2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

A Game Theoretical Model of Traffic with Multiple Interacting Drivers for Use in Autonomous Vehicle Development

Dave W. Oyler[1], Yildiray Yildiz[2], Anouck R. Girard[1], Nan I. Li[1] and Ilya V. Kolmanovsky[1]

Abstract— **This paper describes a game theoretical model of traffic where multiple drivers interact with each other. The model is developed using hierarchical reasoning, a game theoretical model of human behavior, and reinforcement learning. It is assumed that the drivers can observe only a partial state of the traffic they are in and therefore although the environment satisfies the Markov property, it appears as non-Markovian to the drivers. Hence, each driver implicitly has to find a policy, i.e. a mapping from observations to actions, for a Partially Observable Markov Decision Process. In this paper, a computationally tractable solution to this problem is provided by employing hierarchical reasoning together with a suitable reinforcement learning algorithm. Simulation results are reported, which demonstrate that the resulting driver models provide reasonable behavior for the given traffic scenarios.**

I. INTRODUCTION

As is apparent from the prior literature (see e.g., [1] [2] [3]), models of human driver actions in a given traffic scenario can be exploited for the development of decision making algorithms for autonomous driving and for the implementation of high-fidelity simulators that can facilitate the validation and testing of competing autonomous driving policies. A comprehensive list of existing human driver models, control based and behavioral based, can be found in [4]. Many of the proposed models lack driver interaction dynamics, which are important for operating in traffic and simulating real driving. One example of a model that incorporates interactions between drivers can be found in [5], where built in logical rules (if-then-else) are used to represent the decision making process. Although this approach successfully incorporates multi-tasking behavior and interaction of human drivers, it is not clear how the logical rules are obtained, and furthermore, the dynamic behavior (multi step decision making) is not considered. In [6], several logical algorithms are used to model decision making during lane changes. The resulting actions of the drivers are predefined with strict rules, and driver aggressiveness can be incorporated into the model by tuning certain parameters.

In this paper, we present a method to model the collective behavior of vehicles in traffic by microscopic modeling of human drivers. The main advantages of the proposed approach are that a) driver actions are not assumed to be known a priori but determined based on a human decision making process, b) multiple interactions between drivers, vehicles and automation (for example driverless cars) can be modeled simultaneously and, therefore, a traffic scenario with many vehicles can be analyzed realistically. Using the proposed method, collective behavior of the overall system can be quantified. For example, the percentage increase in accidents based on the traffic density can be determined. To achieve this, hierarchical reasoning from game theory and reinforcement learning is employed. Hierarchical reasoning helps model the interactions between intelligent decision makers (drivers) while reinforcement learning helps simulate a time-extended scenario with multiple actions. It is noted that such an approach has not been used before for road traffic modeling. At the core of this approach is a method known as "semi network-form games" [7] which help us obtain the probable outcomes of a complex traffic scenario in the presence of multiple driver-driver interactions.

It is noted that there are other game theoretical approaches to model highway driving such as [8] and [9]. Although these approaches exploit driver interaction models using a game theoretical setting, they do not consider dynamic (multi-move) scenarios. The latter are exploited in [10] for Hybrid Electric Vehicle (HEV) energy management where the driver and the powertrain are considered to be two players in a game. A major advantage of our hierarchical reasoning approach is that it is easily scalable to multiple players, e.g., see [11] where a 50 player game is treated.

The organization of this paper is as follows. The problem definition is given in Section II. The process of obtaining driver policies utilizing reinforcement learning and game theory is explained in Section III. Simulation results are provided in Section IV and a summary is given in Section V.

II. PROBLEM DEFINITION

The problem we treat in this paper is to predict the behavior of drivers in a traffic scenario where the cars are driven on a 3-lane highway. Fig. 1 shows an example scenario with 6 cars. It is noted that using the method we propose, scenarios with more cars or more lanes can be simulated. Below, we provide the information to precisely define the scenario we want to simulate.

In this scenario, the cars are assumed to be traveling in the same direction and driven by human drivers who obey the general traffic laws.

[1] Dave Oyler, Nan Li, Anouck Girard and Ilya Kolmanovsky are with the Department of Aerospace Engineering, University of Michigan, 1320 Beal Avenue, 48109-2140 Ann Arbor, MI, USA {dwoyler, nanli, anouck, ilya}@umich.edu

The research of I. Kolmanovsky and N. Li is supported by the National Science Foundation under award number CNS-1544844.

[2] Yildiray Yildiz is with the Department of Mechanical Engineering, Bilkent University, 06800 Cankaya, Ankara, Turkey yyildiz@bilkent.edu.tr

978-1-4673-8682-1/$31.00 ©2016 AACC

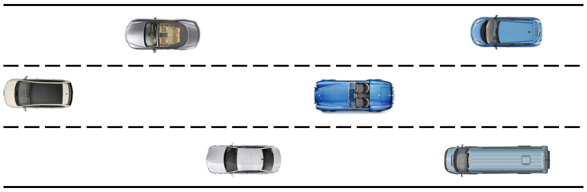


Fig. 1: Traffic in a 3-lane highway.

A. Action space

Drivers have 5 basic actions:

1) 'Maintain' current speed
2) 'Accelerate', provided velocity does not exceed 110 $\mathrm{km/h}$
3) 'Decelerate', provided velocity is above $50\ \mathrm{km/h}$
4) Move to the lane on the left
5) Move to the lane on the right

Remark: It is noted that the actions mentioned above are high level decisions. Acceleration, deceleration, and lane changes are not immediate and occur in response to these high level decisions as determined by the vehicle level control and dynamics.

In this paper, acceleration and deceleration occur at rates of $2.5\ \mathrm{m/s^2}$ and $-2.5\ \mathrm{m/s^2}$, respectively, and lane changes occur with constant lateral velocity such that the total time to change lanes is 3s. During lane changes, the longitudinal velocity remains constant. Cars are able to change actions while accelerating/decelerating before the target speed is reached, but once a lane change begins, it always continues to completion.

B. Observation space

In real traffic flow, a driver can neither observe nor process all the information about all the cars on the road. A human can possibly observe and use the information he/she obtains from the cars in a certain vicinity of him/her. Therefore, we assign the following observation space for the drivers:

1) The distance from the car in front, quantified as "close" (distance $\leq$ 40m), "nominal" (40m$<$ distance $\leq$ 70m), or "far away" (distance $>$ 70m).
2) The distance from the car on the front left, quantified as "close", "nominal" or "far away."
3) The distance from the car on the front right, quantified as "close", "nominal" or "far away."
4) The distance from the car on the rear left, quantified as "close", "nominal" or "far away."
5) The distance from the car on the rear right, quantified as "close", "nominal" or "far away."
6) The relative motion of the car in front, quantified as "approaching" (distance decreasing), "stable" (distance not changing), or "moving away" (distance increasing).
7) The relative motion of the car in front left, quantified as "approaching", "stable" or "moving away."
8) The relative motion of the car in front right, quantified as "approaching", "stable" or "moving away."
9) The relative motion of the car in rear left, quantified as "approaching", "stable" or "moving away"
10) The relative motion of the car in rear right, quantified as "approaching", "stable" or "moving away."

C. Reward function

The "reward function" is a mathematical representation of the goals of a driver. Basic goals of the drivers in real traffic are to not have a collision, to maximize their headway, and to minimize their driving effort.

The reward function *R* reflecting these goals is defined as

$$R = w_1 c + w_2 h + w_3 e, \tag{1}$$

where $w_i, i = 1, 2, 3$ are the weights for each term and c, h and e represents "collision", "headway" and "effort." These terms are explained below.

c (collision): The term c gets the value of "-10" when a collision occurs and the value of "0", otherwise.

h (headway): The term h gets the following values depending on the headway distance

$$h = \begin{cases} 10 & \text{if headway}\geq\text{70m} \\ 0 & \text{if 40m}\leq\text{headway}<\text{70m} \\ -5 & \text{if headway}<\text{40m} \end{cases} \tag{2}$$

e (effort): The term e gets the value of -10 if the driver's action is different than his/her previous action and 0 otherwise.

The weighting terms w_i may change depending on the aggressiveness of the driver but intuitively, collision avoidance must always be the most important factor and keeping a safe headway distance should typically be more important than effortless driving. So, although the weights may vary from driver to driver, the following relationship between the weights is reasonable:

$$w_1 > w_2 > w_3 \tag{3}$$

III. Obtaining Driver Policies

A policy is defined as a map from the observation space to the action space. In this paper, the map and hence the policies are stochastic: As in the case of real life, the drivers have a probability distribution over their possible actions. To obtain this stochastic map, two main tools are utilized: Level-k approach, a subset of hierarchical reasoning, and Jaakkola reinforcement algorithm. In this section both of these tools are explained.

A. Level-k reasoning

The driver models developed in this work are based on the observation that humans use various levels of reasoning. The lowest level, level-0, represents an intelligent agent (driver) who chooses his/her actions without considering the possible actions of other agents. For example, if a driver decides to make a lane change, say, from lane A to lane B, without considering the possible actions of the other drivers in lane B, that driver is designated as a level-0 thinker.

However, if the same driver assumes that the other drivers are level-0 thinkers, and then chooses his/her actions as the best response to the possible actions of other drivers, then he/she is designated as a level-1 thinker. So, a level-k driver assumes that the rest of the drivers are level-(k-1) and acts accordingly. More detailed information about this approach can be found in [12] and [13].

1) Level-0 policy: In general, level-0 policies are considered as "reflexive" behavior, the kind of actions one takes without really taking into account other players' possible actions. These actions can be random, meaning that every possible action is given the same probability of realization given a state, or it can be a very simple behavior that is formed using very basic principles of the scenario one is in. In our scenario, level-0 behavior can be considered as follows:

$$\text{action}_{l0} = \begin{cases} \text{slow down, if front car is approaching} \\ \text{accelerate to/drive at nominal speed, otherwise} \end{cases} \quad (4)$$

B. Jaakkola reinforcement learning

Jaakkola reinforcement learning (RL) algorithm (see [14]) is similar to other conventional RL methods (see [15]) in terms of having a policy evaluation step, where state-action pairs of a policy are assigned values based on the rewards gained, and a policy improvement step where the existing policy is refined so that higher valued actions for the states have increased probability of actually being played. The main distinguishing feature of Jaakkola RL algorithm is that although conventional approaches rely on the Markov property of the system for convergence guarantees, the Jaakkola RL method is developed for cases where although the underlying dynamics are Markov, the agents can not observe all of the system states and therefore the system does not appear to be Markov to the agent. This defines a Partially Observable Markov Decision Process (POMDP) and Jaakkola RL is guaranteed to converge to at least a local maximum in terms of average rewards. It is noted that the highway problem defined in this paper is a POMDP, where although the overall system is Markov, it appears non-Markovian to the drivers due to their restricted observation spaces (see Section II-B).

C. Putting everything together

To obtain a level-k driver policy, we assign level-(k-1) policies to all the drivers in the scenario except the driver we want to "train" to a level-k policy. By training we mean that we run the Jaakkola RL algorithm where the trained driver is the *learner* and the rest of the drivers, together with the vehicles, constitute the *environment*.

To start the procedure, we first assign a level-0 policy to all of the drivers except the one we want to train, then train a level-1 policy and save it. We then train a level-2 policy by assigning all of the other drivers the level-1 policy we just saved and then save the newly obtained level-2 policy. This can continue until we reach the depth of reasoning (level-k) we want to achieve. It is shown in some experimental studies (see [13]) that the probability of finding a level-3 human player is low. Humans are generally level-1, less frequently level-2 and even less frequently level-0, according to the study. Therefore, we train driver policies up to level-2 in this paper.

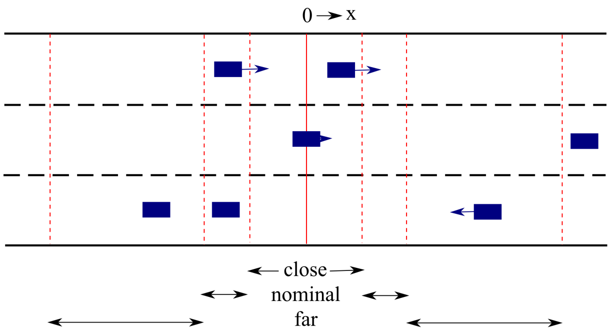


Fig. 2: Simulation Environment

IV. Simulation Results

A. Environment and Set-up

For the purposes of these simulations, the width of a lane is 3.6 meters, and all cars are 2m x 6m. Cars always drive at the center of a lane unless they are changing lanes. The longitudinal axis is called x, and its origin is colocated with the car that is to be trained or evaluated. Cars more than 400m away are considered to be out of visual range and unobservable. If no car can be observed in a position, this is considered equivalent to a car that is "far" and "moving away."

Figure 2 shows a snapshot of an example simulation with three lanes. The rectangles represent cars, which are all moving to the right, and the arrows show the velocities of the cars relative to the car under evaluation, which is located in the center lane at $x = 0$. The observation is as follows:

- Front left: close, moving away.
- Front center: far, moving away.
- Front right: far, approaching.
- Rear left: nominal, approaching.
- Rear right: nominal, stable.

Notice that two cars are unobservable in this scenario. The car in the front center position is beyond visual range, so its observed status is "moving away" even though it is actually "stable". Also, the car in the rear right "far" position is hidden by the car in the rear right "nominal" position. This reflects the POMDP nature of the problem, as discussed previously.

Initialization of a simulation requires the specification of the following values:

1) n_ℓ: the number of lanes;
2) n_c: the number of cars;
3) x^0_{max}: the maximum allowable initialization distance;
4) t_f: the simulation duration.

When a car is initialized, it is assigned to a lane randomly with uniform distribution, and then it is placed within that lane randomly with uniform distribution in $[-x^0_{max}, x^0_{max}]$

such that the distance between all previously initialized cars is at least 20m. The car is also assigned a policy to follow (level 0, 1, or 2). This process repeats until all cars have been initialized, and then the simulation proceeds according to Algorithm 1.

```
t=0;
while $t < t_f$ do
    foreach car do
        Get observation from environment.
        Select action according to policy and
        observation.
        Update position and relative velocity according
        to action.
    end
    if training a policy then
        Evaluate reward function for trainee.
        Update value function.
    end
    if trainee/evaluatee is in a collision state then
        End the simulation.
    end
    $t = t + \Delta t$
end
```

Algorithm 1: Single Episode Simulation

B. Initialization of Training

When training a new policy, the observation value function, V, for observed message m, and the action value function, Q, for message/action pair (m, a), are initialized as follows:

$$\begin{aligned} &\forall m, \quad V(m) = 0; \\ &\forall m, \forall a, \quad Q(m, a) = 0. \end{aligned} \tag{5}$$

For each observation, the actions are assigned equal probability of selection at initialization, and during each policy improvement step, if

$$\max_a Q(m, a) > V(m), \tag{6}$$

then 0.01 is added to the probability of selecting $\operatorname{argmax}_a Q(m, a)$, after which the action probabilities are normalized.

The observation space described in Section II-B has 3^{10} unique observations. In order to ensure that the learning algorithm is exposed to a large portion of the observation space, the trainee needs to be exposed to both sparsely and densely populated roads. Therefore, during training, the number of cars is selected randomly, with uniform distribution, where $0 \leq n_c \leq n_c^{max}$. The maximum number of cars, n_c^{max}, is chosen based on the number of lanes and x_{max}^0 such that if n_c^{max} cars are placed in the environment, the road will be near capacity.

Training then proceeds according to Algorithm 2.

As training progresses, the functions $V(m)$ and $Q(m, a)$ converge, and as the policy is improved, the average reward received increases. For example, consider a training scenario with $w_1 = 0.6, w_2 = 0.3$, and $w_3 = 0.1$.

```
step=0;
while step < desired training cycles do
    Randomly select $n_c \in [0, n_c^{max}]$.
    Initialize all cars with level $k - 1$ policies.
    Evaluate the level $k$ policy using Algorithm 1.
    Improve the policy.
    step=step+1.
end
```

Algorithm 2: Training Process

Figure 3 shows $V(m)$ and $Q(m, a)$ for the observation that all five vehicles are "far" and "moving away". Six values are plotted, including $V(m)$ and $Q(m, a)$ for each of the five actions associated with m. All values converge as the training progresses, and the values of actions "maintain", "move left", and "move right" converge to approximately the same value, which is higher than the value of actions "accelerate" and "decelerate". This is reasonable because w_3 is small, so taking an action other than "maintain" has only a minimal effect on the reward. However, by accelerating or decelerating, the car might move toward other cars that were previously unobservable, which could lead to decreased rewards due to less headway or possible collisions.

Figure 4 shows the average reward as training progresses. Note that in this scenario with $w_1 = 0.6, w_2 = 0.3$, and $w_3 = 0.1$, the maximum reward is $w_1(0) + w_2(10) + w_3(0) = 3$. Thus, the average reward should converge to a value near 3 if the policy leads to few collisions, maximum headway, and minimal effort. The average reward can also be expected to be slightly below 3 due to the occasional requirement to take an action at a cost of $w_3(-10) = -1$ in order to avoid a collision with a cost of $w_1(-10) = -6$.

An example simulation can be seen in Figure 5, which shows a level-2 driver, represented by ○ driving on a 3 lane road with level-1 drivers, represented by □. The direction of travel is to the right, and the vertical dotted lines represent the boundaries between the "close", "nominal", and "far" regions for the level-2 driver. Note that for clarity, this figure limits the amount of road shown, and there are additional cars farther ahead of the level-2 driver, as well as behind it.

Figure 5a shows the initial configuration, where a car is located in the "nominal" region ahead of the level-2 driver, causing reduced rewards. The level-2 driver therefore moves to the right lane, as shown in Figure 5b. The level-1 car in the upper left corner of the figure can also be seen changing lanes to improve its reward.

During the simulation, a car approaches from the front from outside the figure, which causes the car in the upper right corner of Figure 5b to decelerate. The two cars in front of the level-2 driver begin to interact and change lanes. This also causes the level-2 driver to change lanes in order to maintain headway. Figure 5c shows a later moment where the two front cars have moved into the "close" region and one of them is entering the level-2 driver's lane. This causes the level-2 driver to decelerate, which can be seen by comparing the relative vehicle positions in Figures 5c and 5d.

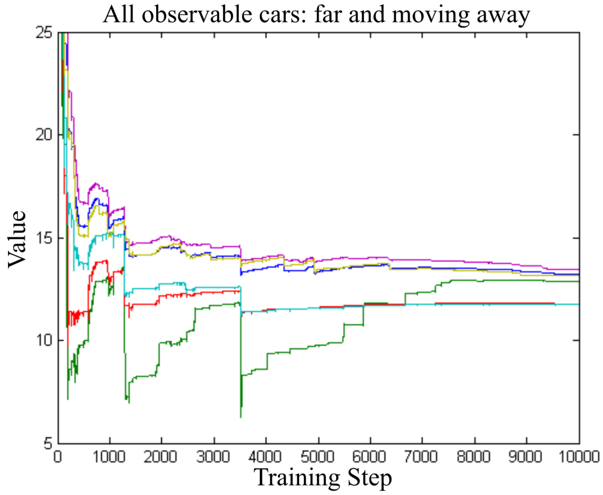


Fig. 3: Values of message and associated message/action pairs

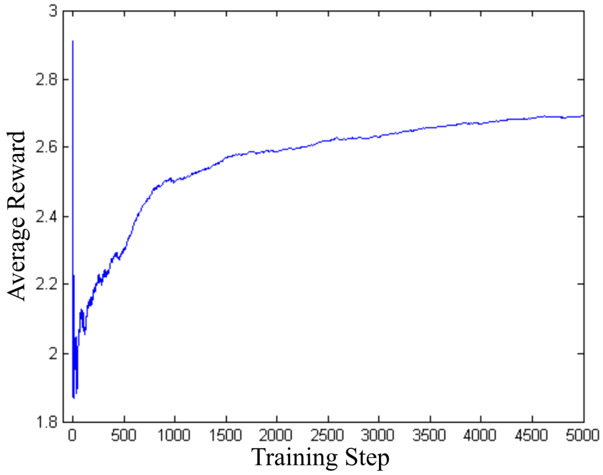
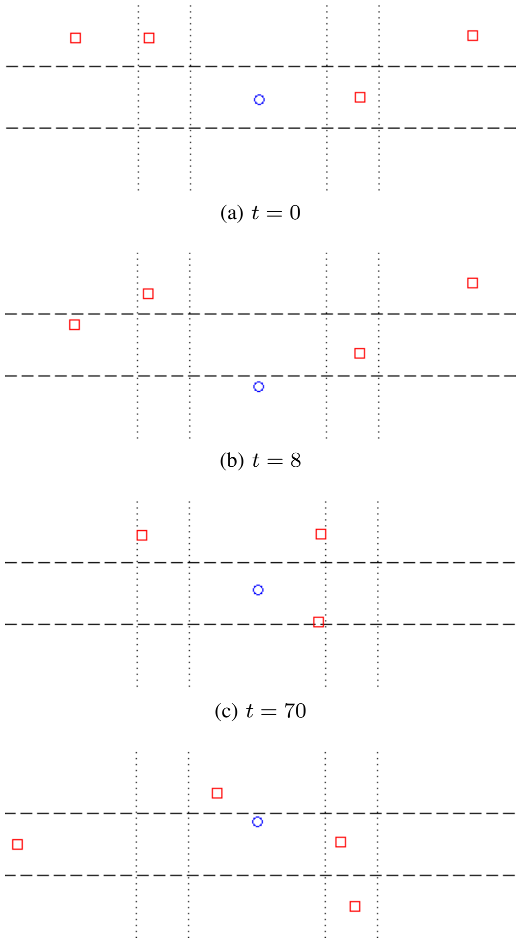


Fig. 4: Average Reward vs. Training Steps



(a) $t = 0$

(b) $t = 8$

(c) $t = 70$

(d) $t = 90$

Fig. 5: Example Simulation

C. *Comparison of Driver Profiles*

Different driver profiles can be compared by simulating multiple episodes and tracking metrics such as the number of collisions or the number of lane changes, among others. For example, consider two level-1 policies that were trained using the following weights:

Profile 1: $w_1 = 0.6, w_2 = 0.3, w_3 = 0.1$

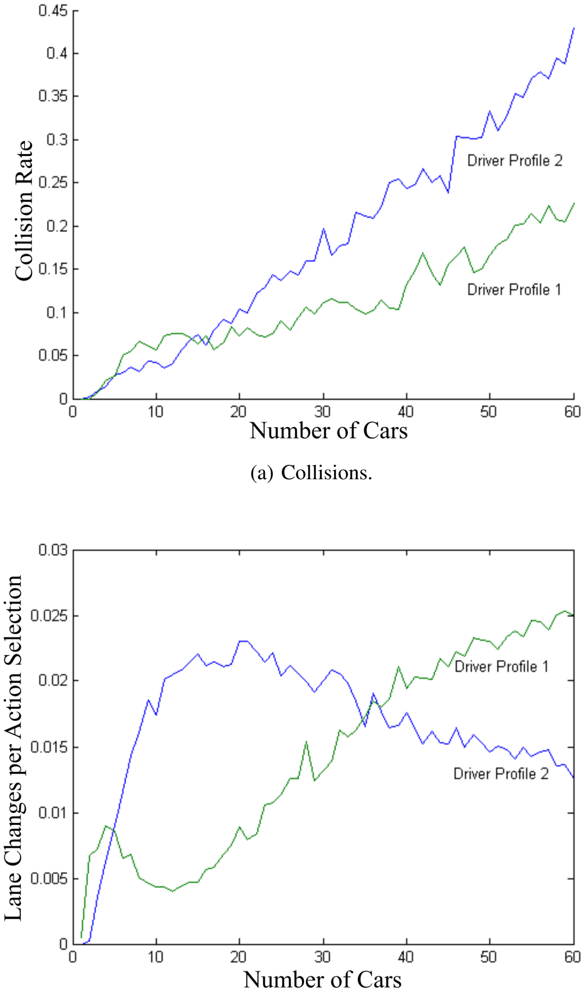
Profile 2: $w_1 = 0.4, w_2 = 0.3, w_3 = 0.3$

These profiles value headway equally, but Profile 1 gives a stricter penalty for collisions, while Profile 2 gives a relatively large penalty for changing actions at the cost of a more lenient penalty for collisions. Note that these two profiles are selected to illustrate the sensitivity of the resulting policies to the weights used for training. They are not necessarily optimal driver profiles or representative of most drivers. One area for future work is the characterization of actual driver profiles, and the primary use of these profiles is to illustrate the usefulness of the proposed methods for analyzing and comparing different driver behaviors.

Figure 6 presents the results from a series of simulations, where each curve shows the average over 1000 trials for each $n_c \in [1, 60]$, which is the abscissa in the figures. The simulations have 3 lanes, and the duration is set to $t_f = 200$sec with $x^0_{max} = 600$m and the maximum visual range set to 400m.

Figure 6a shows the fraction of simulations that end in collision for both of the driver profiles. As expected, as the traffic density increases, so does the collision rate. Also, Profile 1 has a lower collision rate than Profile 2 due to the higher penalty placed on collisions during training.

The collision rates in Figure 6a are higher than the rates typically observed in traffic. This could be due to insufficient control authority (e.g., an additional action could be added to decelerate at a faster rate). It could also be due to insufficient exploration of the observation space during policy training. Alternatively, the high rates could be due to the limited information available in the observations, which leads to a reduced ability to take necessary deconflicting actions. However, as previously discussed, the proposed method provides high level commands, and it can be paired with lower level controllers to improve local interactions while maintaining



(a) Collisions.

(b) Lane Changes.

Fig. 6: Driver Profile Comparison.

its advantages for large scale scenarios. Finally, note that these collision rates do not account for relative velocity, and therefore they include contact with zero relative velocity. These types of collisions can be addressed by simply adding a small safety region around each vehicle to avoid physical contact. These ideas will be explored in future work.

Figure 6b shows the number of lane changes per action selection for the two driver profiles. After an initial transient, the number of lane changes for Profile 2 decreases as the rate of collisions increases due to the fact that w_1 and w_3 have similar weights. Profile 1, on the other hand, prefers to change lanes more often if it reduces the collision rate, which is consistent with the expected behavior when collisions are associated with a much larger penalty than changes to the current action.

V. Summary

In this paper, we presented a game theoretical approach to represent and model interacting driver behavior in traffic. We utilized hierarchical reasoning to model the interaction between drivers and reinforcement learning to obtain driver policies for time-extended (multi-move) scenarios. We demonstrated, via simulations of a 60-car scenario, that the proposed method provides reasonable driver behavior under the given traffic conditions. In addition, we provided statistical analysis where the effects of driver aggressiveness on the number of collisions and on number of lane changes were investigated. It is noted that the proposed approach is easily scalable in terms of both the number of drivers and the number of actions (in time). This makes the proposed method more suitable for real life traffic modeling as well as implementation in high-fidelity simulators to evaluate competing autonomous driving algorithms. In addition, the proposed game theoretic approach results in control policies that can be incorporated into the autonomous driving algorithms.

References

[1] A. Carvalho, S. Lefevre, G. Schildbach, J. Kong, and F. Borrelli, “Automated driving: The role of forecasts and uncertainty - a control perspective,” *European Journal of Control*, vol. 24, pp. 14–32, 2015.

[2] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, N. Zych, E. Garcia, M. Kurdziel, and H. Fujishima, “Team cornell’s skynet: Robust perception and planning in an urban environment,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.

[3] A. G. Cunnigham, E. Galceran, R. Eustice, and E. Olson, “Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving,” in *Proceedings of ICRA*, 2015.

[4] M. Plchl and J. Edelmann, “Driver models in automobile dynamics application,” *Vehicle System Dynamics*, vol. 45, no. 7-8, pp. 699–741, 2007.

[5] D. Salvucci, E. Boer, and A. Liu., “Toward an integrated model of driver behavior in cognitive architecture,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1779, pp. 9–16, 2001.

[6] P. Hidas, “Modelling lane changing and merging in microscopic traffic simulation,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5, pp. 351–371, 2002.

[7] R. Lee and D. Wolpert, *Chapter: Game theoretic modeling of pilot behavior during mid-air encounters*. in Decision making with multiple imperfect decision makers. Intelligent Systems Reference Library Series. Springer, 2011.

[8] J. H. Yoo and R. Langari, “Stackelberg game based model of highway driving,” in *Proc. ASME Dynamic Systems and Control Conference joint with JSME Motion and Vibration Conference*, Fort Lauderdale, Florida, Oct. 2012.

[9] ——, “A stackelberg game theoretic driver model for merging,” in *Proc. ASME Dynamic Systems and Control Conference*, Palo Alto, California, Oct. 2013.

[10] C. Dextreit and I. V. Kolmanovsky, “Game theory controller for hybrid electric vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 652–663, 2014.

[11] Y. Yildiz, A. Agogino, and G. Brat, “Predicting pilot behavior in medium-scale scenarios using game theory and reinforcement learning,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1335–1343, 2014.

[12] D. Stahl and P. Wilson, “On players models of other players: Theory and experimental evidence,” *Games and Economic Behavior*, vol. 10, no. 1, p. 218254, 1995.

[13] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberri, “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] T. Jaakkola, P. S. Satinder, and I. Jordan., “Reinforcement learning algorithm for partially observable markov decision problems,” *Advances in Neural Information Processing Systems 7: Proceedings of the 1994 Conference*, 1994.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge: MIT press, 1998.