

A Traffic Simulation Model with Interactive Drivers and High-fidelity Car Dynamics

Guankun Su* Nan Li* Yildiray Yildiz** Anouck Girard*
Ilya Kolmanovsky*

* *Department of Aerospace Engineering, University of Michigan,
Ann Arbor, MI 48109, USA (e-mail:
{guanksu,nanli,anouck,ilya}@umich.edu).*

** *Department of Mechanical Engineering, Bilkent University,
Ankara, Turkey (e-mail: yyildiz@bilkent.edu.tr).*

Abstract: We integrate a set of game-theoretic driver decision-making models with the high-fidelity car driving simulator *The Open Racing Car Simulator* (TORCS). The game-theoretic driver models simulate the interactive decision making processes of the drivers and TORCS simulates vehicle dynamics in multi-vehicle highway traffic scenarios. We use the integrated simulator to collect human driving data and then use these data to validate and re-calibrate our driver and traffic models. Such an integrated simulator can be used in the development, verification and validation of automated driving functions.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Driver models, Simulators, Autonomous vehicles, Human-machine interface

1. INTRODUCTION

Advanced driver assistance systems (ADAS) and autonomous driving functions (ADF) have been rapidly advancing in recent years, with the promise to provide safer, cleaner, and more efficient everyday transportation (Anderson et al. (2014)). Traffic and vehicle simulators, where real traffic scenarios are modeled, can support the development of these systems.

One of the most significant challenges faced by ADAS and ADF developers is the verification and validation of these systems and functions in terms of safety and performance. Hundreds of millions of miles of driving tests are required to demonstrate the same level of reliability of these systems to that of average human drivers (Kalra and Paddock (2016)). Simulators may be used for fast and safe virtual tests of these systems to reduce the time and effort needed for road tests (Gechter et al. (2012); Li et al. (2018b, 2019)). Simulators may also be used as platforms to collect human driving data in different traffic scenarios and different road and weather conditions in order to better understand human driving behavior (Lowden et al. (2009); Bella et al. (2014)). Furthermore, simulators may be used to train perception systems and control algorithms for autonomous driving exploiting machine learning techniques (Chen et al. (2015); Zhang and Cho (2017)).

In traffic scenarios where multiple vehicles are involved, modeling of their interactive behavior is important. In our previous publications (Oyler et al. (2016); Li et al. (2016, 2018b)), a game-theoretic approach is used for modeling the interactive driver decision making processes in multi-vehicle highway traffic scenarios.

* This research has been supported by the National Science Foundation award number CNS 1544844.

In this paper, we describe the integration of our game-theoretic driver decision-making models in Li et al. (2018b) with the vehicle simulator called TORCS (Wymann et al. (2000)). Our driver models simulate drivers' interactive decision making processes and TORCS simulates vehicle dynamics, so that their integration provides a platform for higher-fidelity highway traffic simulations. After the integrated simulator is developed, we use it to collect human driving data and then use the data to validate and re-calibrate the driver decision-making and traffic models.

2. GAME-THEORETIC DRIVER DECISION-MAKING MODEL

The approach to model drivers' interactive decision making in highway traffic scenarios developed in Li et al. (2018b) is based on level- \mathcal{K} game theory (Costa-Gomes and Crawford (2006); Costa-Gomes et al. (2009)), and is inspired by the "semi network-form game" approach proposed by Lee and Wolpert (2012). In this section, we briefly review the driver decision-making models. Similar modeling approaches have also been previously implemented in cyber-security (Backhaus et al. (2013)) and aerospace (Yildiz et al. (2014); Musavi et al. (2016)) domains, and they have been recently investigated for modeling driver/vehicle interactions in urban traffic scenarios (Li et al. (2018a)).

2.1 Driver model

We first model the car dynamics using the following point-mass discrete-time model:

$$\begin{aligned}x[k+1] &= x[k] + v_x[k] \Delta t, \\v_x[k+1] &= v_x[k] + a[k] \Delta t, \\y[k+1] &= y[k] + v_y[k] \Delta t,\end{aligned}\tag{1}$$

where k represents the discrete time, $x[k]$ and $y[k]$ represent, respectively, the vehicle's longitudinal position and lateral position, $v_x[k]$ and $v_y[k]$ represent, respectively, the vehicle's longitudinal velocity and lateral velocity, and $a[k]$ represents the vehicle's longitudinal acceleration. The Δt is the time step. We note that $y[k] = 0$ [m] corresponds to the right road boundary.

The driver decision-making model is a stochastic policy, $\pi(\gamma|m)$, that maps a driver's observations to probabilities of selecting different actions, $\pi(\gamma|m) : m \mapsto \mathbb{P}(\gamma|m)$, where $m \in M$ denotes an observation message taking values in a finite observation space, and $\gamma \in \Gamma$ denotes an action taking values in a finite action space.

The definitions of the action and observation spaces are given below:

2.1.1 Action space

- (1) *Maintain*: To keep the current speed and the current lane.
- (2) *Accelerate*: To accelerate at a rate $a = a_1$ [m/s²], provided that the speed does not exceed v_{\max} [m/s].
- (3) *Decelerate*: To decelerate at a rate $a = -a_1$ [m/s²], provided that the speed is above v_{\min} [m/s].
- (4) *Hard accelerate*: To accelerate at a rate $a = a_2$ [m/s²], provided that the speed does not exceed v_{\max} [m/s].
- (5) *Hard decelerate*: To decelerate at a rate $a = -a_2$ [m/s²], provided that the speed is above v_{\min} [m/s].
- (6) *Move to the left*: To change lanes to the left with velocity $v_y = \frac{w_{\text{lane}}}{2}$ [m/s], where w_{lane} represents the lane width, provided that there is a lane on the left.
- (7) *Move to the right*: To change lanes to the right with velocity $v_y = -\frac{w_{\text{lane}}}{2}$ [m/s], provided that there is a lane on the right.

At each discrete time instant k , one action is selected from the action space and is applied to the model (1) to update the vehicle's state (x, v_x, y) . The acceleration rate values a_1 and a_2 and the speed bounds v_{\min} and v_{\max} are tunable parameters. The lateral speed during a lane change is equal to $\frac{w_{\text{lane}}}{2}$ hence a lane change takes 2 [s] to complete.

2.1.2 Observation space

- (1) *Front center range*, d_{fc} : The longitudinal distance from the ego car to the car directly in its front.
- (2) *Front left range*, d_{fl} : The longitudinal distance from the ego car to the car in its front and in its left lane.
- (3) *Front right range*, d_{fr} : The longitudinal distance from the ego car to the car in its front and in its right lane.
- (4) *Rear left range*, d_{rl} : The longitudinal distance from the ego car to the car in its back and in its left lane.
- (5) *Rear right range*, d_{rr} : The longitudinal distance from the ego car to the car in its back and in its right lane.
- (6) *Front center range rate*, v_{fc} : The rate of change of d_{fc} .
- (7) *Front left range rate*, v_{fl} : The rate of change of d_{fl} .
- (8) *Front right range rate*, v_{fr} : The rate of change of d_{fr} .
- (9) *Rear left range rate*, v_{rl} : The rate of change of d_{rl} .
- (10) *Rear right range rate*, v_{rr} : The rate of change of d_{rr} .
- (11) *Lane*, I : The current lane of the ego car.

To account for the fact that a human driver may not be able to accurately measure these quantities, we assume that these quantities are encoded into discrete values, as

follows:

$$\widehat{d}_{\zeta}[k] = \begin{cases} \text{"close"} & \text{if } 0 \leq d_{\zeta}[k] \leq \bar{d}_c, \\ \text{"nominal"} & \text{if } \bar{d}_c < d_{\zeta}[k] \leq \underline{d}_f, \\ \text{"far"} & \text{if } d_{\zeta}[k] > \underline{d}_f \text{ or there} \\ & \text{is no car in the } \zeta \text{ position,} \end{cases} \quad (2)$$

$$\widehat{v}_{\zeta}[k] = \begin{cases} \text{"approaching"} & \text{if } v_{\zeta}[k] < -v_m, \\ \text{"stable"} & \text{if } -v_m \leq v_{\zeta}[k] \leq v_m, \\ \text{"moving away"} & \text{if } v_{\zeta}[k] > v_m \text{ or there} \\ & \text{is no car in the } \zeta \text{ position,} \end{cases} \quad (3)$$

where $\zeta \in \{\text{fl}, \text{fc}, \text{fr}, \text{rl}, \text{rr}\}$ indicates the range or range rate to a specific surrounding car, and

$$I[k] = \begin{cases} \text{"right lane"} & \text{if } y[k] < w_{\text{lane}}, \\ \text{"left lane"} & \text{if } y[k] > w_{\text{road}} - w_{\text{lane}}, \\ \text{"middle lane(s)"} & \text{otherwise.} \end{cases} \quad (4)$$

The threshold values \bar{d}_c , \underline{d}_f , v_m are tunable parameters, and the road width w_{road} is a multiple of the lane width w_{lane} .

After the encoding, the vector

$$m[k] = [\widehat{d}_{\text{fl}}[k], \widehat{d}_{\text{fc}}[k], \widehat{d}_{\text{fr}}[k], \widehat{d}_{\text{rl}}[k], \widehat{d}_{\text{rr}}[k], \widehat{v}_{\text{fl}}[k], \widehat{v}_{\text{fc}}[k], \widehat{v}_{\text{fr}}[k], \widehat{v}_{\text{rl}}[k], \widehat{v}_{\text{rr}}[k], I[k]]^{\top} \quad (5)$$

represents the observation message. The set of all message values constitutes the observation space M . Because each component of m can take 3 different values based on (2)–(4), and m has 11 dimensions, M has 3^{11} elements.

The driver policy $\pi(\gamma|m)$ is obtained based on average reward reinforcement learning (Mahadevan (1996)) to maximize the average reward associated with a Markov decision process. In particular, since the ego car can only partially observe the traffic state through the message m , the optimization problem is a partially observable Markov decision process (POMDP) and we use a specific reinforcement learning algorithm for POMDP problems (Jaakkola et al. (1995)) to obtain the policy $\pi(\gamma|m)$. The single-step reward function is defined to reflect the basic goals of a driver, including : 1) to not have an accident, such as a car crash (safety); 2) to minimize the time needed to reach the destination (performance); 3) to keep a reasonable headway from preceding cars (safety and comfort); and 4) to minimize driving effort (comfort). The definition of the reward function and more details on the applied reinforcement learning algorithm can be found in Li et al. (2018b, 2019).

2.2 Interactive traffic model

A traffic scenario usually involves multiple drivers/vehicles. To represent the drivers' interactive behavior, we use a game-theoretic approach.

In particular, the driver interaction model is based on level- \mathcal{K} game theory, where the level \mathcal{K} indicates a player's (a driver's, in our setting) reasoning depth in a multi-player game. A level- \mathcal{K} driver policy anchors its beliefs in a nonstrategic level-0 driver policy, which represents a driver's instinctive responses to traffic conditions without accounting for the interactions between herself and the other drivers. Then, a level-1 driver assumes that all the

other drivers she is interacting with are level-0, and takes optimal responses based on this assumption. Similarly, a level- \mathcal{K} driver optimally responds to a traffic model where all the drivers, except for the level- \mathcal{K} driver herself, are using the level- $(\mathcal{K}-1)$ policy. This way, after a level-0 driver policy, π_0 , is defined, level- \mathcal{K} driver policies, $\pi_{\mathcal{K}}$, can be obtained sequentially for $\mathcal{K} = 1, 2, \dots$.

Specifically, a level- $(\mathcal{K}-1)$ traffic model is constructed, consisting of multiple drivers/vehicles using the level- $(\mathcal{K}-1)$ policy. Then it provides a training environment for the level- \mathcal{K} driver policy, where the level- \mathcal{K} policy is solved for using a reinforcement learning algorithm. For more details on level- \mathcal{K} game theory and its role in obtaining the driver interaction model, see Li et al. (2018b, 2019).

After the level- \mathcal{K} driver policies for $\mathcal{K} = 0, 1$ and 2 are obtained, a heterogeneous and interactive highway traffic model can be constructed using a mixture of level-0, 1 and 2 driver models.

3. INTEGRATING DRIVER DECISION-MAKING MODELS WITH TORCS

We integrate the game-theoretic driver decision-making models described in Section 2 with the vehicle simulator called TORCS. TORCS models 1) rigid-body dynamics, including the mass and rotational inertia of the vehicle; 2) chassis dynamics, including suspensions, links and differentials; 3) tire dynamics for different ground types; and 4) aerodynamics, including slip-streaming and ground effects (Wymann et al. (2000)).

Sensor information provided by TORCS API related to our driver decision-making models includes: the longitudinal distance from the start line, the relative lateral position with respect to the center of the track, and the speed of every car in a simulation. These signals are used to calculate the quantities in Section 2.1.2. The effectors in TORCS related to the control of a car are listed in Table 1.

Effectors	Description
Accelerator	0=none, 1=full throttle
Brake	0=none, 1=full brake
Steering	-1=full right, 1=full left

Table 1: Effectors in TORCS

We design the controls for the effectors in Table 1 to represent human driving behavior. The actions generated by the high-level decision-making model, which take values in the finite action space in Section 2.1.1, represent the desired states the driver wants to reach, such as desired speeds and desired lanes. We design low-level controllers to realize smooth maneuvers. In particular, we design a proportional-integral-derivative (PID) controller to control the vehicle's longitudinal speed and design a proportional-derivative (PD) controller to control the vehicle's lateral motion for lane keeping and lane change. They are presented in the following subsections.

3.1 Longitudinal speed control

At the discrete time instant k , an action command generated from the decision-making policy defines a reference

longitudinal speed through the model (1), denoted by $v_x^{\text{ref}}[k+1]$. The actual longitudinal speed of the vehicle is denoted by $v_x(t)$, where $t \in [k\Delta t, (k+1)\Delta t)$ represents continuous time. We note that TORCS uses the update frequency of 50 Hz, which is much higher than our decision update frequency. Thus, we represent the state updates in TORCS simulations as continuous to simplify the presentation, although the actual updates are discrete-time.

We define the normalized error between the reference speed and the actual speed as

$$e^v(t) = \frac{v_x^{\text{ref}}[k+1] - v_x(t)}{v_x^{\text{ref}}[k+1]}. \quad (6)$$

We normalize the error so that $e^v(t)$ is a dimensionless quantity. Then, the control input is calculated as

$$u^v(t) = k_p^v e^v(t) + k_i^v \int_{k\Delta t}^t e^v(\tau) d\tau + k_d^v \frac{de^v}{dt}(t), \quad (7)$$

saturated to the range $[-1, 1]$. If $u^v(t) \in [0, 1]$, we set the effector “accelerator” to the value of $u^v(t)$ and set “brake” to 0; if $u^v(t) \in [-1, 0)$, we set “brake” to the value of $-u^v(t)$ and set “accelerator” to 0. The PID gains for longitudinal speed control are tuned to provide satisfactory performance.

3.2 Lateral motion control

At the discrete time instant k , an action command generated from the decision-making policy defines a target lane, $I_{\text{target}} \in \{\text{“right”}, \text{“middle”}, \text{“left”}\}$. The lateral motion controller has two objectives including 1) lane keeping, if the target lane is the same as the current lane, and 2) lane change, if the target lane is different from the current lane.

The lateral motion control schematics are illustrated in Fig. 1. We let the center of the target lane be the reference lateral position, $y^{\text{ref}}[k+1]$, and define an angle error $e^\phi(t)$, which is the angle between the direction of the vehicle's current velocity and the line connecting the vehicle's current position $(x(t), y(t))$ to a virtual reference point $(x^{\text{ref}}(t), y^{\text{ref}}(t))$, i.e.,

$$e^\phi(t) = \tan^{-1} \frac{y^{\text{ref}}(t) - y(t)}{x^{\text{ref}}(t) - x(t)} - \tan^{-1} \frac{v_y(t)}{v_x(t)}, \quad (8)$$

where

$$y^{\text{ref}}(t) = y(k\Delta t) + \frac{t - k\Delta t}{T} (y^{\text{ref}}[k+1] - y(k\Delta t)),$$

$$x^{\text{ref}}(t) = \begin{cases} x(t) + l_{\text{car}} + 0.01 T v_x(t), & \text{if lane keeping,} \\ x(t) + 1.2 l_{\text{car}} + 0.1 T v_x(t), & \text{if lane change,} \end{cases}$$

for $t \in [k\Delta t, k\Delta t + T)$, where T is a time constant approximately equal to the time for a lane change to complete, and l_{car} represents the length of the vehicle. Note that $t = k\Delta t$ is the continuous time when the action decision is made, which corresponds to the discrete time k . In the case of lane change, $t = k\Delta t$ represents the time to start changing lanes. Note also that $y^{\text{ref}}(t)$ is designed in such a way that $y^{\text{ref}}(k\Delta t + T) = y^{\text{ref}}[k+1]$. Then, the control input is calculated as

$$u^y(t) = k_p^y e^\phi(t) + k_d^y \frac{de^\phi}{dt}(t), \quad (9)$$

saturated to the range $[-1, 1]$. We set the effector “steering” to the value of $u^y(t)$. The PD gains for lateral motion control are tuned to provide satisfactory performance.

The controlled response of the vehicle during a lane change is shown in Fig. 2. The blue solid curve represents the vehicle's $(x(t), y(t))$ -trajectory during the lane change. The two red dashed lines represent the center of the right lane and the center of the middle lane. We can observe that the lane change is performed stably and smoothly, implying that the combination of our designed longitudinal speed controller and lateral motion controller is effective.

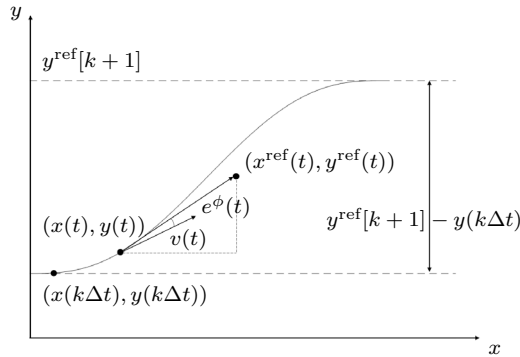


Fig. 1: Lateral motion control schematics.

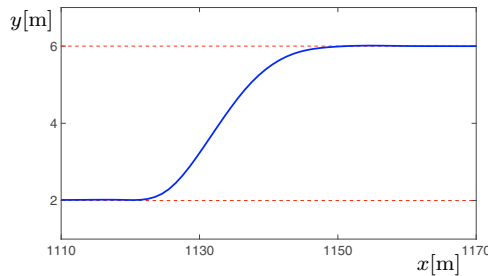


Fig. 2: Lane change response.

4. APPLICATIONS

A heterogeneous and interactive highway traffic simulation model is constructed using a mixture of level-0, 1 and 2 driver models. Such a model has been used for the verification, validation, and calibration of autonomous vehicle path planning systems in our previous publications (Li et al. (2018b, 2019)). The performance of an autonomous vehicle path planning system in terms of user-defined metrics can be evaluated based on simulation results. Critical scenarios can be extracted and recorded for post-analysis. In Li et al. (2018b, 2019), the simulations are based on the point-mass discrete-time model (1). Now, with the higher-fidelity car dynamics, road conditions, and other environmental factors such as light conditions simulated in TORCS, an integrated autonomous vehicle control system for highways that may include a perception subsystem, a high-level path planning subsystem, and several low-level dynamics and actuation control subsystems can be tested using our developed traffic simulation model. Such a comprehensive testing is left as a topic of our future research.

In this paper, we use the integrated simulator as a platform to collect human driving data. These data may be used to analyze human driving behavior. In particular, in this paper we compare the level- \mathcal{K} driver policies and human

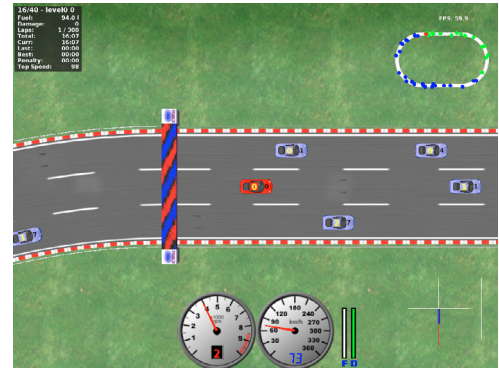


Fig. 3: Simulator interface. In this simulation, the cars are driving on an oval track. Other track types can also be simulated.

driving data, and use human driving data to re-calibrate the highway traffic model constructed based on the level- \mathcal{K} driver policies.

4.1 Data collection

A human operator drives a car of the simulator using a driver control setup consisting of a steering wheel and a pair of gas and brake pedals. The human-driven car interacts with our traffic model where 10% drivers are modeled using the level-0 policy, 60% drivers are modeled using the level-1 policy, and 30% drivers are modeled using the level-2 policy. These percentages of various levels are set, motivated by the experimental results in Costa-Gomes and Crawford (2006); Costa-Gomes et al. (2009).

We record traffic data including the states (the longitudinal distance from the start line, the relative lateral position with respect to the center of the track, and the speed) of the human-driven car and of all other cars in traffic, based on which we calculate the observation quantities in Section 2.1.2 and the encoded observation message (5). The data acquisition sampling rate is 5 Hz. Furthermore, we decode the human driver's behavior into the discrete actions in Section 2.1.1, to facilitate comparing human driving behavior and the level- \mathcal{K} decision-making policies, as follows:

- 1) *Maintain*: No lane change occurs and the rate of change of speed is in the range $[-a_m, a_m]$ $[\text{m/s}^2]$.
- 2) *Accelerate*: No lane change occurs and the rate of change of speed is in the range $(a_m, a_a]$ $[\text{m/s}^2]$.
- 3) *Decelerate*: No lane change occurs and the rate of change of speed is in the range $[-a_a, -a_m]$ $[\text{m/s}^2]$.
- 4) *Hard accelerate*: No lane change occurs and the rate of change of speed is in the range $(a_a, +\infty)$ $[\text{m/s}^2]$.
- 5) *Hard decelerate*: No lane change occurs and the rate of change of speed is in the range $(-\infty, -a_a)$ $[\text{m/s}^2]$.
- 6) *Move to the left*: A left lane change occurs.
- 7) *Move to the right*: A right lane change occurs.

The threshold values a_m and a_a are set in accordance with the values of a_1 and a_2 . The current lane of the human-driven car is identified according to the distances from the car's current lateral position to the center of each lane and is set to the lane corresponding to the smallest distance.

Parameter	Value	Parameter	Value
Δt	1	k_p^v	10
a_1	2.5	k_i^v	0.1
a_2	5	k_d^v	0.01
v_{\max}	98/3.6	k_p^y	0.7
v_{\min}	62/3.6	k_d^y	0.01
\bar{d}_c	21	T	1.5
\bar{d}_f	42		
\underline{v}_m	2/3.6	a_m	0.1
w_{lane}	4	a_a	2.5
w_{road}	12		

Table 2: All parameter values used in the simulations (in SI units).

4.2 Data analysis

To compare human driving behavior and the level- \mathcal{K} policies, we first define a policy that represents average human driving behavior in the form of $\pi_{\mathcal{H}}(\gamma|m) : m \mapsto \mathbb{P}_{\mathcal{H}}(\gamma|m)$, for each $m \in M$ that is encountered in the data collection phase. The probability $\mathbb{P}_{\mathcal{H}}(\gamma|m)$ is estimated by

$$\mathbb{P}_{\mathcal{H}}(\gamma|m) \approx \frac{n(\gamma|m)}{n(m)}, \quad (10)$$

where $n(m)$ is the number of encounters of the observation message m , and $n(\gamma|m)$ is the number of encounters of the message-action pair (m, γ) . Then, we use a weighted Hellinger distance to represent the difference between the human driver policy, $\pi_{\mathcal{H}}$, and each level- \mathcal{K} policy, $\pi_{\mathcal{K}}$, as follows:

$$D_{\mathcal{H},\mathcal{K}} = \sum_{m \in M} w(m) H(\pi_{\mathcal{H}}(\cdot|m), \pi_{\mathcal{K}}(\cdot|m)). \quad (11)$$

Here the Hellinger distance $H(\pi_{\mathcal{H}}(\cdot|m), \pi_{\mathcal{K}}(\cdot|m)) \in [0, 1]$ is used to quantify the similarity between the two probability distributions $\pi_{\mathcal{H}}(\cdot|m)$ and $\pi_{\mathcal{K}}(\cdot|m)$ (the smaller it is, the more similar the two probability distributions are) and is defined as (Nikulin (2001)):

$$H(\pi_{\mathcal{H}}(\cdot|m), \pi_{\mathcal{K}}(\cdot|m)) = \frac{1}{\sqrt{2}} \left(\sum_{i=1}^7 (\sqrt{\mathbb{P}_{\mathcal{H}}(\gamma_i|m)} - \sqrt{\mathbb{P}_{\mathcal{K}}(\gamma_i|m)})^2 \right)^{\frac{1}{2}}, \quad (12)$$

where γ_i , $i = 1, \dots, 7$, represent the seven actions in Section 2.1.1.

The weights $w(m)$ are chosen based on the frequency of message m being encountered, according to

$$w(m) = \frac{n(m)}{\sum_{m' \in M} n(m')}. \quad (13)$$

In particular, we assume that $w(m)$ is determined mainly by the overall traffic and is not significantly influenced by the policy of a single car. Thus, to obtain an estimate of $w(m)$, we can run a simulation where there are only level- \mathcal{K} cars and no human-driven cars in the traffic. Since no human operator needs to be involved in such a simulation, a large period of simulation time can be run in a small period of real time, to obtain more accurate estimates of the frequencies.

For comparison, we also compute the weighted Hellinger distances between the level- \mathcal{K} policies,

$$D_{\mathcal{K}_1, \mathcal{K}_2} = \sum_{m \in M} w(m) H(\pi_{\mathcal{K}_1}(\cdot|m), \pi_{\mathcal{K}_2}(\cdot|m)). \quad (14)$$

Furthermore, since the traffic model consists of 10% level-0 drivers, 60% level-1 drivers, and 30% level-2 drivers, we define the average policy of the traffic, $\pi_{\mathcal{T}}$, as

$$\pi_{\mathcal{T}}(\cdot|m) = 0.1 \pi_0(\cdot|m) + 0.6 \pi_1(\cdot|m) + 0.3 \pi_2(\cdot|m), \quad (15)$$

for each $m \in M$, and compute the weighted Hellinger distance between $\pi_{\mathcal{H}}$ and $\pi_{\mathcal{T}}$,

$$D_{\mathcal{H},\mathcal{T}} = \sum_{m \in M} w(m) H(\pi_{\mathcal{H}}(\cdot|m), \pi_{\mathcal{T}}(\cdot|m)). \quad (16)$$

We have collected approximately 50 [min] of human driving data. The weighted Hellinger distances between each pair of policies are summarized in Table 3.

Pair	Distance	Pair	Distance
$D_{\mathcal{H},0}$	0.79384	$D_{0,1}$	0.79665
$D_{\mathcal{H},1}$	0.54581	$D_{1,2}$	0.70617
$D_{\mathcal{H},2}$	0.70393	$D_{0,2}$	0.79601
$D_{\mathcal{H},\mathcal{T}}$	0.50615		

Table 3: Weighted Hellinger distances.

From Table 3 we can observe that among the level- \mathcal{K} policies, the human driving behavior is most similar to level-1. Reasons for this may include: 1) level-1 reasoners are observed to be most common among the population in the experiments in Costa-Gomes and Crawford (2006); Costa-Gomes et al. (2009), and 2) the level-1 policy is most aggressive among level-0, 1 and 2 policies (Li et al. (2018b)), and human drivers tend to behave aggressively in simulated driving (Matthews et al. (1998)).

On the other hand, we can observe that the traffic model $\pi_{\mathcal{T}}$, based on a mixture of 10% level-0 drivers, 60% level-1 drivers, and 30% level-2 drivers, matches the human driver policy $\pi_{\mathcal{H}}$ best (with the smallest weighted Hellinger distance). We examine different percentage combinations of the level- \mathcal{K} policies, i.e., define

$$\pi_{\mathcal{T}'}(\cdot|m) = p_0 \pi_0(\cdot|m) + p_1 \pi_1(\cdot|m) + (1 - p_0 - p_1) \pi_2(\cdot|m), \quad (17)$$

and plot the weighted Hellinger distance between the human driver policy $\pi_{\mathcal{H}}$ and the mixed policy $\pi_{\mathcal{T}'}$ as a function of (p_0, p_1) in Fig. 4. The optimal traffic model in terms of matching the human driver policy $\pi_{\mathcal{H}}$ estimated from our current simulated driving data, is composed of 20% level-0 drivers, 70% level-1 drivers, and 10% level-2 drivers, with a weighted Hellinger distance $D_{\mathcal{H},\mathcal{T}'} = 0.4870$.

5. CONCLUDING REMARKS

In this paper, we described the integration of our previously developed game-theoretic driver decision-making models with the car driving simulator TORCS, to construct a heterogeneous and interactive traffic simulation model with higher-fidelity car dynamics.

We used the traffic model to collect human driving data and decoded human driving data to create a human driver policy, which was used for the validation and re-calibration of the driver and traffic models. It is worth mentioning that it is in general difficult to validate human driver models (Brackstone and McDonald (1999)).

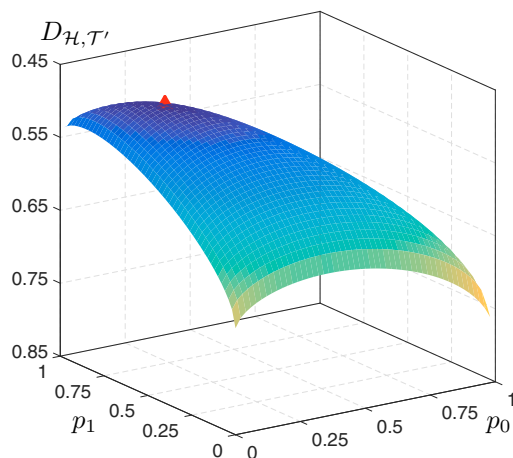


Fig. 4: Weighted Hellinger distance $D_{\mathcal{H},\mathcal{T}'}$ as a function of (p_0, p_1) . The red point denotes the highest point on the surface, corresponding to $(p_0, p_1) = (0.2, 0.7)$ and $D_{\mathcal{H},\mathcal{T}'} = 0.4870$.

On the other hand, another application of the developed traffic simulation model could be to the verification, validation, and calibration of integrated autonomous vehicle control systems. This is left for our future research.

REFERENCES

- Anderson, J.M., Nidhi, K., Stanley, K.D., Sorensen, P., Samaras, C., and Oluwatola, O.A. (2014). *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation.
- Backhaus, S., Bent, R., Bono, J., Lee, R., Tracey, B., Wolpert, D., Xie, D., and Yildiz, Y. (2013). Cyber-physical security: A game theory model of humans interacting over control systems. *IEEE Transactions on Smart Grid*, 4(4), 2320–2327.
- Bella, F., Calvi, A., and D’Amico, F. (2014). Analysis of driver speeds under night driving conditions using a driving simulator. *Journal of safety research*, 49, 45–e1.
- Brackstone, M. and McDonald, M. (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4), 181–196.
- Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Computer Vision (ICCV), 2015 International Conference on*, 2722–2730. IEEE.
- Costa-Gomes, M.A. and Crawford, V.P. (2006). Cognition and behavior in two-person guessing games: An experimental study. *American Economic Review*, 96(5), 1737–1768.
- Costa-Gomes, M.A., Crawford, V.P., and Iriberry, N. (2009). Comparing models of strategic thinking in van huyck, battalio, and beil’s coordination games. *Journal of the European Economic Association*, 7(2-3), 365–376.
- Gechter, F., Contet, J.M., Galland, S., Lamotte, O., and Koukam, A. (2012). Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation. *Simulation Modelling Practice and Theory*, 24, 103–114.
- Jaakkola, T., Singh, S.P., and Jordan, M.I. (1995). Reinforcement learning algorithm for partially observable markov decision problems. In *Advances in Neural Information Processing Systems*, 345–352.
- Kalra, N. and Paddock, S.M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94, 182–193.
- Lee, R. and Wolpert, D. (2012). Game theoretic modeling of pilot behavior during mid-air encounters. In *Decision Making with Imperfect Decision Makers*, 75–111. Springer.
- Li, N., Kolmanovsky, I., Girard, A., and Yildiz, Y. (2018a). Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control. In *2018 American Control Conference (ACC)*, 3215–3220. IEEE.
- Li, N., Oyler, D., Zhang, M., Yildiz, Y., Girard, A., and Kolmanovsky, I. (2016). Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems. In *Decision and Control (CDC), 2016 55th Conference on*, 727–733. IEEE.
- Li, N., Oyler, D.W., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A.R. (2018b). Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on Control Systems Technology*, 26(5), 1782–1797.
- Li, N., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A. (2019). Game theory-based traffic modeling for calibration of automated driving algorithms. In *Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions*, 89–106. Springer.
- Lowden, A., Anund, A., Kecklund, G., Peters, B., and Åkerstedt, T. (2009). Wakefulness in young and elderly subjects driving at night in a car simulator. *Accident Analysis & Prevention*, 41(5), 1001–1007.
- Mahadevan, S. (1996). Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1-3), 159–195.
- Matthews, G., Dorn, L., Hoyes, T.W., Davies, D.R., Glendon, A.I., and Taylor, R.G. (1998). Driver stress and performance on a driving simulator. *Human Factors*, 40(1), 136–149.
- Musavi, N., Onural, D., Gunes, K., and Yildiz, Y. (2016). Unmanned aircraft systems airspace integration: A game theoretical framework for concept evaluations. *Journal of Guidance, Control, and Dynamics*, 96–109.
- Nikulin, M.S. (2001). Hellinger distance. *Encyclopedia of mathematics*, 151.
- Oyler, D.W., Yildiz, Y., Girard, A.R., Li, N.I., and Kolmanovsky, I.V. (2016). A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development. In *2016 American Control Conference (ACC)*, 1705–1710. IEEE.
- Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., and Sumner, A. (2000). TORCS: The open racing car simulator. <http://torcs.sourceforge.net>.
- Yildiz, Y., Agogino, A., and Brat, G. (2014). Predicting pilot behavior in medium-scale scenarios using game theory and reinforcement learning. *Journal of Guidance, Control, and Dynamics*, 37(4), 1335–1343.
- Zhang, J. and Cho, K. (2017). Query-efficient imitation learning for end-to-end simulated driving. In *Conference on Artificial Intelligence*, 2891–2897. AAAI.