

# Chapter 5

## Game Theory-Based Traffic Modeling for Calibration of Automated Driving Algorithms



Nan Li, Mengxuan Zhang, Yildiray Yildiz, Ilya Kolmanovsky  
and Anouck Girard

**Abstract** Automated driving functions need to be validated and calibrated so that a self-driving car can operate safely and efficiently in a traffic environment where interactions between it and other traffic participants constantly occur. In this paper, we describe a traffic simulator capable of representing vehicle interactions in traffic developed based on a game-theoretic traffic model. We demonstrate its functionality for parameter optimization in automated driving algorithms by designing a rule-based highway driving algorithm and calibrating the parameters using the traffic simulator.

### 5.1 Introduction

Significant efforts have been put into research, development, and implementation of automated driving algorithms, aiming at making self-driving cars a viable option for everyday transportation, for example, see [2, 16]. Several challenges and issues in

---

N. Li (✉) · M. Zhang · I. Kolmanovsky · A. Girard  
Department of Aerospace Engineering, University of Michigan,  
1320 Beal Avenue, Ann Arbor, MI 48109, USA  
e-mail: nanli@umich.edu

M. Zhang  
e-mail: mengxuan@umich.edu

I. Kolmanovsky  
e-mail: ilya@umich.edu

A. Girard  
e-mail: anouck@umich.edu

Y. Yildiz  
Department of Mechanical Engineering, Bilkent University, 06800 Ankara, Turkey  
e-mail: yyildiz@bilkent.edu.tr

technical, legal, and social aspects are faced and must be addressed by automated driving researchers to achieve this goal—some of these challenges and issues are summarized in [8, 11].

To validate an automated driving algorithm and calibrate its parameters, hundreds of thousands of miles of driving tests may be required to cover a sufficiently diverse set of traffic scenarios and road conditions [1]. Consequently, simulations, model-based developments, and hardware-in-the-loop techniques need to be relied upon to reduce the need for extensive road testing and to maintain short time-to-market for automated driving technologies.

Traffic simulators can facilitate the initial calibration of automated driving algorithms before actual road tests and reduce the time and effort needed for the overall algorithm testing. As an example, in [18], a virtual environment for testing advanced driver assistance systems is proposed, which is configured using real-world driving data.

Since vehicle operation in traffic is inherently interactive, that is, the actions of car drivers are influenced by and also influence the actions of other traffic participants, a simulator used for testing and discovering faults in automated driving algorithms must reflect such interactions. In [9, 10], we have developed a simulator capable of representing vehicle interactions in traffic based on a hierarchical reasoning theory and level- $k$  games. We have illustrated the use of the simulator to support quantitative analyses and comparisons of various automated driving decision and control systems, and support their initial calibrations. In this paper, we focus on the demonstration of the simulator's functionality for parameter optimization in automated driving algorithms. As a specific case study, a rule-based algorithm for automated highway driving is considered and its parameters are calibrated using the developed traffic simulator.

In the level- $k$  game theory, each player in the game is assumed to have bounded rationality, and his/her level of rationality is represented by a finite reasoning depth  $k$ , see [15]. This assumption is arguably representative of human interactions in real-world situations and is supported by experimental evidence, see also [3–5]. The level- $k$  game theory-based framework is particularly useful to model the interactive behavior of multiple players in a game when the game is complex, which is the case in multi-vehicle traffic scenarios. Similar in spirit implementations have been reported for modeling human pilot-to-unmanned aerial vehicle interactions in [13, 17]. In this paper, the traffic on a highway with  $n$  lanes,  $n \in \mathbb{N}_+$ , composed of up to 30 interactive vehicles, is modeled.

This paper is organized as follows: In Sect. 5.2, we describe our highway traffic modeling approach exploiting the level- $k$  game theory. In Sect. 5.3, we introduce the simulator developed based on the proposed game-theoretic traffic model. In Sect. 5.4, we describe a rule-based automated highway driving algorithm to be tested and calibrated. In Sect. 5.5, we present an illustration of the use of the simulator to calibrate algorithm parameters. Finally, Sect. 5.6 presents a summary and concluding remarks.

## 5.2 Traffic Modeling Based on Level- $k$ Game Theory

The traffic we are modeling in this work is  $n$ -lane highway traffic. Each car in the traffic can cruise with a steady speed, accelerate to a higher speed, decelerate to a lower speed, and change lanes to overtake other cars. Each car is assumed to be controlled by a human driver who obeys the general traffic rules and pursues safe, efficient, and comfortable travel.

### 5.2.1 Modeling of a Single Car

A car (including the driver) is modeled as a hierarchical system composed of two levels of controllers. A higher-level controller, that represents the human driver's decision-making processes, selects a maneuver (that in this paper is called an "action," denoted by  $\gamma$ ) from a finite action set, according to the current state of the car in the traffic, to execute at each time step. Then a lower-level controller controls the engine, powertrain and vehicle dynamics according to the action command from the higher-level controller to let the car realize the prescribed maneuver. The structure of the system is shown in Fig. 5.1. In this work, we focus on the modeling of the higher-level decision making. The lower-level control and the car dynamics are represented by the following discrete-time dynamics,

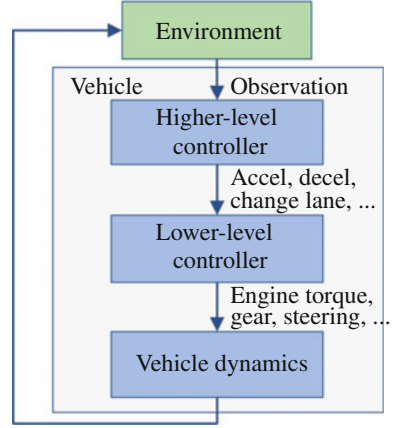
$$\begin{aligned} x(t+1) &= x(t) + v_x(t) \Delta t, \\ v_x(t+1) &= v_x(t) + a(t) \Delta t, \\ y(t+1) &= y(t) + v_y(t) \Delta t, \end{aligned} \tag{5.1}$$

where  $x$  and  $v_x$  represent, respectively, the position and velocity of the car on the highway in the longitudinal direction, while  $y$  represents the position of the car on the highway in the lateral direction. The longitudinal acceleration,  $a(t)$ , and the lateral velocity,  $v_y(t)$ , are two control inputs that are decided by the higher-level controller corresponding to the selected action commands.

In traffic, a driver can only observe and process a limited amount of information, due to the limits of human vision and the human brain. In particular, a driver can observe the motion of the cars in an immediate vicinity of his/her own and decide on his/her next maneuver based on these observations. We assume that the following observations are used by a driver in his/her decision making, as supported by [7, 12]:

- The range and range rate to the car in front and in the same lane,  $d_{fc}$  and  $v_{fc}$ ,
- The range and range rate to the car in front and in the left lane,  $d_{fl}$  and  $v_{fl}$ ,
- The range and range rate to the car in front and in the right lane,  $d_{fr}$  and  $v_{fr}$ ,
- The range and range rate to the car in the rear and in the left lane,  $d_{rl}$  and  $v_{rl}$ ,
- The range and range rate to the car in the rear and in the right lane,  $d_{rr}$  and  $v_{rr}$ ,
- The lane index of the ego car,  $i \in \{1, 2, \dots, n\}$ ,

**Fig. 5.1** Control hierarchy of the car model



where “range” is defined as the longitudinal distance from the ego car to another car. Therefore, the observation state of a car is defined by a vector

$$x_{\text{ob}} = [d_{\text{fl}} \ v_{\text{fl}} \ d_{\text{fc}} \ v_{\text{fc}} \ d_{\text{fr}} \ v_{\text{fr}} \ d_{\text{rl}} \ v_{\text{rl}} \ d_{\text{rr}} \ v_{\text{rr}} \ i]^\top \in \mathbb{R}^{11}. \quad (5.2)$$

It is noted that a human driver may not be able to accurately measure his/her range or range rate to another car. He/she can only estimate and specify them, respectively, as “close,” “far,” and as “approaching,” “moving away,” etc. In this paper, we introduce categorical values to encode the ranges as follows:

$$\text{Spec}(d) = \begin{cases} \text{“close”} & \text{if } d \leq d_c, \\ \text{“nominal”} & \text{if } d_c < d \leq d_f, \\ \text{“far”} & \text{if } d > d_f, \end{cases} \quad (5.3)$$

and introduce categorical values to encode the range rates as follows:

$$\text{Spec}(v) = \begin{cases} \text{“approaching”} & \text{if } v < 0, \\ \text{“stable”} & \text{if } v = 0, \\ \text{“moving away”} & \text{if } v > 0. \end{cases} \quad (5.4)$$

We remark that the observation states and encoding levels must be introduced judiciously. More states and levels may improve the fidelity of the car model but can impede the computations and the implementation due to the “curse of dimensionality.”

The action set, denoted by  $\Gamma$ , covers the major maneuver actions a human driver normally applies in highway traffic:

- “Maintain” current lane and speed:  $a = 0, v_y = 0$ ;
- “Accelerate” at a nominal rate:  $a = a_1, v_y = 0$ ;
- “Decelerate” at a nominal rate:  $a = -a_1, v_y = 0$ ;

- “Hard accelerate” at a large rate:  $a = a_2, v_y = 0$ ;
- “Hard decelerate” at a large rate:  $a = -a_2, v_y = 0$ ;
- Change lane “to the Left”:  $a = 0, v_y = v_{cl}$ ;
- Change lane “to the Right”:  $a = 0, v_y = -v_{cl}$ .

The nominal rate,  $\pm a_1$  ( $\text{m/s}^2$ ), reflects the acceleration/deceleration a human driver would apply in normal situations, while the “hard” rate,  $\pm a_2$  ( $\text{m/s}^2$ ), reflects the acceleration/deceleration a human driver would apply in aggressive driving situations. The latter values depend on the maximum acceleration/deceleration capability of a car.

Additional modeling assumptions are made as follows: The longitudinal velocity of a car is assumed to be bounded to a range  $v_x \in [v_{\min}, v_{\max}]$ , and is saturated to this range during the simulations. The lateral velocity of a car during a lane change is assumed to be constant such that the total time to change lanes is  $t_{cl}$ , i.e.,

$$v_{cl} = \frac{w}{t_{cl}}, \quad (5.5)$$

where  $w$  is the width of a lane. Also, we assume that a car always drives at the center of a lane unless the car is performing a lane change. Once a lane change begins, it always continues to completion.

The model of a driver is a map from the observations to the actions, also called a “policy.” The driver selects an action,  $\gamma^* \in \Gamma$ , to execute based on his/her current observation state, to pursue his/her driving goals at each time step. Basic goals of a driver in highway traffic are (1) to maintain sufficient separation with other vehicles and avoid involvement in accidents, such as car crashes (safety), (2) to minimize the time needed to reach his/her destination (performance), (3) to keep a reasonable headway from preceding cars (safety and comfort), and (4) to minimize driving effort (comfort).

The pursuit of these goals can be reflected in a reward function that is to be maximized. The reward function, in this paper, is designed as

$$R = w_1 \hat{c} + w_2 \hat{v} + w_3 \hat{h} + w_4 \hat{e}, \quad (5.6)$$

where  $w_i, i = 1, 2, 3, 4$ , are the weights for each term and  $\hat{c}, \hat{v}, \hat{h}$ , and  $\hat{e}$  represent “(safety) constraint violation,” “(longitudinal) velocity,” “headway,” and “effort” metrics, respectively. The weights,  $w_i$ , may change depending on the aggressiveness of the driver; however, the safety typically has the highest priority. Hence, the following relationship between the weights should be kept:

$$w_1 \gg w_2, w_3, w_4. \quad (5.7)$$

The terms,  $\hat{c}, \hat{v}, \hat{h}, \hat{e}$ , are explained further below.

**$\hat{c}$  (constraint violation):** We define a safe zone for each car (a rectangular area that over-bounds the geometric contour of the car with a safety margin) whose boundaries are treated as safety constraints. The term  $\hat{c}$  is assigned a value of  $-1$  when a constraint

violation occurs, that is, the safe zone is invaded by another car, and a value of 0 otherwise.

$\hat{v}$  (**velocity**): The term  $\hat{v}$  is assigned the value

$$\hat{v} = \frac{v_x - v_{\text{nominal}}}{a_1}, \quad v_{\text{nominal}} = \frac{v_{\min} + v_{\max}}{2}. \quad (5.8)$$

Dividing by  $a_1$  makes this term of the same order of magnitude as other terms and facilitates the design of the weights.

$\hat{h}$  (**headway**): The term  $\hat{h}$  takes the following values depending on the headway distance (the range to the car directly in front):

$$\hat{h} = \begin{cases} -1 & \text{if } \text{Spec}(d_{fc}) = \text{"close,"} \\ 0 & \text{if } \text{Spec}(d_{fc}) = \text{"nominal,"} \\ 1 & \text{if } \text{Spec}(d_{fc}) = \text{"far."} \end{cases} \quad (5.9)$$

$\hat{e}$  (**effort**): The term  $\hat{e}$  takes the value 0 if the driver's action is "Maintain,"  $\hat{e} = \hat{e}_h$  if the driver's action is "Hard accelerate" or "Hard decelerate," and  $\hat{e} = \hat{e}_n$  otherwise. This term discourages the driver from making unnecessary maneuvers. In particular, if the driver is able to maintain safety with other actions, a higher penalty would discourage the driver from applying "Hard accelerate" or "Hard decelerate," that decrease the comfort. But in the case where another maneuver cannot avoid constraint violation, the driver would apply "Hard accelerate" or "Hard decelerate" to maintain safety. Note that the ratio between  $\hat{e}_n$  and  $\hat{e}_h$  has an influence on the driver's behavior and may be adjusted to match the driving behavior of different human drivers. In this paper, we choose  $\hat{e}_n = -1$  and  $\hat{e}_h = -5$ .

In the above reward function (5.6), we impose a penalty on constraint violations. We also note that there are some combinations of observation states and actions that obviously lead to undesired behaviors, such as constraint violations. We can impose ancillary constraints to avoid the occurrence of such combinations. The imposition of such constraints also benefits the numerical convergence of the reinforcement learning algorithm, that is, applied to compute the optimal control policy. These ancillary constraints are as follows:

- If a car in the left lane is in a parallel position, the ego car cannot select the "to the Left" action,
- If a car in the right lane is in a parallel position, the ego car cannot select the "to the Right" action,
- If  $\text{Spec}(d_{fl}) = \text{"close"}$  and  $\text{Spec}(v_{fl}) = \text{"approaching,"}$  or,  $\text{Spec}(d_{rl}) = \text{"close"}$  and  $\text{Spec}(v_{rl}) = \text{"approaching,"}$  the ego car cannot select the "to the Left" action,
- If  $\text{Spec}(d_{fr}) = \text{"close"}$  and  $\text{Spec}(v_{fr}) = \text{"approaching,"}$  or,  $\text{Spec}(d_{rr}) = \text{"close"}$  and  $\text{Spec}(v_{rr}) = \text{"approaching,"}$  the ego car cannot select the "to the Right" action.

Two cars are assumed to be in a "parallel" position when their safe zones intersect in the longitudinal direction.

We remark that when selecting actions, the ego driver is supposed to not only consider his/her current observation state, but also consider the potential actions of the other cars in his/her vicinity, as their actions influence the evolution of the ego car's states. This reflects the interactive nature of traffic dynamics. In this work, we exploit a hierarchical reasoning and level- $k$  game theory-based approach to model the vehicle-to-vehicle interactions in highway traffic.

### 5.2.2 *Modeling of Interactive Traffic*

In this work, the scheme for modeling interactive highway traffic is premised on the assumptions that (1) when a strategic agent makes its own decisions in a multi-agent interactive scenario (such as a traffic scenario composed of multiple cars), it takes into consideration the predicted decisions of the other agents; (2) an agent predicts the decisions of the other agents through a finite depth of reasoning (called “level”), and different agents may have different reasoning levels. The reader is referred to [4, 15] for more comprehensive discussions on hierarchical reasoning and level- $k$  game theory.

To model the drivers of different reasoning levels, one starts by specifying a “non-strategic” driver model, which is referred to as a “level-0” driver model. A “level-0” driver makes instinctive action decisions to pursue his/her own goals without considering the potential actions or reactions of the other drivers. Then, we use a reinforcement learning algorithm to determine the model of a “level-1” driver; that is, the optimal observation-to-action map (policy) based on the reward function (5.6) and assuming that all of the drivers in the traffic but him/herself are level-0 drivers. Similarly, a level- $k$  driver takes optimal actions assuming that all of the other drivers react as level- $(k - 1)$  drivers. It is suggested by experimental studies in [4] that humans are usually level-0, 1, or 2 reasoners in their interactions. So we generate driver models up to level-2. In general, our traffic simulator can be configured based on certain fractions of drivers of each level. Different reasoning levels may reflect different driving habits and proficiency levels of different human drivers. In principle, the reward functions of the drivers may also not need to be the same.

In this scheme, the underlying dynamics of the traffic composed of multiple cars are modeled as a Markov decision process, whose state is determined by the states  $(x, v_x, y)$  of all cars. In particular, as discussed in the previous subsection, a car only obtains a limited amount of information from its vicinity through its observations, i.e., the whole state of the traffic is only partially observable to each car. Hence, determining a control policy in this setting is a partially observable Markov decision process (POMDP) problem. In this work, as in [10], we employ the Jaakkola reinforcement learning (RL) algorithm [6], that distinguishes itself from conventional approaches by guaranteeing convergence to a local maximum in terms of average rewards when the problem is of POMDP type and states and actions admit a finite number of values, to solve for the level- $k$  driver models.

**Fig. 5.2** Reinforcement learning algorithm to obtain the level- $k$  driver models

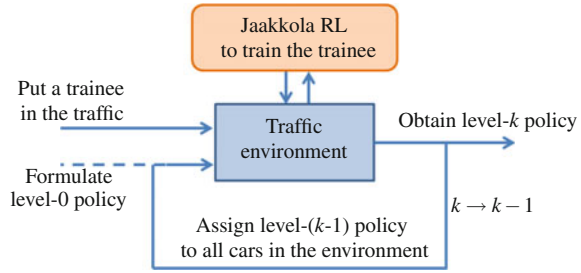


Figure 5.2 shows the procedures to obtain the level- $k$  driver models using the Jaakkola RL algorithm. To obtain the level- $k$  policy, one assigns the level- $(k - 1)$  policy to all of the drivers in the traffic except for the driver being trained (trainee), lets the trainee interact with those level  $(k - 1)$  drivers and exploits the Jaakkola RL algorithm to gradually improve the trainee’s policy. The optimal policy, that assigns a probability distribution over all possible actions to each observation state, is expected to gain the highest average reward if it is executed over an infinitely long simulation. Through Jaakkola RL, the optimal policy is obtained when the average reward converges during the training. For more details on the Jaakkola RL algorithm and its implementation to obtain the level- $k$  policies, see [10].

The development of level- $k$  policies starts by specifying a “level-0” policy. In this paper, we formulate the level-0 policy by prescribing minimally rational behaviors for a range of observation states, as follows:

$$\mathcal{V}_{10}^* = \begin{cases} \text{“Decelerate,”} & \text{if } Spec(d_{fc}) = \text{“nominal”} \ \& \ Spec(v_{fc}) = \text{“approaching,”} \\ & \text{or } Spec(d_{fc}) = \text{“close”} \ \& \ Spec(v_{fc}) = \text{“stable,”} \\ \text{“Hard decelerate,”} & \text{if } Spec(d_{fc}) = \text{“close”} \ \& \ Spec(v_{fc}) = \text{“approaching,”} \\ \text{“Maintain,”} & \text{otherwise.} \end{cases} \quad (5.10)$$

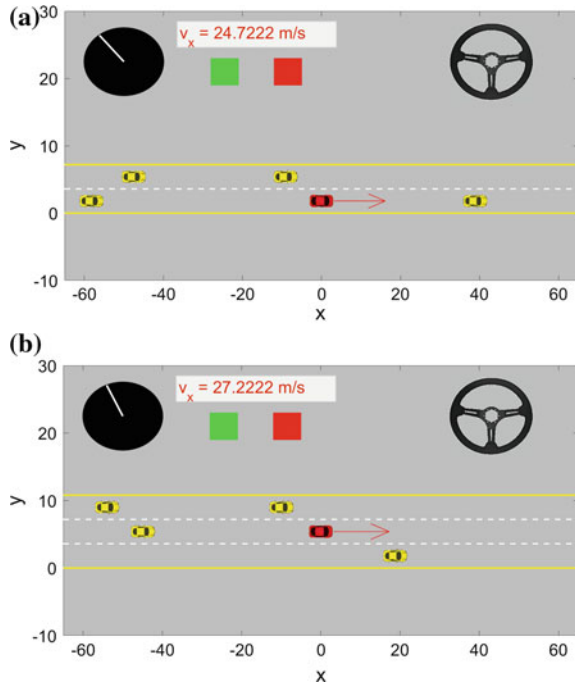
### 5.3 Traffic Simulator Based on Level- $k$ Game Theory

The simulator has been configured with vehicles responding according to level- $k$  driving policies to represent traffic on a 2-lane highway and a 3-lane highway. Note that we have considered case studies for a 3-lane highway also in previous publications [9, 10, 14], but have not reported the results for a 2-lane highway case.

Figure 5.3 shows snapshots of a 2-lane highway traffic simulation and of a 3-lane highway traffic simulation. In the figures, the red car in the center is the ego car—it can be controlled by an automated driving algorithm being tested; the yellow cars constitute the traffic environment and all of them are modeled using the level- $k$



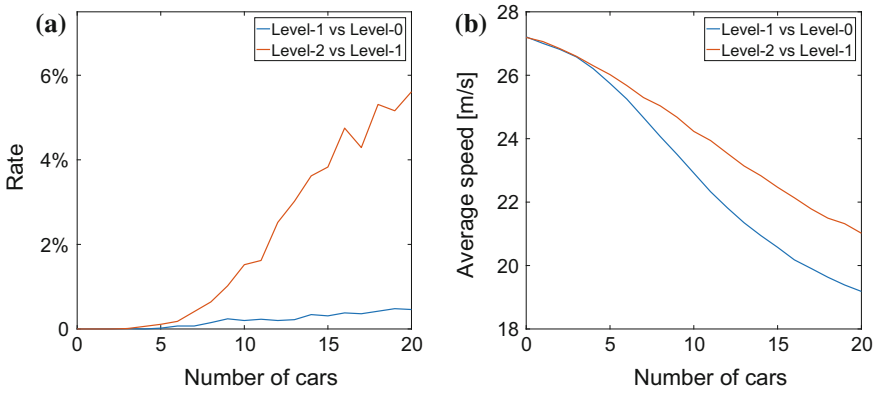
**Fig. 5.3** Traffic simulator. **a** Simulation of a traffic scenario on a 2-lane highway. **b** Simulation of a traffic scenario on a 3-lane highway



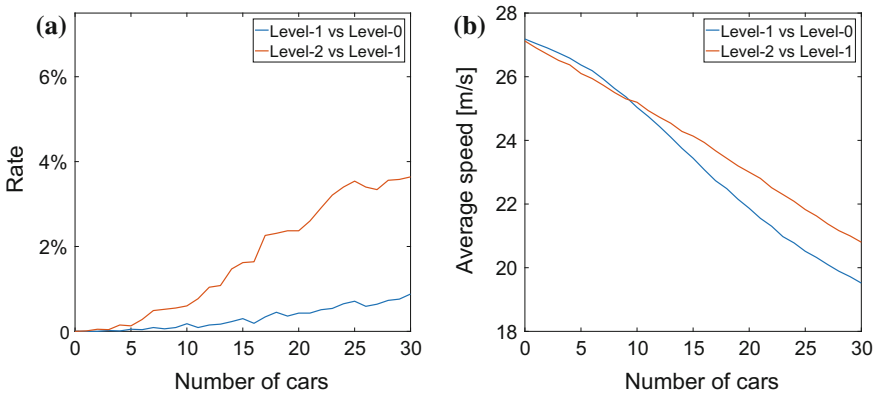
policies—they interact with each other as well as with the red automated car. The red arrow attached to the red car indicates its travel direction and arrow length indicates the car’s travel speed. On the left is a speedometer, and on the right the steering wheel indicates the lateral motion of the red car. The green box and red box in the middle indicate, respectively, the gas pedal and the brake pedal.

We evaluate the level- $k$  game-theoretic driver models by letting the red test car be controlled by a level- $k$  policy and testing it in a level- $(k - 1)$  traffic environment (all of the yellow cars are modeled using level- $(k - 1)$  policies). When training the level- $k$  policies, we use  $w_1 = 10,000$ ,  $w_2 = 5$ ,  $w_3 = 1$ ,  $w_4 = 1$ . We remark that we use a larger weight  $w_2$  for travel speed metric  $\hat{v}$  compared to the weights  $w_3$ ,  $w_4$  for headway distance metric  $\hat{h}$  and for driving effort metric  $\hat{e}$ ; as a result, the obtained policy is of increased aggressiveness—tending to make more frequent maneuvers for overtaking other cars in order to achieve higher travel speeds. This setup is synergistic with the needs for testing automated driving control algorithms, because it provides more complex and challenging scenarios. We use constraint violation rate and average travel speed as metrics to quantitatively evaluate the safety and performance of the level- $k$  policies. Here, “constraint violation” refers to the test car’s safe zone being entered by any of the cars in the simulations. To obtain these rates, 10,000 simulations are run for each number of cars (that reflects the traffic density). Each simulation is 200[s] long and the rates are provided as the percentage of simulation runs during which at least one “constraint violation” occurs.

We remark that for the results presented in Figs. 5.4 and 5.5, the level-2 policy is evaluated in a traffic environment composed of level-1 cars, and the level-1 policy is evaluated in a traffic environment composed of level-0 cars. It is observed that the level-2 policy exhibits higher constraint violation rates than the level-1 policy. One explanation for this is that the dynamics of the level-1 traffic where the level-2 policy is tested are more aggressive and harder to predict because of the complicated interactions simultaneously happening among the level-1 cars and between the level-1 and the level-2 cars; while, the dynamics of the level-0 traffic where the level-1 policy is tested are relatively easy to predict due to the non-strategic behavior of the level-0 cars. Also, it is observed that the level-2 car has higher average travel speed compared to the level-1 car. One explanation for this is that the flow speed of the level-1 traffic is, on average, higher than that of the level-0 traffic, because the level-



**Fig. 5.4** Simulation results for level- $k$  cars driving on a 2-lane highway: **a** Constraint violation rate. **b** Average travel speed



**Fig. 5.5** Simulation results for level- $k$  cars driving on a 3-lane highway: **a** Constraint violation rate. **b** Average travel speed

1 cars can change lanes to overtake slower cars while the level-0 cars cannot. As a consequence, the level-2 car in the level-1 traffic flow also has more possibilities to travel faster, compared to the level-1 car driving in the level-0 traffic flow, that may block the level-1 car from traveling faster. We note that these observations hold for both a 2-lane highway and a 3-lane highway traffic.

We then consider a traffic model composed of a mix of cars operating with different level- $k$  policies. Specifically, in accordance with the results of an experimental study conducted in [4], we consider a traffic model where 10% of the drivers make decisions based on level-0 policies, 60% of the drivers act based on level-1 policies and 30% use level-2 policies. This traffic model is used hereafter to evaluate automated driving control algorithms and support their optimal parameter calibration.

We remark that the level- $k$  cars in the traffic model utilize the 11 observations (5.2) to determine their actions from the discrete action set  $\Gamma$ , as described in Sect. 5.2.1; the test car can use a different observation set (referred to as the “sensing system”) and a different control set that are decided by the algorithm developer to achieve a good balance between performance and complexity. In the rule-based automated highway driving algorithm developed in the next section, we utilize the same sensing system (5.2).

## 5.4 Rule-Based Automated Highway Driving Algorithm

To illustrate the use of the simulator for optimal parameter calibration of automated driving algorithms, we consider a specific rule-based highway driving algorithm, that includes both the longitudinal motion control and the lateral lane change decision making. We optimize its parameters based on an objective function that reflects both safety and performance requirements.

The proposed automated highway driving algorithm is a finite state machine (FSM) that has three states (also called “modes”), see Fig. 5.6. In each mode, the vehicle’s longitudinal motion and lateral motion are controlled based on specific laws; mode switches are triggered when certain conditions get satisfied.

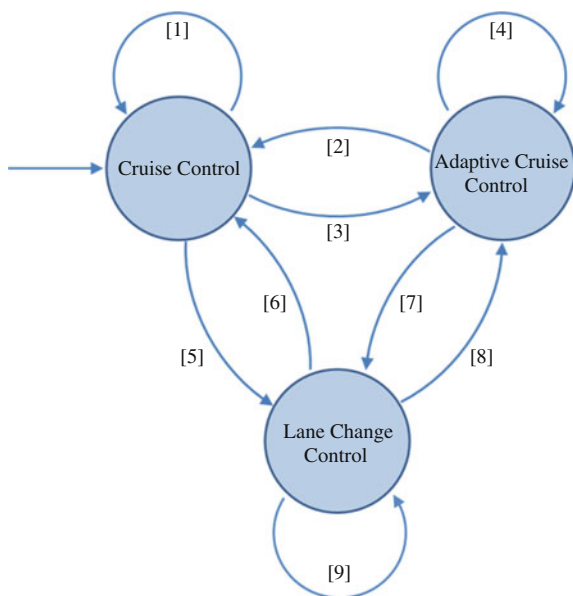
Now we introduce the control laws for each mode and the switch conditions between each pair of modes. In the sequel, the mode of the car at time  $t$  is denoted by  $M(t)$ ; the  $p_i$ ,  $i = 1, \dots, 6$ , are boolean variables and each of them denotes the truth value of a specific condition.

### 5.4.1 Cruise Control Mode - C

When in the cruise control mode (C), the car maintains a reference speed  $v_{\text{ref}}$  by adjusting the acceleration according to

$$a(t) = K_c(v_{\text{ref}} - v_x(t)), \quad (5.11)$$

**Fig. 5.6** Finite state machine diagram for automated highway driving. The labels [1–9] designate switch conditions between different modes



where  $K_c$  is the gain to match the car's actual speed  $v_x(t)$  to a reference speed  $v_{\text{ref}}$ .

In the cruise control mode  $M(t) = C$ , allowable mode switches are:

$$M(t+1) = \begin{cases} A & \text{if } p_1, \\ C & \text{if } \neg p_1, \end{cases} \quad (5.12)$$

where  $p_1$  denotes the truth value of the condition

$$p_1 : d_{\text{fc}} \leq d_{\text{acc}},$$

where  $d_{\text{acc}}$  represents a critical headway distance. Under the above condition, the car switches from the cruise control mode to the adaptive cruise control mode (A) to perform car following when there is a car directly in front within this critical distance.

### 5.4.2 Adaptive Cruise Control Mode - A

When in the adaptive cruise control mode (A), the car follows a leader ( $l$ ) while keeping a desired distance  $d_{\text{des}}$  to the leader by adjusting the acceleration according to

$$a(t) = K_p(x^l(t) - x(t) - d_{\text{des}}) + K_v(v_x^l(t) - v_x(t)), \quad (5.13)$$

where  $K_p$  is the gain to match the actual car-following distance  $x^l(t) - x(t)$  to a desired distance  $d_{des}$ , and  $K_v$  is the gain to match the ego car's speed  $v_x(t)$  to the speed of the leader  $v_x^l(t)$ . The acceleration policy (5.13) can alternatively be defined in terms of time headway.

The acceleration/deceleration of the car is assumed to be bounded by  $a(t) \in [-a_2, a_2]$  (if the computed  $a(t)$  is outside the bounds, it is saturated to the bounds).

In the adaptive cruise control mode  $M(t) = A$ , allowable mode switches are:

$$M(t+1) = \begin{cases} C & \text{if } p_2, \\ L & \text{if } (\neg p_2)p_3p_4p_5, \\ A & \text{otherwise,} \end{cases} \quad (5.14)$$

where

$$p_2: d_{fc} \geq d_{cc},$$

$p_3$ : the predicted acceleration that can be obtained in the target lane  
is larger than the acceleration obtained in the current lane,

$$p_4: d_{fc} \geq d_{win},$$

$$p_5: (d_{fl} \geq d_{win} \text{ and } d_{rl} \geq d_{win}) \text{ or } (d_{fr} \geq d_{win} \text{ and } d_{rr} \geq d_{win}),$$

where  $d_{cc}$  represents a critical headway distance such that when the car directly in front moves away and gets outside this critical distance, the car switches to cruise control mode to maintain the reference speed  $v_{ref}$ , and  $d_{win}$  represents a window of safe distance to complete a lane change.

The motivation for lane changes is for overtaking slower cars and pursuing higher travel speeds. When a predicted acceleration that can be achieved if the ego car travels in another lane is larger than the acceleration that is obtained by the ego car traveling in the current lane ( $p_3$ ), the ego car may switch to the lane change mode ( $L$ ) to perform a lane change. If only one of the left/right lanes satisfies the lane change condition  $(\neg p_2)p_3p_4p_5$ , that lane is set to be the target lane for lane change. If both satisfy the condition, the lane that leads to a larger acceleration is set to be the target lane.

### 5.4.3 Lane Change Mode - L

When in the lane change mode ( $L$ ), the car changes lanes from the original lane to a target lane. In the lane change mode, the car's longitudinal speed remains constant and its lateral motion is determined by

$$v_y(t) = \pm \frac{w}{t_{cl}}, \quad (5.15)$$

where  $w$  is the width of a lane,  $t_{cl}$  is the time needed to make a lane change, and the plus/minus sign indicates the lane change direction. We assume that once a lane change begins, it always continues to completion.

In the lane change mode  $M(t) = L$ , allowable mode switches are:

$$M(t+1) = \begin{cases} A & \text{if } p_1 p_6, \\ C & \text{if } (\neg p_1) p_6, \\ L & \text{if } \neg p_6, \end{cases} \quad (5.16)$$

where

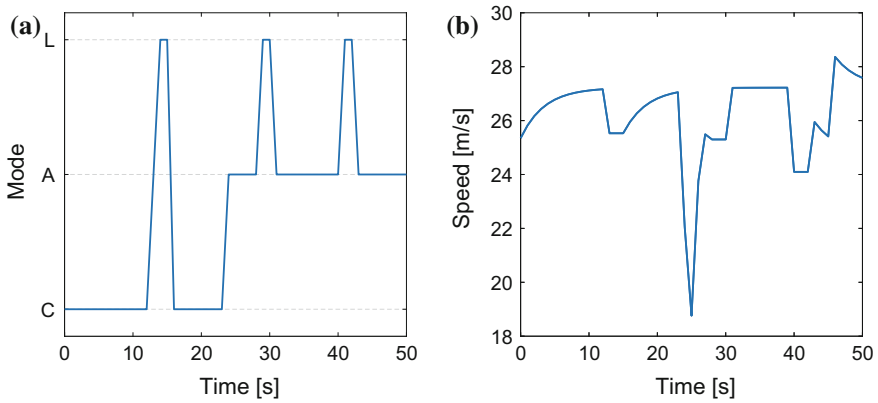
$p_6$ : the car reaches the center of the target lane within a deviation tolerance.

#### 5.4.4 Basic Simulation of Rule-Based Automated Highway Driving Controller

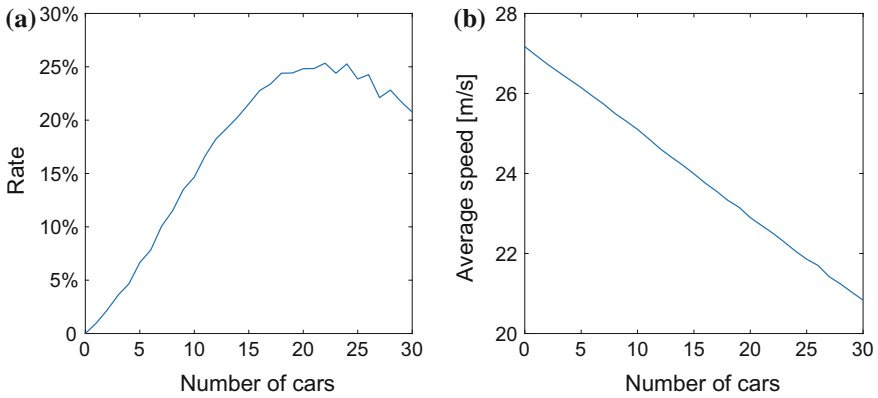
The operation of this rule-based automated highway driving controller in a simulation that is 50 [s] long is presented in Fig. 5.7. The traffic environment is modeled using the developed interactive traffic model composed of a mix of level- $k$  cars, that has been discussed at the end of Sect. 5.3, on a 3-lane highway. Figure 5.7a shows the time history of the mode  $M(t)$ . Figure 5.7b shows the longitudinal speed of the car during the simulation. The parameter values used in the simulation are given below:

$$\begin{aligned} K_c &= 0.25 \text{ [s}^{-1}\text{]}, \quad K_p = 0.25 \text{ [s}^{-2}\text{]}, \quad K_v = 1 \text{ [s}^{-1}\text{]}, \\ v_{\text{ref}} &= v_{\text{max}} = 98 \text{ [km/h]}, \quad v_{\text{min}} = 62 \text{ [km/h]}, \quad t_{cl} = 2 \text{ [s]}, \\ d_{\text{acc}} &= 37 \text{ [m]}, \quad d_{cc} = 47 \text{ [m]}, \quad d_{\text{des}} = 31.5 \text{ [m]}, \quad d_{\text{win}} = 21 \text{ [m]}. \end{aligned} \quad (5.17)$$

Figure 5.8 quantitatively evaluates the performance of this rule-based controller in the modeled 3-lane highway traffic. The metrics are the same as those used to evaluate the level- $k$  policies in Figs. 5.4 and 5.5. It is observed that the test car exhibits a significant number of constraint violations. On one hand, this may result from some deficiencies in the designed control strategy. For example, lack of some necessary observations, such as front cars' turning signals, may limit the test car's capability of handling uncertainties emanating from uncertain behaviors of other drivers. We remark that the level- $k$  policies may not have such a problem because predictions of how neighbor cars behave have been taken into account implicitly during the RL training. On the other hand, the parameters of our rule-based controller may not be calibrated to the best values. The values of these parameters can be optimized with the support of the simulator as discussed next.



**Fig. 5.7** The operation of the rule-based automated highway driving controller: **a** Mode history. **b** Travel speed history



**Fig. 5.8** Simulation results for the rule-based automated driving controller on a 3-lane highway: **a** Constraint violation rate. **b** Average travel speed

## 5.5 Optimal Automated Driving Controller Calibration

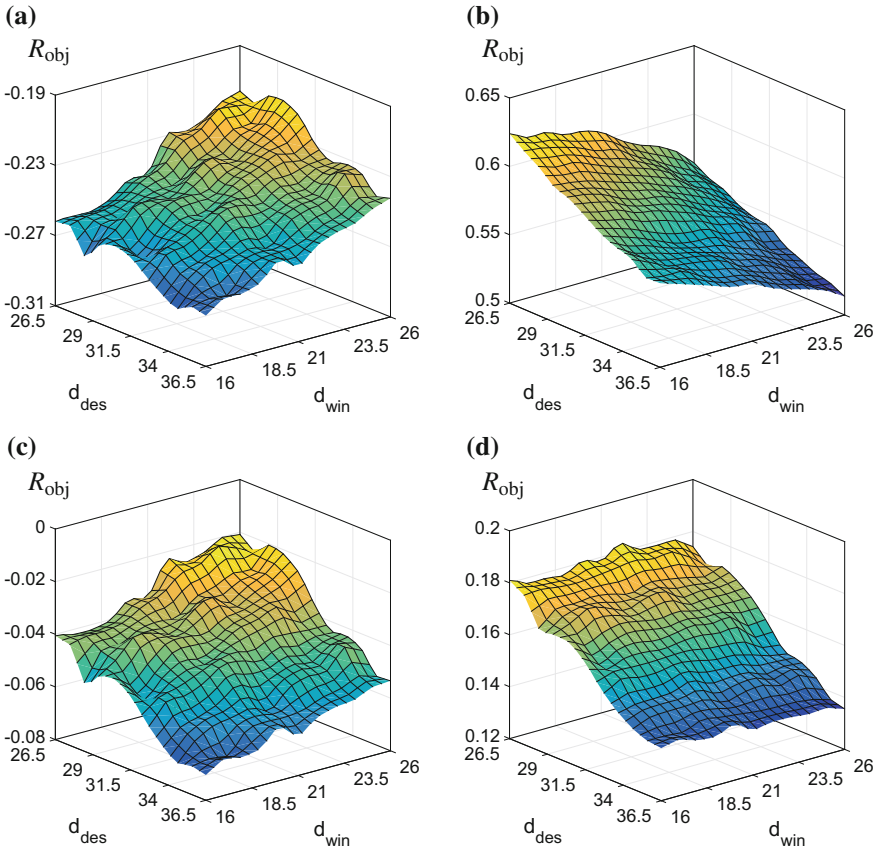
In this section, we calibrate the parameters of the rule-based automated highway driving controller developed in the previous section to improve performance, with the support of the simulator constructed based on the level- $k$  game-theoretic traffic model. We remark that the approach discussed in this section can be used to calibrate other automated driving algorithms as well. Another example can be found in [10].

The performance of the controller can be represented by the value of a predefined objective function that accounts for safety, performance, comfort, fuel economy, etc. As an example, we define the objective function to be maximized as follows,

$$R_{\text{obj}} = k_1(-\bar{c}) + k_2 \frac{\bar{v}_x - v_{\min}}{v_{\max} - v_{\min}}, \quad (5.18)$$

where the weights  $k_1$  and  $k_2$  are determined by the user,  $\bar{c}$  is the constraint violation rate defined as in Figs. 5.4 and 5.5,  $\bar{v}_x$  is the average travel speed during the simulations, and  $v_{\min}$  and  $v_{\max}$  represent, respectively, a speed lower bound and a speed upper bound for highway driving. We remark that (5.18) is designed such that each of its terms is dimensionless and normalized.

The parameters to be optimized in this example are: (1) the desired car-following distance  $d_{\text{des}}$  in (5.13), and (2) the safe distance window to make a lane change  $d_{\text{win}}$  in (5.14). The traffic scenario is modeled as 20 cars driving on a 3-lane highway. The value of (5.18) as a function of  $(d_{\text{des}}, d_{\text{win}})$  is shown in Fig. 5.9. The numbers in the plots are obtained as the average objective function values over 1000 simulation runs for each pair of  $(d_{\text{des}}, d_{\text{win}})$  values on a grid.



**Fig. 5.9** Objective function values versus parameter values corresponding to different weight choices: **a**  $k_1 = 1, k_2 = 0$ . **b**  $k_1 = 0, k_2 = 1$ . **c**  $k_1 = 0.75, k_2 = 0.25$ . **d**  $k_1 = 0.5, k_2 = 0.5$



Figure 5.9 can be used to select the best pair of  $(d_{\text{des}}, d_{\text{win}})$  for a specific objective function design. For example, for maximum safety  $k_1 = 1$  and  $k_2 = 0$ , it can be observed that the larger  $d_{\text{win}}$  is, the higher value one obtains, i.e., the safety is improved with a larger separation between vehicles. When both safety and speed are considered, for example,  $k_1 = 0.75$ ,  $k_2 = 0.25$ , the best pair is  $(d_{\text{des}}, d_{\text{win}}) = (26.5, 25)$  (m). The other parameters of the algorithm, for example,  $K_c$ ,  $K_p$ ,  $K_v$ ,  $v_{\text{ref}}$ ,  $d_{\text{des}}$ ,  $d_{\text{acc}}$ ,  $d_{\text{cc}}$ ,  $d_{\text{win}}$  and  $t_{\text{cl}}$ , can be optimized similarly.

## 5.6 Summary and Concluding Remarks

Simulators that account for interactions of vehicles in traffic can facilitate testing, validation, verification, and calibration of automated driving algorithms in the virtual world. They can also be used to uncover situations and scenarios that are particularly challenging for automated driving or are likely to result in faults in particular automated driving policies, thereby informing their future testing in the simulated or on-the-road conditions. Consequently, the extent of the required time and effort consuming on the road testing may potentially be reduced, thereby addressing one of the major current challenges in the automated vehicle development.

In this paper, we have described an approach to modeling vehicle interactions in traffic based on an application of the level- $k$  game theory. Case studies of configuring the simulator to represent traffic on 2-lane and 3-lane highways and to evaluate and improve parameters in a rule-based automated highway driving policy have been presented. Some qualitative trends observed in the modeled traffic for different number of lanes and for different levels of vehicles involved have been presented. Future work will focus on extending our approach to represent city traffic.

**Acknowledgements** Nan Li and Ilya Kolmanovsky acknowledge the support of this research by the National Science Foundation under Award CNS 1544844 to the University of Michigan. Yildiray Yildiz acknowledges the support of this research by the Scientific and Technological Research Council of Turkey under Grant 114E282 to Bilkent University.

## References

1. Anderson, J.M., Nidhi, K., Stanley, K.D., Sorensen, P., Samaras, C., Oluwatola, O.A.: Autonomous Vehicle Technology: A Guide for Policymakers. Rand Corporation (2014)
2. Campbell, M., Egerstedt, M., How, J.P., Murray, R.M.: Autonomous driving in urban environments: approaches, lessons and challenges. *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* **368**(1928), 4649–4672 (2010)
3. Costa-Gomes, M.A., Crawford, V.P.: Cognition and behavior in two-person guessing games: an experimental study. *Am. Econ. Rev.* **96**(5), 1737–1768 (2006)
4. Costa-Gomes, M.A., Crawford, V.P., Iriberri, N.: Comparing models of strategic thinking in van huyck, battalio, and beil's coordination games. *J. Eur. Econ. Assoc.* **7**(2–3), 365–376 (2009)

5. Hedden, T., Zhang, J.: What do you think I think you think? Strategic reasoning in matrix games. *Cognition* **85**(1), 1–36 (2002)
6. Jaakkola, T., Singh, S.P., Jordan, M.I.: Reinforcement learning algorithm for partially observable markov decision problems. In: *Advances in Neural Information Processing Systems*, pp. 345–352. Citeseer (1995)
7. Kikuchi, S., Chakroborty, P.: Car-following model based on fuzzy inference system. *Transp. Res. Rec.* 82–82 (1992)
8. Langari, R.: Autonomous vehicles. In: 2017 American Control Conference (ACC), pp. 4018–4022 (2017). <https://doi.org/10.23919/ACC.2017.7963571>
9. Li, N., Oyler, D., Zhang, M., Yildiz, Y., Girard, A., Kolmanovsky, I.: Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems. In: *IEEE 55th Conference on Decision and Control (CDC)*, pp. 727–733. IEEE (2016)
10. Li, N., Oyler, D.W., Zhang, M., Yildiz, Y., Kolmanovsky, I., Girard, A.R.: Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Trans. Control Syst. Technol.* **99**, 1–16 (2017). <https://doi.org/10.1109/TCST.2017.2723574>
11. Maurer, M., Gerdes, J.C., Lenz, B., Winner, H.: *Autonomous driving: Technical, Legal and Social Aspects*. Springer Publishing Company, Incorporated (2016)
12. McDonald, M., Wu, J., Brackstone, M.: Development of a fuzzy logic based microscopic motorway simulation model. In: *IEEE Conference on Intelligent Transportation System*, pp. 82–87. IEEE (1997)
13. Musavi, N., Onural, D., Gunes, K., Yildiz, Y.: Unmanned aircraft systems airspace integration: a game theoretical framework for concept evaluations. *J. Guidance Control Dyn.* **40**(1), 96–109 (2016)
14. Oyler, D.W., Yildiz, Y., Girard, A.R., Li, N.I., Kolmanovsky, I.V.: A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development. In: 2016 American Control Conference (ACC), pp. 1705–1710 (2016)
15. Stahl, D.O., Wilson, P.W.: On players models of other players: theory and experimental evidence. *Games Econ. Behav.* **10**(1), 218–254 (1995)
16. Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., et al.: Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Rob.* **25**(8), 425–466 (2008)
17. Yildiz, Y., Agogino, A., Brat, G.: Predicting pilot behavior in medium-scale scenarios using game theory and reinforcement learning. *J. Guidance Control Dyn.* **37**(4), 1335–1343 (2014)
18. Zhou, J., Schmied, R., Sandalek, A., Kokal, H., del Re, L.: A framework for virtual testing of ADAS. *SAE Int. J. Passeng. Cars Electron. Electr. Syst.* **9**(2016-01-0049), 66–73 (2016)