Adaptive Game-Theoretic Decision Making
for Autonomous Vehicle Control at Roundabouts

Ran Tian, Sisi Li, Nan Li, Ilya Kolmanovsky, Anouck Girard and Yildiray Yildiz

Abstract— In this paper, we propose a decision making algorithm for autonomous vehicle control at a roundabout intersection. The algorithm is based on a game-theoretic model representing the interactions between the ego vehicle and an opponent vehicle, and adapts to an online estimated driver type of the opponent vehicle. Simulation results are reported.

I. INTRODUCTION

Autonomous vehicle control in urban traffic is still facing enormous challenges. Many of these challenges involve intersection traffic scenarios. According to [1], almost 40% of car crashes in the U.S. are intersection related. In an intersection scenario, it is typical to have multiple traffic participants interacting with each other. A driver or an automation controlling a vehicle at an intersection should account for these interactions in her/its decision making.

Intersections are usually categorized into two types: signalized intersections and unsignalized intersections, of which roundabout is a particular kind [2]. At a signalized intersection, the motions of all vehicles and of all other traffic participants are guided by traffic lights or traffic signs, which act as centralized traffic controls. On the other hand, at an unsignalized intersection, the driver/automation controlling a vehicle needs to decide on her/its own, whether, when and how to enter and cross the intersection, in which accounting for the interactions between traffic participants is particularly important, as each participant's actions influence and are also influenced by the actions of the other participants.

Game theory is a useful tool to model the interactions between strategic decision makers. Game-theoretic modelings of driver and vehicle interactions in highway traffic scenarios for use in autonomous vehicle control development are discussed in [3], [4], [5]. Game-theoretic autonomous vehicle control algorithms for intersection traffic scenarios are proposed in [6], [7]. In [6], vehicle-to-vehicle interactions at an unsignalized four-way intersection are modeled using a game-theoretic framework, and then an autonomous vehicle controller for such an intersection is developed based on the vehicle interaction model.

A roundabout intersection involves a merge/de-merge type traffic where a vehicle first merges into the center circle,

travels counter-clockwise (in the U.S.), and then de-merges from the circle [8]. Autonomous vehicle control particularly for roundabouts have been investigated in [9], [10], where vehicle interactions in multi-vehicle scenarios are not considered.

In this paper, we exploit the game-theoretic vehicle interaction modeling framework in [6] to develop an autonomous vehicle controller for a roundabout intersection. The contributions of this paper are: 1) We develop a game-theoretic model representing the interactions between two vehicles at a roundabout intersection. Such a model can have multiple uses, such as for developing autonomous vehicle control systems [6] or for the verification, validation, and calibration of such systems [3], [4]. We focus on the former in this paper. 2) We propose a decision making algorithm for an ego vehicle at such a roundabout intersection that is based on the vehicle interaction model and adapts to an estimated driver type of an opponent vehicle. 3) We describe an explicit online implementation scheme exploiting function approximation techniques to avoid the need for solving optimization problems related to the algorithm in real time.

II. VEHICLE MODEL

A. Vehicle kinematics

The control of a vehicle is typically modeled using a hierarchical architecture [11]: a high-level decision making layer plans the desired path for the vehicle, and then a low-level dynamics and actuation control layer controls the subsystems, e.g., engine, transmission, steering, etc., to track the references generated by the high level.

In this paper, we focus on the high-level decision making and we use a discrete-time model to represent the vehicle kinematics at a roundabout intersection as follows,

$$x(t+1) = x(t) + v(t)\cos\big(\theta(t)\big)\,\Delta t, \qquad (1a)$$
$$y(t+1) = y(t) + v(t)\sin\big(\theta(t)\big)\,\Delta t, \qquad (1b)$$
$$v(t+1) = v(t) + a(t)\,\Delta t, \qquad (1c)$$
$$\theta(t+1) = \theta(t) + \omega(t)\,\Delta t, \qquad (1d)$$

where $x(t)$ and $y(t)$ represent, respectively, the vehicle's position in the x-direction and y-direction at the discrete time t; $v(t)$, $a(t)$, $\theta(t)$, and $\omega(t)$ are, respectively, the vehicle's speed, acceleration, yaw angle and yaw rate at t; Δt is the time step size. An action, denoted by γ, is an input pair (a, ω) to the model (1). In this paper, we assume that the vehicle can choose actions from a finite action set, $\boldsymbol{\Gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$, to execute.

B. Scenario of interest

In this paper, we model the vehicle-to-vehicle interactions at a roundabout intersection, see Fig. 1(a). A vehicle can enter or exit the roundabout in the directions indicated by the blue arrows. When in the roundabout, all vehicles must travel counter-clockwise (indicated by the orange arrows). In this paper, we consider the interactions between two vehicles, the ego vehicle (represented by the blue solid box in Fig. 1(a)) and the opponent vehicle (represented by the red solid box in Fig. 1(a)). The double lines indicate the front ends of the vehicles.
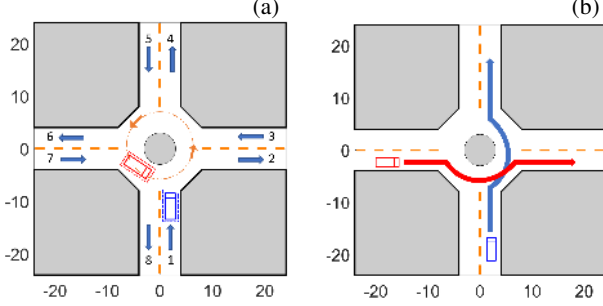

Fig. 1. The roundabout intersection and traffic scenario to be considered.

C. Reward function

The vehicle decision making process is based on receding-horizon optimal control: At each time step t, the vehicle selects a sequence of actions $\Gamma(t) = \{\gamma(t), \cdots, \gamma(t+n-1)\}$ to maximize a cumulative reward over a horizon of length n, $\mathcal{R}(t) = \sum_{j=1}^{n} \lambda^{j-1} R(t+j)$, where $R(t+j)$ represents a stage reward at step $t+j$ over the horizon, and $\lambda \in [0, 1]$ is a discount factor.

The stage reward function, $R(t)$, is defined as

$$R(t) = \mathbf{w}^{\mathsf{T}} \mathbf{\Phi}(t), \tag{2}$$

where $\mathbf{\Phi}(t) = [\phi_1(t), \phi_2(t), \phi_3(t), \phi_4(t), \phi_5(t), \phi_6(t)]^{\mathsf{T}}$ is the feature vector at t, and $\mathbf{w}$ is the weight vector that contains the weights for the features.

The feature $\phi_1(t)$ is an indicator that characterizes the collision status of the vehicle. We bound the geometric contour of the vehicle by a rectangle (the dashed boxes in Fig. 1(a)). We refer to this rectangle as the collision avoidance zone (c-zone). If the ego vehicle's c-zone has an overlap with the opponent vehicle's c-zone, which indicates a danger of collision, $\phi_1(t) = -1$; $\phi_1(t) = 0$ otherwise. In this paper, the size of the c-zone is 5 [m] $\times 2$ [m].

The feature $\phi_2(t)$ is an indicator that characterizes the on-road status of the vehicle. If the ego vehicle's c-zone crosses the road boundaries, i.e., the ego vehicle's c-zone has an overlap with the gray areas in Figure 1(a), $\phi_2(t) = -1$; $\phi_2(t) = 0$ otherwise.

The feature $\phi_3(t)$ characterizes the distance-to-objective status of the vehicle. We define $\phi_3(t)$ as

$$\phi_3(t) = -|x_r - x(t)| - |y_r - y(t)|, \tag{3}$$

where (x_r, y_r) are the coordinates of a reference point on the vehicle's objective lane.

The feature $\phi_4(t)$ characterizes the safe separation of the ego vehicle from the opponent vehicle. When driving in traffic, a vehicle is supposed to keep a reasonable distance from its surrounding vehicles to improve safety. We define a safety zone (s-zone) that over-bounds the vehicle's c-zone with a safety margin. If the ego vehicle's s-zone has an overlap with the opponent vehicle's s-zone, $\phi_4(t) = -1$; $\phi_4(t) = 0$ otherwise. In this paper, the s-zone is concentric with the c-zone and is 8 [m] $\times 2.4$ [m] in size.

The feature $\phi_5(t)$ penalizes the crossing the lane markings that separate traffic of opposite directions and driving into a wrong lane (not the vehicle's objective lane) when exiting the roundabout. If either occurs, $\phi_5(t) = -1$; $\phi_5(t) = 0$ otherwise.

The feature $\phi_6(t)$ rewards the vehicle's speed, defined as $\phi_6(t) = v(t)$.

III. GAME THEORETIC DECISION MAKING

Based on the definitions of the features, the rewards of a vehicle not only depend on its own states and actions, but also depend on the states and actions of its opponent vehicle (e.g., $\phi_1(t)$ and $\phi_4(t)$). Such an interdependence reflects the interactive nature of vehicle decision making in a multi-vehicle traffic scenario. The receding-horizon optimal control problem is thus formulated as: At each time step t, to select

$$\Gamma_{\text{ego}}^*(t) = \{\gamma_{\text{ego}}^*(t), \cdots, \gamma_{\text{ego}}^*(t + n - 1)\} = \tag{4}$$
$$\arg\max_{\gamma_{\text{ego}}(t+j) \in \mathbf{\Gamma}} \sum_{j=0}^{n-1} \lambda^j R\big(\mathbf{X}(t+j), \gamma_{\text{ego}}(t+j), \gamma_{\text{opp.}}(t+j)\big),$$

and execute $\gamma_{\text{ego}}^*(t)$ over one time step, where $\mathbf{X}(t) = [x_{\text{ego}}(t), y_{\text{ego}}(t), \theta_{\text{ego}}(t), v_{\text{ego}}(t), x_{\text{opp.}}(t), y_{\text{opp.}}(t), \theta_{\text{opp.}}(t), v_{\text{opp.}}(t)]^{\mathsf{T}}$ represents the state of the traffic, which contains both the ego vehicle's states and the opponent vehicle's states; $\gamma_{\text{ego}}(t+j)$ is the ego vehicle's action at $t+j$ over the horizon and is to be optimized, and $\gamma_{\text{opp.}}(t+j)$ is the opponent vehicle's action at $t+j$ over the horizon and is to be predicted. We note that for the two interacting vehicles, either is the "ego vehicle" from its own perspective, and is also the "opponent vehicle" from the other's perspective, that is, (4) can be used to describe the decision making of either of the two vehicles.

To obtain $\Gamma_{\text{ego}}^*(t)$, the ego vehicle needs to predict the sequence $\Gamma_{\text{opp.}}(t) = \{\gamma_{\text{opp.}}(t), \cdots, \gamma_{\text{opp.}}(t + n - 1)\}$, which contains the actions that the opponent vehicle executes over the prediction horizon. Approaches such as data-driven driver behavior prediction may be used for such predictions [12]. In this paper, we exploit a game-theoretic approach for such predictions as it accounts for vehicle-to-vehicle interactions.

Our approach is based on the level-k game theory [13]. Level-k game theory relies on a hierarchical cognitive structure to model human reasoning in games. Each player is assumed to be of a particular reasoning level, indicated by $k \in \{0, 1, \cdots\}$. A level-k player assumes that all other players can be modeled as level-$(k-1)$ reasoners and acts

accordingly. The development of the reasoning hierarchy starts from a level-0 player's model, which usually represents instinctive decision making without accounting for the interactions between players. Then a level-1 player's model can be obtained by assuming that all the players except for the ego player can be modeled as level-0 players. Based on this assumption, a level-1 player's model can predict the actions of the other level-0 players, e.g., the sequence $\Gamma_{\text{opp.}}(t)$ in (4), and then compute its own optimal actions $\Gamma_{\text{ego}}^*(t)$ based on (4). This procedure continues – a level-k player's model is generated by assuming that all other players act according to level-$(k-1)$ models, until the desired highest level is obtained. This reasoning hierarchy has been observed in human interactions for other application domains by experimental studies and it is shown that humans are commonly level-0, 1 and 2 reasoners [13], [14].

In [6], such a level-k game-theoretic framework is employed to model the driver and vehicle interactive behavior at an unsignalized four-way intersection. By assuming that a level-0 driver treats the other vehicles on the road as stationary obstacles (this way, a level-0 driver may be viewed as an aggressive driver in human traffic, who usually assumes that the others will yield the right of way) and constructing the corresponding level-1 and 2 driver models using the procedure described above, we observe that the behavior of a level-0 driver and that of a level-2 driver are similar as they both represent aggressive drivers. Therefore, instead of considering three driver levels (level-0, 1 and 2), we consider two driver types, type-1 and 2, in this paper.

A type-1 driver model represents a conservative driver who predicts the opponent vehicle's actions based on the assumption that the opponent vehicle's driver is an instinctive decision maker, who maximizes her cumulative reward (4) by treating the other vehicles on the road as stationary obstacles. On the other hand, a type-2 driver model represents an aggressive driver who predicts the opponent vehicle's actions by assuming that the opponent vehicle's driver is a type-1 driver. Indeed, the type-1/2 driver model meets the level-1/2 driver model defined in the level-k game-theoretic framework described above.

Specifically, a type-1 driver selects actions, $\Gamma_{\text{ego}}^{(1)}(t) = \{\gamma_{\text{ego}}^{(1)}(t), \cdots, \gamma_{\text{ego}}^{(1)}(t+n-1)\}$, based on

$$\Gamma_{\text{ego}}^{(1)}(t) = \tag{5}$$
$$\arg\max_{\gamma_{\text{ego}}(t+j)\in\Gamma} \sum_{j=0}^{n-1} \lambda^j R\big(\mathbf{X}(t+j), \gamma_{\text{ego}}(t+j), \gamma_{\text{opp.}}^{(0)}(t+j)\big),$$

where $\gamma_{\text{opp.}}^{(0)}(t+j)$ represents the predicted opponent vehicle's action at $t+j$ over the horizon under the assumption that the opponent vehicle's driver treats the ego vehicle as a stationary obstacle. Note that based on this assumption, the actions $\gamma_{\text{opp.}}^{(0)}(t+j)$, $j = 0, \cdots, n-1$, are independent of $\Gamma_{\text{ego}}^{(1)}(t)$, and thus can be determined first.

Similarly, a type-2 driver selects actions, $\Gamma_{\text{ego}}^{(2)}(t)$, based on

$$\Gamma_{\text{ego}}^{(2)}(t) = \tag{6}$$
$$\arg\max_{\gamma_{\text{ego}}(t+j)\in\Gamma} \sum_{j=0}^{n-1} \lambda^j R\big(\mathbf{X}(t+j), \gamma_{\text{ego}}(t+j), \gamma_{\text{opp.}}^{(1)}(t+j)\big),$$

where $\{\gamma_{\text{opp.}}^{(1)}(t), \cdots, \gamma_{\text{opp.}}^{(1)}(t+n-1)\}$ are computed using (5) by switching the roles of "ego" and "opp."

IV. ADAPTIVE GAME-THEORETIC AUTONOMOUS VEHICLE CONTROL

A. Adaptive control strategy

Based on the type-1, 2 driver models, we develop an autonomous vehicle (AV) control algorithm. At each time step t, the AV controller predicts the opponent vehicle's actions over the horizon, $\Gamma_{\text{opp.}}(t)$, based on an estimate of the opponent vehicle's driver type. Specifically, $\Gamma_{\text{opp.}}(t) = \Gamma_{\text{opp.}}^{(1)}(t)$ computed using (5) if the opponent vehicle's driver is estimated as type-1, and $\Gamma_{\text{opp.}}(t) = \Gamma_{\text{opp.}}^{(2)}(t)$ computed using (6) if the opponent vehicle's driver is estimated as type-2 (with the roles of "ego" and "opp." switched). Then, the controller computes the optimal actions $\Gamma_{\text{ego}}^*(t)$ for the AV to execute using (4) with the predicted $\Gamma_{\text{opp.}}(t)$ substituted in.

The estimate of the opponent vehicle's driver type is updated after each time step. Specifically, at time $t + 1$, we compare the opponent vehicle's actually applied action $\gamma_{\text{opp.}}(t)$ to the first action $\gamma_{\text{opp.}}^{(1)}(t)$ in the sequence $\Gamma_{\text{opp.}}^{(1)}(t)$ and to the first action $\gamma_{\text{opp.}}^{(2)}(t)$ in the sequence $\Gamma_{\text{opp.}}^{(2)}(t)$. If $\gamma_{\text{opp.}}(t)$ matches $\gamma_{\text{opp.}}^{(\eta)}(t)$ better for some $\eta \in \{1, 2\}$, we increase the probability that the opponent vehicle's driver can be modeled as type-η.

Such an adaptive game-theoretic AV control strategy has been tested versus different driver models for the opponent vehicle at a non-roundabout type four-way intersection in [6], and exhibits reasonable performance. The action sequences $\Gamma_{\text{opp.}}^{(1)}(t)$, $\Gamma_{\text{opp.}}^{(2)}(t)$, and $\Gamma_{\text{ego}}^*(t)$ are computed based on (5), (6), and (4) using a tree-search based method, which can be computationally demanding when the prediction horizon is large. Thus, in what follows, we propose an explicit online implementation scheme exploiting function approximation techniques, to avoid the need for solving optimization problems in real time.

B. Explicit online implementation

We note that a function $g : \big(\mathbf{X}(t), \eta(t)\big) \mapsto \gamma_{\text{opp.}}^{(\eta(t))}(t)$, where $\eta(t) \in \{1, 2\}$ indicates the opponent vehicle's driver type at t, is implicitly defined by (5) and (6) (with the roles of "ego" and "opp." switched). Given the traffic state $\mathbf{X}(t)$, we denote the $\mathbf{X}(t)$-section of g by $g^{\mathbf{X}(t)} : \eta(t) \mapsto \gamma_{\text{opp.}}^{(\eta(t))}(t)$. The procedure to update the opponent vehicle's driver type estimate described in Section IV-A requires the inverse map of $g^{\mathbf{X}(t)}$, $\big(g^{\mathbf{X}(t)}\big)^{-1} : \gamma_{\text{opp.}}^{(\eta(t))}(t) \mapsto \eta(t) \in \{1, 2\}$. Such an inverse map $\big(g^{\mathbf{X}(t)}\big)^{-1}$ may be multi-valued because at some traffic states, the opponent vehicle may select the same action $\gamma_{\text{opp.}}(t)$ regardless of its driver types, e.g., when the

two vehicles are far from each other. To have a single-valued function so that it can be fitted using a function approximator, we want to restrict $\left(g^{\mathbf{X}(t)}\right)^{-1}$ to be defined on a subset of the state space of $\mathbf{X}(t)$, denoted by $\mathcal{X}_{\text{critical}}$, such that $\left(g^{\mathbf{X}(t)}\right)^{-1}$ is single-valued on $\mathcal{X}_{\text{critical}}$. To find the set $\mathcal{X}_{\text{critical}}$ is to find the subset of the state space of $\mathbf{X}(t)$ where the action $\gamma_{\text{opp.}}^{(1)}(t)$ selected by a type-1 driver model is different from the action $\gamma_{\text{opp.}}^{(2)}(t)$ selected by a type-2 driver model. More specifically, for each $\mathbf{X}(t)$, we compare $\gamma_{\text{opp.}}^{(1)}(t)$ and $\gamma_{\text{opp.}}^{(2)}(t)$: if $\gamma_{\text{opp.}}^{(1)}(t) \neq \gamma_{\text{opp.}}^{(2)}(t)$, $\mathbf{X}(t) \in \mathcal{X}_{\text{critical}}$; $\mathbf{X}(t) \notin \mathcal{X}_{\text{critical}}$ otherwise. The construction of $\mathcal{X}_{\text{critical}}$ can be done offline, then approximated and described using a clustering neural network (NN_B).

After the set $\mathcal{X}_{\text{critical}}$ is identified, when $\mathbf{X}(t) \in \mathcal{X}_{\text{critical}}$, through the function $\left(g^{\mathbf{X}(t)}\right)^{-1}\left(\gamma_{\text{opp.}}(t)\right)$, we obtain an estimate of the opponent vehicle's driver type at t, $\eta(t)$ (assuming that $\gamma_{\text{opp.}}(t) = \gamma_{\text{opp.}}^{(\eta)}(t)$, for some $\eta \in \{1, 2\}$). The function $g^{-1}\big|_{\mathcal{X}_{\text{critical}}} : \left(\mathbf{X}(t), \gamma_{\text{opp.}}(t)\right) \mapsto \eta(t) \in \{1, 2\}$ is constructed offline and approximated using a neural network (NN_C).

Taking historical data into account, we update the AV controller's belief on the opponent vehicle's driver type based on

$$\mathbb{P}^{(2)}(t+1) = (1-\beta)\mathbb{P}^{(2)}(t) + \beta\,\mathbb{I}\{\eta(t) = 2\}, \quad (7)$$

where $\mathbb{P}^{(2)}(t)$ represents the probability that the opponent vehicle's driver can be modeled as type-2, $\beta \in [0, 1]$ is an estimate update step size, and $\mathbb{I}\{\eta(t) = 2\}$ is an indicator function, taking 1 if the event $\{\eta(t) = 2\}$ is true and 0 otherwise.

Based on $\mathbb{P}^{(2)}(t)$, the AV controller predicts the opponent vehicle's actions over the horizon, $\Gamma_{\text{opp.}}(t)$, using different models: If $\mathbb{P}^{(2)}(t) < 0.5$, the opponent vehicle's driver is more likely to be a type-1 driver, and thus the AV controller sets $\Gamma_{\text{opp.}}(t) = \Gamma_{\text{opp.}}^{(1)}(t)$. Otherwise, the opponent vehicle's driver is more likely to be a type-2 driver, and thus the AV controller sets $\Gamma_{\text{opp.}}(t) = \Gamma_{\text{opp.}}^{(2)}(t)$. Finally, the AV controller computes the optimal actions $\Gamma_{\text{ego}}^*(t)$ using (4) with the predicted $\Gamma_{\text{opp.}}(t)$ substituted in. Similar to the function g implicitly defined by (5) and (6), a control policy $\pi_{\text{ego}} : \left(\mathbf{X}(t), \eta(t)\right) \mapsto \gamma_{\text{ego}}^*(t) \in \Gamma$, where $\eta(t) \in \{1, 2\}$ indicates the estimated opponent vehicle's driver type at t, is implicitly defined by (4). The policy π_{ego} is constructed offline and approximated using a neural network (NN_A).

With the use of the above three neural networks, we move the computations to solve the optimization problems (4), (5), and (6) from online to offline. The online control is policy-based and requires only neural network evaluations, where the policy adapts to the opponent vehicle's driver by the adaptation law (7). The overall structure of the AV controller is shown in Fig 2.

C. Controller training

We randomly create states $\mathbf{X}(t)$, and compute $\Gamma_{\text{opp.}}^{(1)}(t)$ using (5) and $\Gamma_{\text{opp.}}^{(2)}(t)$ using (6) (with the roles of "ego" and "opp." switched). If $\gamma_{\text{opp.}}^{(1)}(t) \neq \gamma_{\text{opp.}}^{(2)}(t)$, we label $\mathbf{X}(t)$

by $\zeta\left(\mathbf{X}(t)\right) = 1$; and $\zeta\left(\mathbf{X}(t)\right) = 0$ otherwise. This labeled data set is used to train NN_B.
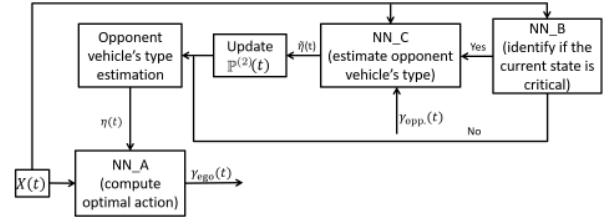

Fig. 2.   Structure of the proposed autonomous vehicle controller.

If $\zeta\left(\mathbf{X}(t)\right) = 1$, we label the pair $\left(\mathbf{X}(t), \gamma_{\text{opp.}}^{(1)}(t)\right)$ by $\eta\left(\mathbf{X}(t), \gamma_{\text{opp.}}^{(1)}(t)\right) = 1$, and label the pair $\left(\mathbf{X}(t), \gamma_{\text{opp.}}^{(2)}(t)\right)$ by $\eta\left(\mathbf{X}(t), \gamma_{\text{opp.}}^{(2)}(t)\right) = 2$. This labeled data set is used to train NN_C.

Finally, we compute the optimal actions $\Gamma_{\text{ego}}^*(t, \eta(t)) = \{\gamma_{\text{ego}}^*(t, \eta(t)), \cdots, \gamma_{\text{ego}}^*(t + n - 1, \eta(t))\}$ based on (4) with $\Gamma_{\text{opp.}}(t) = \Gamma_{\text{opp.}}^{(\eta(t))}(t)$, $\eta(t) = 1, 2$, substituted in, and label the pair $\left(\mathbf{X}(t), \eta(t)\right)$ by $\gamma\left(\mathbf{X}(t), \eta(t)\right) = \gamma_{\text{ego}}^*(t, \eta(t))$. This labeled data set is used to train NN_A.

Table I lists the features and labels for the three neural networks. Each data set is split into a training set and a validation set with a ratio of 8:2. We use the same architecture for each neural network (2 convolutional layers followed by 6 fully connected layers) with different hyperparameters. The validation accuracy of each neural network is shown in Table I.

Neural Net	Features	Label	Accuracy
NN_A	$\mathbf{X}(t), \eta(t)$	$\gamma_{\text{ego}}^*(t, \eta(t))$	97.4%
NN_B	$\mathbf{X}(t)$	$\zeta\left(\mathbf{X}(t)\right)$	98.2%
NN_C	$\mathbf{X}(t), \gamma_{\text{opp.}}^{(\eta)}(t)$	$\eta\left(\mathbf{X}(t), \gamma_{\text{opp.}}^{(\eta)}(t)\right)$	96.7%

TABLE I

TRAINING FEATURES, LABELS AND VALIDATION ACCURACY.

V. SIMULATION RESULTS

In this section, we present simulation results to show the performance of the adaptive game-theoretic AV controller. The traffic scenario to be considered is shown in Fig. 1(b). The vehicles are assumed to select actions from the set in Table II. The weight in (2) is chosen as $\mathbf{w} = [1000, 500, 5, 100, 50, 1]^{\mathsf{T}}$. The discount factor in the cumulative reward function is $\lambda = 0.8$, and the update step size in the adaptation law (7) is $\beta = 0.6$.

action γ	a [m²/s]	ω [rad/s]
maintain (γ_1)	0	0
accelerate (γ_2)	2.5	0
decelerate (γ_3)	-2.5	0
hard brake (γ_4)	-5	0
turn left (γ_5)	0	$\pi/4$
turn right (γ_6)	0	$-\pi/4$

TABLE II

ACTION SET Γ.

We first test the AV controller's performance versus opponent vehicles controlled by type-1 or 2 drivers. We then test the controller's performance versus driver models that may not act exactly as type-1 or 2 models. In particular, we let human operators control the opponent vehicle using a keyboard.
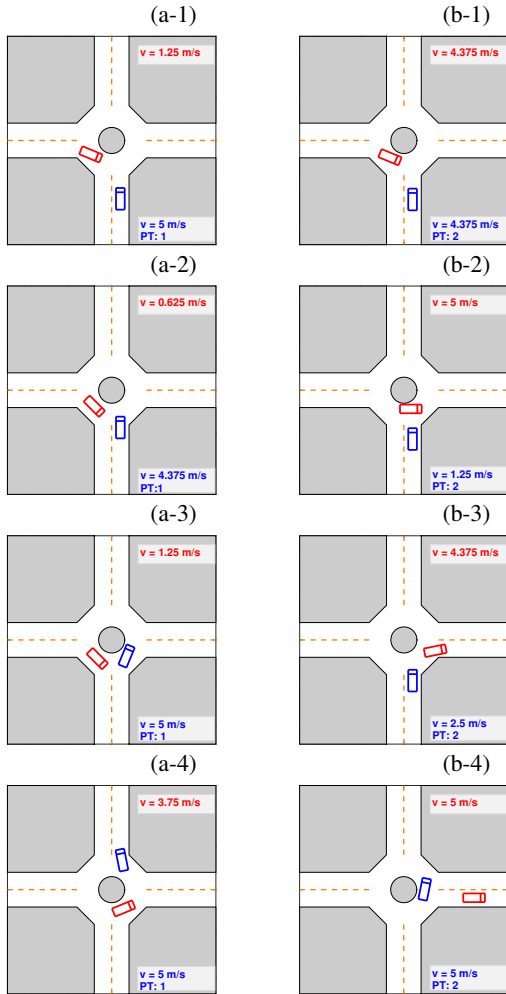
A. AV controller versus type-1/2 drivers

Fig. 3. Interactions between the ego vehicle (blue) controlled by the proposed AV controller and a type-1 opponent vehicle (red in (a-1) - (a-4)), and a type-2 opponent vehicle (red in (b-1) - (b-4)), at $t = 1.75$ s, $t = 2.5$ s, $t = 3.75$ s and $t = 6.25$ s.
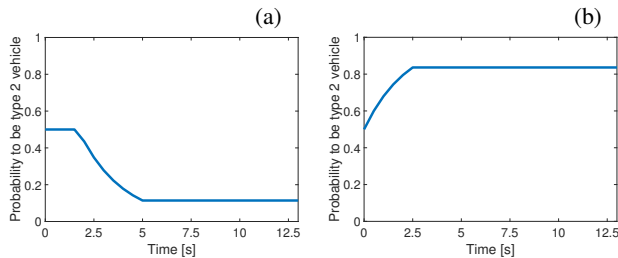

Fig. 4. Time histories of $\mathbb{P}^{(2)}(t)$ corresponding to the simulations in Fig. 3. (a) Versus a type-1 opponent vehicle. (b) Versus a type-2 opponent vehicle.

Fig. 3 (a-1) - (a-4) and (b-1) - (b-4) show the responses of the ego vehicle controlled by the AV controller when it encounters, respectively, a type-1 opponent vehicle and a type-2 opponent vehicle. The initial conditions, $\mathbf{X}(0)$, for both cases are the same. Fig. 4 shows the controller's belief histories on the opponent vehicle's type, $\mathbb{P}^{(2)}(t)$, over the simulations. When interacting with a type-1, conservative driver, the ego vehicle chooses to pass the roundabout first, since it predicts that the opponent vehicle will yield the right of way. When interacting with a type-2, aggressive driver, the ego vehicle chooses to decelerate, yields the right of way, and then accelerates to pass the roundabout after the opponent vehicle passes.

We run 1000 simulations, where the initial conditions and the types of the opponent vehicle are randomly generated, to statistically evaluate the controller's performance. The success rate is 93.4%, i.e., in 934 out of 1000 simulation runs, the ego and opponent vehicles successfully reach their objective lanes without colliding with each other, without driving off the road or crossing the lane markings that separate traffic of opposite directions, and without causing a deadlock (neither vehicle decides to enter the roundabout or both vehicles get stuck in the middle of the roundabout).

B. AV controller versus human drivers

We next test the proposed AV controller versus opponent vehicles controlled by human operators. We note that neither do the operators know the mechanism behind the controller, nor does the controller know the operators' driving styles in advance.

Fig. 5 shows the interactions between the ego AV and two opponent human-controlled vehicles, and Fig. 6 shows the controller's belief histories $\mathbb{P}^{(2)}(t)$ over the simulations. In the first experiment ((c-1) - (c-5)), the human operator acts aggressively, so the ego vehicle yields the right of way and the controller identifies the opponent vehicle as type-2 (Fig. 6(a)). In the second experiment ((d-1) - (d-5)), the human operator accelerates and tries to pass the roundabout first at the beginning. The controller thus identifies the opponent vehicle as type-2 and decelerates to avoid collision. However, the human operator realizes soon that he gets too close to the AV, thus decelerates. Then the AV controller updates its belief on the opponent vehicle's type to type-1 (Fig. 6(b)) and accelerates to pass the roundabout first.

We run 70 simulations conducted by 7 different human operators (10 trials for each person) to statistically evaluate the controller's performance. The success rate is 88.6%. We note that the human operators may have driven the vehicle more aggressively than they usually do in real driving since there are no safety issues [15].

We note that our neural network-based online implementation is also computationally feasible – the average computation time for the AV controller to identify the traffic state status, update the opponent vehicle's type estimate, and then generate the ego vehicle's action, is 34.1 [ms] in total, running on a laptop with Intel Core I7 processor and NVIDIA GeForce GTX GPU.
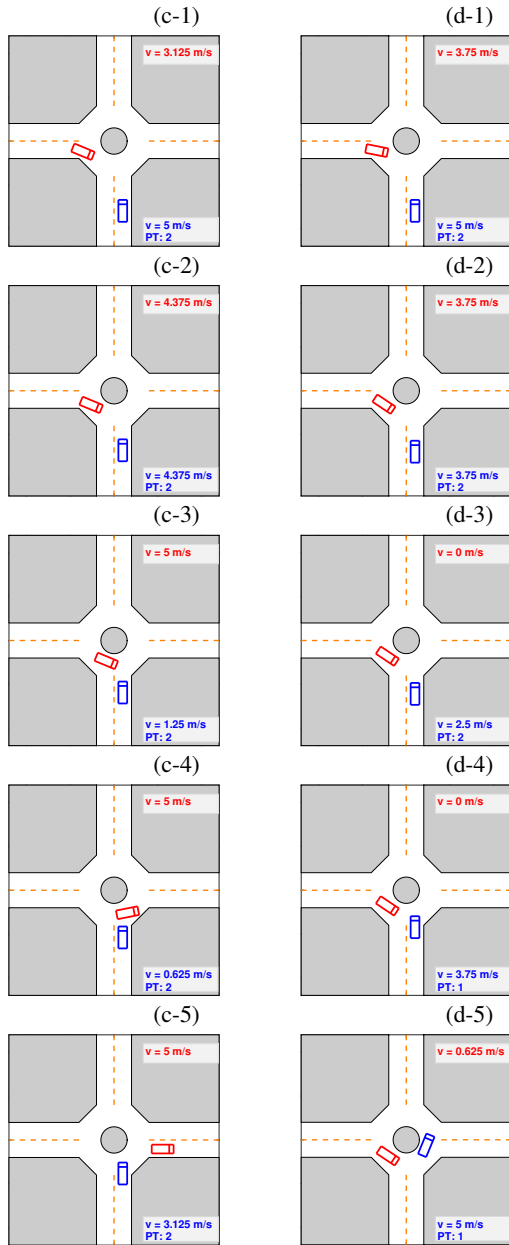
Fig. 5. Interactions between the ego vehicle controlled by the proposed AV controller (blue) and opponent vehicles controlled by human operator 1 (red in (c-1) - (c-5)), and by human operator 2 (red in (d-1) - (d-5)), at $t = 1$ s, $t = 1.5$ s, $t = 2.5$ s, $t = 3.25$ s and $t = 5$ s.
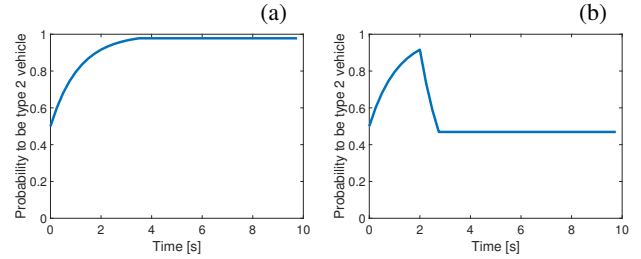

Fig. 6. Time histories of $\mathbb{P}^{(2)}(t)$ corresponding to the simulations in Fig. 5. (a) Versus human operator 1. (b) Versus human operator 2.

VI. CONCLUDING REMARKS

In this paper, we described an algorithm for autonomous vehicle control at a roundabout intersection. The algorithm is based on a game-theoretic model representing the interactions between the ego vehicle and an opponent vehicle, and adapts to an online estimated driver type of the opponent vehicle. We further proposed an explicit online implementation scheme exploiting function approximation techniques. Simulation results were reported to show the feasibility of the proposed control algorithm.

REFERENCES

[1] Federal Highway Administration, "Intersection safety needs identification report," Tech. Rep., Available: https://safety.fhwa.dot.gov/intersection/other_topics/needsidrpt/needsidrpt.pdf [Sep. 20, 2018].

[2] ——, "Roundabouts and mini roundabouts," Tech. Rep., Available: https://safety.fhwa.dot.gov/intersection/innovative/roundabouts/ [Sep. 20, 2018].

[3] 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 Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.

[4] N. Li, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. Girard, "Game theory-based traffic modeling for calibration of automated driving algorithms," in *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*. Springer, 2019, pp. 89–106.

[5] A. Talebpour, H. S. Mahmassani, and S. H. Hamdar, "Modeling lane-changing behavior in a connected environment: A game theory approach," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 216–232, 2015.

[6] 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 *2018 Annual American Control Conference (ACC)*. IEEE, June 2018, pp. 3215–3220.

[7] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *18th International Conference on Intelligent Transportation Systems*. IEEE, Sep. 2015, pp. 343–347.

[8] Washington State Department of Transportation, "Roundabouts," Tech. Rep., Available: https://www.wsdot.wa.gov/Safety/roundabouts/ [Sep. 20, 2018].

[9] J. Pérez Rastelli, V. Milanés, T. De Pedro, and L. Vlacic, "Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts," in *Proceedings of the 18th World Congress The International Federation of Automatic Control*, Milan, Italy, Aug. 2011. [Online]. Available: https://hal.inria.fr/hal-00737439

[10] J. Pérez Rastelli and M. Santos, "Fuzzy logic steering control of autonomous vehicles insideroundabouts," *Applied Soft Computing*, Oct. 2015. [Online]. Available: https://hal.inria.fr/hal-01232623

[11] N. Li, H. Chen, I. Kolmanovsky, and A. Girard, "An explicit decision tree approach for automated driving," in *Dynamic Systems and Control Conference*. ASME, 2017.

[12] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1759–1764, July 2018.

[13] M. A. Costa-Gomes, N. Iriberri, and V. P. Crawford, "Comparing models of strategic thinking in van huyck, battalio, and beil's coordination games," *Journal of the European Economic Association*, vol. 7, no. 2/3, pp. 365–376, 2009.

[14] M. A. Costa-Gomes and V. P. Crawford, "Cognition and behavior in two-person guessing games: An experimental study," *American Economic Review*, vol. 96, no. 5, pp. 1737–1768, Dec. 2006.

[15] G. Matthews, L. Dorn, T. W. Hoyes, D. R. Davies, A. I. Glendon, and R. G. Taylor, "Driver stress and performance on a driving simulator," *Human Factors*, vol. 40, no. 1, pp. 136–149, 1998.