Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment

Journal Article

Author(s):
Galceran, Enric; Cunningham, Alexander G.; Eustice, Ryan M.; Olson, Edwin

Publication date:
2017-08

Permanent link:
https://doi.org/10.3929/ethz-b-000128913

Originally published in:
Autonomous Robots 41(6), https://doi.org/10.1007/s10514-017-9619-z

Auton Robot (2017) 41:1367–1382
DOI 10.1007/s10514-017-9619-z

Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment

Enric Galceran[1] · **Alexander G. Cunningham**[2] · **Ryan M. Eustice**[3] · **Edwin Olson**[4]

Received: 16 December 2015 / Accepted: 11 January 2017 / Published online: 9 February 2017

Abstract This paper reports on an integrated inference and decision-making approach for autonomous driving that models vehicle behavior for both our vehicle and nearby vehicles as a discrete set of closed-loop policies. Each policy captures a distinct high-level behavior and intention, such as driving along a lane or turning at an intersection. We first employ Bayesian changepoint detection on the observed history of nearby cars to estimate the distribution over potential policies that each nearby car might be executing. We then sample policy assignments from these distributions to obtain high-likelihood actions for each participating vehicle, and perform closed-loop forward simulation to predict the outcome for each sampled policy assignment. After evaluating these predicted outcomes, we execute the policy with the maximum expected reward value. We validate behavioral prediction and decision-making using simulated and real-world experiments.

Keywords Robotics · Autonomous driving

1 Introduction

Decision-making for autonomous driving is challenging due to uncertainty on the continuous state of nearby vehicles and, especially, over their potential discrete intentions, such as turning at an intersection or changing lanes (Fig. 1). The large state space of environments with many vehicles is computationally expensive to evaluate given the set of actions other vehicles can take.

Previous decision-making approaches have employed hand-tuned heuristics (Montemerlo 2008; Miller 2008; Urmson et al. 2008) and numerical optimization (Ferguson et al. 2008; Xu et al. 2012; Hardy and Campbell 2013), but these methods do not account for the coupled dynamic effects of interacting traffic agents. For example, a car abruptly initiating a passing maneuver might induce a preceding car to reconsider its decision to start passing.

Partially observable Markov decision processes (POMDP) offer a theoretically-grounded framework to capture these interactions, however solvers (Kurniawati et al. 2008; Silver and Veness 2010; Bai et al. 2014) often have difficulty scaling computationally to real-world scenarios. In addition, current approaches for anticipating future intentions of other traffic agents (Kim et al. 2011; Joseph et al. 2011; Aoude et al. 2013;

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Robotics Science and Systems".

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

✉ Enric Galceran
enricg@ethz.ch

Alexander G. Cunningham
alex.cunningham@tri.global

Ryan M. Eustice
eustice@umich.edu

Edwin Olson
ebolson@umich.edu

[1] Autonomous Systems Lab, Institute of Robotics and Intelligent Systems, ETH Zurich, Leonhardstrasse 21, Zurich 8092, Switzerland

[2] Toyota Research Institute, 2311 Green Rd, Ann Arbor, MI 48105, USA

[3] Department of Naval Architecture and Marine Engineering, University of Michigan, 2600 Draper Dr, Ann Arbor, MI 48109, USA

[4] Department of Computer Science and Engineering, University of Michigan, 2260 Hayward St. BBB 3737, Ann Arbor, MI 48109, USA

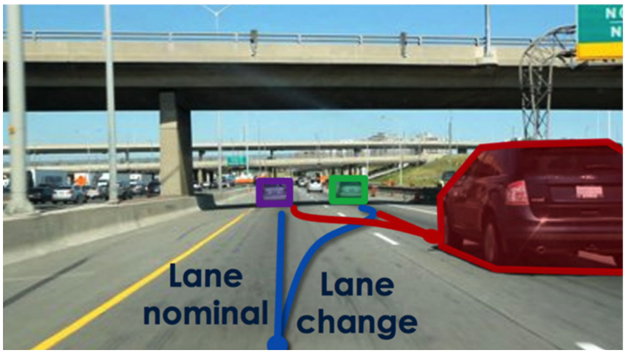


Fig. 1 Our multipolicy approach leverages the fact that not all possible actions of traffic participants are equally likely. Therefore, we can factor the actions of the egovehicle and traffic vehicles into a set of policies that capture common behaviors like lane following, lane changing, or turning. This way we can inform our action search to focus on the likely interactions of traffic agents

Havlak and Campbell 2014) either consider only the current state of a neighboring vehicle, ignoring the history of its past actions, or rather require onerous collection of training data.

In this paper, we present an integrated behavioral anticipation and decision-making system that models behavior for *both* the egovehicle and nearby vehicles as the result of closed-loop policies[1] applied to each. This approach is made tractable by considering only a finite set of *a priori* known policies (as illustrated in Fig. 1). Each policy is designed to capture a different high-level behavior, such as following a lane, changing lanes, or turning at an intersection. Our system proceeds in a sequence of two interleaved stages: behavioral prediction and policy selection. First, we leverage Bayesian changepoint detection to estimate the policy that a given vehicle was executing at each point in its history of actions, and then inferring the likelihood of each potential intention of the vehicle. Furthermore, we propose a statistical test based on changepoint detection to identify anomalous behavior of other vehicles, such as driving in the wrong direction or swerving out of lane. Therefore, we can detect when our policies fail to model observed behavior, and individual policies can therefore adjust their prescribed control actions to react to anomalous cars. Next, using the inferred distribution over policies for other vehicles, we select a policy to execute by sampling over policy assignments to the egovehicle and traffic vehicles and simulating forward to evaluate the outcomes of each policy decision. The reward function for a policy choice combines multiple user-defined metrics, and the final policy for the egovehicle maximizes the reward over all the sampled outcomes. The policy anticipation and selection procedure repeats in a receding horizon manner. As a result, our system is able to anticipate and exploit coupled interactions with other vehicles, allowing us to avoid overly-conservative decisions.

The central contributions of this paper are:

- A behavioral prediction approach that uses Bayesian changepoint detection to leverage the observed state history of vehicles to infer the likelihood of their possible future actions.
- A statistical test for detecting anomalous behavior online.
- A decision-making algorithm approximating the POMDP solution that evaluates the predicted outcomes of interactions between vehicles through forward simulation.
- An evaluation of the proposed system using real-world traffic data, and a traffic scenario in both simulation and on a real-world autonomous vehicle platform.

This work extends our earlier work which introduces and refines multipolicy decision-making. In our ICRA 2015 (Cunningham et al. 2015) paper, we introduced the multipolicy approach for decision-making, which we demonstrated in a real-world autonomous car under assumed known vehicle behaviors. We extended the approach to incorporate integrated prediction of other vehicle policies in our RSS 2015 (Galceran et al. 2015a) paper, though the full-system verification was limited to simulation. This paper presents new experimental results, including anticipation and decision-making on a real-world autonomous vehicle. Additionally, we have carefully extended the description of our approach to facilitate its implementation by other practitioners.

2 Related work

2.1 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 (Petti and Fraichard 2005; Ohki et al. 2010; Choi et al. 2010). 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 (Fulgenzi et al. 2008; Toit and Burdick 2010). 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. continuing straight, merging, or passing). Some researchers have explored using Gaussian mixture

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

model (GMMs) to account for nonlinearities and multiple discrete decisions (Toit and Burdick 2012; Havlak and Campbell 2014); 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 Bayeseian networks have been also utilized for behavioral anticipation (Dagli et al. 2003). Gindele et al. (2015) 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, Broadhurst et al. (2005), Ferguson et al. (2008), or Hardy and Campbell (2013), 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.

Gaussian process (GP) regression has been utilized to learn typical motion patterns for classification and prediction of agent trajectories (Trautman and Krause 2010; Kim et al. 2011; Joseph et al. 2011), particularly in autonomous driving (Aoude et al. 2013; Tran and Firl 2013, 2014). In more recent work, Kuderer et al. (2015) 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.

2.2 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 (DARPA 2007). In that event, participants tackled decision making using a variety of solutions ranging from finite state machine (FSMs) (Montemerlo 2008) and decision trees (Miller 2008) to several heuristics (Urmson et al. 2008). However, these approaches were tailored for specific and simplified situations and were, even according to their authors, “not robust to a varied world” (Urmson et al. 2008).

More recent approaches have addressed the decision making problem for autonomous driving through the lens of trajectory optimization (Ferguson et al. 2008; Werling et al. 2010; Xu et al. 2012; Hardy and Campbell 2013). However, these methods do not model the closed-loop interactions between vehicles, failing to reason about their potential outcomes.

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 POMDPs is intractable (Papadimitriou and Tsitsiklis 1987; Madani et al. 2003). A variety of general POMDP solvers exist in the literature that seek to approximate the solution (Thrun 2000; Kurniawati et al. 2008; Silver and Veness 2010; Bai et al. 2014). 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 (Candido et al. 2010), there has been some recent progress that exploits GPU parallelization (Lee and Kim 2016).

However, some researchers have proposed approximate solutions to the POMDP formulation to tackle decision-making in autonomous driving scenarios. Wei et al. (2011) proposed a point-based Markov decision process (MDP) for single-lane driving and merging, and Ulbrich and Maurer (2013) applied a POMDP formulation to handle highway lane changes. An MDP formulation was employed by Brechtel et al. (2011) for highway driving; similarly to our *policies*, they utilize *behaviors* that react to other objects. The POMDP approach of Bandyopadhyay et al. (2013a) considers partial observability of road users’ intentions, while Brechtel et al. (2014) 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 (Brechtel et al. 2011; He et al. 2011; Somani et al. 2013; Bandyopadhyay et al. 2013; Brechtel et al. 2014). Similarly, we propose to exploit domain knowledge from autonomous driving to design a set of policies that are readily available at planning time.

3 Problem formulation

As a decision problem, the goal is to choose egovehicle actions that maximize a reward function over time within a dynamic, uncertain environment with tightly coupled interactions between multiple agents. We initially formulate this problem as a full POMDP which we then approximate by exploiting driving domain knowledge to reformulate the problem as a discrete decision over a small set of high-level policies for the egovehicle.

Let V denote the set of vehicles near the egovehicle including the egovehicle. In our particular system, we consider all tracked vehicles within the range of our LIDAR sensors, 50m. At time t, a vehicle $v \in V$ can take an action $a_t^v \in \mathcal{A}^v$ to transition from state $x_t^v \in \mathcal{X}^v$ to x_{t+1}^v. 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. As a notational convenience, let $x_t \in \mathcal{X}$ include all state variables x_t^v for all vehicles at time t, and similarly let $a_t \in \mathcal{A}$ be the actions of all vehicles.

We model the vehicle dynamics with a conditional probability function

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

Similarly, we model observation uncertainty as

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

where $z_t^v \in \mathcal{Z}^v$ is the observation made by vehicle v at time t, and $z_t \in \mathcal{Z}$ is the vector of all sensor observations made by all vehicles. In our system, 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. These observations are provided by the perception module (see Sect. 4.1) to the egovehicle. For the rest of the vehicles considered during planning, transform the observations into each agent's coordinate frame, considering the egovehicle's state as an observation. In addition, our observation model considers the limited field of view of each agent, not being able to account for observations that are far away (beyond 50m). While in some recent work we have considered the effect of occlusions (Galceran et al. 2015b), we do not consider them in this paper. Further, we model uncertainty on the behavior of other agents with the following driver model:

$$D(x_t, z_t^v, a_t^v) = p(a_t^v|x_t, z_t^v), \tag{3}$$

where $a_t^v \in \mathcal{A}$ is a latent variable that must be inferred from sensor observations.

The egovehicle'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 $\pi : \mathcal{X} \times \mathcal{Z}^v \rightarrow \mathcal{A}^v$ that yields an action from the current maximum *a posteriori* (MAP) estimate of the state and an observation:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t=t_0}^{H} R(x_t, \pi(x_t, z_t^v))\right], \tag{4}$$

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

$$p(x_{t+1}) = \iiint\limits_{\mathcal{X}\,\mathcal{Z}\,\mathcal{A}} p(x_t)p(z_t|x_t)p(x_{t+1}|x_t, a_t) \\ p(a_t|x_t, z_t)\, da_t\, dz_t\, dx_t. \tag{5}$$

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

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

and the independence assumption finally leads to

$$p(x_{t+1}) = \prod_{v\in V} \iiint\limits_{\mathcal{X}^v\,\mathcal{Z}^v\,\mathcal{A}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v)\, da_t^v\, dz_t^v\, dx_t^v. \tag{7}$$

Despite assuming independent vehicle actions, marginalizing over the large state, observation, and action spaces in Eq. 7 is still too expensive. A possible approximation to speed up the process, commonly used by general POMDP solvers (Thrun 2000; Bai et al. 2014) is to solve Eq. 4 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.

4 Multipolicy approach

The key observation we leverage is that, in the vast majority of traffic situations, traffic participants behave in a regular, predictable manner, following traffic rules. Thus, we can structure the decision process to reason over a limited space of closed-loop policies for both the egovehicle and traffic vehicles. Closed-loop policies allow approximation of vehicle dynamics and observation models from Sect. 3 through deterministic, coupled forward simulation of all vehicles with their assigned policies. Therefore, we can evaluate the consequences of our decisions over available policies (for both our vehicle and other agents), without needing to evaluate for every control input of every vehicle.

This assumption does not preclude our system from handling situations where reaction time is key, as we engineer all policies to produce vehicle 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 driving mode. The internal formulation of a given policy can include a variety of local planning and control algorithms, but because our approach is invariant to the implementation-specific details, we will not cover them in this paper. The key requirement for policy execution is that it works under forward simulation, which allows for a very broad class of algorithms. In the most general formulation, let each policy π_i be parameterized by a parameter vector θ_i, and be a function of the current world state estimates x and an internal policy state s_i. The parameter vector θ_i can capture, for example, the "driving style" of the policy by regulating its acceleration profile to be more or less aggressive. The internal state s_i allows reuse of internal planning data, though for conciseness of notation, we will leave this implicit in policy formulations. Each policy has a state-dependent applicability check $\textsc{Applicable}(\pi_i, x)$ that determines whether the policy can start in the given state x. Applicability reduces the set of policies to evaluate by removing both illogical behaviors like parking while on a highway and unsafe behaviors like performing a lane change when traffic is not clear. We thus reduce the search in Eq. 4 to a limited set of policies.

By assuming each vehicle $v \in V$ is executing a policy $\pi_t^v \in \Pi$ at time t, the driver model for other agents in Eq. 3 can be now expressed as:

$$D(x_t, z_t^v, a_t^v, \pi_t^v) = p(a_t^v | x_t, z_t^v, \pi_t^v) p(\pi_t^v | x_t, \mathbf{z_{1:t}}), \tag{8}$$

where $p(\pi_t^v | x_t, \mathbf{z_{1:t}})$ is the probability that vehicle v is executing the policy π_t^v, which is conditioned on the current state x_t and prior observations $\mathbf{z_{1:t}}$ from our vehicle's standpoint. Inferring this probability is a key component of our approach, which we present in Sect. 5. Thus, the per-vehicle joint density from Eq. 6 can now be approximated in terms of π_t^v:

$$\begin{aligned} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v) &= p(x_t^v) p(z_t^v | x_t^v) p(x_{t+1}^v | x_t^v, a_t^v) \\ &\quad p(\pi_t^v | x_t, \mathbf{z_{1:t}}) p(a_t^v | x_t^v, z_t^v, \pi_t^v). \end{aligned} \tag{9}$$

Finally, since we have full authority over the policy executed by our controlled car $q \in V$, we can separate our vehicle from the other agents in $p(x_{t+1})$ as follows, using the per-vehicle distributions of Eq. 9:

$$\begin{aligned} p(x_{t+1}) \approx & \iint_{\mathcal{X}^q \mathcal{Z}^q} p^q(x_t^q, x_{t+1}^q, z_t^q, a_t^q, \pi_t^q)\, dz_t^q\, dx_t^q \\ & \prod_{v \in V | v \neq q} \left[\sum_{\Pi} \iint_{\mathcal{X}^v \mathcal{Z}^v} p^v(x_t^v, x_{t+1}^v, z_t^v, a_t^v, \pi_t^v)\, dz_t^v\, dx_t^v \right]. \end{aligned} \tag{10}$$

We have thus far factored the action space from $p(x_{t+1})$ by assuming actions are given by the available policies. However, Eq. 10 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 car, we simulate forward both the egovehicle and traffic vehicles under their assigned policies to obtain sequences of predicted states and observations. We evaluate the expected sum of rewards using these sample rollouts over the entire decision horizon in a computationally feasible manner.

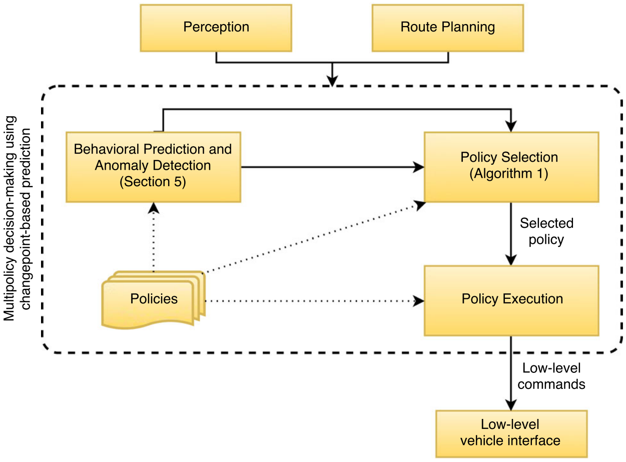

Perception
Route Planning
Multipolicy decision-making using changepoint-based prediction
Behavioral Prediction and Anomaly Detection (Section 5)
Policy Selection (Algorithm 1)
Selected policy
Policies
Policy Execution
Low-level commands
Low-level vehicle interface

Fig. 2 Multipolicy decision-making via changepoint-based prediction system diagram. The system takes as input a route to the user's desired destination and perceptual data (including localization and dynamic object tracks), and outputs low-level control commands (e.g., forward speed and steering wheel angle) to the vehicle. The key components of our approach are the behavioral anticipation and anomaly detection module, described in Sect. 5, and the policy selection algorithm described in Sect. 6

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 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. 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 pre-determined policy set.

4.1 System design

Figure 2 illustrates the complete system, where the key components are a behavioral anticipation and anomaly detection module (Sect. 5), our policy selection algorithm (Sect. 6), and a set of policies capturing different driving behaviors. The system takes as input a high-level route plan (in the form of driving directions from start to the user's desired destination) and perception, and continuously outputs low-level control commands (in our case, nominal forward speed and steering wheel angle) to the vehicle platform. In particular, the per-

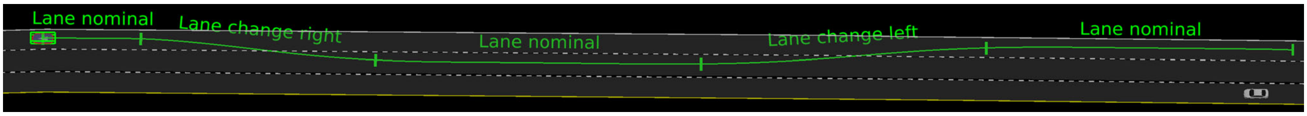


Fig. 3 Policy changepoint detection on a simulated passing maneuver on a highway. Our vehicle (*far right*) tracks the behavior of another traffic agent (*far left*) as it navigates through the highway segment from *right* to *left*. Using the tracked vehicle's history of past observations for the last 30s (*green curve*), we are able to infer which policies are most likely to have generated the maneuvers of the tracked vehicle

ception module provides the egovehicle's pose and velocity through localization, and Gaussian tracks of the poses and velocities of other agents within the sensor field of view of the egovehicle.

In this work we use a set of hand-engineered policies that covers many in-lane and intersection driving situations, comprising the following policies: *lane-nominal*, drive in the current lane and maintain distance to the car directly in front; *lane-change-right*/*lane-change-left*, separate policies for a single lane change in each direction; and *turn-right*, *turn-left*, *go-straight*, or *yield* at an intersection. Of course, this set of policies can be easily extended to handle more driving requirements.

In practice, each policy is implemented as a computer program with planning and control loops that use a suitable choice of algorithms for the policy's task (like parking or changing lanes). At all times the policy execution block in Fig. 2 runs the currently selected policy to generate control actions at rates (on the order of 30 to 50 Hz) suitable for smooth and safety-critical vehicle control at all times. This policy execution module also only allows valid policy transitions, and that policies in the middle of maneuvers (such as lane changes) are not preempted. This design choice minimizes the need for a complex centralized error handling module, since failure cases are handled on a per-policy basis. In parallel to policy execution and at a lower rate (on the order of 1 Hz), our policy selection algorithm (Sect. 6) evaluates which policy we should execute at the current time. Low-level control of the vehicle is not constrained by the decision-making process, since the lower-level controls are continuously prescribed by the current policy.

From a software architecture standpoint, the multipolicy approach provides an inherent modularization that allows to reuse the same code (the policies) for behavior prediction, for decision-making via forward simulation, and for low-level planning and control for the vehicle.

5 Behavioral prediction and anomaly detection via changepoint detection

This section describes how we infer the probability of the policies executed by other cars and their parameters. Our behavioral anticipation method segments the history (i.e., time-series data) of observed states of each vehicle, where each segment is associated with the policy most likely to have generated the observations in the segment. We obtain this segmentation using Bayesian changepoint detection, which infers the points in the recent history of observations of the state of other vehicles where the underlying policy generating the observations changes, as illustrated by the simulation in Fig. 3. In our system the perception module provides Gaussian estimates of the pose and velocity of other vehicles, to which we apply a sliding window to keep the the most recent n seconds (we use $n = 30$ in our system) of observations of the state of each vehicle. Thus, we can compute the likelihood of all available policies for each tracked car given the observations in the most recent segment, capturing the distribution $p(\pi_t^v | x_t, \mathbf{z_{1:t}})$ over the car's potential policies at the current timestep. This yields a probability distribution of the policies that a tracked car might execute in the near future, allowing our system to draw samples from this distribution and evaluate them through forward simulation in time. Further, full history segmentation allows us to detect anomalous behavior that is not explained by the set of policies in our system. We next describe the anticipation method for a single vehicle, which we then apply successively to all nearby vehicles.

5.1 Changepoint detection

To segment a tracked car's history of observed states, we adopt the recently proposed Changepoint detection using Approximate Model Parameters (CHAMP) algorithm by Niekum et al. (2014, 2015), which builds upon the work of Fearnhead and Liu (2007). Given the set of available policies Π and a time series of the observed states of a given vehicle $\mathbf{z_{1:n}} = (z_1, z_2, \ldots, z_n)$, CHAMP infers the MAP set of times $\tau_1, \tau_2, \ldots, \tau_m$, at which changepoints between policies have occurred, yielding $m + 1$ segments. Thus, the i^{th} segment consists of observations $\mathbf{z_{\tau_i+1:\tau_{i+1}}}$ and has an associated policy $\pi_i \in \Pi$ with parameters θ_i.

The changepoint positions are modeled as a Markov chain where the transition probabilities are a function of the time since the last changepoint:

$$p(\tau_{i+1} = t | \tau_i = s) = g(t - s), \tag{11}$$

where $g(\cdot)$ is the pdf of a prior distribution over segment length, and $G(\cdot)$ denotes its cdf. Specifically, CHAMP employs a truncated Gaussian as a prior over segment length:

$$g(t) = \frac{\frac{1}{\sigma}\phi(\frac{t-\mu}{\sigma})}{1-\Phi(\frac{\alpha-\mu}{\sigma})} \tag{12}$$

$$G(t) = \Phi(\frac{t-\mu}{\sigma}) - \Phi(\frac{\alpha-\mu}{\sigma}), \tag{13}$$

where ϕ is the standard normal pdf, Φ is its cdf, and α is the minimum segment length.

Given a segment from time s to t and a policy π, the policy evidence for that segment is defined as:

$$L(s,t,\pi) = p(\mathbf{z_{s+1:t}}|\pi) = \int p(\mathbf{z_{s+1:t}}|\pi,\theta)p(\theta)\,d\theta. \tag{14}$$

To avoid marginalizing over parameters, CHAMP approximates the logarithm of the policy evidence for that segment via the BIC (Bishop 2007) as:

$$\log L(s,t,\pi) \approx \log p(\mathbf{z_{s+1:t}}|\pi,\hat{\theta}) - \frac{1}{2}k_\pi \log(t-s), \tag{15}$$

where k_π is the number of parameters of policy π and $\hat{\theta}$ are estimated parameters for policy π. The BIC is a well-known approximation that avoids marginalizing over the model (policy, in our case) parameters and provides a principled penalty against complex policies by assuming a Gaussian posterior around the estimated parameters $\hat{\theta}$. Thus, only the ability to fit policies to the observed data is required, which can be achieved via maximum likelihood estimation (MLE) (described in Sect. 5.2).

As shown by Fearnhead and Liu (2007), the distribution C_t over the position of the first changepoint before time t can be estimated efficiently using standard Bayesian filtering and an online Viterbi algorithm. Defining

$$P_t(j,\pi) = p(C_t = j, \pi, \mathcal{E}_j, \mathbf{z_{1:t}}) \tag{16}$$

$$P_t^{\mathrm{MAP}} = p(\text{Changepoint at } t, \mathcal{E}_t, \mathbf{z_{1:t}}), \tag{17}$$

where $\mathcal{E}_j$ is the event that the MAP choice of changepoints has occurred prior to a given changepoint at time j, results in:

$$P_t(j,\pi) = (1 - G(t-j-1))L(j,t,\pi)p(\pi)P_j^{\mathrm{MAP}} \tag{18}$$

$$P_t^{\mathrm{MAP}} = \max_{j,\pi}\left[\frac{g(t-j)}{1-G(t-j-1)}P_t(j,\pi)\right]. \tag{19}$$

At any time, the most likely sequence of latent policies (called the Viterbi path) that results in the sequence of observations can be recovered, recursively, by finding the timestep and policy pair (j,π) that maximize P_t^{MAP}, and then repeating the maximization for P_j^{MAP} successively until time zero is reached. Further details on this changepoint detection method are provided by Niekum et al. (2014, 2015).

5.2 Behavioral prediction

In contrast to other anticipation approaches in the literature that consider only the current state of the target vehicle and assign equal likelihood to all its potential intentions (Ferguson et al. 2008; Hardy and Campbell 2013; Havlak and Campbell 2014), here we compute the likelihood of each latent policy by leveraging changepoint detection on the history of observed vehicle states.

Given the segmented history of observations of a given vehicle obtained via changepoint detection, consider the $(m+1)^{\text{th}}$ segment (the most recent), consisting of observations $\mathbf{z_{\tau_m+1:n}}$. The likelihood and parameters of each latent policy $\pi \in \Pi$ for the target vehicle given the present segment can be computed by fitting the policy models as follows:

$$\forall\pi \in \Pi, \qquad \mathcal{L}(\pi) = \operatorname*{argmax}_{\theta} \log p(\mathbf{z_{\tau_m+1:n}}|\pi,\theta). \tag{20}$$

Specifically, we assume $p(\mathbf{z_{\tau_m+1:n}}|\pi,\theta)$ to be a multivariate Gaussian with mean at the trajectory $\psi^{\pi,\theta}$ obtained by simulating forward in time the execution of policy π under parameters θ from timestep $\tau_m + 1$:

$$p(\mathbf{z_{\tau_m+1:n}}|\pi,\theta) = \mathcal{N}(\mathbf{z_{\tau_m+1:n}}; \psi^{\pi,\theta}, \sigma I), \tag{21}$$

where σ is a hyperparameter set by hand capturing modeling error and I is a suitable identity matrix (we discuss our forward simulation of policies further in Sect. 6.2). That is, Eq. 21 essentially *measures the deviation of the observed states from those in the trajectory prescribed by the given policy*. The same model fitting procedure is employed for computing Eq. 15 during changepoint detection.

The policy likelihoods obtained via Eq. 20 capture the probability distribution over the possible policies that the observed vehicle might be executing at the current timestep, which can be represented, using delta functions, as a mixture distribution:

$$p(\pi_t^v|x_t, \mathbf{z_{0:t}}) = \eta \sum_{i=1}^{|\Pi|} \delta(\alpha_i)\cdot\mathcal{L}(\pi_i), \tag{22}$$

where α_i is the hypothesis over policy π_i and η is a normalizing constant. We can therefore compute the approximated posterior of Eq. 10 by sampling from this distribution for each vehicle, obtaining high-likelihood samples from the coupled interactions of traffic agents.

5.3 Anomaly detection

The time-series segmentation obtained via changepoint detection allows us to perform online detection of anomalous behavior not modeled by our policies. Inspired by prior work on anomaly detection (Piciarelli and Foresti 2006; Chandola et al. 2009; Kim et al. 2011), we first define the properties of anomalous behavior in terms of policy likelihoods, and then compare the observed data against labeled normal patterns in previously-recorded vehicle trajectories. Thus, we define the following two criteria for anomalous behavior:

1. *No likely available policies* Anomalous behavior is not likely to be explained by any of the available policies as we design policies to abide by traffic rules and provide a smooth riding experience. Therefore, behaviors like driving in the wrong direction or crossing a solid line on the highway will not be captured by the available policies. We thus measure the average likelihood among all segments in the vehicle's history as the global similarity of the observed history to all available policies:

$$\mathcal{S} = \frac{1}{m+1} \sum_{i=1}^{m+1} \mathcal{L}(\pi_i), \tag{23}$$

   where π_i is the policy associated with the i^{th} segment.
2. *Ambiguity among policies* A history segmentation that fluctuates frequently among different policies might be a sign of ambiguity on the segmentation. To express this criterion formally, we first construct a histogram capturing the occurrences of each policy in the vehicle's segmented history. A histogram with a broad spread indicates frequent fluctuation, whereas one with a single mode is more likely to correspond to normal behavior. We measure this characteristic as the excess kurtosis of the histogram, $\kappa = \frac{\mu_4}{\sigma^4} - 3$, where μ_4 is the fourth moment of the mean and σ is the standard deviation. The excess kurtosis satisfies $-2 < \kappa < \infty$. If $\kappa = 0$, the histogram resembles a normal distribution, whereas if $\kappa < 0$, the histogram presents a broader spread. That is, we seek to identify changepoint sequences where there is no dominant policy.

Using these criteria, we define the following normality measure given a vehicle's MAP choice of changepoints:

$$N = \frac{1}{2} \left[(\kappa + 2)\mathcal{S} \right]. \tag{24}$$

This normality measure on the target car's history can then be compared to that of a set of previously recorded trajectories of other vehicles. We thus define the normality test for the current vehicle's history as $N < 0.5\gamma$, where γ is the minimum normality measure evaluated on the prior time-series.

6 Multipolicy decision-making

The policy selection algorithm for our car (Algorithm 1), implements the formulation and approximations given in Sects. 3 and 4 by leveraging the anticipation scheme from Sect. 5. The algorithm begins by drawing a set of samples $s \in S$ from the distribution over policies of other cars via Eq. 22, where each sample assigns a policy $\pi^v \in \Pi$ to each nearby vehicle v, excluding our car. For each policy π available to our car and for each sample s, we simulate forward all vehicles under policy assignments (π, s) until the decision horizon H, which yields a set Ψ of simulated trajectories ψ. We then evaluate the reward $r_{\pi,s}$ for each rollout Ψ, and finally select the policy π^* maximizing the expected reward. The process continuously repeats in a receding horizon manner. Note that policies that are not applicable given the current state x_0, such as an intersection handling policy when driving on the highway, are not considered for selection (line 5).

Algorithm 1: Policy selection.

```
Input:
  • Current MAP estimate of the state, $x_0$.
  • Set of available policies $\Pi$.
  • Policy assignment probabilities (Eq. 22).
  • Planning horizon $H$.
1 Draw a set of samples $s \in S$ via Eq. 22, where each sample
  assigns a policy to each nearby vehicle. $\mathcal{R} \leftarrow \emptyset$ // Rewards
  for each rollout
2 foreach $\pi \in \Pi$ do // Policies for our car
3    foreach $s \in S$ do // Policies for other cars
4       if APPLICABLE($\pi, x_0$) then
5          $\Psi^{\pi,s} \leftarrow$ SIMULATEFORWARD($x_0, \pi, s, H$)
           // $\Psi^{\pi,s}$ captures all vehicles
6          $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\pi, s, $ COMPUTEREWARD$(\Psi^{\pi,s}))\}$
7 return $\pi^* \leftarrow$ SELECTBEST($\mathcal{R}$)
```

6.1 Accounting for multiple possible route plans

The destination objectives of each vehicle considered in our decision making approach, including the ego-vehicle, are captured by route plans, which consist of a set of driving directions that a vehicle must follow to reach its destination, similarly to the driving directions given by GPS navigation devices. The route plan for the ego-vehicle is given as input to the decision making system, from an external module that computes the driving directions necessary to reach the destination desired by the user.

For other traffic participants, we extract possible partial route plans deterministically from a prior road network map, with driving directions covering until the decision horizon.

6.2 Sample rollout via forward simulation

While high-fidelity vehicle simulation techniques exist, in practice (Cunningham et al. 2015), a lower-fidelity simulation can capture the necessary interactions between vehicles to make reasonable choices for egovehicle behavior, while providing faster performance. Our simulation model for each vehicle assumes an idealized steering controller, but nonetheless, this simplification still faithfully describes the high-level behavior of the between-vehicle interactions. We simulate traffic vehicles classified as anomalous using a single policy accounting only for their current state and local obstacles, since they are not likely to be modeled by the set of behaviors in our system. Note that policies selected for all vehicles and remain constant from the start of the sample rollout, which prevents the approach from anticipating policy changes of traffic vehicles. As a partial solution for this problem, we allow policies to internally switch to other policies, for instance switching from a lane-change to a lane-nominal policy upon completion.

6.3 Reward function

The reward function for evaluating the outcome of a rollout Ψ involving all vehicles is a weighted combination of metrics $m_q(\cdot) \in \mathcal{M}$, with weights w_q that express user importance. The construction of a reward function based on a flexible set of metrics derives from our previous work (Cunningham et al. 2015), which we extend here to handle multiple potential policies for other vehicles. Typical metrics include measures of accomplishment (distance to goal), safety (minimum distance to obstacles) and passenger comfort (maximum yaw rate). For a full policy assignment (π, s) with rollout $\Psi^{\pi,s}$, we compute the rollout reward $r_{\pi,s}$ as the weighted sum $r_{\pi,s} = \sum_{q=1}^{|\mathcal{M}|} w_q m_q(\Psi^{\pi,s})$. We normalize each $m_q(\Psi_{\pi,s})$ to the interval [0, 1] across all rollouts to ensure comparability between metrics of different units. Because normalization can amplify insignificant variations in metric values, we set the weight w_q to zero when the range of $m_q(\cdot)$ across all samples is too small to be informative.

We finally evaluate each egovehicle policy reward r_π as the expected reward over all rollout rewards $r_{\pi,s}$, computed as $r_\pi = \sum_{k=1}^{|S|} r_{\pi,s_k} p(s_k)$, where $p(s_k)$ is the joint probability of the policy assignments in sample s_k, computed as a product of the per-vehicle assignment probabilities (Eq. 22). We use expected reward to target better average-case performance, as it is easy to become overly conservative if one only accounts for worst-case behavior.

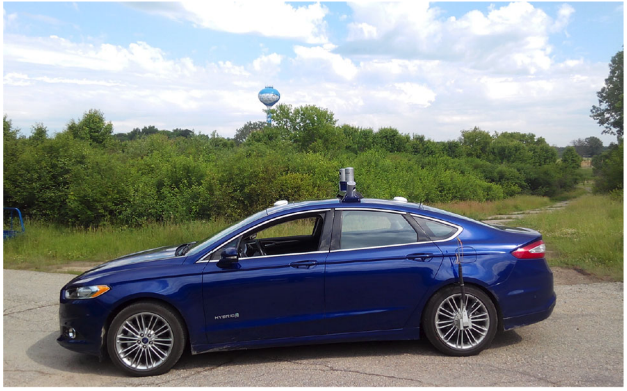


Fig. 4 Our autonomous car platform. The vehicle is a Ford Fusion equipped with a sensor suite including four LIDAR units and survey-grade INS. All perception, planning, and control is performed onboard

7 Experimental evaluation

We evaluate our approach in two parts: the behavioral anticipation method by itself and then the integrated anticipation and policy selection approach. Both evaluations use the same instrumented autonomous vehicle platform (Fig. 4) for data collection and active autonomous driving. To evaluate our behavior prediction and anomaly detection method we use traffic-tracking data collected using our autonomous vehicle platform, which we describe below. Finally, we evaluate our multipolicy approach performing integrated behavioral analysis and decision-making on highway traffic scenarios.

7.1 Autonomous vehicle platform, dataset, and setup

Our autonomous vehicle platform (Fig. 4) for data collection and autonomous testing consists of a drive-by-wire Ford Fusion 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.

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.

The traffic-tracking dataset used to evaluate behavior anticipation consists of 67 dynamic object trajectories

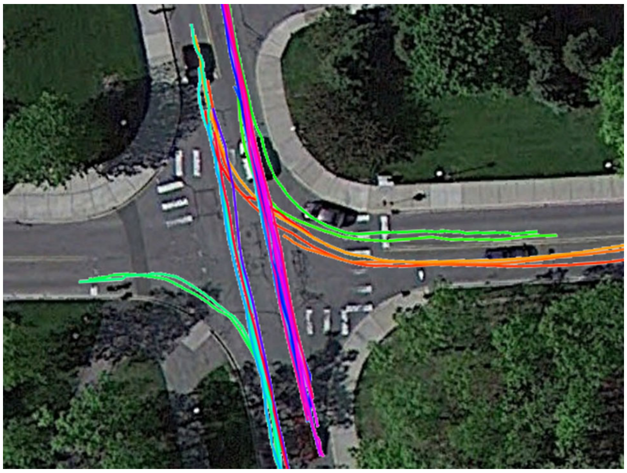


Fig. 5 29 trajectories in the traffic-tracking dataset used to evaluate our multipolicy framework, overlaid on satellite imagery

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. 5) 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 3.5h of total tracking data.

In all experiments we use a C implementation of our system running on a single 2.8GHz Intel i7 laptop computer.

7.2 Behavioral prediction

For our system, we are interested in correctly identifying the behavior of target vehicles by associating it to the most likely policy according to the observations. Thus, we evaluate our behavioral analysis method in the context of a classification problem, where we want to map each trajectory to the underlying policy (class) that is generating it at the current timestep. The available policies used in this evaluation are:

$$\begin{aligned}\Pi = \{&\text{lane-nominal, lane-change-left, lane-change-right}\}\\ &\cup\\ &\{\text{turn-right, turn-left, go-straight, yield}\},\end{aligned}\tag{25}$$

where the first subset applies to in-lane maneuvers and the second subset applies to intersection maneuvers. For all policies we use a fixed set of empirically-tuned parameters including maximum longitudinal and lateral accelerations, and allowed distances to nearby cars.

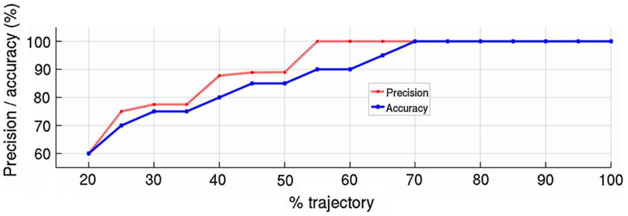


Fig. 6 Precision and accuracy curves of current policy identification via changepoint detection, evaluated at increasing subsequences of the trajectories. Our method provides over 85% accuracy and precision after only 50% of trajectory completion, while the closed-loop nature of our policies produce vehicle behavior that seeks safety in a timely manner regardless of anticipation performance

To assess each classification as correct or incorrect, we leverage the road network map and compare the final lane where the trajectory actually ends to that predicted by the declared policy. In addition, we assess behavioral prediction performance on subsequences of incremental duration of the input trajectory, measuring classification performance on increasingly longer observation sequences. Classification performance is measured in terms of *precision* and *accuracy*, defined as usual in terms of total positives P, total negatives N, true positives TP, and false positives FP as follows:

- precision or positive predictive value (PPV),

  $$PPV = TP/(TP + FP),$$

- accuracy (ACC),

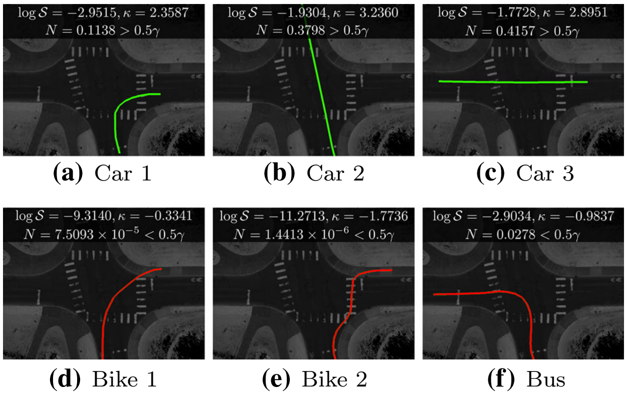
  $$ACC = (TP + TN)/(P + N).$$

Figure 6 shows the accuracy and precision curves for policy classification over the entire dataset. The ambiguity among hypotheses results in poor performance when only an early stage of the trajectories is used, especially under 30% completion. However, we are able to classify the trajectories with over 85% accuracy and precision after only 50% of the trajectory has been completed. Note, however, that the closed-loop nature of our policies allows us to produce vehicle behavior that seeks safety at all times regardless of anticipation performance.

A qualitative evaluation of our behavioral analysis and prediction method is shown in Fig. 3, where we run changepoint detection on a simulated passing maneuver executed by a tracked vehicle on a three-lane highway. This simulation allows us to evaluate the method independently of the potential tracking errors present in the real-world traffic tracking dataset. As shown in Fig. 3, we are able to correctly segment the passing maneuver into the available policies (Eq. 25).

7.3 Anomaly detection

We now qualitatively explore the performance of our anomaly detection test. We recorded three additional trajectories corresponding to two bicycles and a bus. The bikes crossed the



(a) Car 1 **(b)** Car 2 **(c)** Car 3

(d) Bike 1 **(e)** Bike 2 **(f)** Bus

Fig. 7 Anomaly detection examples. *Top row* normal trajectories driven by cars from the intersection dataset. *Bottom row* anomalous trajectories driven by bikes (**d**), (**e**), and a bus (**f**). Our test is able to correctly detect the anomalous trajectories not modeled by our intersection policies ($\gamma = 0.1233$)

intersection from the sidewalk, while the bus made a significantly wide turn. We run the test on these trajectories and on three additional intersection trajectories using the minimum normality value on the intersection portion of the dataset, $\gamma = 0.1233$. As shown by the results in Fig. 7, our test is able to correctly detect the anomalous behaviors not modeled in our system.

7.4 Decision-making via changepoint-based prediction results

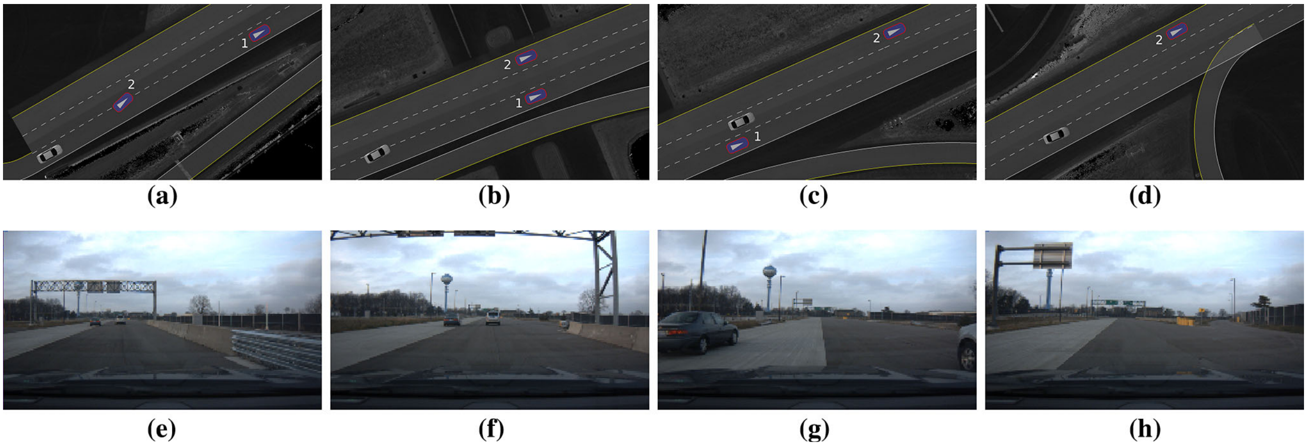
We tested the full behavioral anticipation and decision-making system in both real-world and simulated environments to demonstrate feasibility in a real vehicle environment and evaluate the effect of policy sampling strategies on decision results. The two-vehicle scenario we used is illustrated in Fig. 8, 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 Cunningham et al. (2015), which demonstrated many trials of simple overtaking of a vehicle on a two-lane road assuming a single possible behavior for the passed vehicle.

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 separate 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:

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

7.4.1 Evaluating policy rewards

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:



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

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

Fig. 8 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*

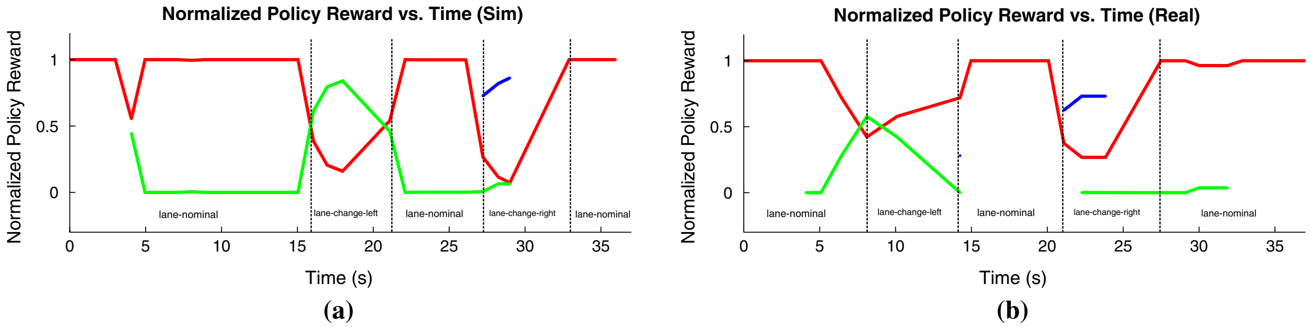


Fig. 9 These time-series plots show rewards for each policy (where policies *lane-nominal*, *lane-change-left* and *lane-change-right* are red, green and blue, respectively) is available to the egovehicle for both the simulated (*top*) and real-world version of the test scenario, with policy rewards 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*

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. As can be seen through the policy reward trends in Fig. 9, 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 (Sect. 4.1), such as in Fig. 9(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.

7.4.2 Sampling computational performance

In addition to showing that decision process makes reasonable decisions, we evaluate the computational performance of the sampling process and investigate strategies for maintaining real-time operation.

We target a rate of 1 Hz for policy selection, though in practice though there are occasional spikes in the number of samples when there are more applicable policies available. The median time to evaluate a sample on the compute platform used in our vehicle platform is 270 ms, with a maximum time of 686 ms during the test scenario. We note that if policy selection takes longer than our target time, the currently running policy still timely produces vehicle behavior that seeks safety, so computational delays only result in suboptimal decision timing.

At peak, exhaustive sampling of all permutations all feasible policy assignments uses 12 samples in the simulated scenario, with a median of 5 samples. Adding more cars and policies would increase the computational cost further, so to further scale the system we need to limit how many samples we evaluate. The sampling strategies each choose a subset of samples from the exhaustive evaluation, so we can evaluate the procedure by post-processing the logged scenarios and recomputing policy rewards.

- *Most likely*: Only highest probability policy assignment to traffic vehicles.
- *Most likely + ambiguous*: Evaluates additional samples if there are multiple assignments of near-equal probability.
- *N best assignments*: Bound the number of assignments (set to 3).
- *N best samples*: Bound total number of samples (set to 6).

These approaches trade off between fidelity in approximating the full policy reward distribution and the computation cost. Choosing the most likely assignment to evaluate results in only needing to evaluate a maximum of three samples, which is a lower bound with three active policies. Increasing the sample set to only include ambiguous assignments increases the maximum number of samples to four. For both *N best assignments* and *N best samples*, the maximum number of samples is six, which is expected given the parameters.

There is a trade-off in sampling strategy selection where using fewer samples produces a less accurate approximation of the policy reward, but is computationally cheaper. Because the computed rewards will be different if fewer samples are included, we need to determine if the decision made by the system actually changes with sampling strategy. We use the exhaustive sampling approach as the standard for correct decisions, and for each sampling strategy we count the number of discrepancies. Within the simulated scenario results, using only the most likely sample yields incorrect decisions 3% but expanding the number of samples to account for ambiguity makes no errors. The *N best assignments* and *N best samples* approaches both are incorrect for 6% of results, which can be expected as both strategies under-sample when ambiguity is present.

8 Limitations and further work

While we have demonstrated our approach in proof-of-concept scenarios, there remain several limitations to the current methodology that motivate our future work.

8.1 Limitations

We have shown that at the moment our system is able to detect anomalous behavior of traffic participants. While this information could be used to, for example, make a conservative decision, such as slow down, in the presence of an anomaly, we have not implemented such capability yet. Such decisions could be implemented within each individual policy, as well as part of the policy selection algorithm.

While each policy in our system has collision avoidance built in and will therefore produce vehicle behavior that seeks safety, our decision making approach does not provide strict safety guarantees when dealing with unexpected events that are far ahead in the future. Exploring safety guarantees for our system is an avenue for further work.

While our decision-making approach can choose between a set of possible policies, it does not yet allow for either the egovehicle or traffic vehicles to switch policies after initial policy assignment. As such, our approach does not solve a full MDP in this formulation, a problem we intend to address in future work through branching searches through policies.

Our decision-making method currently takes the relatively simple approach of reasoning over all cars that are observable by the egovehicle, but this can result in computational limitations as the number of policy assignments increase. With the computational timing from (Sect. 7.4.2) we can perform exhaustive sampling on 4 vehicles, and switch to other sampling approaches when there is more traffic. This could be improved by more a more selective choice of nearby agents likely to interact with the egovehicle to consider.

8.2 Further work

In future work we plan to explore explicitly modeling unexpected behavior, such as the appearance of a pedestrian or vehicles occluded by large objects. Further, additional metrics for anomaly detection we wish to explore in future work, beyond segment likelihood and excess kurtosis, include mean segment length. Exploring principled methods for reacting to detected anomalous behavior is also an avenue for future work.

9 Conclusion

We introduced a principled framework for integrated behavioral anticipation and decision-making in environments with extensively coupled interactions between agents as an approximate POMDP solver. By explicitly modeling reasonable behaviors of both our vehicle and other vehicles as policies, we make informed high-level behavioral decisions that account for the consequences of our actions. As we have shown, this approach is feasible in real-world test cases.

We presented a behavior analysis and anticipation system based on Bayesian changepoint detection that infers the likelihood of policies of other vehicles. Furthermore, we provided a normality test to detect unexpected behavior of other traffic participants. We have shown that our behavioral anticipation approach can identify the most-likely underlying policies that explain the observed behavior of other cars, and to detect anomalous behavior not modeled by the policies in our system.

Acknowledgements This work was supported in part by a grant from Ford Motor Company via the Ford-UM Alliance under award N015392 and in part by DARPA under award D13AP00059. The authors are sincerely grateful to Patrick Carmody, Ryan Wolcott, Steve Vozar, Jeff Walls, Gonzalo Ferrer, and Igor Gilitschenski for help collecting experimental data and for valuable comments.

References

Aoude, G. S., Luders, B. D., Joseph, J. M., Roy, N., & How, J. P. (2013). Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, *35*(1), 51–76.

Bai, H., Hsu, D., & Lee, W. S. (2014). Integrated perception and planning in the continuous space: A POMDP approach. *International Journal of Robotics Research*, *33*(9), 1288–1302.

Bandyopadhyay, T., Jie, C. Z., Hsu, D., Ang, M. H., Rus, D., & Frazzoli, E. (2013a). In *Experimental robotics: The 13th international symposium on experimental robotics* (pp. 963–977). Springer International Publishing, chap Intention-Aware Pedestrian Avoidance.

Bandyopadhyay, T., Won, K., Frazzoli, E., Hsu, D., Lee, W., & Rus, D. (2013b). Intention-aware motion planning. In E. Frazzoli, T. Lozano-Perez, N. Roy, & D. Rus (Eds.), *Proceedings of the international workshop on the algorithmic foundations of robotics, Springer tracts in advanced robotics* (Vol. 86, pp. 475–491). Berlin: Springer.

Bishop, C. M. (2007). *Pattern recognition and machine learning. Information science and statistics*. Berlin: Springer.

Brechtel, S., Gindele, T., & Dillmann, R. (2011). Probabilistic MDP-behavior planning for cars. In *Proceedings of the IEEE intelligent transportation systems conference* (pp. 1537–1542).

Brechtel, S., Gindele, T., & Dillmann, R. (2014). Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In *Proceedings of the IEEE intelligent transportation systems conference* (pp. 392–399).

Broadhurst, A., Baker, S., & Kanade, T. (2005). Monte carlo road safety reasoning. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 319–324). Las Vegas, NV: IEEE.

Candido, S., Davidson, J., & Hutchinson, S. (2010). Exploiting domain knowledge in planning for uncertain robot systems modeled as pomdps. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 3596–3603). Anchorage, AK: IEEE.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 15.

Choi, J., Eoh, G., Kim, J., Yoon, Y., Park, J., & Lee, B. H. (2010). Analytic collision anticipation technology considering agents' future behavior. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1656–1661). Taipei, Taiwan: IEEE.

Cunningham, A. G., Galceran, E., Eustice, R. M., & Olson, E. (2015). MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In *Proceedings of the IEEE international conference on robotics and automation*, Seattle, WA.

Dagli, I., Brost, M., & Breuel, G. (2003). Agent technologies, infrastructures, tools, and applications for E-Services: NODe 2002 agent-related workshops. Springer Berlin Heidelberg, Chap Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks, pp. 179–194.

DARPA (2007) DARPA Urban Challenge. http://archive.darpa.mil/grandchallenge/

Du Toit, N., & Burdick, J. (2010). Robotic motion planning in dynamic, cluttered, uncertain environments. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 966–973). Anchorage, AK: IEEE.

Du Toit, N. E., & Burdick, J. W. (2012). Robot motion planning in dynamic, uncertain environments. *IEEE Transactions on Robotics*, *28*(1), 101–115.

Fearnhead, P., & Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *69*(4), 589–605.

Ferguson, D., Darms, M., Urmson, C., & Kolski, S. (2008a). Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 1149–1154). Eindhoven, Netherlands: IEEE.

Ferguson, D., Howard, T. M., & Likhachev, M. (2008b). Motion planning in urban environments. *Journal of Field Robotics*, *25*(11–12), 939–960.

Fulgenzi, C., Tay, C., Spalanzani, A., & Laugier, C. (2008). 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* (pp. 1056–1062). Nice, France: IEEE.

Galceran, E., Cunningham, A.G., Eustice, R.M., & Olson, E. (2015a). Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Proceedings of the robotics: science & systems conference*. Rome, Italy: Robotics: Science and Systems Foundation.

Galceran, E., Olson, E., & Eustice, R. M. (2015b). Augmented vehicle tracking under occlusions for decision-making in autonomous driving. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 3559–3565). Hamburg, Germany: IEEE.

Gindele, T., Brechtel, S., & Dillmann, R. (2015). Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, *7*, 69–79.

Hardy, J., & Campbell, M. (2013). Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. *IEEE Transactions on Robotics*, *29*(4), 913–929.

Havlak, F., & Campbell, M. (2014). Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. *IEEE Transactions on Robotics*, *30*(2), 461–474.

He, R., Brunskill, E., & Roy, N. (2011). Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, *40*, 523–570.

Joseph, J., Doshi-Velez, F., Huang, A. S., & Roy, N. (2011). A Bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, *31*(4), 383–400.

Kim, K., Lee, D., & Essa, I. (2011). Gaussian process regression flow for analysis of motion trajectories. In *Proceedings of the IEEE international conference on computer vision* (pp. 1164–1171). Barcelona, Spain: IEEE.

Kuderer, M., Gulati, S., & Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. In *Proceedings of the IEEE international conference on robotics and automation* (pp 2641–2646). Seattle, WA: IEEE.

Kurniawati, H., Hsu, D., & Lee, W. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of the robotics: Science & systems conference*. Zurich, Switzerland: IEEE.

Lee, T., & Kim, Y. J. (2016). Massively parallel motion planning algorithms under uncertainty using POMDP. *International Journal of Robotics Research*, *35*(8), 928–942.

Madani, O., Hanks, S., & Condon, A. (2003). On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, *147*(1–2), 5–34.

Miller, I., et al. (2008). Team Cornell's Skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, *25*(8), 493–527.

Montemerlo, M., et al. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, *25*(9), 569–597.

Niekum, S., Osentoski, S., Atkeson, C. G., & Barto, A. G. (2014). CHAMP: Changepoint detection using approximate model parameters. Tech. Rep. CMU-RI-TR-14-10, Robotics Institute, Carnegie Mellon University.

Niekum, S., Osentoski, S., Atkeson, C. G., & Barto, A. G. (2015). Online bayesian changepoint detection for articulated motion models. In

Proceedings of the IEEE international conference on robotics and automation. Seattle, WA: IEEE.

Ohki, T., Nagatani, K., & Yoshida, K. (2010). 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* (pp. 1819–1824). Taipei, Taiwan: IEEE.

Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. *Mathematics of Operations Research*, *12*(3), 441–450.

Petti, S., & Fraichard, T. (2005). Safe motion planning in dynamic environments. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2210–2215). Edmonton, AB: IEEE.

Piciarelli, C., & Foresti, G. (2006). On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, *27*(15), 1835–1842.

Silver, D., & Veness, J. (2010). Monte-carlo planning in large POMDPs. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems 23* (pp. 2164–2172). Red Hook, NY: Curran Associates Inc.

Somani, A., Ye, N., Hsu, D., & Lee, W. S. (2013). DESPOT: Online POMDP planning with regularization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in neural information processing systems 26* (pp. 1772–1780). Red Hook, NY: Curran Associates Inc.

Thrun, S. (2000). Monte Carlo POMDPs. In *Proceedings of the advances in neural information processing systems Conference* (pp 1064–1070).

Tran, Q., & Firl, J. (2013). Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 334–339). Gold Coast City, Australia: IEEE.

Tran, Q., & Firl, J. (2014). Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 918–923). Dearborn, MI: IEEE.

Trautman, P., & Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 797–803). Taipei, Taiwan: IEEE.

Ulbrich, S., & Maurer, M. (2013). Probabilistic online pomdp decision making for lane changes in fully automated driving. In *Proceedings of the IEEE intelligent transportation systems conference* (pp 2063–2067).

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., et al. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, *25*(8), 425–466.

Wei, J., Dolan, J. M., Snider, J. M., & Litkouhi, B. (2011). A point-based MDP for robust single-lane autonomous driving behavior under uncertainties. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 2586–2592). Shanghai, China: IEEE.

Werling, M., Ziegler, J., Kammel, S., & Thrun, S. (2010). Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 987–993). Anchorage, AK: IEEE.

Xu, W., Wei, J., Dolan, J., Zhao, H., & Zha, H. (2012). A real-time motion planner with trajectory optimization for autonomous vehicles. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 2061–2067). Saint Paul, MN: IEEE.

Enric Galceran is a Senior Researcher with the Autonomous Systems Lab (ASL) at ETH Zurich, Zurich, Switzerland. He received the B.S. (2008) and M.S. (2010) degrees in Computer Engineering and the Ph.D. (2014) in Robotics from the University of Girona, Girona, Catalonia (Spain). He was a Postdoctoral Researcher with the Perceptual Robotics Laboratory (PeRL) and the APRIL laboratory at the University of Michigan, Ann Arbor, MI. His research interests include integrated perception and planning for autonomous navigation, and he is active in projects involving autonomous driving, agricultral robotics, multi-robot collaboration, and search and rescue operations.

Alexander G. Cunningham is an autonomy researcher at Toyota Research Institute, in Ann Arbor, Michigan. He received his B.S. degree in Electrical and Computer Engineering from Worcester Polytechnic Institute in Worcester MA in 2008, and completed his M.S. and Ph.D. at Georgia Institute of Technology in Atlanta GA in 2010 and 2014 respectively for his work on decentralized simultaneous localization and mapping. He was previously a Research Investigator at the University of Michigan with Perceptual Robotics Laboratory (PeRL), Department of Naval Architecture and Marine Engineering. His research interests include autonomous driving, focusing primarily on robust perception and reasoning under uncertainty.

Ryan M. Eustice received the B.S. degree in Mechanical Engineering from Michigan State University, East Lansing, MI in 1998, and the Ph.D. degree in Ocean Engineering from the Massachusetts Institute of Technology/Woods Hole Oceanographic Institution Joint Program, Woods Hole, MA, in 2005. Currently, he is an Associate Professor with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, with joint appointments in the Department of Electrical Engineering and Computer Science, and in the Department of Mechanical Engineering. His research interests include simultaneous localization and mapping, computer vision and robot perception, marine and mobile robotics, and autonomous driving.

Edwin Olson is an Associate Professor of Computer Science and Engineering at the University of Michigan. He is the director of the APRIL robotics lab, which studies Autonomy, Perception, Robotics, Interfaces, and Learning. His active research projects include mapping methane in landfills, multi-robot search and rescue, communication, railway safety, and automobile autonomy and safety. He received a Ph.D. from the Massachusetts Institute of Technology in 2008 for his work in robust robot mapping.