robot has been used to visually indicate the policy chosen at any time.

During two days of testing, a group of 8 volunteers was asked to patrol the hallway, given random initial and goal positions, similar to the experiments proposed in §4.1. The robot alternated between using MPDM and using the *Go-Solo* policy exclusively every five minutes. The performance metrics were recorded every second, constituting a total of 4.8k measurements.

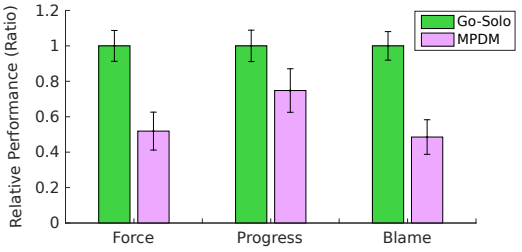


Fig. 10 The mean and standard error for the performance metrics over 10s intervals form real world experiments. All measures are normalized based on the corresponding mean value for the *Go-Solo* policy. MPDM shows much better *Force* and *Blame* costs than only *Go-Solo* at the price of slightly reducing its *Progress*.

Fig. 9 dipicts some of the challenging situations that our approach has tackled successfully. On the *Right* and *Left* scenes, the robot chooses to *Stop* avoiding the "freezing robot behavior" which would result in high values of *Blame* and *Force*. As soon as the dynamic obstacles are no longer an hindrance, the robot changes the policy to execute and *Goes-Solo*. In Fig. 9-*Center* we show an example of the robot executing the *Follow* policy, switching between leaders in order to avoid inconveniencing the person standing by the wall. The video provided[2] clearly shows the limitations of the *Go-Solo* and how MPDM solves these limitations.

Fig. 10 shows the results of MPDM compared to a constant navigation policy - *Go-Solo*. We show that our observations based on simulations hold in real environments. Specifically, MPDM performs much better, roughly 50%, in terms of *Force* and *Blame* while sacrificing roughly 30% in terms of *Progress*. This results in the more desirable behavior for navigation in social environments that is qualitatively evident in the video provided.

5 Related Work

5.1 Related Work on Behavioral Prediction

Despite the probabilistic nature of the anticipation problem, several methods in the literature assume no uncertainty on the future states of other participants [54, 51, 10]. Such an approach could be justified in a scenario where vehicles broadcast their intentions over some communications channel, but it is an unrealistic assumption otherwise.

Some approaches assume a dynamic model of the obstacle and propagate its state using standard filtering techniques such as the extended Kalman filter [24, 14]. Despite providing rigorous probabilistic estimates over an obstacle's future states, these methods often perform poorly when dealing with nonlinearities in the assumed dynamics model and the multimodalities induced by discrete decisions (e.g. contin-

[2] https://april.eecs.umich.edu/media/mehta2016iros.mp4

uing straight, merging, or passing). Some researchers have explored using GMM to account for nonlinearities and multiple discrete decisions [15, 31]; however, these approaches do not consider the history of previous states of the target object, assigning an equal likelihood to each discrete hypothesis and leading to a conservative estimate.

Dynamic Bayesian networks have been also utilized for behavioral anticipation [12]. In [28] is proposed a hierarchical dynamic Bayesian network where some of the models on the network are learned from observations using an EM approach.

A common anticipation strategy in autonomous driving used by, for example, [7], [16], or [30], consists of computing the possible goals of a target vehicle by planning from its standpoint, accounting for its current state. This strategy is similar to our factorization of potential driving behavior into a set of policies, but lacks closed-loop simulation of vehicle interactions.

GP regression has been utilized to learn typical motion patterns for classification and prediction of agent trajectories [64, 38, 36], particularly in autonomous driving [1, 61, 62]. In more recent work, [41] use inverse reinforcement learning to learn driving styles from trajectory demonstrations in terms of engineered features. They then use trajectory optimization to generate trajectories for their autonomous vehicle that resemble the learned driving styles. Nonetheless, these methods require the collection of training data to reflect the many possible motion patterns the system may encounter, which can be time-consuming. For instance, a lane change motion pattern learned in urban roads will not be representative of the same maneuver performed at higher speeds on the highway. In this paper we focus instead on hand-engineered policies.

5.2 Related Work on Decision-Making

Early instances of decision-making systems for autonomous vehicles capable of handling urban traffic situations stem from the 2007 DARPA Urban Challenge [13]. In that event, participants tackled decision-making using a variety of solutions ranging from FSM [50] and decision trees [49] to several heuristics [66]. However, these approaches were tailored for specific and simplified situations and were, even according to their authors, “not robust to a varied world” [66].

More recent approaches have addressed the decision-making problem for autonomous driving through the lens of trajectory optimization [17, 69, 70, 30].

However, these methods do not model the closed-loop interactions between vehicles, failing to reason about their potential outcomes.

Partially observable Markov decision processes (POMDP) offer a theoretically-grounded framework to incorporate these interactions in the planning process, however solvers [44, 55, 2] often have difficulty scaling computationally to real-world scenarios. The POMDP model provides a mathematically rigorous formalization of the decision-making problem in dynamic, uncertain scenarios such as autonomous driving. Unfortunately, finding an optimal solution to most POMDP is

intractable [53, 47]. A variety of general POMDP solvers exist in the literature that seek to approximate the solution [60, 44, 55, 2]. Although these methods typically require computation times on the order of several hours for problems with even small state, observation, and action spaces compared to real-world scenarios [9], there has been some recent progress that exploits GPU parallelization [45].

However, some researchers have proposed approximate solutions to the POMDP formulation to tackle decision-making in autonomous driving scenarios. [68] proposed a point-based MDP for single-lane driving and merging, and [65] applied a POMDP formulation to handle highway lane changes. An MDP formulation was employed by [5] for highway driving; similarly to our *policies*, they utilize *behaviors* that react to other objects. The POMDP approach of [3] considers partial observability of road users' intentions, while [6] solve a POMDP in continuous state space reasoning about potentially hidden objects and observation uncertainty, considering the interactions of road users.

The idea of assuming finite sets of policies to speed up planning has appeared previously [5, 32, 57, 4, 6]. Similarly, we propose to exploit domain knowledge from autonomous driving to design a set of policies that are readily available at planning time.

5.3 Related Work on Social Navigation

In a simulated environment, van den Berg *et al.* [67] proposed a multi-agent navigation technique using *velocity obstacles* that guarantees a collision-free solution assuming a fully-observable world. From the computer graphics community, Guy *et al.* [29] extended this work using *finite-time velocity obstacles* to provide a locally collision-free solution that was less conservative as compared to [67]. However, the main drawback of these methods is that they are sensitive to imperfect state estimates and make strong assumptions that may not hold in the real world.

Several approaches attempt to navigate in social environments by traversing a Potential Field (PF) [37] generated by a set of pedestrians [56, 59, 19]. Huang *et al.* [35] used visual information to build a PF to navigate. In the field of neuroscience, Helbing and Molnár [33] proposed the Social Force Model, a kind of PF approach that describes the interactions between pedestrians in motion.

Unfortunately, PF approaches have some limitations, such as local minima or oscillation under certain configurations [39]. These limitations can be overcome to a certain degree by using a global information plan to avoid local minima [8]. We use this same idea in our method by assuming that a global planner provides reachable goals, i.e., there is a straight line connection to those positions ensuring feasibility in the absence of other agents.

Inverse Reinforcement Learning-based approaches [71, 43, 46, 40] can provide good solutions by predicting social environments and planning through them. However, their effectiveness is limited by the training scenarios considered which might not be a representative set of the diverse situations that may arise in the real world.

An alternative approach looks for a pedestrian leader to follow, thus delegating the responsibility of finding a path to the leader, such as the works of [58, 42, 18]. In this work, *Follow* becomes one of the policies that the robot can choose to execute as an alternate policy to navigating.

Some approaches [23, 63, 20, 21] plan over the predicted trajectories of other agents. However predicting the behavior of pedestrians is challenging and the underlying planner must be robust to prediction errors.

POMDPs provide a principled approach to deal with uncertainty, but they quickly become intractable. Foka *et al.* [22] used POMDPs for robot navigation in museums. Cunningham *et al.* [11] show that, by introducing a number of approximations (in particular, constraining the policy to be one of a finite set of known policies), that the POMDP can be solved using MPDM. In their original paper, they use a small set of lane-changing policies; in this work, we explore an indoor setting in which the number and complexity of candidate policies is much higher. [48]

6 Conclusion

We have introduced a principled framework for decision-making in environments under uncertainty with extensively coupled interactions between agents as an approximate POMDP solver. By explicitly modeling reasonable behaviors of both our system and other agents' policies, we make informed high-level behavioral decisions that account for the consequences of our actions.

In this chapter we have also presented two cases, an autonomous car driving and a robot navigation. MPDM has been successfully applied to each of these very different cases by carefully taking different assumptions. In the case of autonomous driving, policies are low level maneuvers where complex interactions take place. We therefore predict intentions over a longer history as well as allow a longer planning budget. While navigating in indoor social environments, in order to compensate for the inaccuracies in the prediction model, we required a faster update of the policy selection, but tolerant to higher levels of uncertainty. As we have shown, this approach is feasible in real-world test cases and can be implemented online as it is required for autonomous driving and robot navigation in real environments.

Acknowledgments

This work was supported by a grant from Ford Motor Company via the Ford-UM Alliance under award N015392, DARPA YIP grant under award D13AP00059, CyberSEES grant award 1442773, and ARIA (TRI) grant award N021563.

References

1. Georges S. Aoude, Brandon D. Luders, Joshua M. Joseph, Nicholas Roy, and Jonathan P. How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35(1):51–76, 2013.
2. Haoyu Bai, David Hsu, and Wee Sun Lee. Integrated perception and planning in the continuous space: A POMDP approach. *International Journal of Robotics Research*, 33(9):1288–1302, 2014.
3. Tirthankar Bandyopadhyay, Chong Zhuang Jie, David Hsu, Marcelo H. Ang, Daniela Rus, and Emilio Frazzoli. *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, chapter Intention-Aware Pedestrian Avoidance, pages 963–977. Springer International Publishing, 2013.
4. Tirthankar Bandyopadhyay, KokSung Won, Emilio Frazzoli, David Hsu, WeeSun Lee, and Daniela Rus. Intention-aware motion planning. In Emilio Frazzoli, Tomas Lozano-Perez, Nicholas Roy, and Daniela Rus, editors, *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, volume 86 of *Springer Tracts in Advanced Robotics*, pages 475–491. Springer Berlin Heidelberg, 2013.
5. S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic MDP-behavior planning for cars. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1537–1542, 2011.
6. S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 392–399, 2014.
7. A. Broadhurst, S. Baker, and T. Kanade. Monte carlo road safety reasoning. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 319–324, Las Vegas, NV, USA, June 2005.
8. Oliver Brock and Oussama Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 341–346, 1999.
9. S. Candido, J. Davidson, and S. Hutchinson. Exploiting domain knowledge in planning for uncertain robot systems modeled as pomdps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3596–3603, Anchorage, AK, USA, May 2010.
10. J.S. Choi, Gyuho Eoh, Jimin Kim, Younghwan Yoon, Junghee Park, and B.-H. Lee. Analytic collision anticipation technology considering agents' future behavior. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1656–1661, Taipei, Taiwan, October 2010.
11. Alexander G. Cunningham, Enric Galceran, Ryan M. Eustice, and Edwin Olson. MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, WA, USA, May 2015.
12. Ismail Dagli, Michael Brost, and Gabi Breuel. *Agent Technologies, Infrastructures, Tools, and Applications for E-Services: NODe 2002 Agent-Related Workshops*, chapter Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks, pages 179–194. Springer Berlin Heidelberg, 2003.
13. DARPA. DARPA Urban Challenge. http://archive.darpa.mil/grandchallenge/, 2007.
14. N.E. Du Toit and J.W. Burdick. Robotic motion planning in dynamic, cluttered, uncertain environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 966–973, Anchorage, AK, USA, May 2010.
15. Noel E. Du Toit and Joel W. Burdick. Robot motion planning in dynamic, uncertain environments. 28(1):101–115, 2012.
16. Dave Ferguson, M. Darms, C. Urmson, and S. Kolski. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1149–1154, Eindhoven, Netherlands, June 2008.
17. Dave Ferguson, Thomas M. Howard, and Maxim Likhachev. Motion planning in urban environments. *Journal of Field Robotics*, 25(11-12):939–960, 2008.

18. G. Ferrer, A. Garrell, F. Herrero, and A. Sanfeliu. Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous Robots*, pages 1–19, 2016.
19. G. Ferrer, A. Garrell, and A. Sanfeliu. Social-aware robot navigation in urban environments. In *European Conference on Mobile Robotics*, pages 331–336, 2013.
20. G. Ferrer and A. Sanfeliu. Multi-objective cost-to-go functions on robot navigation in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3824–3829, 2015.
21. Gonzalo Ferrer. *Social robot navigation in urban dynamic environments*. PhD thesis, Universitat Politècnica de Catalunya, Spain, October, 2015.
22. AF Foka and PE Trahanias. Probabilistic Autonomous Robot Navigation in Dynamic Environments with Human Motion Prediction. *International Journal of Social Robotics*, 2(1):79–94, 2010.
23. C. Fulgenzi, A. Spalanzani, and C. Laugier. Probabilistic motion planning among moving obstacles following typical motion patterns. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4027–4033. IEEE, 2009.
24. C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1056–1062, Nice, France, September 2008.
25. Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Proceedings of the Robotics: Science & Systems Conference*, Rome, Italy, July 2015.
26. Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, page submitted, 2016.
27. Enric Galceran, Edwin Olson, and Ryan M. Eustice. Augmented vehicle tracking under occlusions for decision-making in autonomous driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3559–3565, Hamburg, Germany, October 2015.
28. T. Gindele, S. Brechtel, and R. Dillmann. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, pages 69–79, 2015.
29. Stephen J Guy, Jatin Chhugani, Changkyu Kim, Nadathur Satish, Ming Lin, Dinesh Manocha, and Pradeep Dubey. Clearpath: highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187. ACM, 2009.
30. J. Hardy and M. Campbell. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. 29(4):913–929, 2013.
31. F. Havlak and M. Campbell. Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. 30(2):461–474, 2014.
32. Ruijie He, Emma Brunskill, and Nicholas Roy. Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, 40:523–570, 2011.
33. D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.
34. Albert S Huang, Edwin Olson, and David C Moore. LCM: Lightweight communications and marshalling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4057–4062, 2010.
35. Wesley H Huang, Brett R Fajen, Jonathan R Fink, and William H Warren. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4):288–299, 2006.
36. Joshua Joseph, Finale Doshi-Velez, Albert S. Huang, and Nicholas Roy. A Bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383–400, 2011.
37. Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

38. Kihwan Kim, Dongryeol Lee, and I Essa. Gaussian process regression flow for analysis of motion trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1164–1171, Barcelona, Spain, November 2011.
39. Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1398–1404, 1991.
40. Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 2016.
41. M. Kuderer, S. Gulati, and W. Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2641–2646, 2015.
42. Markus Kuderer and Wolfram Burgard. An approach to socially compliant leader following for mobile robots. In *International Conference on Social Robotics*, pages 239–248. Springer, 2014.
43. Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proc. of Robotics: Science and Systems (RSS)*, 2012.
44. H. Kurniawati, D. Hsu, and W. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of the Robotics: Science & Systems Conference*, Zurich, Switzerland, June 2008.
45. Taekhee Lee and Young J. Kim. Massively parallel motion planning algorithms under uncertainty using POMDP. *International Journal of Robotics Research*, 35(8):928–942, 2016.
46. Matthias Luber, Luciano Spinello, Jens Silva, and Kai O Arras. Socially-aware robot navigation: A learning approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 902–907, 2012.
47. Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1–2):5–34, 2003.
48. Dhanvin Mehta, Gonzalo Ferrer, and Edwin Olson. Autonomous navigation in dynamic social environments using multi-policy decision making. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1190–1197, 2016.
49. Isaac Miller et al. Team Cornell's Skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, 25(8):493–527, 2008.
50. Michael Montemerlo et al. Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.
51. T. Ohki, Keiji Nagatani, and Kazuya Yoshida. Collision avoidance method for mobile robot considering motion and personal spaces of evacuees. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1819–1824, Taipei, Taiwan, October 2010.
52. Edwin Olson, Johannes Strom, Ryan Morton, Andrew Richardson, Pradeep Ranganathan, Robert Goeddel, Mihai Bulic, Jacob Crossman, and Bob Marinier. Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792, 2012.
53. Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.
54. S. Petti and T. Fraichard. Safe motion planning in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2210–2215, Edmonton, AB, Canada, August 2005.
55. David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2164–2172. Curran Associates, Inc., 2010.
56. Emrah Akin Sisbot, Luis Felipe Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.

57. Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. DESPOT: Online POMDP planning with regularization. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1772–1780. Curran Associates, Inc., 2013.
58. Procpio Stein, Anne Spalanzani, Vtor Santos, and Christian Laugier. Leader following: A study on classification and selection. *Robotics and Autonomous Systems*, 75, Part A:79 – 95, 2016.
59. Mikael Svenstrup, Thomas Bak, and Hans Jørgen Andersen. Trajectory planning for robots in dynamic human environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4293–4298, 2010.
60. S. Thrun. Monte Carlo POMDPs. *Proceedings of the Advances in Neural Information Processing Systems Conference*, pages 1064–1070, 2000.
61. Q. Tran and J. Firl. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 334–339, Gold Coast City, Australia, June 2013.
62. Quan Tran and J. Firl. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 918–923, Dearborn, MI, USA, June 2014.
63. Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.
64. Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803, Taipei, Taiwan, October 2010.
65. S. Ulbrich and M. Maurer. Probabilistic online pomdp decision making for lane changes in fully automated driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2063–2067, 2013.
66. Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William Red Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
67. Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. *Robotics Research, Springer Tracts in Advanced Robotics*, 70:3–19, 2011.
68. J. Wei, J. M. Dolan, J. M. Snider, and B. Litkouhi. A point-based MDP for robust single-lane autonomous driving behavior under uncertainties. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2586–2592, Shanghai, China, May 2011.
69. M. Werling, J. Ziegler, S. Kammel, and S. Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 987–993, Anchorage, AK, USA, May 2010.
70. Wenda Xu, Junqing Wei, J.M. Dolan, Huijing Zhao, and Hongbin Zha. A real-time motion planner with trajectory optimization for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2061–2067, Saint Paul, MN, USA, May 2012.
71. Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, James A Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936, 2009.