

Game-Theoretic Modeling of Traffic in Unsignalized Intersection Network for Autonomous Vehicle Control Verification and Validation

Ran Tian^{1b}, Nan Li^{1b}, *Member, IEEE*, Ilya Kolmanovsky^{2b}, *Fellow, IEEE*,
Yildiray Yildiz^{2b}, *Senior Member, IEEE*, and Anouck R. Girard^{2b}, *Senior Member, IEEE*

Abstract—For a foreseeable future, autonomous vehicles (AVs) will operate in traffic together with human-driven vehicles. Their planning and control systems need extensive testing, including early-stage testing in simulations where the interactions among autonomous/human-driven vehicles are represented. Motivated by the need for such simulation tools, we propose a game-theoretic approach to modeling vehicle interactions, in particular, for urban traffic environments with unsignalized intersections. We develop traffic models with heterogeneous (in terms of their driving styles) and interactive vehicles based on our proposed approach, and use them for virtual testing, evaluation, and calibration of AV control systems. For illustration, we consider two AV control approaches, analyze their characteristics and performance based on the simulation results with our developed traffic models, and optimize the parameters of one of them.

Index Terms—Autonomous vehicles, decision making, game theory, human factors, multi-agent systems, system testing.

I. INTRODUCTION

AUTONOMOUS driving technologies have greatly advanced in recent years with the promise of providing safer, more efficient, environment-friendly, and easily accessible transportation [1]–[3]. To fulfill such a commitment requires developing advanced planning and control algorithms to navigate autonomous vehicles (AVs), as well as comprehensive testing procedures to verify their safety and performance characteristics [4]–[6]. It is estimated based on the collision fatalities rate that to confidently verify an AV control system, hundreds of millions of miles need to be driven [4], which can be highly time and resource consuming if these driving tests are all conducted in the physical world. Therefore, an alternative solution is to use simulation tools to conduct early-stage testing and evaluation

in a virtual world so that the overall verification process can be accelerated [7], [8]. The work of this paper is motivated by the need for virtual testing of AV control systems.

In the near to medium term, AVs are expected to operate in traffic together with human-driven vehicles. Therefore, accounting for the interactions among autonomous/human-driven vehicles is important to achieve safe and efficient driving behavior of an AV.

Control strategies for AVs that account for vehicle interactions include the ones based on Markov decision processes [9]–[12], model predictive control [13], [14], game-theoretic models [15]–[18], [18], [19], as well as data-driven approaches [20], [21]. To evaluate the effectiveness of these algorithms requires simulation environments that can represent the interactions among autonomous/human-driven vehicles.

In our previous work [22], we exploited a game-theoretic approach to modeling vehicle interactions in highway traffic. Compared to highway traffic, urban traffic environments with intersections are considered to be more challenging for both human drivers and AVs, as they involve more extensive and complex interactions among vehicles. For instance, almost 40% of traffic accidents in the U.S. are intersection-related [23].

In this paper, we extend the game-theoretic approach of [22] to modeling vehicle interactions in urban traffic. In particular, we consider urban traffic environments with unsignalized intersections. Firstly, unsignalized intersections may be even more challenging than signalized intersections because, due to the lack of guidance from traffic signals, a driver/automation needs to decide on its own, whether, when and how to enter and drive through the intersection. According to the U.S. Federal Highway Administration’s report, almost 70% of fatalities due to intersection-related traffic accidents happened at unsignalized intersections [24]. Thus, well-verified autonomous driving systems for unsignalized intersections may deliver significant safety benefits. Indeed, many previous works in the literature on AV control for intersections, including [19], [25]–[28], deal with unsignalized intersections, although they do not always explicitly point this out.

Our approach formulates the decision-making processes of drivers/vehicles as a dynamic game, where each vehicle

Manuscript received October 13, 2019; revised July 12, 2020; accepted October 8, 2020. This work was supported by the National Science Foundation under Grant CNS 1544844. The Associate Editor for this article was N. Geroliminis. (*Corresponding author: Ran Tian.*)

Ran Tian, Nan Li, Ilya Kolmanovsky, and Anouck R. Girard are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: tianran@umich.edu; nanli@umich.edu; ilya@umich.edu; anouck@umich.edu).

Yildiray Yildiz is with the Department of Mechanical Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: yyildiz@bilkent.edu.tr).

Digital Object Identifier 10.1109/TITS.2020.3035363

interacts with other vehicles by observing their states, predicting their future actions, and then planning its own actions. In addition to the difference in traffic scenarios being considered (i.e., urban traffic in this paper versus highway traffic in [22]), this paper contains the following methodological contribution compared to [22]: Due to the much larger state space for urban traffic environments with intersections compared to that for highway traffic, the reinforcement learning approach used in [22] to solve for control policies is computationally prohibitive. Therefore, we develop in this paper an alternative approach that uniquely integrates a game-theoretic formalism, receding-horizon optimization, and an imitation learning algorithm to obtain control policies. This new approach is shown to be computationally effective for the large state space of urban traffic.

Our model representing vehicle interactions falls in-between macroscopic traffic models and microscopic driver behavior models. On the one hand, macroscopic traffic models typically assume a large number (e.g., hundreds to thousands) of vehicles and study the average or statistical properties of traffic flow, such as traffic flux (vehicles/hour) versus traffic density (vehicles/km) [29]–[31]. Individual vehicle behavior is usually not represented in such models. On the other hand, microscopic driver behavior models typically focus on modeling the decision-making as well as control processes of individual drivers [32]–[34], such as the responses of a human driver to various traffic situations. The interactions among multiple vehicles are usually not incorporated in such models. Consequently, neither models are particularly suitable for virtual testing of AV control systems. In contrast, our game-theoretic model represents the interactive decision-making processes of multiple drivers/vehicles, where each individual vehicle’s behavior is represented using a kinematic vehicle model. This way, we can model traffic scenes with a medium number (e.g., dozens) of interacting vehicles, suitable as test scenarios for AV control systems.

In [35], we modeled the interactions among vehicles at unsignalized intersections, but using a different game-theoretic approach from the one used in this paper. Specifically, in [35], we modeled vehicle interactions based on a formulation of a leader-follower game; while in this paper, we consider the application of level- k game theory [36], [37]. The control strategies of all interacting vehicles modeled using the framework of [35] are homogeneous; while the control strategies of different vehicles modeled using the scheme of this paper are heterogeneous, differentiated by their level- k control policies with different $k = 0, 1, 2, \dots$. This heterogeneity can be used to represent the different driving styles among different drivers, e.g., aggressive driving versus cautious/conservative driving. In addition, [35] models a single intersection with up to 10 interacting vehicles; while in this paper, thanks to the effective application of the aforementioned solution approach integrating game theory, receding-horizon optimization, and imitation learning to obtain control policies, the scheme of this paper can be used to model much larger road systems involving many intersections and many vehicles with manageable online computational effort. This enables the investigation into driving characteristics that are exhibited

when a vehicle drives through multiple road segments, such as overall travel time, fuel consumption, etc. A road system with 15 intersections and 30 vehicles is illustrated as an example in Section IV. Furthermore, application of the developed traffic models to verification and validation of AV control systems is comprehensively discussed in this paper, but not in [35].

Preliminary results of this paper have been reported in the conference papers [38] and [39]. The results modeling the interactions between two vehicles at a four-way intersection are reported in [38] and those for two vehicles at a roundabout intersection are in [39]. This paper generalizes the methodology to modeling the interactions among multiple (more than two) vehicles and to an additional intersection type – T-shaped intersection. Constructing larger road systems based on the models of these three intersections is reported for the first time in this paper. This paper also demonstrates how the developed traffic models can be used for virtual testing, evaluation, and calibration of AV control systems, which is not provided in [38] and [39].

In summary, the contributions of this paper are: 1) We describe an approach based on level- k game theory to modeling the interactions among vehicles in urban traffic environments with unsignalized intersections. 2) We propose an algorithm based on imitation learning to obtain level- k control policies so that our approach to modeling vehicle interactions is scalable – able to model traffic scenes with many intersections and many vehicles. In particular, this new imitation learning approach is compared with the supervised learning approach used in our previous work [39] and is shown to provide better results. 3) We demonstrate the use of the developed traffic models for virtual testing, evaluation, and calibration of AV control systems. For illustration purposes, we consider two AV control approaches, analyze their characteristics and performance based on the simulation results with our traffic models, and optimize the parameters of one of them.

This paper is organized as follows: The models representing vehicle kinematics and driver decision-making processes are introduced in Section II. The game-theoretic model representing vehicle interactions and obtaining its explicit approximation via imitation learning are discussed in Section III. The procedure to construct traffic models of larger road systems based on the models of three basic intersection scenarios is described in Section IV. We then propose two AV control approaches in Section V, used as case studies to illustrate the application of our developed traffic models to AV control verification and validation. Simulation results are reported in Section VI, and finally, the paper is concluded in Section VII.

II. TRAFFIC DYNAMICS AND DRIVER DECISION-MAKING MODELING

In this section, we describe our models to represent the traffic dynamics and the decision-making processes of interacting drivers.

A. Traffic Dynamics

Firstly, we describe the evolution of a traffic scenario using a discrete-time model as follows:

$$\mathbf{s}_{t+1} = \mathcal{F}(\mathbf{s}_t, \mathbf{u}_t), \quad (1)$$

where $\mathbf{s} = (s^1, s^2, \dots, s^m)$ denotes the traffic state, composed of the states s^i , $i \in \mathcal{M} = \{1, 2, \dots, m\}$, of all interacting vehicles in the scenario, $\mathbf{u} = (u^1, u^2, \dots, u^m)$ denotes the collection of all vehicles' actions u^i , and the subscript t represents the discrete-time instant. In particular, the state of a vehicle is composed of two parts, $s^i = (s^{i,1}, s^{i,2})$. The first part $s^{i,1} = (x^i, y^i, v^i, \theta^i)$ represents the state of vehicle kinematics, modeled using the following unicycle model:

$$\begin{bmatrix} x_{t+1}^i \\ y_{t+1}^i \\ v_{t+1}^i \\ \theta_{t+1}^i \end{bmatrix} = f(s_t^i, u_t^i) = \begin{bmatrix} x_t^i + v_t^i \cos(\theta_t^i) \Delta t \\ y_t^i + v_t^i \sin(\theta_t^i) \Delta t \\ v_t^i + a_t^i \Delta t \\ \theta_t^i + \omega_t^i \Delta t \end{bmatrix}, \quad (2)$$

where (x^i, y^i) , v^i , and θ^i represent, respectively, the vehicle's position in the ground-fixed frame, its speed, and its heading angle, the inputs a^i and ω^i represent, respectively, the vehicle's acceleration and heading angle rate, while Δt is the sampling period for decision-making. The second part $s^{i,2} = (r^i, \zeta^i)$ contains additional information related to the vehicle's decision-making objective, including $r^i = (r_x^i, r_y^i)$, representing a target/reference position to go, and ζ^i , a feature vector containing key information about the road layout and geometry such as the road width, the angle of intersection, and etc [35]. When vehicle i is driving toward, in the middle of, or exiting a specific intersection, $s^{i,2}$ stays constant with r^i being a point located in the center of the vehicle's target lane; $s^{i,2}$ gets updated after the vehicle has returned to straight road and is driving toward the next intersection.

We remark that the unicycle model (2) is suitable for our purpose of modeling the interactive decision-making processes and the resulting dynamic behavior of multiple vehicles in intersection traffic scenarios. This model is simple while it can sufficiently accurately represent vehicle kinematics at low to medium vehicle speeds and involving turning behavior [16], [40]. In Section VI we show that 1) driving behaviors planned based on the model (2) can be accurately executed by vehicle systems with lower-level control, and 2) vehicle trajectories extracted from real-world traffic data can be satisfactorily reproduced by simulating the model (2) along with the decision logic developed following our approach.

B. Driver Decision-Making

An action u^i is a pair of values of the inputs (a^i, ω^i) , i.e., $u^i = (a^i, \omega^i)$. We assume that the drivers of the vehicles make sequential decisions based on receding-horizon optimization as follows: At each discrete-time instant t , the driver of vehicle i solves for

$$\begin{aligned} (\mathbf{u}_t^i)^* &= \{(u_{0|t}^i)^*, (u_{1|t}^i)^*, \dots, (u_{N-1|t}^i)^*\} \\ &\in \arg \max_{\mathbf{u}_t^i \in \mathcal{U}^N} \sum_{\tau=0}^{N-1} \lambda^\tau R(s_{\tau|t}^i, \mathbf{s}_{\tau|t}^{-i}, u_{\tau|t}^i, \mathbf{u}_{\tau|t}^{-i}), \end{aligned} \quad (3)$$

where $\mathbf{u}_t^i = \{u_{0|t}^i, u_{1|t}^i, \dots, u_{N-1|t}^i\}$ represents a sequence of predicted actions of vehicle i , with $u_{\tau|t}^i$ denoting the predicted action for time step $t+\tau$ and taking values in a finite action set \mathcal{U} ; the notations $s_{\tau|t}^i$, $\mathbf{s}_{\tau|t}^{-i}$ and $\mathbf{u}_{\tau|t}^{-i}$ represent, respectively, the predicted state of vehicle i , and the collections of predicted

states and actions of the other vehicles $j \in \mathcal{M}$, $j \neq i$, i.e., $\mathbf{s}_{\tau|t}^{-i} = (s_{\tau|t}^j)_{j \in \mathcal{M}, j \neq i}$ and $\mathbf{u}_{\tau|t}^{-i} = (u_{\tau|t}^j)_{j \in \mathcal{M}, j \neq i}$; R is a reward function depending on the states and actions of all interacting vehicles, which will be introduced in detail in the following section; and $\lambda \in (0, 1]$ is a factor discounting future reward.

Once an optimal action sequence $(\mathbf{u}_t^i)^*$ is determined, vehicle i applies the first element $(u_{0|t}^i)^*$ for one time step, i.e., $u_t^i = (u_{0|t}^i)^*$. After the states of all vehicles have been updated, vehicle i repeats this procedure at $t+1$.

The fact that R depends not only on the ego vehicle's state and action but also on those of the other vehicles determines the interactive nature of the drivers' decision-making processes in a multi-vehicle traffic scenario. Note that, due to the unknowns $\mathbf{u}_{\tau|t}^{-i}$ and $\mathbf{s}_{\tau|t}^{-i}$ for $\tau = 0, 1, \dots, N-1$, the problem (3) has not been well-defined yet and cannot be solved. To be able to solve for $(\mathbf{u}_t^i)^*$, we will exploit a game-theoretic approach in Section III to predict the values of $\mathbf{u}_{\tau|t}^{-i}$ and $\mathbf{s}_{\tau|t}^{-i}$.

C. Reward Function

We use the reward function R in (3) to represent vehicles' decision-making objectives in traffic. In this paper, we consider R defined as follows:

$$R(s_{\tau|t}^i, \mathbf{s}_{\tau|t}^{-i}, u_{\tau|t}^i, \mathbf{u}_{\tau|t}^{-i}) = \mathbf{w}^\top \Phi(s_{\tau+1|t}^i, (s_{\tau+1|t}^j)_{j \in \mathcal{M}, j \neq i}), \quad (4)$$

where $\Phi = [\phi_1, \phi_2, \dots, \phi_6]^\top$ is the feature vector and $\mathbf{w} \in \mathbb{R}_+^6$ is the weight vector. Note that $s_{\tau+1|t}^j = f(s_{\tau|t}^j, u_{\tau|t}^j)$ for all $j \in \mathcal{M}$ based on the kinematic vehicle model (2).

The features $\phi_1, \phi_2, \dots, \phi_6$ are designed to encode common considerations in driving, such as safety, comfort, travel time, etc. They are defined as follows.

The feature ϕ_1 characterizes the collision status of the vehicle. In particular, we over-bound the geometric contour of each vehicle by a rectangle, referred to as the collision-zone (c -zone). Then, $\phi_1 = -1$ if vehicle i 's c -zone at the predicted state $s_{\tau+1|t}^i$ overlaps with any of the other vehicles' c -zones at their predicted states $s_{\tau+1|t}^j$, which indicates a danger of collision; and $\phi_1 = 0$ otherwise. The c -zone over-bounds the vehicle's geometric contour by a small margin to compensate for the effect of perception errors. Note that the size of these perception errors is small at low to medium vehicle speeds when compared to the resolution of the actions.

The feature ϕ_2 characterizes the on-road status of the vehicle, taking -1 if vehicle i 's c -zone crosses any of the road boundaries, and 0 otherwise. And similarly, ϕ_3 characterizes the in-lane status of the vehicle. If vehicle i 's c -zone crosses a lane marking that separates the traffic of opposite directions or enters a lane different from its target lane when exiting an intersection, then $\phi_3 = -1$; $\phi_3 = 0$ otherwise.

To characterize the status of maintaining a safe and comfortable separation between vehicles, we further define a separation-zone (s -zone) for each vehicle, which over-bounds the vehicle's c -zone with a separation margin. The feature ϕ_4 takes -1 if vehicle i 's s -zone overlaps with any of the other vehicles' s -zones at their predicted states, and takes 0 otherwise.

The features ϕ_5 and ϕ_6 characterize the vehicle's status of driving toward its target lane. They are defined as $\phi_5 = -|r_x^i - x^i| - |r_y^i - y^i|$ and $\phi_6 = v^i$, so that the vehicle is encouraged to approach the reference point r^i in its target lane as quickly as it can.

The above reward function design reflects common driving objectives in traffic. The weight vector \mathbf{w} can be tuned to achieve reasonable driving behavior, or be calibrated using traffic data and approaches such as inverse reinforcement learning [41], [42].

III. GAME-THEORETIC DECISION-MAKING AND EXPLICIT REALIZATION VIA IMITATION LEARNING

Game theory is a useful tool for modeling intelligent agents' strategic interactions. In this paper, we exploit the level- k game theory [36], [37] to model vehicles' interactive decision-making.

A. Level- k Reasoning and Decision-Making

In level- k game theory, it is assumed that players make decisions based on finite depths of reasoning, called "level," and different players may have different reasoning levels. In particular, a level-0 player makes non-strategic decisions – decisions without regard to the other players' decisions. Then, a level- k , $k \geq 1$, player makes strategic decisions by assuming that all of the other players are level- $(k-1)$, predicting their decisions based on such an assumption, and optimally responding to their predicted decisions. It is verified by experimental results from cognitive science that such a level- k reasoning process can model human interactions with higher accuracy than traditional analytic methods in many cases [37].

To incorporate level- k reasoning in our decision-making model (3), we start with defining a level-0 decision rule. According to the non-strategic assumption on level-0 players, we let a level-0 decision of a vehicle i , $i \in \mathcal{M}$, depend only on the traffic state \mathbf{s}_t , including its own state s_t^i and the other vehicles' states \mathbf{s}_t^{-i} , but not on the other vehicles' actions \mathbf{u}_t^{-i} . In this paper, a level-0 decision, $(\mathbf{u}_t^i)^0 = \{(u_{0|t}^i)^0, (u_{1|t}^i)^0, \dots, (u_{N-1|t}^i)^0\}$, is a sequence of predicted actions that maximizes the cumulative reward in (3) with treating all of the other vehicles as stationary obstacles over the planning horizon, i.e., $v_{\tau|t}^j = 0$, $\omega_{\tau|t}^j = 0$ for all $j \neq i$, $\tau = 0, 1, \dots, N$. This way, a level-0 vehicle represents an aggressive vehicle which assumes that all of the other vehicles will yield the right of way to it.

On the basis of the formulated level-0 decision rule, the level- k decisions of the vehicles are obtained based on

$$\begin{aligned} (\mathbf{u}_t^i)^k &= \{(u_{0|t}^i)^k, (u_{1|t}^i)^k, \dots, (u_{N-1|t}^i)^k\} \\ &\in \arg \max_{\mathbf{u}_t^i \in \mathcal{U}^N} \sum_{\tau=0}^{N-1} \lambda^\tau R(s_{\tau|t}^i, \mathbf{s}_{\tau|t}^{-i}, u_{\tau|t}^i, (\mathbf{u}_{\tau|t}^{-i})^{k-1}), \end{aligned} \quad (5)$$

for every $i \in \mathcal{M}$, and for every $k = 1, 2, \dots, k_{\max}$ through sequential, iterated computations, where $(\mathbf{u}_{\tau|t}^{-i})^{k-1}$ denotes the level- $(k-1)$ decisions of the other vehicles $j \neq i$, which have been determined either in the previous iteration or based on

the level-0 decision rule (for $k = 1$), and k_{\max} is the highest reasoning level for computation.

Given a finite action set \mathcal{U} , the problem (5) for every $i \in \mathcal{M}$ and $k = 1, 2, \dots, k_{\max}$ can be solved with exhaustive search, e.g., based on a tree structure [43].

B. Explicit Level- k Decision-Making via Imitation Learning

A level- k vehicle drives in traffic by applying $u_t^i = (u_{0|t}^i)^k$ at every time step, where $(u_{0|t}^i)^k$ is determined according to (5) with the current state as the initial condition, i.e., $s_{0|t}^i = s_t^i$ and $\mathbf{s}_{0|t}^{-i} = \mathbf{s}_t^{-i}$.

On the one hand, the problem (5) needs to be numerically solved. The required computational effort to solve (5) grows for larger k and larger numbers of interacting vehicles, because in order to compute the level- k decision of vehicle i , the level- $(k-1)$ decisions of all other vehicles $j \neq i$ need to be determined first, which, in turn, need the prerequisite determination of level- $(k-2)$ decisions for $k \geq 2$, and etc. On the other hand, for virtual testing of AV control systems, fast simulations are desired so that a large number of test scenarios can be covered within a short period of real time. To achieve fast simulations, we exploit machine learning techniques to move the computational tasks associated with (5) offline and obtain explicit level- k decision policies for online use.

In particular, we define a policy as a map from a triple of the ego vehicle's state s_t^i , the other vehicles' states \mathbf{s}_t^{-i} , and the ego vehicle's reasoning level k to the level- k action of the ego vehicle, i.e.

$$\pi_k : (s_t^i, \mathbf{s}_t^{-i}, k) \mapsto (u_t^i)^k. \quad (6)$$

This map is algorithmically determined by solving the problem (5) and letting $(u_t^i)^k = (u_{0|t}^i)^k$. In what follows, we pursue an explicit approximation of π_k , denoted by $\hat{\pi}_k$, exploiting the approach called "imitation learning."

Imitation learning is an approach for an autonomous agent to learn a control policy from expert demonstrations to imitate expert's behavior. The expert can be a human expert [44] or a well-behaved artificial intelligence [45]. In this paper, we treat the algorithmically determined map π_k as the expert.

Imitation learning can be formulated as a standard supervised learning problem, in which case it is also commonly referred to as "behavioral cloning," where the learning objective is to obtain a policy from a pre-collected dataset of expert demonstrations that best approximates the expert's behavior at the states contained in the dataset. Such a procedure can be described as

$$\hat{\pi}_k \in \underset{\pi_\theta}{\operatorname{argmin}} \mathbb{E}_{\bar{\mathbf{s}} \sim \mathbb{P}(\bar{\mathbf{s}}|\pi_k)} [\mathcal{L}(\pi_k(\bar{\mathbf{s}}), \pi_\theta(\bar{\mathbf{s}}))], \quad (7)$$

where $\bar{\mathbf{s}}$ denotes the triple $(s^i, \mathbf{s}^{-i}, k)$, π_k denotes the expert policy (6), π_θ denotes a policy parameterized by θ (e.g., the weights of a neural network) that is being evaluated and optimized, \mathcal{L} is a loss function, and the notation $\mathbb{E}_{\bar{\mathbf{s}} \sim \mathbb{P}(\bar{\mathbf{s}}|\pi_k)}(\cdot)$ is defined as

$$\mathbb{E}_{\bar{\mathbf{s}} \sim \mathbb{P}(\bar{\mathbf{s}}|\pi_k)}(\cdot) = \int (\cdot) d\mathbb{P}(\bar{\mathbf{s}}|\pi_k). \quad (8)$$

We remark that a key feature of the procedure (7) is that the expectation is with respect to the probability distribution $\mathbb{P}(\bar{\mathbf{s}}|\pi_k)$ of the data $\bar{\mathbf{s}}$ determined by the expert policy π_k , which is essentially the empirical distribution of $\bar{\mathbf{s}}$ in the pre-collected dataset.

In our previous work [39], we have explored the procedure (7) to obtain an explicit policy that imitates level- k decisions for an autonomous vehicle to drive through a roundabout intersection.

Using (7) to train the policy $\hat{\pi}_k$ has a drawback in that only the states that can be reached by executing the expert policy π_k will be included in the dataset, and such a sampling bias may cause the error between $\hat{\pi}_k$ and π_k to propagate in time. In particular, a small error may cause the vehicle to reach a state that is not exactly included in the dataset and, consequently, a large error may occur at the next time step.

Therefore, in this paper we consider an alternative approach, based on the ‘‘Dataset Aggregation’’ (DAgger) algorithm, to train the policy $\hat{\pi}_k$. DAgger is an iterative algorithm that optimizes the policy under its induced state distribution [46]. The learning objective of DAgger can be described as

$$\hat{\pi}_k \in \underset{\pi_\theta}{\operatorname{argmin}} \mathbb{E}_{\bar{\mathbf{s}} \sim \mathbb{P}(\bar{\mathbf{s}}|\pi_\theta)} [\mathcal{L}(\pi_k(\bar{\mathbf{s}}), \pi_\theta(\bar{\mathbf{s}}))], \quad (9)$$

$$\mathbb{E}_{\bar{\mathbf{s}} \sim \mathbb{P}(\bar{\mathbf{s}}|\pi_\theta)}(\cdot) = \int (\cdot) d\mathbb{P}(\bar{\mathbf{s}}|\pi_\theta), \quad (10)$$

where the distinguishing feature from (7) is that the expectation is with respect to the probability distribution $\mathbb{P}(\bar{\mathbf{s}}|\pi_\theta)$ induced from the policy π_θ that is being evaluated and optimized. DAgger can effectively resolve the aforementioned issue with regard to the propagation of error in time, since there will be data points $(\bar{\mathbf{s}}, \pi_k(\bar{\mathbf{s}}))$ for states $\bar{\mathbf{s}}$ reached by executing $\hat{\pi}_k$.

The procedure to obtain explicit level- k decision policies based on an improved version of DAgger algorithm [45] is presented as Algorithm 1. In Algorithm 1, n_{\max} denotes the maximum number of simulation episodes and t_{\max} represents the length of a simulation episode. By ‘‘initialize the simulation environment,’’ we mean constructing a traffic scene, including specifying the road layout and geometry as well as the number of vehicles. By ‘‘initialize vehicle i ,’’ we mean putting the vehicle in a lane entering the scene while satisfying a minimum separation distance constraint from the other vehicles, and specifying a sequence of target lanes for the vehicle to traverse and finally leave the scene. By ‘‘vehicle i fails,’’ we mean the occurrence of 1) vehicle i ’s c -zone overlapping with any of the other vehicles’ c -zones, 2) crossing any of the road boundaries, or 3) crossing a lane marking that separates the traffic of opposite directions. And, by ‘‘vehicle i succeeds,’’ we mean vehicle i gets to the last target lane in its target lane sequence so that it can leave the scene without further interactions with the other vehicles.

IV. TRAFFIC IN UNSIGNALIZED INTERSECTION NETWORK

We model urban traffic where the road system is composed of straight roads and three most common types of unsignalized intersections: four-way, T-shaped, and roundabout [47]. Such traffic models can be used as simulation environments for

Algorithm 1 Imitation Learning Algorithm to Obtain Explicit Level- k Decision Policies

```

1 Initialize  $\hat{\pi}_k^0$  to an arbitrary policy;
2 Initialize dataset  $\mathcal{D} \leftarrow \emptyset$ ;
3 for  $n = 1 : n_{\max}$  do
4   Initialize the simulation environment;
5   for  $i \in \mathcal{M}$  do
6     Initialize vehicle  $i$ ;
7   end for
8   for  $t = 0 : t_{\max} - 1$  do
9     for  $i \in \mathcal{M}$  do
10      if vehicle  $i$  fails or succeeds then
11        Re-initialize vehicle  $i$ ;
12      end if
13      for  $k = 1 : k_{\max}$  do
14        if  $\hat{\pi}_k^{n-1}(s_t^i, \mathbf{s}_t^{-i}, k) \neq \pi_k(s_t^i, \mathbf{s}_t^{-i}, k)$  then
15           $\mathcal{D} \leftarrow \mathcal{D} \cup ((s_t^i, \mathbf{s}_t^{-i}, k), \pi_k(s_t^i, \mathbf{s}_t^{-i}, k))$ 
16        end if
17      end for
18      Randomly generate  $k_t \in \{1, \dots, k_{\max}\}$ ;
19       $s_{t+1}^i = f(s_t^i, \hat{\pi}_k^{n-1}(s_t^i, \mathbf{s}_t^{-i}, k_t))$ ;
20    end for
21  end for
22  Train classifier  $\hat{\pi}_k^n$  on  $\mathcal{D}$ ;
23 end for
24 Output  $\hat{\pi}_k = \hat{\pi}_k^{n_{\max}}$ .

```

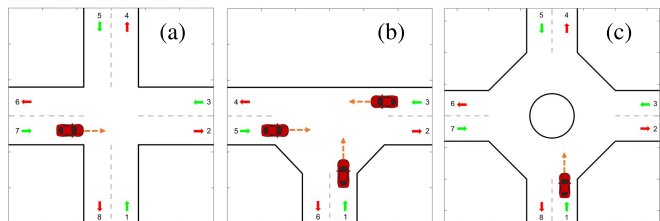


Fig. 1. Unsignalized intersections to be modeled: (a) four-way, (b) T-shaped, and (c) roundabout.

virtual testing of AV control systems, which will be introduced in Section V.

The three unsignalized intersections to be modeled are illustrated in Fig. 1. A vehicle can come from any of the entrance lanes (marked by green arrows) to enter an intersection and go to any of the exit lanes (marked by red arrows) to leave it, except that U-turns are not allowed for four-way and T-shaped intersections.

When training the level- k policy $\hat{\pi}_k$ using Algorithm 1, we treat these three unsignalized intersections separately. Specifically, when initializing the simulation environment in step 4, we select one of these three unsignalized intersections as the traffic scene for the current simulation episode. In addition, since in this paper we only consider these three unsignalized intersections, their layout and geometry features can be characterized and distinguished using a label $\zeta \in \{1, 2, 3\}$, i.e., the state ζ^i of vehicle i takes the value 1 when vehicle i operates in the area of the four-way intersection,

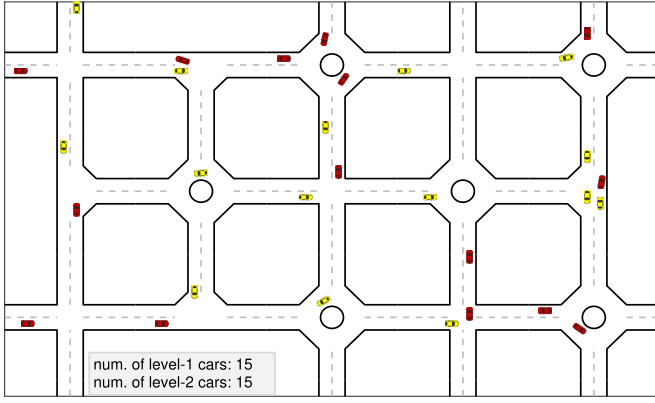


Fig. 2. An urban traffic scenario with 15 level-1 cars (yellow) and 15 level-2 cars (red).

2 for the T-shaped intersection, and 3 for the roundabout. For more intersection types with various layout and geometry features, a higher dimensional vector ζ may be used (e.g., see the intersection model in [35]).

Once the policy $\hat{\pi}_k$ for each of these three unsignalized intersections has been obtained, we can model larger road systems using these three unsignalized intersections as modules and assembling them in arbitrary ways. Fig. 2 shows an example of assembly. When a vehicle operates at/near to a specific intersection, it uses a local coordinate system, accounts for its interactions with only the vehicles in an immediate vicinity, and applies the $\hat{\pi}_k$ corresponding to this intersection.

To model the heterogeneity in driving styles of different drivers, we let different vehicles be of different reasoning levels. Specifically, a level- k vehicle is controlled by the policy:

$$\hat{\pi}_k = \hat{\pi}_k(\cdot, \cdot, k) : (s_t^i, \mathbf{s}_t^{-i}) \mapsto (u_t^i)^k. \quad (11)$$

For instance, in Fig. 2 the 15 yellow cars are level-1 and the 15 red cars are level-2.

V. AUTONOMOUS VEHICLE CONTROL APPROACHES

In this section, we describe two AV control approaches for urban traffic environments with unsignalized intersections. These approaches will be tested and calibrated using our traffic model, thereby demonstrating its utility for verification and validation.

A. Adaptive Control Based on Level- k Models

In this approach, the autonomous ego vehicle treats the other drivers as level- k drivers. As different drivers may behave corresponding to different reasoning levels, the ego vehicle estimates their levels and adapts its own control strategy based on the estimation results.

The control strategy of the autonomous ego vehicle, i , can be described as: At each discrete-time instant t , vehicle i

solves for

$$\begin{aligned} (\mathbf{u}_t^i)^a &= \{(u_{0|t}^i)^a, (u_{1|t}^i)^a, \dots, (u_{N-1|t}^i)^a\} \\ &\in \arg \max_{\mathbf{u}_t^i \in \mathcal{U}^N} \sum_{\tau=0}^{N-1} \lambda^\tau R(s_{\tau|t}^i, \mathbf{s}_{\tau|t}^{-i}, u_{\tau|t}^i, (\mathbf{u}_{\tau|t}^{-i})^{\tilde{k}}), \end{aligned} \quad (12)$$

where $(\mathbf{u}_{\tau|t}^{-i})^{\tilde{k}} = ((u_{\tau|t}^j)^{\tilde{k}_t^j})_{j \in \mathcal{M}, j \neq i}$ denotes the collection of predicted actions of the other vehicles. In particular, the actions of vehicle j , $u_{\tau|t}^j$, $\tau = 0, 1, \dots, N-1$, are predicted by modeling vehicle j as level- \tilde{k}_t^j and are solved for according to (5), where \tilde{k}_t^j is determined according to the following maximum likelihood principle:

$$\tilde{k}_t^j \in \arg \max_{k \in \mathcal{K}} \mathbb{P}^i(k^j = k|t), \quad (13)$$

in which $\mathbb{P}^i(k^j = k|t)$ represents vehicle i 's belief at time t in that vehicle j can be modeled as level- k , with k taking values in a model set \mathcal{K} . The beliefs $\mathbb{P}^i(k^j = k|t)$ get updated after each time step according to the following algorithm: If there exist $k, k' \in \mathcal{K}$ such that $\pi_k(s_t^j, \mathbf{s}_t^{-j}, k) \neq \pi_{k'}(s_t^j, \mathbf{s}_t^{-j}, k')$, then

$$\begin{aligned} \mathbb{P}^i(k^j = k|t+1) &= \frac{p^i(k^j = k|t+1)}{\sum_{k' \in \mathcal{K}} p^i(k^j = k'|t+1)}, \\ p^i(k^j = k|t+1) &= \begin{cases} (1-\beta)\mathbb{P}^i(k^j = k|t) + \beta & \text{if } k = \hat{k}_t^j, \\ \mathbb{P}^i(k^j = k|t), & \text{otherwise,} \end{cases} \end{aligned} \quad (14)$$

where $\beta \in [0, 1]$ represents an update step size, and

$$\begin{aligned} \hat{k}_t^j &\in \operatorname{argmin}_{k \in \mathcal{K}} \operatorname{dist}(u_t^j, (u_t^j)^k), \\ &= \sqrt{(a_t^j - (a_t^j)^k)^2 + (\omega_t^j - (\omega_t^j)^k)^2}; \end{aligned} \quad (15)$$

if $\pi_k(s_t^j, \mathbf{s}_t^{-j}, k) = \pi_{k'}(s_t^j, \mathbf{s}_t^{-j}, k')$ for all $k, k' \in \mathcal{K}$, then $\mathbb{P}^i(k^j = k|t+1) = \mathbb{P}^i(k^j = k|t)$ for all $k \in \mathcal{K}$.

The level estimation algorithm (13)-(15) has the following three features: 1) If the actions predicted by all of the models in \mathcal{K} are the same, then the autonomous ego vehicle has no information to distinguish their relative accuracy and thus maintains its previous beliefs. 2) Otherwise, the ego vehicle identifies the model(s) in \mathcal{K} whose prediction $(u_t^j)^k$ matches vehicle j 's actually applied action u_t^j for time t with the highest accuracy. 3) The ego vehicle improves its belief(s) in that model(s) from its previous beliefs. This way, it takes into account both its previous estimates and the current, latest estimate.

Similar to (6) defined by (5), we can define a policy to represent the control determined by (12) as follows:

$$\pi_a : (s_t^i, \mathbf{s}_t^{-i}, \tilde{\mathbf{k}}_t^{-i}) \mapsto (u_t^i)^a, \quad (16)$$

where $\tilde{\mathbf{k}}_t^{-i} = (\tilde{k}_t^j)_{j \in \mathcal{M}, j \neq i}$ denotes the collection of level estimates of the other vehicles and $(u_t^i)^a = (u_{0|t}^i)^a$ is determined by (12). Furthermore, we can train an explicit approximation $\hat{\pi}_a$ to π_a using a similar imitation learning procedure as that for training the explicit approximation $\hat{\pi}_k$ to π_k . This way, together with replacing π_k with $\hat{\pi}_k$ in the level estimation

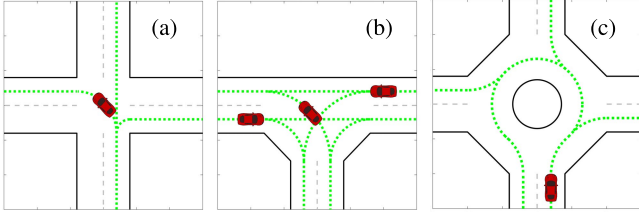


Fig. 3. Reference paths for the autonomous ego vehicle to drive through (a) four-way, (b) T-shaped, and (c) roundabout intersections.

algorithm (13)-(15), we can move the major computational tasks involved in this adaptive control approach (12)-(15) offline, and thus, render its online computational feasibility.

The algorithm to train $\hat{\pi}_a$ using DAgger with π_a as the expert policy is similar to Algorithm 1 and is omitted.

B. Rule-Based control

The second AV control approach that we consider is a rule-based solution. Compared to many other approaches, rule-based control has the advantage of interpretability and can often be calibrated by tuning a small number of parameters.

In this approach, the autonomous ego vehicle drives by following a pre-planned reference path and accounts for its interactions with other vehicles by adjusting its speed along the path correspondingly. Examples of reference paths for the autonomous ego vehicle to drive through intersections are illustrated by the green dotted curves in Fig. 3.

The basic control rules can be explained as follows: The autonomous ego vehicle pursues a higher speed along the reference path if there is no other vehicle in conflict with it. If there are other vehicles in conflict with it, then the autonomous ego vehicle yields to them by maximizing distances from them. Specifically, at each discrete-time instant t , the autonomous ego vehicle, i , selects and applies for one time step an acceleration value from a finite set of accelerations, \mathcal{A} , according to Algorithm 2.

Algorithm 2 Rule-Based Autonomous Vehicle Control Algorithm

```

1 Initialize  $\mathcal{M}_c \leftarrow \emptyset$ ;
2 for  $j \in \mathcal{M}, j \neq i$  do
3   if the estimated future path of  $j$  intersects with  $i$ 's
   future path and  $\text{dist}((x_t^i, y_t^i), (x_t^j, y_t^j)) \leq R_c$  then
4      $\mathcal{M}_c \leftarrow \mathcal{M}_c \cup \{j\}$ ;
5   end if
6 end for
7 if  $\mathcal{M}_c \neq \emptyset$  then
8    $(a_t^i)^r =$ 
    $\arg \max_{a \in \mathcal{A}} [\min_{j \in \mathcal{M}_c} \text{dist}((x_{1|t}^i, y_{1|t}^i), (x_{1|t}^j, y_{1|t}^j))]$ ;
9 else
10   $(a_t^i)^r = \max\{a \in \mathcal{A}\}$ ;
11 end if
12 Output  $(a_t^i)^r$ .
```

In Algorithm 2, \mathcal{M}_c represents the set of vehicles that are in conflict with the ego vehicle. In particular, the ego vehicle estimates each of the other vehicles' future paths based on their current positions and their target lanes and using the same path planning algorithm that is used by the ego vehicle to create its own path. If the estimated future path of a vehicle j intersects with the ego vehicle's own future path and the current distance between these two vehicles is smaller than a threshold value R_c , then vehicle j is identified as a vehicle in conflict, i.e., $j \in \mathcal{M}_c$. In particular, the distance function $\text{dist}(\cdot, \cdot)$ measures the Euclidean distance.

If there are vehicles in conflict, $\mathcal{M}_c \neq \emptyset$, then the ego vehicle maximizes the minimum among the predicted distances from these vehicles to improve safety. In step 8, $(x_{1|t}^i, y_{1|t}^i)$ represents the predicted position of the ego vehicle i by driving along its reference path for one step with the speed after applying the acceleration a , and $(x_{1|t}^j, y_{1|t}^j)$ represents the predicted position of vehicle j by driving along its current heading direction with its current speed. If there is no vehicle in conflict, $\mathcal{M}_c = \emptyset$, then the ego vehicle maximizes its speed.

Note that the key parameter for this rule-based control approach is the threshold value R_c . It determines both whether a vehicle will be identified as in conflict with the ego vehicle and the separation distance that the ego vehicle tries to keep from other vehicles. We will utilize our traffic model to calibrate R_c in Section VI-C.

VI. RESULTS

In this section, we show simulation results of our level-k game theory-based vehicle interaction model, and illustrate its application to the verification, validation and calibration of AV control systems.

A. Level-k Vehicle Models

We consider a sampling period $\Delta t = 0.25$ [s] and an action set \mathcal{U} consisting of 6 actions representing common driving maneuvers in urban traffic, listed in Table I. The weight vector, the planning horizon, and the discount factor for the reward function (4) are $\mathbf{w} = [1000, 500, 50, 100, 5, 1]^T$, $N = 4$, and $\lambda = 0.8$. When evaluating the features ϕ_1 and ϕ_4 , we consider the c -zone of a vehicle as a $5[\text{m}] \times 2[\text{m}]$ rectangle centered at the vehicle's position (x, y) and stretched along its heading direction θ , and the s -zone of a vehicle as a rectangle concentric with its c -zone and $8[\text{m}] \times 2.4[\text{m}]$ in size. Furthermore, we consider a speed range $[v_{\min}, v_{\max}] = [0, 5][\text{m/s}]$. When the speed calculated based on the model (2) gets outside $[v_{\min}, v_{\max}]$, it is saturated to this range. We note that $[v_{\min}, v_{\max}] = [0, 5][\text{m/s}]$ is a reasonable range to represent common speeds for vehicles to drive through unsignalized intersections. For instance, in California it is suggested to maintain the vehicle speed below 15[mpg] when traversing an uncontrolled highway intersection [48].

Experimental studies [37], [49] suggest that humans are most commonly level-1 and -2 reasoners in their interactions. Therefore, we model vehicles in traffic using level-1 and -2 policies in this paper. In particular, on the basis of our level-0 decision rule (see Section III-A), a level-1 vehicle represents a

TABLE I
ACTION SET \mathcal{U}

action u	a [m/s ²]	ω [rad/s]
maintain (u_1)	0	0
accelerate (u_2)	2.5	0
decelerate (u_3)	-2.5	0
hard brake (u_4)	-5	0
turn left (u_5)	0	$\pi/4$
turn right (u_6)	0	$-\pi/4$

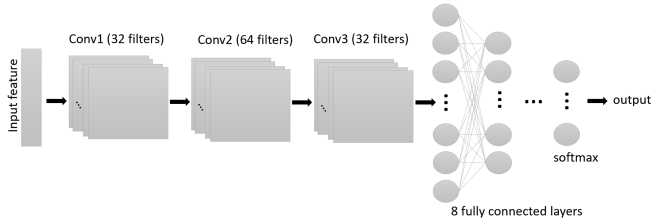


Fig. 4. Architecture of the neural network.

cautious/conservative vehicle and a level-2 vehicle represents an aggressive vehicle. Indeed, as level-0 and level-2 vehicles both represent aggressive vehicles, they behave similarly in many situations.

We use a neural network with the architecture shown in Fig. 4 to represent a policy π_θ and train its weights θ using Algorithm 1 to obtain an explicit approximation $\hat{\pi}_k$ to the level- k policy π_k , which is algorithmically determined based on (5). The accuracy of the obtained $\hat{\pi}_k$ in terms of matching π_k on the training dataset is 98.3%. Then, we generate 30% more data points of $((s_t^i, s_t^{-i}, k), \pi_k(s_t^i, s_t^{-i}, k))$ for testing. The accuracy of $\hat{\pi}_k$ in terms of matching π_k on the test dataset is 97.8%. As has been discussed at the beginning of Section III-B, the reason for generating $\hat{\pi}_k$ is to move the numerical computations for determining the level- k decisions through (5) offline. With $\hat{\pi}_k$, the interactive decision-making processes of vehicles are reduced to function evaluations (here, the function is expressed as a neural network). This way, the online simulations of traffic scenarios, used as environments for virtual testing of AV control systems, can be significantly accelerated.

To show the advantage of using the DAgger algorithm (9) over using a standard supervised learning procedure (7) to obtain the policy $\hat{\pi}_k$, we show a case observed in our simulations where the policy trained using standard supervised learning fails but the one trained using DAgger succeeds. In Fig. 5(a-3), the blue vehicle controlled by $\hat{\pi}_k$ trained using standard supervised learning fails in making an adequate right turn to get around the central island. This is due to a significant error of $\hat{\pi}_k$ from π_k at certain states encountered by the blue vehicle when entering the roundabout, and the encounter with such states results from the issue of error propagation in time that has been discussed in Section III-B. In contrast, the blue vehicle in Fig. 5(b-3) controlled by $\hat{\pi}_k$ trained using DAgger succeeds in making a proper right turn, illustrating the effectiveness of DAgger in avoiding such an issue.

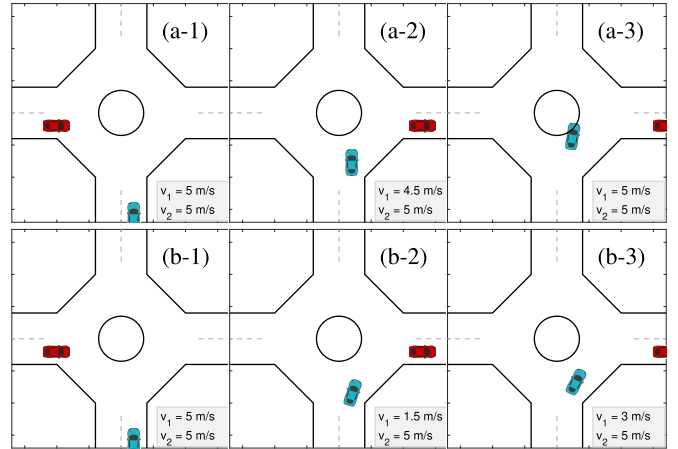


Fig. 5. (a-1)-(a-3) show three sequential steps in a simulation where the blue vehicle controlled by $\hat{\pi}_k$ trained using standard supervised learning fails in making an adequate right turn to get around the central island of a roundabout; (b-1)-(b-3) show steps in a similar simulation where the blue vehicle controlled by $\hat{\pi}_k$ trained using DAgger succeeds in making a proper right turn.

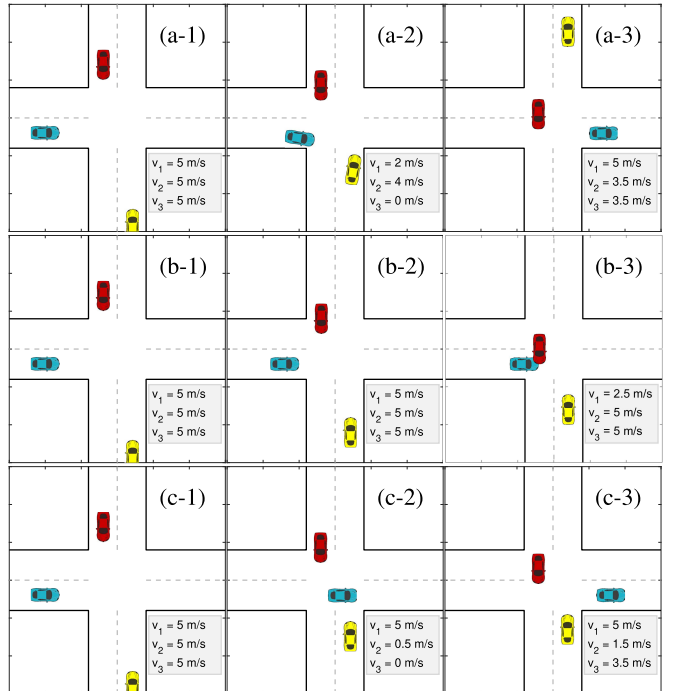


Fig. 6. Interactions of level- k vehicles at the four-way intersection. (a-1)-(a-3) show three sequential steps in a simulation where three level-1 vehicles interact with each other; (b-1)-(b-3) show steps of three level-2 vehicles interacting with each other; (c-1)-(c-3) show steps of a level-2 vehicle (blue) interacting with two level-1 vehicles (yellow and red); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

In what follows we show the interactions between level- k vehicles at the four-way, T-shaped, and roundabout intersections. In particular, we let three vehicles be controlled by different level- k policies and show how the traffic scenarios evolve differently depending on the different combinations of level- k policies.

It can be observed from Figs. 6-8 that, in general, when level-1 and level-2 vehicles interact with each other, the

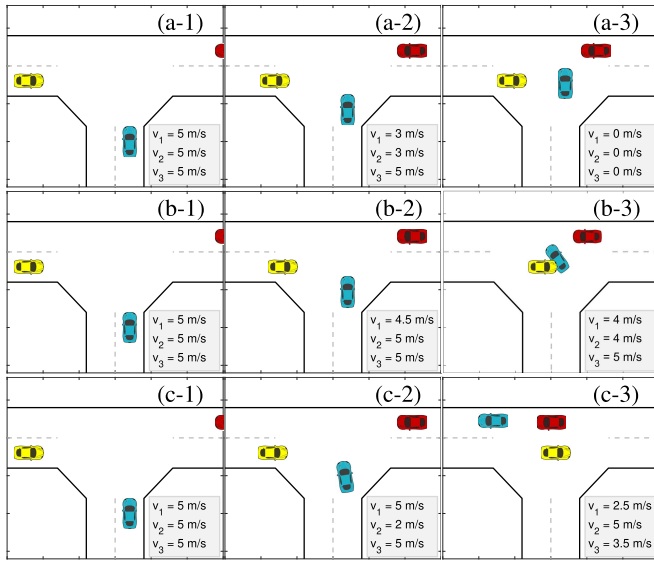


Fig. 7. Interactions of level- k vehicles at the T-shaped intersection. (a-1)-(a-3) show three sequential steps in a simulation where three level-1 vehicles interact with each other; (b-1)-(b-3) show steps of three level-2 vehicles interacting with each other; (c-1)-(c-3) show steps of a level-2 vehicle (blue) interacting with two level-1 vehicles (yellow and red); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

conflicts between them can be resolved. This is expected since level-1 vehicles, representing cautious/conservative vehicles, will yield the right of way and level-2 vehicles, representing aggressive vehicles, will proceed ahead. In contrast, when level-1 vehicles interact with level-1 vehicles, deadlocks may occur, such as the one being observed in the T-shaped intersection in Fig. 7(a), because everyone yields to the others. When level-2 vehicles interact with level-2 vehicles, collisions may occur, such as the ones being observed in panel (b) of Figs. 6-8, because everyone assumes the others would yield.

We remark that deadlocks (collisions) do not always occur in level-1 (level-2) interactions. The initial conditions of Figs. 6-8 are chosen to show such situations. For randomized initial conditions, the rates of success, defined as the proportion of 2000 simulation episodes where neither deadlocks nor collisions occur to the ego vehicle, for different numbers of interacting vehicles and different combinations of level- k policies at the three intersections are shown in Fig. 9. In Fig. 9, “L- k car in L- k' Env.” shows the rate of success of a level- k ego vehicle interacting with other vehicles that are all level- k' ; “L- k car in Mix Env.” shows the rate of success of a level- k ego vehicle interacting with other vehicles whose control policies are randomly chosen between level-1 and level-2 with equal probability.

The following observations can be made: 1) As the number of interacting vehicles increases, the rate of success decreases for all the cases. This is reasonable since a larger number of interacting vehicles represents a more complex traffic scenario. 2) The rates of success of a level-2 ego vehicle interacting with other vehicles that are also level-2 are the lowest among the results of all combinations of level- k policies. This is also reasonable since when all the vehicles are aggressive and assume the others would yield, traffic accidents are more

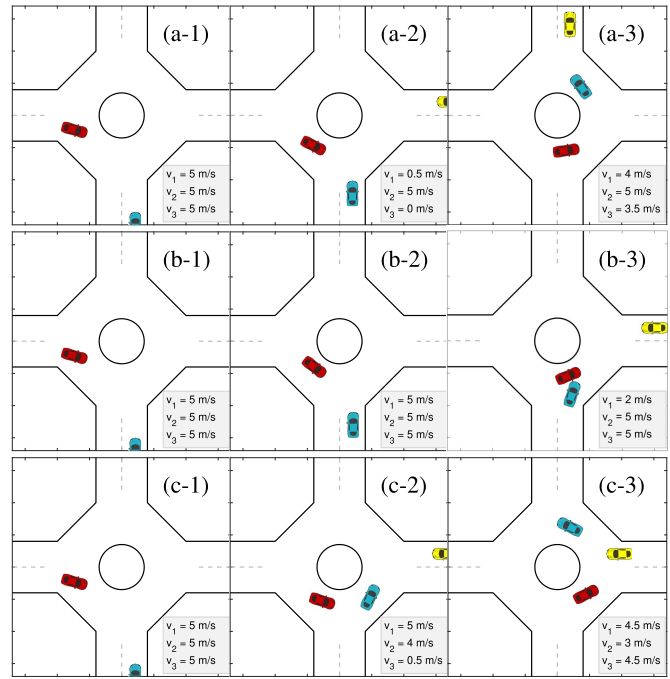


Fig. 8. Interactions of level- k vehicles at the roundabout intersection. (a-1)-(a-3) show three sequential steps in a simulation where three level-1 vehicles interact with each other; (b-1)-(b-3) show steps of three level-2 vehicles interacting with each other; (c-1)-(c-3) show steps of a level-2 vehicle (blue) interacting with two level-1 vehicles (yellow and red); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

likely to occur. 3) Among the results of the three intersection types, the rates of success for the roundabout intersection are the highest. This illustrates the effective functionality of roundabouts in reducing traffic conflicts.

We further remark that although the high rates of failure of “level-2 versus level-2” are not desired in real-world traffic, it is important for a simulation environment for AV control testing to include such cases that represent rational interactions between aggressive vehicles. Note that a level-2 vehicle is a rational decision maker that behaves aggressively, which is fundamentally different from a driver/vehicle model that acts aggressively but in an irrational way, e.g., taking actions randomly. The cases of level-2 vehicle interactions provide challenging test scenarios for AV control systems, which can be more realistic than those provided by some worst-case (i.e., not necessarily rational) models [50].

B. Model Validation

We validate our level- k vehicle models before illustrating how to use them for AV control testing.

1) *Feasibility Validation*: The unicycle model (2) has been used to represent vehicle kinematics and the action set \mathcal{U} in Table I has been used to represent common driving maneuvers. We now show that the trajectories generated by (2) with actions from \mathcal{U} are feasible trajectories for vehicle systems. For this, we use a hybrid kinematic/dynamic bicycle model with the brush tire model [51] to represent high-fidelity vehicle dynamics, and we use a PID-based controller [52] to control the vehicle dynamics to execute the trajectory generated by (2)

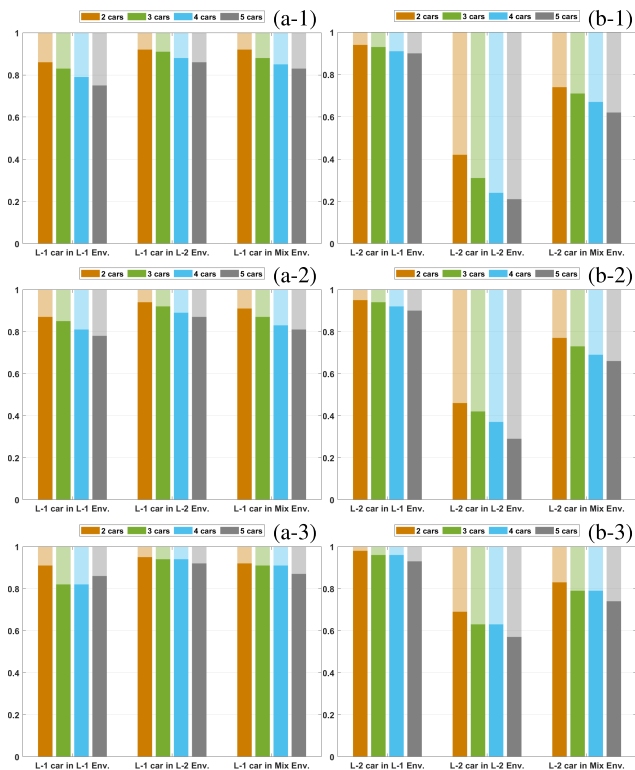


Fig. 9. The rates of success of level- k policies. (a-1)-(a-3) show the rates of success of a level-1 ego vehicle operating in various traffic environments (various in the numbers and policies of interacting vehicles) at the four-way, T-shaped, and roundabout intersections; (b-1)-(b-3) show those of a level-2 ego vehicle; the bars in dark color represent the rates of success.

and \mathcal{U} . Specifically, at each discrete-time instant t the level- k decision policy selects an action from \mathcal{U} , which defines a desired state s_{t+1} for the vehicle system through the unicycle model (2). Then, over the continuous-time interval from t to $t + 1$, the PID-based controller controls the vehicle dynamics to track the desired state s_{t+1} .

Two examples of tracking results for the T-shaped and the roundabout intersections are shown in Fig. 10, where the red solid curves represent the trajectories generated by (2) (referred to as “reference trajectories”) and the black dotted curves represent the tracking trajectories. It can be observed that the tracking trajectories closely match the reference trajectories. This justifies the feasibility of trajectories generated by the unicycle model (2) and the action set \mathcal{U} .

2) *Comparison to Traffic Data:* We next validate our level- k vehicle models using real-world traffic data.

In Fig. 11, we show two traffic scenarios at a T-shaped intersection extracted from the INTERACTION dataset [53] and their reproduction by simulating our level- k vehicle models. Specifically, we initialize the states of our level- k vehicle models according to the initial scene of the scenario, and compare the evolution of the scenario simulated by our models to the actual one from data. It can be seen that the simulated evolution accurately matches the actual evolution for both cases.

We also compare the average speeds of our level- k vehicle models and of actual vehicles in the dataset when traversing

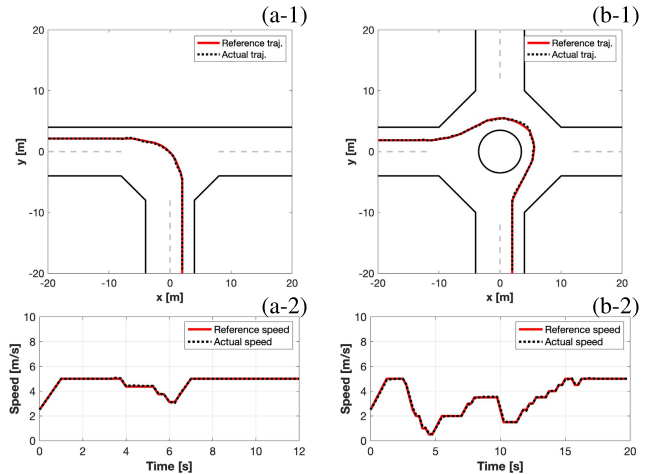


Fig. 10. Feasibility validation of the unicycle model (2) and the action set \mathcal{U} . (a-1)-(a-2) show an example of path and speed tracking result at a T-shaped intersection; (b-1)-(b-2) show that at a roundabout intersection.

T-shaped intersections. The average speeds versus the numbers of interacting vehicles are plotted in Fig. 12, where the 95% confidence intervals of data are indicated by the vertical error bars. It can be seen that the average speeds of our level- k vehicle models are lower than the average speeds of actual vehicles. This is because some vehicles in the dataset drive at a speed that is higher than the speed upper bound $v_{\max} = 5$ [m/s] of our models, and is also due to some differences in the road layout and geometry (e.g., three-lane versus two-lane on the left, see Fig. 11). Also, only 56 scenarios at this T-shaped intersection are contained in the dataset and used to compute the average speed results of the red curve. This causes the relatively large error bars. In contrast, we run 2000 simulation episodes with randomized level- k policy combinations and initial conditions to compute the average speed results of the blue curve. So the error bars are relatively small. In summary, similar trends of average speeds versus numbers of interacting vehicles are exhibited between the simulation results of our level- k vehicle models and the traffic data. Also note that the 95% confidence intervals of our models are contained in the 95% confidence intervals of the traffic data.

C. Evaluation and Calibration of Autonomous Vehicle Control Approaches

We test the two AV control approaches described in Section V using a simulation environment constructed based on level- k vehicle models.

For the first approach of adaptive control based on level- k models, we use the same sampling period Δt , action set \mathcal{U} , reward function including the weight vector \mathbf{w} , planning horizon N , and discount factor λ as those used for the level- k vehicle models. In the level estimation algorithm (13)-(15), we consider the model set $\mathcal{K} = \{1, 2\}$ and the update step size $\beta = 0.6$.

When training the explicit approximation $\hat{\pi}_a$ to the policy π_a that is algorithmically determined by (12), we use the same neural network architecture shown in Fig. 4. The accuracy

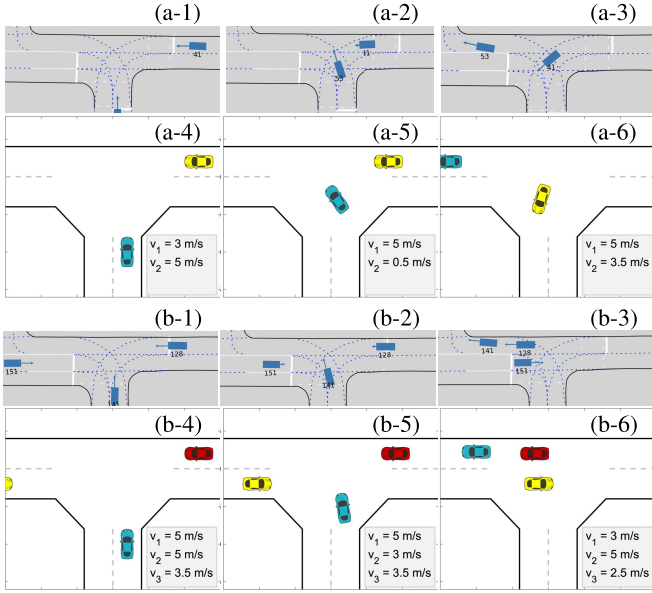


Fig. 11. Reproduction of real-world traffic scenarios using our level- k vehicle models. (a-1)-(a-3) visualize a traffic scenario with two interacting vehicles extracted from the dataset [53] at three sequential time instants; (a-4)-(a-6) show the simulation results of a level-2 vehicle (blue) interacting with a level-1 vehicle (yellow) in a similar scenario. (b-1)-(b-3) visualize a traffic scenario with three interacting vehicles extracted from the same dataset; (b-4)-(b-6) show the simulation results of a level-2 vehicle (blue) interacting with two level-1 vehicles (yellow and red) in a similar scenario.

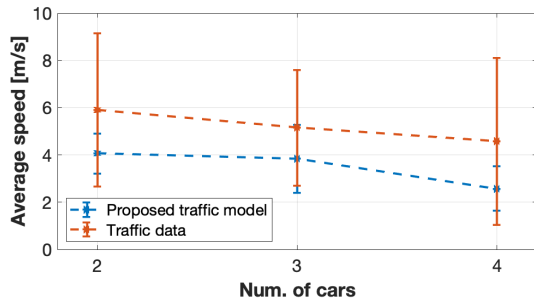


Fig. 12. Average speeds versus numbers of interacting vehicles for traversing T-shaped intersections of level- k vehicle models and traffic data.

of the obtained $\hat{\pi}_a$ in terms of matching π_a is 98.8% on the training dataset and is 98.6% on a test dataset of 30% additional data points that are not used for training.

Firstly, we simulate similar scenarios as those shown in Figs. 6-8, but let the autonomous ego vehicle (blue) be controlled by the adaptive control approach instead of level- k policies. Figs. 13-15 show snapshots of the simulations. It can be observed that the autonomous ego vehicle can resolve the conflicts with the other two vehicles and safely drive through the intersections although the other two vehicles are controlled by varying policies. The bottom panels show the level estimation histories of the simulations. It can be observed that the autonomous ego vehicle can resolve the conflicts because it successfully identifies the level- k models of the other two vehicles. Recall that vehicle j is identified as level-1 (level-2) when $\mathbb{P}(k^j = 2) < 0.5$ ($\mathbb{P}(k^j = 2) \geq 0.5$).

The success of the adaptive control approach in situations where level- k control policies with fixed k fail suggests the

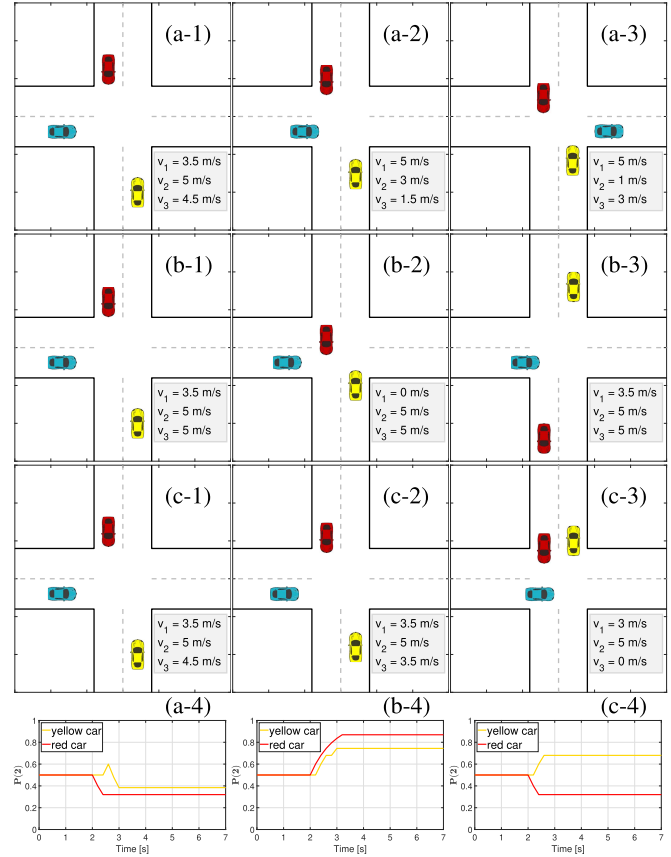


Fig. 13. Interactions of the autonomous ego vehicle (blue) controlled by the adaptive control approach with level- k vehicles at the four-way intersection. (a-1)-(a-3) show three sequential steps in a simulation where the autonomous ego vehicle interacts with two level-1 vehicles, and (a-4) shows the time histories of the two vehicles' level estimates where $\mathbb{P}(2) = \mathbb{P}(k = 2)$ denotes the ego vehicle's belief in the level-2 model; (b-1)-(b-4) show those of the autonomous ego vehicle interacting with two level-2 vehicles; (c-1)-(c-4) show those of the autonomous ego vehicle interacting with a level-1 vehicle (red) and a level-2 vehicle (yellow); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

significance in AV control of intention recognition and action prediction for the other vehicles. Note that these two steps are achieved in our adaptive control approach through the level estimates and the level- k models of the other vehicles.

We then statistically evaluate and compare the two AV control approaches. For the second approach of rule-based control, we consider an acceleration set $\mathcal{A} = \{-5, -2.5, 0, 2.5\}[\text{m/s}^2]$ and an initial design of the threshold value $R_c = 14[\text{m}]$.

In order to cover a rich set of scenarios, we construct a larger traffic scene shown in Fig. 16, which models the road system of an urban area in Los Angeles and consists of one four-way intersection, one roundabout, and two T-shaped intersections. We let an autonomous ego vehicle controlled by the adaptive control approach or the rule-based control approach drive through this traffic scene. Apart from the autonomous ego vehicle, we also put multiple other vehicles controlled by level- k policies in the scene and let them drive through the scene repeatedly. Their initial positions, lanes entering the scene, and sequences of target lanes to traverse the scene are all randomly chosen.

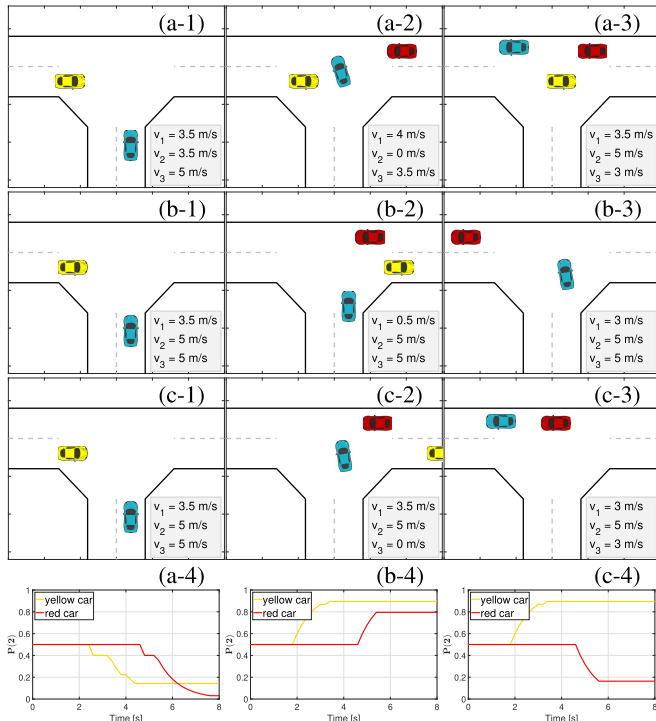


Fig. 14. Interactions of the autonomous ego vehicle (blue) controlled by the adaptive control approach with level- k vehicles at the T-shaped intersection. (a-1)–(a-3) show three sequential steps in a simulation where the autonomous ego vehicle interacts with two level-1 vehicles, and (a-4) shows the time histories of the two vehicles’ level estimates where $\mathbb{P}(2) = \mathbb{P}(k = 2)$ denotes the ego vehicle’s belief in the level-2 model; (b-1)–(b-4) show those of the autonomous ego vehicle interacting with two level-2 vehicles; (c-1)–(c-4) show those of the autonomous ego vehicle interacting with a level-1 vehicle (red) and a level-2 vehicle (yellow); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

We evaluate the two control approaches based on two statistical metrics: the rate of collision (CR) and the rate of deadlock (DR). The rate of collision is defined as the proportion of 2000 simulation episodes where the autonomous ego vehicle collides with another vehicle or with the road boundaries. The rate of deadlock is defined as the proportion of 2000 simulation episodes where no collision occurs to the autonomous ego vehicle but it fails to drive through the scene in 300[s] of simulation time. We consider three traffic models: 1) all of the other vehicles are level-1, called a “level-1 environment,” 2) all of the other vehicles are level-2, called a “level-2 environment,” and 3) the control policy of each of the other vehicles is randomly chosen between level-1 and level-2 with equal probability, called a “mixed environment.”

The CR and DR results of the adaptive control approach and the rule-based control approach for different numbers of other vehicles in the scene are shown in Figs. 17 and 18. The number of other vehicles, n_d , represents traffic density, roughly, $2.87n_d$ [vehicles/mile] (the total length of the roads is about 560 [m]).

From Fig. 17 it can be observed that, for the adaptive control approach, the CR and DR increase as the traffic density increases, which is reasonable. In particular, the increase in CR slows down as the number of other vehicles goes beyond 20. Among the results for different traffic models, the CR and DR

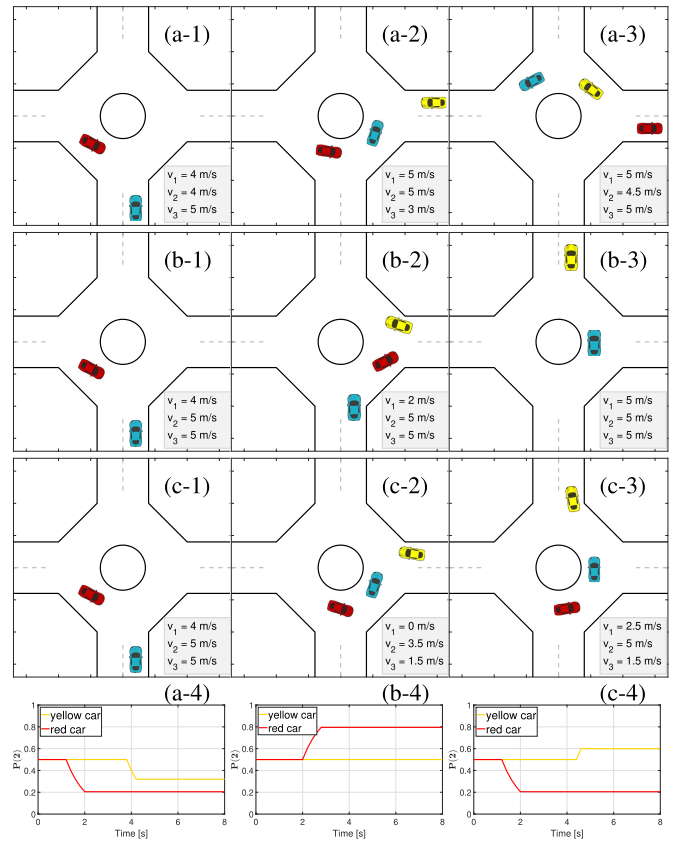


Fig. 15. Interactions of the autonomous ego vehicle (blue) controlled by the adaptive control approach with level- k vehicles at the roundabout intersection. (a-1)–(a-3) show three sequential steps in a simulation where the autonomous ego vehicle interacts with two level-1 vehicles, and (a-4) shows the time histories of the two vehicles’ level estimates where $\mathbb{P}(2) = \mathbb{P}(k = 2)$ denotes the ego vehicle’s belief in the level-2 model; (b-1)–(b-4) show those of the autonomous ego vehicle interacting with two level-2 vehicles; (c-1)–(c-4) show those of the autonomous ego vehicle interacting with a level-1 vehicle (red) and a level-2 vehicle (yellow); v_1 , v_2 and v_3 are the speeds of the blue, yellow and red vehicles, respectively.

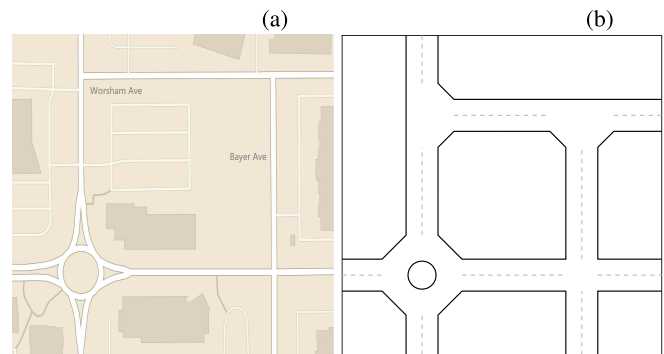


Fig. 16. Traffic scene for evaluating autonomous vehicle control approaches. (a) shows an urban area in Los Angeles (provided by Google Maps) and (b) shows the model of the road system in (a).

for the level-1 environment are the lowest and those for the level-2 environment are the highest. This is also reasonable since the level-1 environment, composed of level-1 vehicles, represents a cautious/conservative traffic model, the level-2 environment represents an aggressive traffic model and is thus most challenging for the autonomous ego vehicle, while the

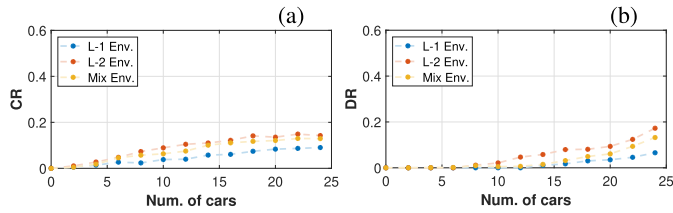


Fig. 17. Evaluation results of the adaptive control approach: (a) the rate of collision (CR) and (b) the rate of deadlock (DR) versus different numbers of environmental vehicles and different traffic models.

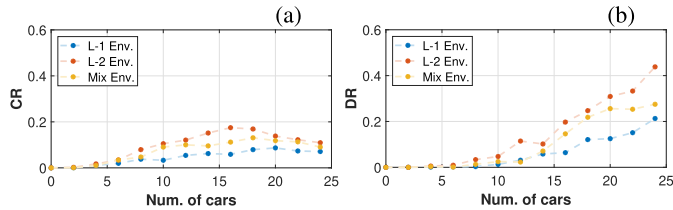


Fig. 18. Evaluation results of the rule-based control approach with $R_c = 14$ [m]: (a) the rate of collision (CR) and (b) the rate of deadlock (DR) versus different numbers of environmental vehicles and different traffic models.

mixed environment lies in between. Furthermore, the results for the adaptive control approach are less sensitive to changes in traffic models than those for level- k policies with fixed k shown in Fig. 9. This shows again the significance of adaptation of AV control strategy to other vehicles' intentions and actions. Note that the rate of success for a single intersection of the adaptive control approach, if estimated as $1 - \frac{CR+DR}{4}$, is close to that of "L-1 car in L-2 Env." and that of "L-2 car in L-1 Env.," which represent the best performance of level- k policies.

For the rule-based control approach, it can be observed from Fig. 18 that as the traffic density increases, the CR first increases and then decreases, while the DR keeps increasing. The decrease in CR when the traffic becomes very dense is due to the constant yielding of the autonomous ego vehicle to other vehicles, which causes the dramatic increase in DR.

Comparing the results of the two approaches, the adaptive control approach performs better than the rule-based control approach in the above experiments. This is attributed to the more sophisticated and complicated algorithm behind the adaptive control approach. However, the rule-based control is more interpretable (e.g., the reason for the decrease in CR is easily understood), and is easier to calibrate.

In Fig. 19, we show two informative cases observed in our simulations. In the first case in Fig. 19(a), the autonomous ego vehicle (blue) controlled by the adaptive control approach and the level-1 vehicle (yellow) on its left both yield to the other and cause a deadlock. Note that a level-1 vehicle represents a vehicle with a cautious/conservative driver, and accordingly, yields to the autonomous ego vehicle. Although the autonomous ego vehicle eventually decides to proceed ahead and successfully drives through the roundabout, it takes too long for such a conflict to be resolved, and thus this scenario falls into our DR category. To avoid such deadlock scenarios, the autonomous ego vehicle may need to identify the driving style of the opponent vehicle faster, which may be

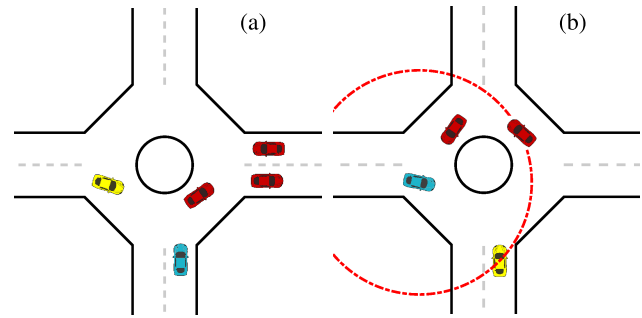


Fig. 19. Failure cases. (a) shows a scenario where the autonomous ego vehicle (blue) controlled by the adaptive control approach gets stuck at the entrance of the roundabout due to the level-1 vehicle (yellow) on its left. (b) shows a scenario where the autonomous ego vehicle (blue) controlled by the rule-based control approach gets hit by the level-2 vehicle (red) on its left.

achieved through a larger update step size β . In the second case in Fig. 19(b), the autonomous ego vehicle controlled by the rule-based control approach stops in the roundabout to yield to the yellow vehicle on its right and within the critical distance R_c (marked by the red dashed circle). However, because the gap between the autonomous ego vehicle and the yellow vehicle is still quite large, the red vehicle on the left of the autonomous ego vehicle expects it to proceed and thus does not slow down, which causes a collision. This scenario shows that a larger critical distance R_c may not always correspond to a safer driving behavior. We remark that failure/corner cases identified by our simulations, such as the above two cases, can also inform the design of specific test trajectories for AV control systems.

We now optimize the threshold value R_c in the rule-based control approach to achieve better performance defined by a performance index as follows:

$$J = \frac{1}{n_{\max}} \sum_{n=1}^{n_{\max}} \left(w_c \phi_c(\mathcal{S}_n) + w_d \phi_d(\mathcal{S}_n) + w_v \frac{\phi_s(\mathcal{S}_n)}{\bar{v}(\mathcal{S}_n) + \epsilon} \right), \quad (17)$$

where \mathcal{S}_n denotes the n th simulation episode; the $\phi_c(\mathcal{S}_n)$, $\phi_d(\mathcal{S}_n)$, and $\phi_s(\mathcal{S}_n)$ are indicator functions, taking 1 if, respectively, a collision occurs to the autonomous ego vehicle, no collision but a deadlock occurs to the autonomous ego vehicle, and neither collision nor deadlock occur and the autonomous ego vehicle successfully drives through the scene in 300[s] of simulation time in the n th simulation episode, and taking 0 otherwise; $\bar{v}(\mathcal{S}_n)$ is the average speed of the autonomous ego vehicle in the n th simulation episode; $w_c, w_d, w_v \geq 0$ are weighting factors, and $\epsilon > 0$ is a constant to adjust the shape of the function with respect to the average speed $\bar{v}(\mathcal{S}_n)$ and to avoid the denominator being 0.

The performance index function (17) imposes penalties for collisions and deadlocks through the first two terms, and rewards higher average speeds through the last term. Note that the last term is designed in such a way that the penalty increases fast for decrease in speed values that are already very low, and decreases slowly for increase in speed values that are already very high. In obtaining the following results, we run simulations in the same scene shown in Fig. 16 with

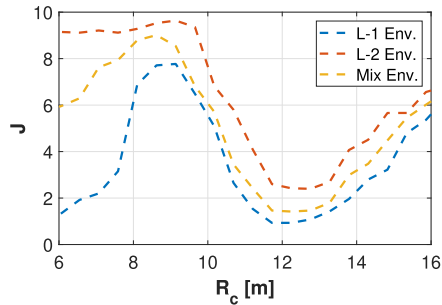


Fig. 20. Performance index J as function of R_c of the rule-based control approach with different traffic models.

15 other vehicles, and we use $w_c = 10$, $w_d = 5$, $w_v = 1$, and $\epsilon = 0.1$.

We plot the values of (17) for different values of R_c in Fig. 20. Specifically, for each value of R_c , we run $n_{\max} = 2000$ simulation episodes and calculate the value of (17) based on the simulation results. Lower values of (17) represent better performance in terms of having less collisions, less deadlocks, and higher average travel speeds.

In Fig. 20, the blue curve represents the result when the autonomous ego vehicle operates in the level-1 environment. It can be observed that the performance is good when R_c takes very small values, i.e., in the range of [6, 7.5][m]. This is because small R_c corresponds to aggressive behavior and the level-1 environment represents a conservative traffic model, thus, the other vehicles almost always yield to the autonomous ego vehicle when there is a conflict. Since the autonomous ego vehicle proceeds ahead while the other vehicles yield, collisions and deadlocks are avoided. However, when operating in the level-2 or mixed environment, small R_c leads to poor performance. This is because both the autonomous ego vehicle and the other vehicles behave aggressively and cause many collisions. When R_c takes values in the range of [7.5, 11][m], the performance is the worst for all of the three traffic models. This is because such R_c values correspond to behaviors between aggressive and conservative, which can cause collisions with both aggressive and conservative interacting vehicles. The range [11.5, 13][m] is suitable for choosing the value of R_c , where the performance is good and insensitive to changes in the traffic models. For larger R_c values, the autonomous ego vehicle becomes overly conservative and almost always yields to the other vehicles, which causes it difficulties to enter the intersections and leads to many deadlocks.

VII. CONCLUSION

In this paper, we described a framework based on level- k game theory for modeling traffic consisting of heterogeneous (in terms of their driving styles) and interactive vehicles in urban environments with unsignalized intersections. An algorithm integrating the level- k decision-making formalism, receding-horizon optimization, and imitation learning was proposed and used to solve for level- k control policies.

The developed traffic models are useful as simulation environments for verification and validation of autonomous vehicle

control systems. In particular, we considered two autonomous vehicle control approaches as case studies: an adaptive control approach based on level- k vehicle models and a rule-based control approach. We analyzed their characteristics and evaluated their performance based on their testing results with our traffic models, and then optimized the parameters of the rule-based approach based on a performance index.

We envision that traffic models developed using the framework proposed in this paper can also be integrated with urban traffic/driving simulators with higher-fidelity car dynamics and environmental representations, such as CARLA [54], using an approach similar to that of [55], to create more realistic urban traffic simulations and support autonomous driving system development.

REFERENCES

- [1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856415000804>
- [2] U.S. Department of Transportation's National Highway Traffic Safety Administration. *Automated Vehicles for Safety*. Accessed: Jun. 18, 2019. [Online]. Available: <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>
- [3] J. Meyer, H. Becker, P. M. Bösch, and K. W. Axhausen, "Autonomous vehicles: The next jump in accessibilities?" *Res. Transp. Econ.*, vol. 62, pp. 80–91, Jun. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0739885917300021>
- [4] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transp. Res. A, Policy Pract.*, vol. 94, pp. 182–193, Dec. 2016.
- [5] J. Zhou and L. del Re, "Reduced complexity safety testing for ADAS & ADF," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5985–5990, 2017.
- [6] H. Waschl, I. Kolmanovsky, and F. Willems, *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*. Cham, Switzerland: Springer, 2019.
- [7] D. Zhao *et al.*, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595–607, Mar. 2017.
- [8] H. Waschl, R. Schmied, D. Reischl, and M. Stolz, "A virtual development and evaluation framework for ADAS—Case study of a P-ACC in a connected environment," in *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*. New York, NY, USA: Springer, 2019, pp. 107–131.
- [9] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 1, pp. 5–17, Mar. 2018.
- [10] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic Foundations of Robotics X*, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds. Berlin, Germany: Springer, 2013, pp. 475–491.
- [11] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1617–1624.
- [12] N. Li, A. Girard, and I. Kolmanovsky, "Stochastic predictive control for partially observable Markov decision processes with time-joint chance constraints and application to autonomous vehicle control," *J. Dyn. Syst., Meas., Control*, vol. 141, no. 7, Jul. 2019, Art. no. 071007.
- [13] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2994–3008, Sep. 2018.
- [14] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 23–35, Fall 2017.
- [15] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A game-theoretic approach to replanning-aware interactive scene prediction and planning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3981–3992, Jun. 2016.

- [16] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," in *Robotics: Science and Systems*, vol. 2, 2016.
- [17] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9590–9596.
- [18] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transp. Res. C, Emerg. Technol.*, vol. 88, pp. 140–158, Mar. 2018.
- [19] A. Dreves and M. Gerdt, "A generalized Nash equilibrium approach for optimal control problems of autonomous cars," *Optim. Control Appl. Methods*, vol. 39, no. 1, pp. 326–342, Jan. 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2348>
- [20] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1590–1595.
- [21] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3530–3538.
- [22] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.
- [23] Federal Highway Administration. *Intersection Safety Needs Identification Report*. Accessed: Jun. 20, 2019. [Online]. Available: https://safety.fhwa.dot.gov/intersection/other_topics/needsidrpt/needsidrpt.pdf
- [24] *Unsignalized Intersections*. Accessed: Jun. 20, 2019. [Online]. Available: <https://safety.fhwa.dot.gov/intersection/conventional/unsignalized/>
- [25] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 825–830.
- [26] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2034–2039.
- [27] S. Pruekprasert, J. Dubut, X. Zhang, C. Huang, and M. Kishida, "A game-theoretic approach to decision making for multiple vehicles at roundabout," 2019, *arXiv:1904.06224*. [Online]. Available: <http://arxiv.org/abs/1904.06224>
- [28] S. Pruekprasert, X. Zhang, J. Dubut, C. Huang, and M. Kishida, "Decision making for autonomous vehicles at unsignalized intersection in presence of malicious vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2299–2304.
- [29] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London. Ser. A. Math. Phys. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.
- [30] A. D. May, *Traffic Flow Fundamentals*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.
- [31] C. F. Daganzo, "The cell transmission model, part II: Network traffic," *Transp. Res. B, Methodol.*, vol. 29, no. 2, pp. 79–93, Apr. 1995.
- [32] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res. B, Methodol.*, vol. 15, no. 2, pp. 105–111, Apr. 1981.
- [33] M. Brackstone and M. McDonald, "Car-following: A historical review," *Transp. Res. F, Traffic Psychol. Behaviour*, vol. 2, no. 4, pp. 181–196, Dec. 1999.
- [34] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," *Hum. Factors: J. Hum. Factors Ergonom. Soc.*, vol. 48, no. 2, pp. 362–380, Jun. 2006.
- [35] N. Li, Y. Yao, I. Kolmanovsky, E. Atkins, and A. R. Girard, "Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2020, doi: 10.1109/TITS.2020.3026160.
- [36] D. O. Stahl and P. W. Wilson, "On players' models of other players: Theory and experimental evidence," *Games Econ. Behav.*, vol. 10, no. 1, pp. 218–254, 1995.
- [37] M. A. Costa-Gomes and V. P. Crawford, "Cognition and behavior in two-person guessing games: An experimental study," *Amer. Econ. Rev.*, vol. 96, no. 5, pp. 1737–1768, Nov. 2006.
- [38] N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 3215–3220.
- [39] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 321–326.
- [40] S. Karimi and A. Vahidi, "Receding horizon motion planning for automated lane change and merge using Monte Carlo tree search and level-K game theory," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020.
- [41] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2000, p. 2.
- [42] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2008, pp. 1433–1438.
- [43] L. Claussmann, A. Carvalho, and G. Schildbach, "A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 2976–2982.
- [44] F. Codevilla, M. Muller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-End driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–9.
- [45] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, "A fast integrated planning and control framework for autonomous driving via imitation learning," in *Proc. ASME Dyn. Syst. Control Conf.*, 2018, Art. no. V003T37A012.
- [46] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 627–635.
- [47] K. Fitzpatrick, M. D. Wooldridge, and J. D. Blaschke, *Urban Intersection Design Guide: Guidelines*, vol. 1. Feb. 2005.
- [48] California Department of Transportation. *California Manual for Setting Speed Limits*. Accessed: Jun. 10, 2020. [Online]. Available: <https://dot.ca.gov/-/media/dot-media/programs/traffic-operations/documents/2019-california-manual-for-setting-speed-limits-a11y.pdf>
- [49] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberry, "Comparing models of strategic thinking in van huyck, battalio, and Beil's coordination games," *J. Eur. Econ. Assoc.*, vol. 7, nos. 2–3, pp. 365–376, Apr. 2009.
- [50] G. Chou, Y. E. Sahin, L. Yang, K. J. Rutledge, P. Nilsson, and N. Ozay, "Using control synthesis to generate corner cases: A case study on autonomous driving," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2906–2917, Nov. 2018.
- [51] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1094–1099.
- [52] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.
- [53] W. Zhan *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," Sep. 2019, *arXiv:1910.03088*. [Online]. Available: <http://arxiv.org/abs/1910.03088>
- [54] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.
- [55] G. Su, N. Li, Y. Yildiz, A. Girard, and I. Kolmanovsky, "A traffic simulation model with interactive drivers and high-fidelity car dynamics," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 384–389, 2019.



Ran Tian received the B.S. degree in aerospace engineering from the University of Michigan, Ann Arbor, MI, USA, in 2016, the B.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017, and the M.S. degree in robotics from the University of Michigan, in 2019. He is currently pursuing the Ph.D. degree with the University of California, Berkeley, CA, USA. His current research interests include decision-making under uncertainty and human-robot interaction.



Nan Li (Member, IEEE) received the B.S. degree in automotive engineering from Tongji University, Shanghai, China, in 2014, and the M.S. degrees in mechanical engineering and in mathematics from the University of Michigan, Ann Arbor, MI, USA, in 2016 and 2020, respectively, where he is currently pursuing the Ph.D. degree in aerospace engineering. His research interests are stochastic control and multiagent systems.



Ilya Kolmanovsky (Fellow, IEEE) received the Ph.D. degree from the University of Michigan in 1995. He is currently a Professor with the Department of Aerospace Engineering, University of Michigan, with research interests in control theory for systems with state and control constraints, and in control applications to aerospace and automotive systems.



Yildiray Yildiz (Senior Member, IEEE) received the B.S. degree (valedictorian) in mechanical engineering from Middle East Technical University, Ankara, in 2002, the M.S. degree in mechatronics from Sabanci University, Istanbul, in 2004, and the Ph.D. degree in mechanical engineering with a mathematics minor from MIT in 2009. He is currently an Assistant Professor with Bilkent University, Ankara. He held postdoctoral associate and associate scientist positions with the NASA Ames Research Center, California, employed by the University of California, Santa Cruz, through its University Affiliated Research Center, from 2009 to 2010 and from 2010 to 2014, respectively. He was a recipient of NASA Honor Award, Young Scientist Award from Science Academy of Turkey, Young Scientist Award from Turkish Academy of Sciences, Research Incentive Award from Prof. Mustafa Parlar Education and Research Foundation, and best student conference paper award from ASME. He is currently serving as an Associate Editor of the *IEEE Control Systems Magazine* and *European Journal of Control*. His research interests include control, machine learning, game theory, and applications of these fields for modeling and control of automotive and aerospace systems.



Anouck R. Girard (Senior Member, IEEE) received the Ph.D. degree in ocean engineering from the University of California, Berkeley, CA, USA, in 2002. She has been with the University of Michigan, Ann Arbor, MI, USA, since 2006, where she is currently an Associate Professor of aerospace engineering. She has coauthored the book *Fundamentals of Aerospace Navigation and Guidance* (Cambridge University Press, 2014). Her current research interests include flight dynamics and control systems. She was a recipient of the Silver Shaft Teaching Award from the University of Michigan and a Best Student Paper Award from the American Society of Mechanical Engineers.