Planning for Autonomous Cars that Leverages Effects on Human Actions

Dorsa Sadigh, Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan
University of California, Berkeley, {dsadigh, sastry, sseshia, anca}@eecs.berkeley.edu

***Abstract*—Traditionally, autonomous cars make predictions about other drivers' future trajectories, and plan to stay out of their way. This tends to result in defensive and opaque behaviors. Our key insight is that an autonomous car's actions will actually affect what other cars will do in response, whether the car is aware of it or not. Our thesis is that we can leverage these responses to plan more efficient and communicative behaviors. We model the interaction between an autonomous car and a human driver as a dynamical system, in which the robot's actions have immediate consequences on the state of the car, but also on human actions. We model these consequences by approximating the human as an optimal planner, with a reward function that we acquire through Inverse Reinforcement Learning. When the robot plans with this reward function in this dynamical system, it comes up with actions that purposefully change human state: it merges in front of a human to get them to slow down or to reach its own goal faster; it blocks two lanes to get them to switch to a third lane; or it backs up slightly at an intersection to get them to proceed first. Such behaviors arise from the optimization, without relying on hand-coded signaling strategies and without ever explicitly modeling communication. Our user study results suggest that the robot is indeed capable of eliciting desired changes in human state by planning using this dynamical system.**

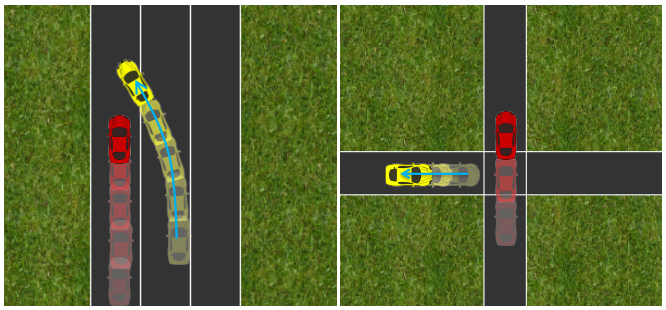
I. Introduction

Currently, autonomous cars tend to be overly *defensive* and obliviously *opaque*. When needing to merge into another lane, they will patiently wait for another driver to pass first. When stopped at an intersection and waiting for the driver on the right to go, they will sit there unable to wave them by. They are very capable when it comes to obstacle avoidance, lane keeping, localization, active steering and braking [5–8, 13, 15, 24]. But when it comes to other human drivers, they tend to rely on simplistic models: for example, assuming that other drivers will be bounded disturbances [9, 20], they will keep moving at the same velocity [16, 21, 26], or they will approximately follow one of a set of known trajectories [10, 25].

These simplistic models lead to predictions about what other cars are going to do, and the autonomous car's task is to do its best to stay out of their way. It will not cut in front of another driver when it is in a rush. It will also be restricted to functional actions, and not execute actions that are communicative.

Our goal is to enable autonomous cars to be more efficient, and better at coordinating with human drivers.

> *Our key insight is that other drivers do not operate in isolation: an autonomous car's actions will actually have effects on what other drivers will do. Leveraging these effects during planning will generate behaviors for autonomous cars that are more efficient and communicative.*



(a) Car merges *ahead* of human; anticipates human *braking*

(b) Car *backs up* at 4way stop; anticipates human *proceeding*

(c) User drives human car

Fig. 1: We enable cars to plan with a model of how human drivers would react to the car's actions. We test the planner in a user study, where the car figures out that (a) it can merge in front of a human and that will slow them down, or (b) it can back up slightly at an intersection and that will make the human go first.

In this work, we develop an optimization-based method for planning an autonomous vehicle's behavior in a manner that is cognizant of the effects it will have on human driver actions. This optimization leads to plans like the ones in Fig.1.

In the top left, the yellow (autonomous) car decides to *cut in front* of a human driver in order to more efficiently reach its goal. It arrives at this plan by anticipating that taking this action will cause the human to brake and make space for it.

In the top right, the yellow car wants to let the human driver go first through the intersection, and it autonomously plans to *back up* slightly before going, an-

ticipating that this will encourage the human to proceed. These can be interpreted as signaling behaviors, but they emerge out of optimizing to affect human actions, without ever explicitly modeling human inferences.

Our contributions are three-fold:

1. Formalizing interaction with drivers as a dynamical system. We model driving in an environment with a human driven car as a dynamical system with both autonomous and human agents. In this model, the autonomous car's actions do not just have immediate effects on the car's state; instead, they also affect human actions. These, in turn, affect the state of the world. We propose a dynamics model for this system by modeling the human as optimizing some reward function, which we learn through Inverse Reinforcement Learning.

This builds on work in social navigation which accounts for interaction potentials with human trajectories [11, 23]: the human and the robot trajectories are jointly planned as the optimum of some reward function in order for everyone to reach their goals and avoid each other. More generally, these works instantiate collaborative planning [19]. In contrast, our work allows for the human and the robot to have different reward functions: the human is optimizing their own reward function, and the robot is leveraging this to better optimize its own.

The practical implications of allowing different reward functions are that the robot now has the ability to decide to be more aggressive (or not overly-defensive) in pursuing its functional goals, as well as to specifically target desired human states/responses.

2. Deriving an approximate optimization solution. We introduce an approximation to the human model, and derive a symbolic representation of the gradient of the robot's reward function with respect to its actions in order to enable efficient optimization.

3. Analyzing planning in the human-autonomous car system. We present the consequences of planning in this dynamical system, showcasing behaviors that emerge when rewarding the robot for certain effects on human state, like making the human slow down, change lanes, or go first through an intersection. We also show that such behaviors can emerge from simply rewarding the robot for reaching its goal state fast – the robot becomes more aggressive by leveraging its possible effects on human actions. Finally, we test our hypothesis that the planner is actually capable of affecting real human actions in the desired way though an in-lab user study.

Overall, this paper takes a first step towards enabling cars to be aware of (and even leverage) the consequences that their actions have on other drivers. Even though admittedly more work is needed to put these ideas in the field, we are encouraged to see planners generate actions that affect humans in a desired way without the need for any hand-coded strategies or heuristics.

II. Problem Statement

We focus on a human-robot system consisting of an autonomous (robot) car interacting in an environment with other human driven vehicles on the road. Our goal is for the autonomous car to plan its actions in a manner that is cognizant of their effects on the human driver actions. We restrict ourselves to the two agent case in this work, we have an autonomous car $\mathcal{R}$ sharing the road with a human driver $\mathcal{H}$.

We model the problem as a fully observable dynamical system, but one in which the robot actions have consequences beyond their immediate effects on the car: they will also affect human actions which in turn will affect state.

A state $x \in X$ in our system is continuous, and includes the positions and velocities of the human and autonomous (robot) car. The robot can apply continuous controls $u_{\mathcal{R}}$, which affect state immediately through a dynamics model $f_{\mathcal{R}}$:

$$x' = f_{\mathcal{R}}(x, u_{\mathcal{R}}) \tag{1}$$

However, the next state the system reaches also depends on the control the human chooses, $u_{\mathcal{H}}$. This control affects the intermediate state through a dynamics model $f_{\mathcal{H}}$:

$$x'' = f_{\mathcal{H}}(x', u_{\mathcal{H}}) \tag{2}$$

The overall dynamics of the system combines the two:

$$x^{t+1} = f_{\mathcal{H}}(f_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}), u^t_{\mathcal{H}}) \tag{3}$$

The robot's reward function depends on the current state, the robot's action, as well as the action that the human takes at that step in response, $r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$.

The key aspect of this formulation is that the robot will have a model for what $u_{\mathcal{H}}$ will be, and use that in planning to optimize its reward.

The robot will use Model Predictive Control (MPC) [17] at every iteration, it will compute a finite horizon sequence of actions to maximize its reward. It will then execute the first one, and replan.

Let $\mathbf{x} = (x^1, \ldots, x^N)^\top$ denote a finite horizon sequence of states, $\mathbf{u}_{\mathcal{H}} = (u^1_{\mathcal{H}}, \ldots, u^N_{\mathcal{H}})^\top$ denote a finite sequence of human's continuous control inputs, and $\mathbf{u}_{\mathcal{R}} = (u^1_{\mathcal{R}}, \ldots, u^N_{\mathcal{R}})^\top$ denote a finite sequence of robot's continuous control inputs. Let $R_{\mathcal{R}}$ be the reward over the MPC time horizon:

$$R_{\mathcal{R}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) \tag{4}$$

where $\mathbf{x}^0$ is the current state (the state at the current iteration), and each state thereafter is obtained through the dynamics model in (3) from the previous and the robot and human controls.

At every iteration, the robot needs to find the $\mathbf{u}_\mathcal{R}$ that maximizes this reward:

$$\mathbf{u}^*_\mathcal{R} = \arg\max_{\mathbf{u}_\mathcal{R}} R_\mathcal{R}(x^0, \mathbf{u}_\mathcal{R}, \mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R})) \tag{5}$$

Here, $\mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R})$ is what the human would do over the next N steps if the robot were to execute $\mathbf{u}_\mathcal{R}$.

The robot does not actually know $\mathbf{u}^*_\mathcal{H}$, but in the next section we propose a *model* for the human behavior that the robot can use, along with an approximation to make (5) tractable.

III. Planning While Cognizant of Effects on Human Action

In order for the robot to solve the finite horizon problem from (5) at every iteration, it needs access to $\mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R})$. This would require the robot to have access to the human's brain, able to simulate what the human would do in various scenarios. And yet, autonomous cars do exist. Typically, we get around this problem by assuming that $\mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R}) = \mathbf{u}^*_\mathcal{H}(x^0)$, e.g. that the human will maintain their current velocity [12]. In this work, we break that assumption.

We embrace that the human will take different actions depending on what actions the robot will choose. To do this, we model the human as *maximizing their own reward function* $r_\mathcal{H}(x^t, u^t_\mathcal{R}, u^t_\mathcal{H})$.

A. General Model

If the robot were to perform $\mathbf{u}_\mathcal{R}$ starting from x^0 for the next N time steps, the human would be planning at every step to maximize their reward for a finite time horizon based on the state x^t that would be reached and the control the robot would apply at that state. For instance, the robot would execute the first control $u^0_\mathcal{R}$, and the human would plan for a finite time horizon based on x^0 and $u^0_\mathcal{R}$. The human would then execute the first control in the planned sequence, reaching a new state x^1, where they would observe the robot control $u^1_\mathcal{R}$, and replan. In general, in this model we have:

$$\mathbf{u}^{*t}_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R}) = \mathbf{u}^{*t}_\mathcal{H}(\mathbf{x}^0, u^{0:t}_\mathcal{R}, u^{*0:t-1}_\mathcal{H}) \tag{6}$$

$$= \arg\max_{\mathbf{u}^{t:t+N-1}_\mathcal{H}} r_\mathcal{H}(x^t, u^t_\mathcal{R}, u^t_\mathcal{H}) + \tag{7}$$

$$\sum_{i=t+1:t+N-1} r_\mathcal{H}(x^i, \tilde{u}^i_R, u^i_\mathcal{H}) \tag{8}$$

Here, $\tilde{u}_R$ is the human's prediction of what the robot will do, which the human needs in order to be able to plan for the next few steps. This could be a simple prediction, like the robot maintaining its velocity, or it could be a complex prediction, relying on the robot also computing the optimal plan, moving us to the full game-theoretic formulation.

B. Simplifying Assumption

We simplify this model with an approximation: we give the human model access to $\mathbf{u}_\mathcal{R}$ from the start, compute the best response for the human, and assume that to be $\mathbf{u}^*_\mathcal{H}$.

Let $R_\mathcal{H}$ be the human reward over the time horizon:

$$R_\mathcal{H}(\mathbf{x}^0, \mathbf{u}_\mathcal{H}, \mathbf{u}_\mathcal{R}) = \sum_{t=1}^{N} r_\mathcal{H}(x^t, u^t_\mathcal{R}, u^t_\mathcal{H}) \tag{9}$$

Our approximation is:

$$\mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R}) = \arg\max_{\mathbf{u}_\mathcal{H}} R_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R}, \mathbf{u}_\mathcal{H}) \tag{10}$$

This approximation is motivated by the short time horizon, meaning we are not assuming the human has access to the overall plan of the robot, just to the first few time steps – this is easier for a human to predict than a full sequence of controls, e.g. that the robot will merge into the human's lane after a certain amount of time.

The general formulation is a two-player game, but this avoids the problem of infinite regress by allowing the robot to play first and force a best response from the human.[1]

C. Solution

Assuming a known human reward function $r_\mathcal{H}$ (which we will obtain later through Inverse Reinforcement Learning (IRL) [1, 14, 18, 28], see below), we can solve the optimization in (5) using L-BFGS [2], which is a quasi-Newton method that stores an approximate inverse Hessian implicitly.

To apply L-BFGS, we need the gradient of (5) with respect to $\mathbf{u}_\mathcal{R}$:

$$\frac{\partial R_\mathcal{R}}{\partial \mathbf{u}_\mathcal{R}} = \frac{\partial R_\mathcal{R}}{\partial \mathbf{u}_\mathcal{H}} \frac{\partial \mathbf{u}^*_\mathcal{H}}{\partial \mathbf{u}_\mathcal{R}} + \frac{\partial R_\mathcal{R}}{\partial \mathbf{u}_\mathcal{R}} \tag{11}$$

$\frac{\partial R_\mathcal{R}}{\partial \mathbf{u}_\mathcal{H}}$ and $\frac{\partial R_\mathcal{R}}{\partial \mathbf{u}_\mathcal{R}}$ can both be computed symbolically through backward propogation, as we have a representation of $R_\mathcal{R}$ in terms of $\mathbf{u}_\mathcal{H}$ and $\mathbf{u}_\mathcal{R}$. For $\frac{\partial \mathbf{u}^*_\mathcal{H}}{\partial \mathbf{u}_\mathcal{R}}$, we use that $\mathbf{u}^*_\mathcal{H}$ is the minimum from (10), which means that the gradient of $R_\mathcal{H}$ evaluated at $\mathbf{u}^*_\mathcal{H}$ is 0:

$$\frac{\partial R_\mathcal{H}}{\partial \mathbf{u}_\mathcal{H}}(x^0, \mathbf{u}_\mathcal{R}, \mathbf{u}^*_\mathcal{H}(x^0, \mathbf{u}_\mathcal{R})) = 0 \tag{12}$$

Now, we can differentiate the expression in equation (12) with respect to $\mathbf{u}_\mathcal{R}$:

$$\frac{\partial^2 R_\mathcal{H}}{\partial \mathbf{u}^2_\mathcal{H}} \frac{\partial \mathbf{u}^*_\mathcal{H}}{\partial \mathbf{u}_\mathcal{R}} + \frac{\partial^2 R_\mathcal{H}}{\partial \mathbf{u}_\mathcal{H} \partial \mathbf{u}_\mathcal{R}} \frac{\partial \mathbf{u}_\mathcal{R}}{\partial \mathbf{u}_\mathcal{R}} = 0 \tag{13}$$

Finally, we can solve for a symbolic expression for $\frac{\partial \mathbf{u}^*_\mathcal{H}}{\partial \mathbf{u}_\mathcal{R}}$:

[1]We enforce turn-taking for convenience, and it is justified in cases where the robot response is immediate and the human response takes longer (thus the human accounts for the robot). However, controls could also be synchronous: the robot would still force a best response for the human, but starting with the next time step.

$$\frac{\partial \mathbf{u}^*_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{R}}} = [-\frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}} \partial \mathbf{u}_{\mathcal{R}}}][\frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}^2_{\mathcal{H}}}]^{-1} \tag{14}$$

and plug it into (11).

D. Implementation Details

In our implementation, we used the software package Theano [3, 4] to symbolically compute all Jacobians and Hessians. Theano optimizes the computation graph into efficient C code, which is crucial for real-time applications. In our implementation, each step of our optimization is solved in approximately 0.3 seconds for horizon length $N = 5$ on a 2.3 GHz Intel Core i7 processor with 16 GB RAM. Future work will focus on achieving better computation time and a longer planning horizon.

E. Human Driver Reward

Thus far, we have assumed access to $r_{\mathcal{H}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$. In our implementation, we learn this reward function from human data. We collect demonstrations of a driver in a simulation environment, and use Inverse Reinforcement Learning [1, 14, 18, 22, 28] to recover a reward function that explains the demonstrations.

To handle continuous states and actions, and the fact that the demonstrations are noisy and possible locally optimal, we use Continuous Inverse Optimal Control with Locally Optimal Examples [14]. In what follows, we recap the algorithm, and present the features we used in our implementation.

IRL. We parametrize the human reward function as a linear combination of features:

$$r_{\mathcal{H}}(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) = \theta^T \phi(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}}) \tag{15}$$

and apply the principle of maximum entropy [27, 28] to define a probability distribution over human demonstrations $\mathbf{u}_{\mathcal{H}}$, with trajectories that have higher reward being more probable:

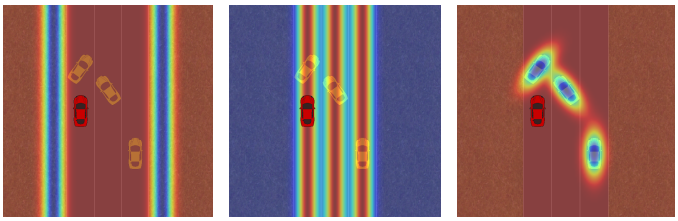
$$P(\mathbf{u}_{\mathcal{H}}|x^0, \theta) = \frac{\exp(R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}))}{\int \exp(R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \tilde{\mathbf{u}}_{\mathcal{H}}))d\tilde{\mathbf{u}}_{\mathcal{H}}} \tag{16}$$

We then do an optimization over the weights θ in the reward function that make the human demonstrations the most likely:

$$\max_{\theta} P(\mathbf{u}_{\mathcal{H}}|x^0, \theta) \tag{17}$$

We approximate the partition function in (16) following [14], by computing a second order Taylor approximation around the demonstration:

$$R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \tilde{\mathbf{u}}_{\mathcal{H}}) \simeq R_{\mathcal{H}}(x^0, \mathbf{u}_{\mathcal{R}}, \mathbf{u}_{\mathcal{H}}) + (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}})^\top \frac{\partial R_{\mathcal{H}}}{\partial \mathbf{u}_{\mathcal{H}}} + (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}})^\top \frac{\partial^2 R_{\mathcal{H}}}{\partial \mathbf{u}^2_{\mathcal{H}}} (\tilde{\mathbf{u}}_{\mathcal{H}} - \mathbf{u}_{\mathcal{H}}), \tag{18}$$



(a) Features for the boundaries of the road (b) Feature for staying inside the lanes. (c) Features for avoiding other vehicles.

Fig. 2: Features used in IRL for the human driven vehicle. In the heat map, the warmer colors correspond to higher reward. In (a), we show the features corresponding to staying within road boundaries, in (b), we show the features for staying within each lane, and in (c) we show non-spherical gaussian features corresponding to avoiding collisions.

which makes the integral in (16) a Gaussian integral, with a closed form solution. See [14] for more details.

Features. Fig.2 shows the heat map of our features. The heat map of features we have used are shown in Figure 2. The warmer colors correspond to higher rewards. In Fig. 2(a), we show the features corresponding to staying within the boundaries of the roads. In Fig. 2(b), we have features corresponding to staying within each lane, and in Fig. 2, we have features corresponding to collision avoidance, which are non-spherical Gaussians, and their major axis is along the vehicle's heading. In addition to the features shown in the figure, we include a quadratic function of the speed to capture efficiency as an objective.

Demonstrations. We collected demonstrations of a single human driver in an environment with multiple autonomous cars, which followed precomputed routes.

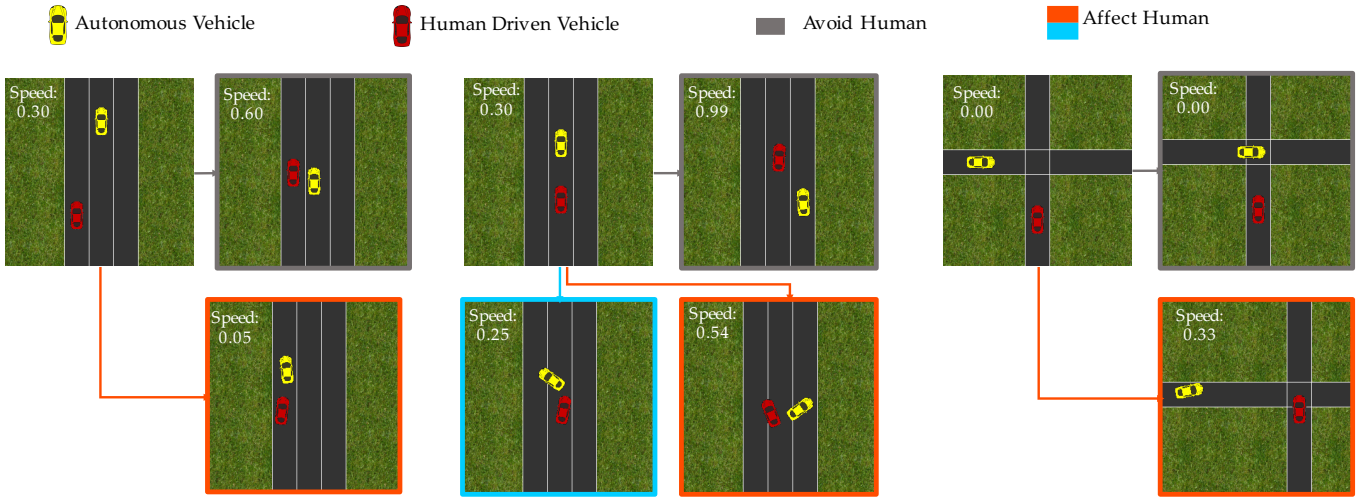
Despite the simplicity of our features and robot actions during the demonstrations, the learned human model is enough for the planner to produce behavior that is human-interpretable (case studies in Sec. IV), and that can affect human action in the desired way (user study in Sec. V).

IV. Case Studies

In this section, we introduce 3 driving scenarios, and show the result of our planner assuming a simulated human driver, highlighting the behavior that emerges from different robot reward functions. In the next section, we put the planner to the test with real users and measure the effects of the robot's plan. Fig.3 illustrates our three scenarios, and contains images from the actual user study data showcasing not just robot actions but also their real effects. Here, the yellow car is the autonomous vehicle, and the red car is the human driven vehicle.

A. Conditions for Analysis Across Scenarios

In all three scenarios, we start from an initial position of the vehicles on the road, as shown in Fig.3. In the ***control*** condition, we give the car the reward function to avoid collisions and have high velocity. We refer to this as R_{control}. In the ***experimental*** condition, we augment this



(a) Scenario 1: make human slow down (b) Scenario 2: make human go left/right (c) Scenario 3: make human go first

Fig. 3: Driving scenarios. In (a), the car plans to merge in front of the human in order to make them slow down. In (b), the car plans to direct the human to another lane, and uses its heading to choose which lane the human will go to. In (c), the car plans to back up slightly in order to make the human proceed first at the intersection. None of these plans use any hand coded strategies. They *emerge* out of optimizing with a learned model of how humans react to robot actions. In the training data for this model, the learned was *never exposed to situations* where another car stopped at an orientation as in (b), or backed up as in (c). However, by capturing human behavior in the form of a reward, the model is able to generalize to these situations, enabling the planner to find creative ways of achieving the desired effects.

reward function with a specific desired human action (e.g. low speed, lateral position, etc.). We refer to this as $R_{\text{control}} + R_{\text{affect}}$. Sections IV-C through IV-E contrast the two plans for each of our three scenarios. Sec. IV-F shows what happens when instead of explicitly giving the robot a reward function designed to trigger certain effects on the human, we simply task the robot with reaching a destination as quickly as possible.

B. Driving Simulator

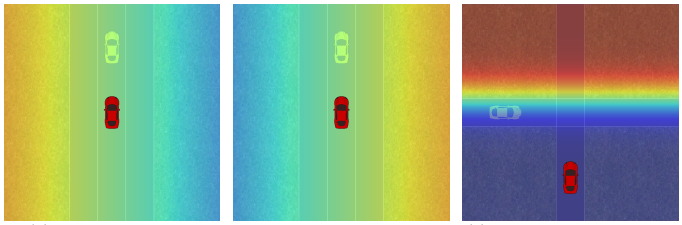
We model the dynamics of the vehicles as a simple point-mass model. Let the state of the system be $\mathbf{x} = [x \;\; y \;\; \theta \;\; v]^\top$, where x, y are the coordinates of the vehicle, θ is the heading, and v is the speed. We let $\mathbf{u} = [u_1 \;\; u_2]^\top$ represent the control input, where u_1 is the steering input and u_2 is the acceleration. We also use α as the friction coefficient, then the dynamics model of the vehicle is:

$$[\dot{x} \;\; \dot{y} \;\; \dot{\theta} \;\; \dot{v}] = [v \cdot \cos(\theta) \quad v \cdot \sin(\theta) \quad v \cdot u_1 \quad u_2 - \alpha \cdot v]. \tag{19}$$

C. Scenario 1: Make Human Slow Down

In this scenario, we show how an autonomous vehicle can plan to make a human driver slow down in a highway driving setting. The vehicles start at the initial conditions depicted on left in Fig. 3 (a), in separate lanes. In the experimental condition, we augment the robot's reward with the negative of the square of the human velocity, which encourages the robot to slow the human down.

Fig.3(a) contrasts our two conditions. In the control condition, the human moves forward uninterrupted. *In the experimental condition, however, the robot plans to move in front of the person, expecting that this will make them slow down.*



(a) Reward for Scenario 2, making the human to turn left. (b) Reward for Scenario 2, making the human to turn right. (c) Reward for Scenario 3, making the human to cross first.

Fig. 4: Heat map of the reward functions in scenarios 2 and 3. The warmer colors show higher reward values. In (a), (b), the reward function of the autonomous vehicle is plotted, which is a function of the human driven vehicle's position. In order to affect the driver to go left, the reward is higher on the left side of the road in (a), and to affect the human to go right in (b), the rewards are higher on the right side of the road. In (c), the reward of the autonomous vehicle is plotted for scenario 3 with respect to the position of the human driven car. Higher rewards correspond to making the human cross the intersection.

D. Scenario 2: Make Human Go Left/Right

In this scenario, we show how an autonomous vehicle can plan to change the human's lateral location or lane. The vehicles start at the initial conditions depicted on left in Fig. 3 (b), in the same lane, with the robot in front of the human. In the experimental condition, we augment the robot's reward with the lateral position of the human, in two ways, to encourage the robot to make the human go either left (orange border image) or right (blue border image). The two reward additions are shown in Fig.4(a) and (b).

Fig.3 (b) contrasts our two conditions. In the control condition, the human moves forward, and might decide to change lanes. *In the experimental condition, however, the robot plans to purposefully occupy two lanes (using either a positive or negative heading), expecting this will make the human move around it by using the unoccupied lane.*

E. Scenario 3: Make Human Go First

In this scenario, we show how an autonomous vehicle can plan to make the human proceed first at an intersection. The vehicles start at the initial conditions depicted on left in Fig. 3 (c), with both human and robot stopped at the 4-way intersection. In the experimental condition, we augment the robot's reward with a feature based on the y position of the human car $y_{\mathcal{H}}$ relative to the middle of the intersection y_0. In particular, we used the hyperbolic tangent of the difference, $\tanh(y_{\mathcal{H}} - y_0)$. The reward addition is shown in Fig.4 (c).

Fig.3 (c) contrasts our two conditions. In the control condition, the car goes in front of the human. *In the experimental condition, however, the robot plans to purposefully back up slightly, expecting this will make the human cross first.* Note that this could be interpreted as a communicative behavior, but communication was never explicitly encouraged in the reward function. Instead, *this behavior emerged out of the goal of affecting human actions.*

This is perhaps the most surprising behavior of the three scenarios, because it is not something human drivers do. However, our user study suggests that human drivers to respond to this in the expected way. Further, pedestrians exhibit this behavior at times, stepping back away from an intersection to let a car go by first.

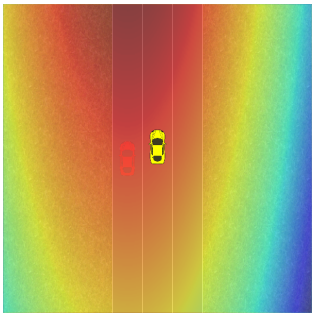
F. Behaviors Also Emerge from Efficiency

Thus far, we explicitly encoded a desired effect on human actions in the reward we gave the robot to optimize. We have also found, however, that behaviors like the ones we have seen so far can emerge out of the need for efficiency.

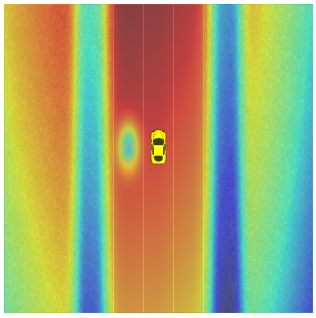
Fig.5 (bottom) shows the generated plan for when the robot is given the goal to reach a point in the left lane as quickly as possible (reward shown in Fig.6). By modeling the effects its actions have on the human actions, the robot plans to merge in front of the person, expecting that they will slow down.

In contrast, the top of the figure shows the generated plan for when the robot uses a simple (constant velocity) model of the person. In this case, the robot assumes that merging in front of the person can lead to a collision, and defensively waits for the person to pass, merging behind them.

We hear about this behavior often in autonomous cars today: they are defensive. Enabling them to plan in a manner that is cognizant that they can affect other driver actions can make them more efficient at achieving their goals.



(a) Single feature corresponding to distance to goal on the top left.



(b) All features present for autonomous vehicle's reward function.

Fig. 6: Heat map of reward function for reaching a final goal at the top left of the road. As shown in the figure, the goal position is darker showing more reward for reaching that point.

V. User Study

The previous section showed the robot's plans when interacting with a simulated user that perfectly fits the robot's model of the human. Next, we present the results of a user study that evaluates whether the robot can successfully have the desired effects on real users.

A. Experimental Design

We use the same 3 scenarios as in the previous section.

Manipulated Factors. We manipulate a single factor: the ***reward*** that the robot is optimizing, as described in Sec. IV-A. This leads to two conditions: the ***experimental*** condition where the robot is encouraged to have a particular effect on human state though the reward $R_{\text{control}} + R_{\text{affect}}$, and the ***control*** condition where that aspect is left out of the reward function and the robot is optimizing only R_{control} (three conditions for Scenario 2, where we have two experimental conditions, one for the left case and one for the right case).

Dependent Measures. For each scenario, we measure the value along the user trajectory of the feature added to the reward function for that scenario, R_{affect}. Specifically, we measure the human's negative squared velocity in Scenario 1, the human's x axis location relative to center in scenario 2 in Scenario 2, and whether the human went first or not through the intersection in Scenario 3 (i.e. a filtering of the feature that normalizes for difference in timing among users and measures the desired objective directly).

Hypothesis. We hypothesize that our method enables the robot to achieve the effects it desires not only in simulation, but also when interacting with real users:

> *The reward function that the robot is optimizing has a significant effect on the measured reward during interaction. Specifically, R_{affect} is higher, as planned, when the robot is optimizing for it.*

Subject Allocation. We recruited 10 participants (2 female, 8 male). All the participants owned drivers license with at least 2 years of driving experience. We

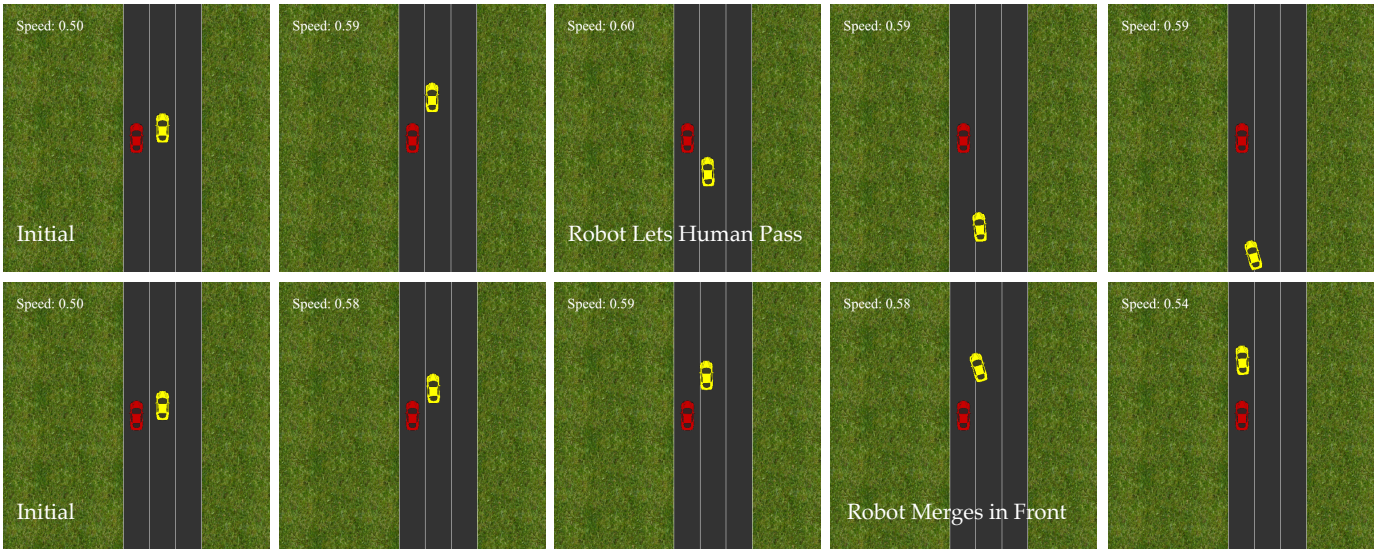


Fig. 5: A time lapse for Sec. IV-F, where the autonomous vehicle's goal is to reach a final point in the left lane. In the top scenario, the autonomous vehicle has a simple model of the human driver that does not account for the influence of its actions on the human actions, so it acts more defensively, waiting for the human to pass first. In the bottom, the autonomous vehicle uses the learned model of the human driver, so it acts more aggressively and reaches its goal faster.

ran our experiments using a 2D driving simulator, we have developed with the driver input provided through driving simulator steering wheel and pedals as shown in Figure 1. We used a within-subjects design and counterbalanced the order of the conditions.

B. Analysis

Scenario 1: A repeated measures ANOVA showed the square speed to be significantly lower in the experimental condition than in the control condition ($F(1,160) = 228.54$, $p < 0.0001$). This supports or hypothesis: the human moved slower when the robot planned to have this effect on the human.

We plot the speed and latitude profile of the human driven vehicle over time for all trajectories in Fig.7. Fig.7(a) shows the speed profile of the control condition trajectories in gray, and of the experimental condition trajectories in orange. Fig.7(b) shows the mean and standard error for each condition. In the control condition, human squared speed keeps increasing. In the experimental condition however, by merging in front of the human, the robot is triggering the human to brake and reduce speed, as planned. The purple trajectory represents a simulated user that perfectly matches the robot's model, showing the ideal case for the robot. The real interaction moves significantly in the desired direction, but does not perfectly match the ideal model, since real users do not act exactly as the model would predict.

The figure also plots the y position of the vehicles along time, showing that the human has not travelled as far forward in the experimental condition.

Scenario 2: A repeated measures ANOVA showed a significant effect for the reward factor ($F(2,227) = 55.58$, $p < 0.0001$). A post-hoc analysis with Tukey HSD showed that both experimental conditions were significantly different from the control condition, with the user car going more to the left than in the control condition when R_{affect} rewards left user positions ($p < 0.0001$), and more to the right in the other case ($p < 0.001$). This supports our hypothesis.

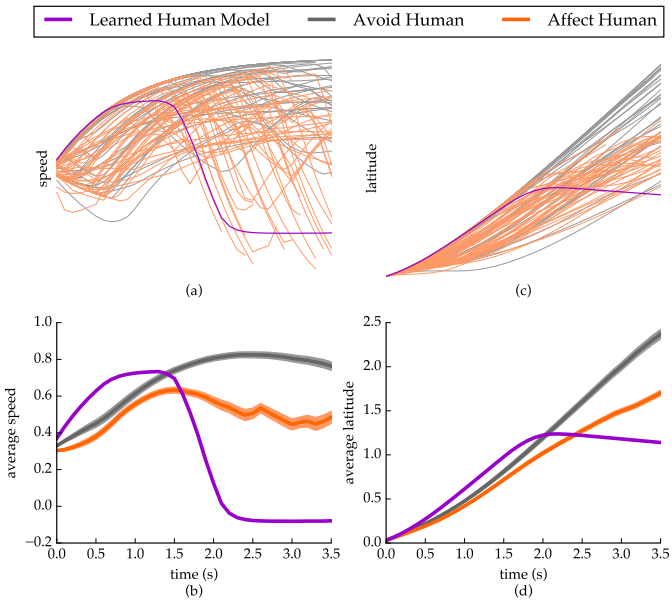


Fig. 7: Speed profile and latitude of human driven vehicle for Scenario 1. The first column shows the speed of all trajectories with its mean and standard errors in the bottom graph. The second column shows the latitude of the vehicle over time; similarly, with the mean and standard errors. The grey trajectories correspond to the control condition, and the orange trajectories correspond to the experimental condition: the robot decides to merge in front of the users and succeeds at slowing them down. The purple plot corresponds to a simulated user that perfectly matches the model that the robot is using.

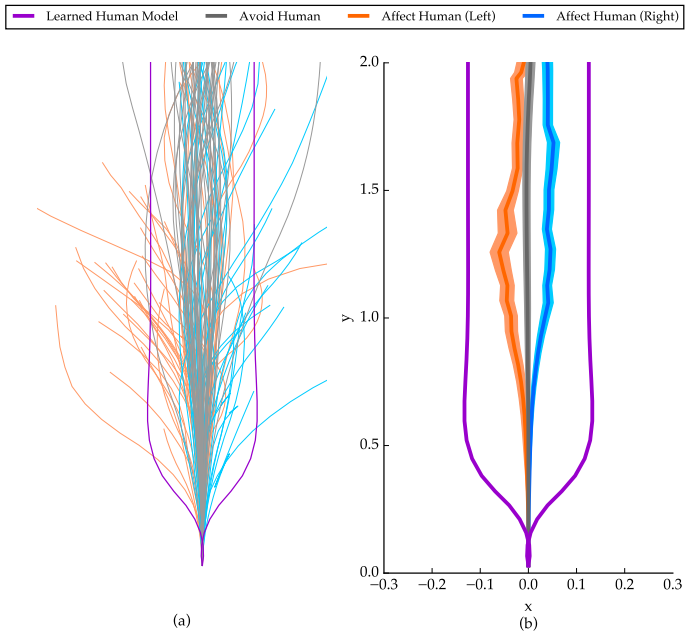


Fig. 8: Trajectories of human driven vehicle for Scenario 2 (a) with mean and standard error (right). Orange (blue) indicates conditions where the reward encouraged the robot to affect the user to go left (right).

We plot all the trajectories collected from the users in Fig.8. Fig.8(a) shows the control condition trajectories in grey, while the experimental conditions trajectories are shown in orange (for left) and blue (for right). By occupying two lanes, the robot triggers an avoid behavior from the users in the third lane. Here again, purple curves show a simulated user, i.e. the ideal case for the robot.

Scenario 3: An ordinal logistic regression with user as a random factor showed that significantly more users went first in the intersection in the experimental condition than in the baseline ($\chi^2(1, 129) = 106.41$, $p < .0001$). This supports our hypothesis.

Fig.9 plots the y position of the human driven vehicle with respect to the x position of the autonomous vehicle. For trajectories that have a higher y position for the human vehicle than the x position for the robot, the human car has crossed the intersection before the autonomous vehicle. The lines corresponding to these trajectories travel above the origin, which is shown with a blue square in this figure. The mean of the orange lines travel above the origin, which means that the autonomous vehicle has successfully affected the humans to cross first. The grey lines travel below the origin, i.e. the human crossed second.

Overall, our results suggest that the robot was able to affect the human state in the desired way, even though it does not have a perfect model of the human.

VI. Discussion

Summary. In this paper, we formalized the interaction between an autonomous (robot) vehicle and a human driver as a dynamical system, in which the actions of the robot affect those of the human and vice-versa. We introduced an approximate solution that enables the robot to optimize its own reward within this system. The resulting plans can purposefully modify human behavior, and can achieve the robot's goal more efficiently. Our user study suggests that this is not only true in simulation, but also true when tested with real users.

Limitations. All this work happened in a simple driving simulator. To put this on the road, we will need more emphasis on safety, as well as a longer planning horizon. The former involves the use of formal methods and safe control as well as better models of users: not all drivers act the same and replanning is not the end-solution to address this. Using a probabilistic dynamics model as opposed to planning with the most probable human actions, as well as estimating driving style, will be important next steps.

An even bigger limitation is that we currently focus on a single human driver. Looking to the interaction among multiple vehicles is not just a computational challenge, but also a modeling one – it is not immediately clear how to formulate the problem when multiple human-driven vehicles are interacting and reacting to each other.

Conclusion. Despite these limitations, we are encouraged to see autonomous cars generate human-interpretable behaviors though optimization, without relying on hand-coded heuristics. We also look forward to applications of these ideas beyond autonomous driving, to mobile robots, UAVs, and in general to human-robot interactive scenarios where robot actions can influence human actions.

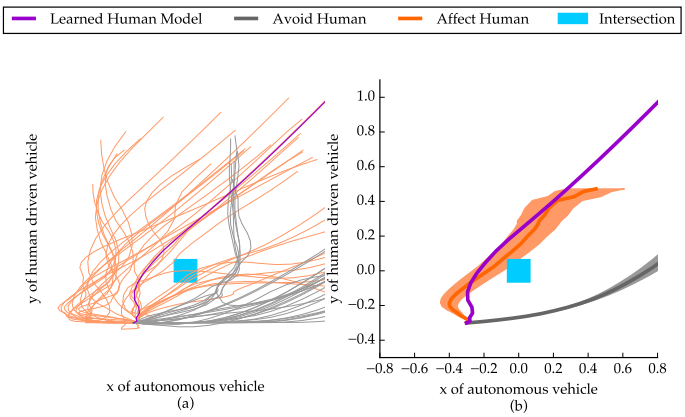


Fig. 9: Plot of $y_{\mathcal{H}}$ with respect to $x_{\mathcal{R}}$. The orange curves correspond to when the autonomous vehicle affects the human to cross the intersection first. The grey curves correspond to when the nominal setting.

VII. Acknowledgments

This work was partially supported by Berkeley DeepDrive, NSF grants CCF-1139138 and CCF-1116993, ONR N00014-09-1-0230, and an NDSEG Fellowship.

References

[1] Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 1–8. ACM, 2005.

[2] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[3] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[4] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[5] MWMG Dissanayake, Paul Newman, Steven Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[6] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3): 566–580, May 2007.

[7] Paolo Falcone, Francesco Borrelli, H Eric Tseng, Jahan Asgari, and Davor Hrovat. Integrated braking and steering model predictive control approach in autonomous vehicles. In *Advances in Automotive Control*, volume 5, pages 273–278, 2007.

[8] Paolo Falcone, H. Eric Tseng, Francesco Borrelli, Jahan Asgari, and Davor Hrovat. MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(sup1):611–628, September 2008.

[9] Alison Gray, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 208–213. IEEE, 2013.

[10] Christoph Hermes, Christian Wohler, Kurt Schenk, and Franz Kummert. Long-term vehicle motion prediction. In *2009 IEEE Intelligent Vehicles Symposium*, pages 652–657, 2009.

[11] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA), Seattle, USA*, volume 134, 2015.

[12] Stéphanie Lefèvre, Ashwin Carvalho, Yiqi Gao, H Eric Tseng, and Francesco Borrelli. Driver models for personalised driving assistance. *Vehicle System Dynamics*, 53(12): 1705–1720, 2015.

[13] John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, et al. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008.

[14] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617*, 2012.

[15] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168.

[16] Brandon Luders, Mangal Kothari, and Jonathan P How. Chance constrained rrt for probabilistic robustness to environmental uncertainty. In *AIAA guidance, navigation, and control conference (GNC), Toronto, Canada*, 2010.

[17] Manfred Morari, CE Garcia, JH Lee, and DM Prett. *Model predictive control*. Prentice Hall Englewood Cliffs, NJ, 1993.

[18] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th international conference on Machine learning*, pages 663–670, 2000.

[19] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 189–196. ACM, 2015.

[20] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M Murray, and Sanjit A Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 239–248. ACM, 2015.

[21] Dorsa Sadigh and Ashish Kapoor. Safe control under uncertainty. *arXiv preprint arXiv:1510.07313*, 2015.

[22] Masamichi Shimosaka, Tetsuya Kaneko, and Kentaro Nishi. Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning. In *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1694–1700. IEEE, 2014.

[23] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 797–803.

[24] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[25] Ramanarayan Vasudevan, Victor Shia, Yiqi Gao, Ricardo Cervera-Navarro, Ruzena Bajcsy, and Francesco Borrelli. Safe semi-autonomous control with enhanced driver modeling. In *American Control Conference (ACC), 2012*, pages 2896–2903. IEEE, 2012.

[26] Michael P Vitus and Claire J Tomlin. A probabilistic approach to planning and control in autonomous urban driving. In *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 2459–2464.

[27] Brian D Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.

[28] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438, 2008.