Transportation Research Part C 88 (2018) 140–158

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A human-like game theory-based controller for automatic lane changing

Hongtao Yu[a,*], H. Eric Tseng[b], Reza Langari[a]

[a] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77840, USA
[b] Ford Research Laboratories, Dearborn, MI 48124, USA

ARTICLE INFO

Keywords:
Autonomous vehicle
Game theory
Lane changing

ABSTRACT

Lane changing is a critical task for autonomous driving, especially in heavy traffic. Numerous automatic lane-changing algorithms have been proposed. However, surrounding vehicles are usually treated as moving obstacles without considering the interaction between vehicles/drivers. This paper presents a game theory-based lane-changing model, which mimics human behavior by interacting with surrounding drivers using the turn signal and lateral moves. The aggressiveness of the surrounding vehicles/drivers is estimated based on their reactions. With this model, the controller is capable of extracting information and learning from the interaction in real time. As such, the optimal timing and acceleration for changing lanes with respect to a variety of aggressiveness in target lane vehicle behavior are found accordingly. The game theory-based controller was tested in Simulink and dSPACE. Scenarios were designed so that a vehicle controlled by a game theory-based controller could interact with vehicles controlled by both robot and human drivers. Test results show that the game theory-based controller is capable of changing lanes in a human-like manner and outperforms fixed rule-based controllers.

1. Introduction

Autonomous vehicles have attracted increasing interest in recent years. Both technology companies and traditional automotive manufacturers are engaged in this transformation. Distinguished members of IEEE predicted up to 75% of vehicles would be autonomous by 2040 (IEEE News Releases, 2012). Fundamental tasks of autonomous driving include car following, lane keeping and lane changing (Khodayari et al., 2010). Car following and lane keeping have been extensively studied. Cruise control, adaptive cruise control (ACC), lane keeping assist and lane centering have been developed (Ozguner et al., 2007). The focus of this paper is lane changing, an essential task for navigating a vehicle in heavy traffic.

Complicated tasks such as lane changing are usually realized through planning algorithms (Zhang et al., 2013). The algorithms can be divided into four hierarchical classes: route planning, path planning, trajectory planning and manoeuver planning. It should be noted that these approaches are often combined to make a complete plan instead of being treated independently (Varaiya, 1993). Route planning is to find a globally optimal path based on traffic situations, which is out of the scope of this paper. Path planning is to determine a collision-free geometric path for the vehicle to follow (Likhachev et al., 2003; Paden et al., 2016; Urmson et al., 2008). Trajectory planning is concerned with real-time transition of vehicle states. It further optimizes the chosen geometric path to assure a smooth and feasible journey, while considering the constraints of vehicle dynamics and obstacles (Borrelli et al., 2005; Kim and Kumar, 2014; Nilsson and Sjöberg, 2013; Wang and Qi, 2001). Maneuver planning deals with a high-level characterization of vehicle

* Corresponding author.
E-mail address: hzy5046@tamu.edu (H. Yu).

https://doi.org/10.1016/j.trc.2018.01.016
Received 7 March 2017; Received in revised form 7 October 2017; Accepted 15 January 2018
Available online 04 February 2018

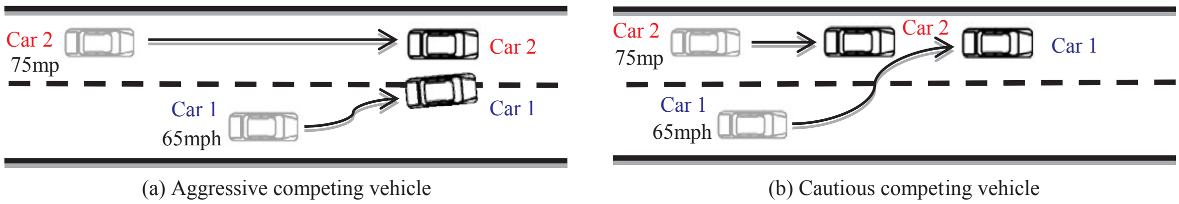

Fig. 1. Change lanes.

motion, such as 'changing lanes' and 'going straight'. Obstacle prediction and risk assessment are usually considered part of the maneuver planning process (Katrakazas et al., 2015).

Though considerable efforts have been made for path planning and trajectory planning, maneuver planning, especially obstacle prediction in the dynamic environment, remains largely unsolved. A typical lane-changing scenario is shown in Fig. 1. Car 1 is the host vehicle. It wants to move to the left lane in which Car 2 is occupying. Car 1 needs to predict the trajectory of Car 2 in order to change lanes safely. It is argued that the prediction could be improved by implementing Vehicle-to-vehicle (V2V) technology and vehicle-to-infrastructure (V2I) technology. In the V2V and V2I applications, it is assumed that vehicles are connected by communication technologies. In such a way, these vehicles can cooperate with each other to increase efficiency and safety (Kato et al., 2002). However, for years to come, it can be expected that there will be a mix of autonomous vehicles and traditional human-driven cars that do not necessarily support V2V or V2I. A self-driving car also needs to interact with human drivers, which are much less predictable than intelligent vehicles. This paper focuses on the mixed automatic/manual traffic scenario. The host vehicle should be able both to detect and predict the actions of surrounding human-driven vehicles on its own and to cooperate with surrounding connected vehicles.

In the literature, a variety of maneuver planning algorithms have been developed to use on-board sensors for obstacle prediction. These models can be classified into physics-based, maneuver-based and interaction-aware models (Lefèvre et al., 2014). The first category estimates the motion of obstacle based on the laws of physics. For instance, it is commonly assumed that surrounding vehicles have constant velocities or acceleration regardless of the motion of the host vehicle. However, this assumption is not always true in the real world, which could cause some trouble. An example is shown in Fig. 1(a). Car 1 wants to change lanes while Car 2 is following far behind in the target lane. Car 1 thinks that if Car 2 has a constant velocity, it is safe to change lanes since the distance between two cars is long enough. However, if Car 2 is much faster and aggressive, it may not want to slow down and allow Car 1 into its lane. Actually, an aggressive and near hostile Car 2 may accelerate and prevent Car 1 from cutting-in upon finding the lane-change intention of Car 1. In this case, it would pose a serious risk for both vehicles.

Maneuver-based models rely on early recognition of the intentions of other traffic participants. For example Liu and Tomizuka (2016) adopted recursive regression to predict the trajectory of the competing vehicle. Alin et al. (2012) applied a grid-based Bayesian filter to inferring the trajectories of other vehicles. Nevertheless, it is still assumed the maneuvers of a vehicle are executed independently from other vehicles in these models, which could also lead to undesired results. Fig. 1(b) illustrates another scenario. Car 2 is close to Car 1 in the beginning. Car 1 thinks Car 2 will continue to move at a constant velocity based on Car 2's previous maneuvers. Therefore, Car 1 should stay in the current lane. Otherwise, it may crash into Car 2. However, Car 2 may be "cautious" or friendly and cooperative. When it finds that Car 1 starts to change lanes, it may slow down and let Car 2 change lanes successfully and safely. In summary, a lane change decision may not produce a comfortable and safe lane change maneuver if the interaction between vehicles is not considered. It is advisable for an autonomous vehicle to treat surrounding traffic as interactive emotional human-driven vehicles rather than just moving obstacles.

For the sake of capturing the dependence between vehicles, efforts have been made to study the interaction-aware models. Coupled HMM (CHMMs) were employed to model pairwise dependencies between multiple moving entities (Brand et al., 1997; Oliver and Pentland, 2000). However, though the model is able to tell when to change lanes, how to change lanes is not discussed. Another approach that has been widely used to study the interaction between vehicles is game theory. Yoo and Langari (2012, 2013) proposed an approach to modelling the interactions between vehicles during lane changing and lane merging based on a Stackelberg game. Wang et al. (2015) presented a model for car-following and lane-changing control based on a differential game. However, motion prediction of the competing vehicle was based on the assumption that the host vehicle knew the cost function of the competing vehicle beforehand in these methods. Sadigh et al. (2016) modelled the interaction between drivers by approximating the human as an optimal planner, with a reward function obtained from inverse reinforcement learning. Kita (1999) and Liu et al. (2007) modelled merging scenarios using a non-cooperative game. However, the reward functions of competing vehicles were learned offline from data sets in these approaches. These functions, whose parameters were fixed after learning, were used for online prediction of the actions of all types of drivers. The prediction might not be reliable since the differences between human drivers (e.g., aggressive or cautious) were not considered. In other words, the models were not adaptive. Bahram et al. (2016), Lawitzky et al. (2013) proposed a prediction and planning loop, which could find the most likely maneuver sequence of the host vehicle and the competing vehicle over multiple time steps. However, the intention-based maneuver probability of the competing vehicle was also learned offline and the framework assumed the host vehicle knew this probability beforehand.

This paper proposes a lane-changing controller based on a game of incomplete information. Though the system does not know the types of other drivers at first, it tries to interact with those drivers (e.g., a small lateral move) and extract information from the interaction. The information contributes to a better understanding of the situation, which helps the controller find the optimal strategy.

The contributions of this paper are:

(1) **Modelled lane-changing behavior based on game theory, which considered the interaction between vehicles.** The proposed game theory-based (GT-based) controller is responsible for lane change decision. Whether to change lanes and how to change lanes (i.e. how to change speed during the lane change) are jointly analyzed. One advantage of the game is that mutual influence of traffic participants is considered. Motion prediction of the competing vehicle is not only based on current traffic conditions, but also based on potential future motion of the host vehicle. Put differently, motion prediction of the competing vehicle and motion planning of the host vehicle are not treated independently, which increases the reliability of decision making. Most interactions between vehicles are modelled by a two-player game. In a special scenario where multiple vehicles intend to move into the same gap, a multi-player game instead of two-player game is also modelled and discussed.
(2) **Differentiated drivers by introducing a parameter called aggressiveness to payoff functions. This parameter could be estimated and updated while changing lanes, which made payoff functions adaptive.** In the literature, payoff functions of competing vehicles are often learned offline from a large data set. One set of parameter values is adopted and used to predict the decisions of all kinds of drivers. In contrast, the aggressiveness of payoff functions could be learned and updated during the lane change in this paper. It provides an opportunity for online adjustment of the structure of payoff functions.
(3) **The model was developed based on the assumption that the aggressiveness of competing vehicles was not known by the host vehicle. It means the game only has incomplete information.** Since the aggressiveness of other cars is unknown, the difficulty of predicting their motion increases. However, it makes the model more realistic.
(4) **A framework was created to estimate aggressiveness during the lane change.** In the beginning, the host vehicle does not know the type of the competing vehicle. It is assumed the competing vehicle is driven by a normal driver and payoff functions of normal drivers are used. The estimated aggressiveness keeps being updated by comparing the predicted motion and actual motion of the competing vehicle.
(5) **A gap selection model was developed and combined with the GT-based model to further mimic the logic of human drivers.** A game could not cover all the logic of a human driver since it mainly studies the interaction. For example, a human driver tends to evaluate multiple gaps in the target lane at the same time. However, it is not guaranteed that all the vehicles that form these gaps are interacting with the host vehicle when the host vehicle selects gaps. In order to mimic this logic, a gap selection model was developed.
(6) **Validated the model in a driving simulator, in which the proposed controller interacted with both robot and human drivers.** The proposed model was tested and compared with fixed rule-based controllers and MPC. Test results show that it outperforms these controllers in certain scenarios.

The rest of this paper is organized as follows. The formulation of the game is demonstrated in Section 2. Test results are presented in Section 3. Section 4 summarizes the control system and outlines a few topics for future research.

2. Control system

Basic motions of a vehicle can be categorized to: car following, lane keeping and lane changing (Khodayari et al., 2010). Car following is realized by an ACC system. The model contains an upper-level controller and a lower-level controller. The lower-level controller is responsible for cruise control. It implements PID control to adjust tire forces and let the vehicle move at the desired velocity. The upper-level controller is in charge of calculating the current desired velocity. During car following, this controller estimates a safe following distance and uses it to calculate the desired velocity. When there is no car ahead, this controller let the vehicle move at the desired speed. Lane keeping is also accomplished by PID controllers, while assuming that cameras are able to tell the accurate lane position of the vehicle by detecting lane markers. The third task is lane changing, which is accomplished by a two-layer planner. The higher layer is rule-based. It generates the initial lane change intention when the vehicle ahead is too slow. The lower layer further analyzes the feasibility of lane changing, especially when there are multiple cars nearby. A GT-based controller was developed to fulfil this task. It uses game theory to account for possible interaction between vehicles and tells when and how to change lanes.

2.1. Game formulation

Game theory is a powerful tool to study the interaction between decision makers. A game is a well-defined mathematical object. It consists of the following elements: players, strategies and payoffs (Nash, 1951). Various types of games, including Cournot Game, Cake cutting and signaling game, have been proposed (Morris, 2012). This paper emphasizes on the application of the Stackelberg game (Von Stackelberg, 2010) to traffic modelling.

The whole process mimics the lane-changing decision of a human driver and its corresponding longitudinal speed modulation strategy, as shown in Fig. 2. A driver begins to consider changing lanes when the car ahead is moving too slowly. Before changing lanes, the driver looks at the rear-view mirror and side-view mirrors to check the surrounding traffic. If the traffic flow in the adjacent lane is faster and the space is large enough, the driver in the host vehicle starts to interact with the surrounding vehicle by using the turn signal or making a small lateral move. Then the driver of the host vehicle looks at mirrors and observes the reactions of the competing vehicle (i.e. the target vehicle that could compete for the same space needed for a lane change). The driver is able to see if the competing vehicle in response is accelerating or decelerating/yielding and by how much. We argue that a human driver in the

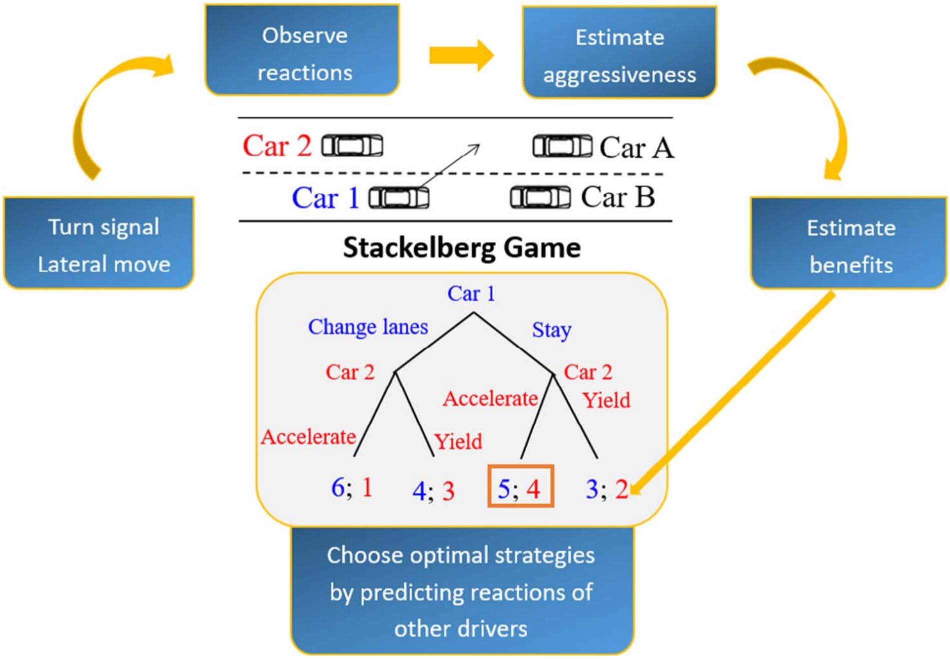

Fig. 2. Game theory.

host vehicle will make the lane change decision according to the aggressiveness of the competing vehicle. Therefore we design automated lane change decision making process based on aggressiveness estimation and the corresponding payoffs to both the host and competing vehicles.

For illustration purpose, example payoffs are given with specific number for all four interaction scenarios between the host/subject vehicle (Car 1) and competing vehicle (Car 2) in Fig. 2. Car 1 chooses its optimal strategy by predicting the future reactions of Car 2. This process is like playing chess, which could be modelled by a Stackelberg Game. There are two players, Car 1 and Car 2. Car 1 is the leader. He can either choose changing lanes or staying in the current lane. Car 2 is the follower. He can react by accelerating or yielding to Car 1. Blue numbers on the left are payoffs of Car 1. Red numbers on the right are payoffs of Car 2. For instance, if Car 1 changes lanes and Car 2 reacts by accelerating, Car 1' payoff is six and Car 2's payoff is one. Both players are trying to maximize their payoffs. According to the example payoffs in Fig. 2, Car 2 will choose yielding to Car 1 when Car 1 changes lanes because of its higher payoff. Similarly, when Car 1 chooses to stay in the current lane, Car 2 will accelerate based on the payoff. In this example, Car 1 gets the highest payoff by staying in the current lane. As described above, the lane change decision making of the leader Car 1 naturally involves evaluation and search of the best payoff for its the strategy pair, (stay, accelerate), assuming rational decisions from Car 2. Therefore, we model the lane change decision making as a Stackelberg Game where there always exits an equilibrium. Note that while the lead vehicle has the first move advantage, the actual benefit depends on its correct assessment of the other vehicle.

While most scenarios in the traffic involve more than two vehicles, as shown in Fig. 2, other cars may influence the decision of the host vehicle, we argue that the vehicle interaction can be treated as two at a time. For example, for Car 1 to assess its lane change movement illustrated in Fig. 2, it often only considers the interaction between Car 2 and itself.[1] While Car 1 has to consider Car A's positions to prevent crashes in this lane change movement. Car A is unlikely to react to vehicles behind. We see that only Car 1 and Car 2 are interacting with each other when Car 1 changing lanes. Car A and Car B only provide constraints to possible movement of Car 1, the host vehicle. Therefore for most cases with multiple vehicles, we can consider the lane change decision as a two-player game with variants of payoff functions.[2]

The complete structure of the game is shown in Table 1, where a denotes the longitudinal acceleration of the vehicle and U denotes the payoff of the combination of strategies. Car 1 determines not only if it will change lanes, but also its acceleration while changing lanes or staying in the current lane. Since acceleration is continuous, there are infinite combinations of strategies. Car 2 also has an infinite number of choices. It can choose any acceleration within the assumed limits of the vehicle, i.e. $(-6, 4)\,\mathrm{m/s^2}$. Both cars are trying to maximize their payoffs. It can be seen that the payoff function plays an important role in the decision making process. In order to produce reasonable logic for change decision, the combination of two payoff functions are designed and considered. The first function, U_{safety}, quantifies the safety payoff that a player can obtain in the game. The second function, U_{space}, estimates the space

[1] While a host car may evaluate several gaps in the target lane when making lane change decision, each gap is an individual two player game in the above construct. A gap selection model is developed and combined when the GT-based model, which is presented in Section 3.6.

[2] In a special scenario where multiple vehicles intend to move into the same gap, a multi-player game instead of two player game must be considered. We will illustrate in Section 2.7 how this work can be extended to the multi-player game.

Table 1
Game formulation.

Decision making		Car 1	
		Change lanes $-6 \leqslant a \leqslant 4$	*Stay* $-6 \leqslant a \leqslant 4$
Car 2	$-6 \leqslant a \leqslant 4$	U_{c0},U_{s0}	U_{c1},U_{s1}

payoff that a player can get in the game.

2.2. Safety payoff

U_{safety} is defined as the change of the safety factor during the lane-changing process between current time and the instant when lane change is completed, as shown below.

$$U_{safety} = \frac{1}{2}(SP_{t=T_{cl}} - SP_{t=0}), \tag{1}$$

where $SP_{t=T_{cl}}$ is the safety factor when the lane-changing process ends, $SP_{t=0}$ is the initial safety factor and T_{cl} is the time needed to complete the lane change, which is estimated by

$$T_{cl} = \frac{y_1 T_{c1}}{LW}, \tag{2}$$

where y_1 is the lane position of Car 1, LW is lane width and T_{c1} is the duration of the complete lane change. The safety factor is a function of the time headway. Its range is between negative one and one. The smaller the time headway, the smaller the safety factor, as illustrated in Fig. 3.

The safety factor of a vehicle at time t is defined by

$$SP_t = \begin{cases} 1 & T_{headway\,t} \leqslant -T_b \\ \frac{2\,|\,T_{headway\,t}|}{T_b} - 1 & -T_b < T_{headway\,t} \leqslant T_b, \\ 1 & T_{headway\,t} > T_b \end{cases} \tag{3}$$

where $T_{headway}$ is the time headway during lane changes at time t and T_b is a breakpoint, which represents the desired time headway of the vehicle. When two cars are far enough from each other, the time headway is bigger than T_b. It is always safe under this condition and the safety factor reaches its maximum value. T_b of the host vehicle is preset, which is similar to the desired speed of the ACC system. T_b of the competing vehicle is estimated by

$$T_b = \min(3, T_0), \tag{4}$$

where T_0 is the initial time headway between the competing vehicle (i.e. Car 2) and the vehicle in front of it (i.e. Car A) when the host vehicle (i.e. Car 1) starts to change lanes. The number three comes from the three-second rule (Qiao et al., 2016), which suggests that

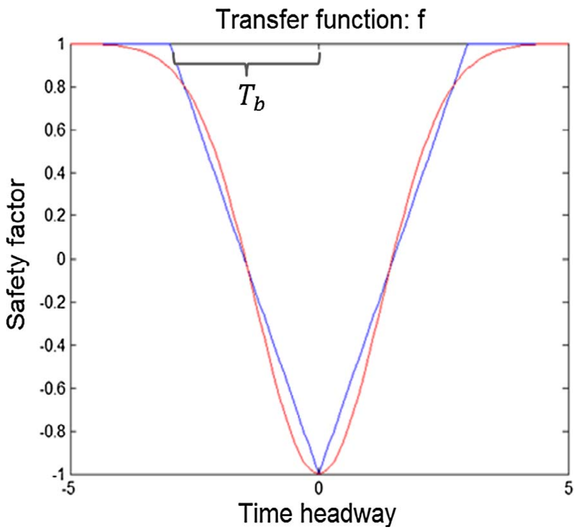

Fig. 3. Safety factor.

a minimum of a three-second interval between vehicles should be kept. Eq. (4) means if the initial time headway of the competing vehicle is bigger than three seconds, it is assumed that its desired time headway is three seconds. Otherwise, the desired time headway of the competing vehicle is T_0. The initial time headway between Car 1 and Car 2 is calculated by

$$T_{headwat=0} = \frac{P_1 - P_2}{v_2}, \tag{5}$$

where P_i denotes the initial longitudinal position relative to the road coordinate system, and v_i is the initial velocity. The time headway of the following vehicle after T_{cl} seconds is given by

$$T_{headwat=T_{cl}} = \begin{cases} \frac{P_{1l}-P_{2l}}{v_2+a_2T_{cl}} & P_{1l} \geqslant P_{2l} \\ \frac{P_{2l}-P_{1l}}{v_1+a_1T_{cl}} & P_{1l} < P_{2l} \end{cases}, \tag{6}$$

where v_i denotes the initial velocity relative to the road coordinate system, a_i denotes the future acceleration, P_{1l} and P_{2l} are longitudinal positions of Car 1 and Car 2 after T_{cl} seconds, which are calculated by Eq. (7) and Eq. (8). It should be noted that a_1 and a_2 in the equations are variables rather than constants. They are brought into the game to find the optimal strategies (i.e. acceleration) for the future. Consequently, a_1 and a_2 may not be the same as the current acceleration of vehicles.

$$P_{1l} = P_1 + v_1T_{cl} + \frac{1}{2}a_1T_{cl}^2, \tag{7}$$

$$P_{2l} = P_2 + v_2T_{cl} + \frac{1}{2}a_2T_{cl}^2. \tag{8}$$

In summary, the safety factor is a piecewise linear function of the time headway. However, a piecewise function is relatively slow in the optimization process. In order to make the model more efficient, a function that is differentiable everywhere is used to approximate the piecewise function. In this study, Gaussian probability distribution functions are adopted. The amplitude is fixed and the standard deviation is tuned to approximate the original function, as shown by the red curves in Fig. 3.

2.3. Space payoff

The second payoff function U_{space} estimates the change of the space factor RP, to be defined as a functions of the relative positions between two interactive vehicles. U_{space} is given by

$$U_{space} = \frac{1}{2}(RP_{t=T_{cl}} - RP_{t=0}), \tag{9}$$

where T_{cl} is the time needed to change lanes, $RP_{t=T_{cl}}$ is the space factor after T_{cl} seconds and $RP_{t=0}$ is the initial space factor. The space factor RP value of a vehicle indicates its competitive advantage to prevent its three seconds headway from being invaded by the other vehicle it interacts with. Its value is designed to be between negative one and one. We will first define RP of a competing vehicle (Car 2) w.r.t. subject vehicle (Car 1) at the competing lane, RP_{21_2}, when the two interactive vehicles are on separate nes. We will then generalize the definition to the situation where both vehicles are on the same lane.

When two cars move in different lanes, the space factor of Car 2, interacting with Car 1 at time instant t is defined as

$$RP_{21\_2}(t) = \begin{cases} -1 & t_{21}(t) \leqslant -3 \\ \frac{2}{3}t_{21}(t) + 1 & -3 < t_{21}(t) \leqslant 0, \\ 1 & t_{21}(t) > 0 \end{cases} \tag{10}$$

where *21* indicates it is the space factor of Car 2 and Car 2 is interacting with Car 1, *_2* indicates that the target lane of both cars is currently occupied by Car 2, t_{21} is defined as

$$t_{21} = \begin{cases} \frac{P_2-P_1}{v_2} & P_2 \leqslant P_1 \\ \frac{P_2-P_1}{v_1} & P_2 > P_1 \end{cases}, \tag{11}$$

where P_i denotes the initial longitudinal position relative to the road coordinate system, v_i denotes the initial velocity. t_{12} is given by

$$t_{12} = \frac{P_1 - P_2}{v_{following}} = -\frac{P_2 - P_1}{v_{following}} = -t_{21}, \tag{12}$$

where $v_{following}$ is the velocity of the following vehicle. Since Eq. (15) is defined based on t_{21} for the scenario where the competing lane is already occupied by Car 2, we define in Equation (18) to relate the space factors of the other interactive vehicle.

$$RP_{12\_2}(t) = -RP_{21\_2}(t). \tag{13}$$

Fig. 4 illustrates why the space factor is defined in this way. When Car 1 is far ahead of Car 2, Car 2 will not lose a lot of space if Car 1 changes lanes. As a result, Car 2 tends to give way to Car 1. If Car 1 wants to change lanes when Car 2 is close to Car 1, Car 2 has to decelerate to give way to Car 1. In this case, Car 2 tends to keep its space factor by accelerating as opposed to yielding.

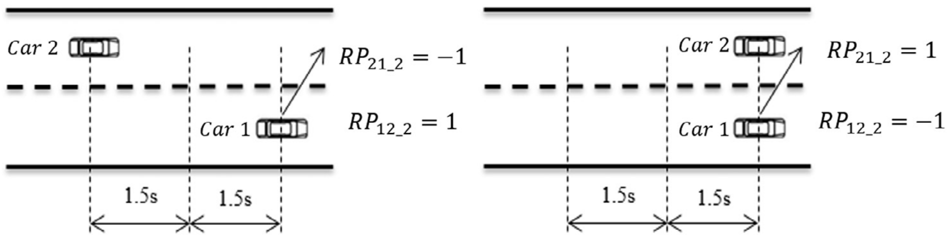

Fig. 4. Space factor when two cars move in different lanes.

When two cars move in the same lane, the space factor of Car 2 in the instant t is calculated by

$$RP_{21\_2}(t) = \begin{cases} -1 & t_{21}(t) \leqslant -3 \\ \frac{t_{21}(t)}{3} & -3 < t_{21}(t) \leqslant 3. \\ 1 & t_{21}(t) > 3 \end{cases} \tag{14}$$

Eq. (14) indicates that the space factor is determined by relative positrons of vehicles. Space factors of moving far behind and moving far ahead are negative one and one, respectively. If two vehicles are not far from each other, the space factor gradually increases as the relative position increases. It follows that the sum of two cars' space factors is still zero. For the sake of solving the game fast and efficiently, Eq. (10) and Eq. (14) are also approximated by functions that are differentiable everywhere as well. In this study, Gaussian cumulative distribution functions are utilized.

2.4. Total payoff

The total payoff function is a linear combination of the safety payoff and space payoff, which is given by

$$U_{payoff} = f_w(a,a_0)((1-\beta(q))*U_{safety}(a) + \beta(q)*U_{space}(a) + 1)-1, \quad 0 \leqslant \beta(q) \leqslant 1 \tag{15}$$

where a denotes the future acceleration of the vehicle, a_0 denotes the current acceleration, q denotes the aggressiveness of the driver that has Gaussian probability distribution of $\mathscr{N}(0,1)$, f_w is the penalty on the change of acceleration (i.e. jerk) and velocity, U_{safety} is the safety payoff of the strategy, U_{space} is the space payoff of the strategy, $\beta(q)$ is the cumulative distribution function of q to be used as the weight of the payoff. $\beta(q)*U_{space}$ is the total space payoff. $(1-\beta(q))U_{safety}$ is the total safety payoff. It can be seen that β is the most important parameter in the total payoff function. It affects the ratio of the total space payoff to the total safety payoff.

One parameter called aggressiveness is introduced into the equation. Aggressiveness of drivers has a significant influence on driving. Substantial research has been done to study human factors of driving, including aggressiveness (Ahmed, 1999; Beck et al., 2006; Shinar and Compton, 2004; Vanlaar et al., 2008). One of the widely used assumptions is that aggressiveness of drivers obeys Gaussian distribution (Huang et al., 1999; Huang et al., 2005), as shown in Fig. 5.

There is a one-to-one mapping between aggressiveness q and weight β. Aggressive drivers care more about space than safety. They have big β, reflecting more weight on space payoff than safety payoff. Cautious drivers care more about safety than space. Their weights of safety payoffs are bigger than the weights of space payoffs. It should be noticed that aggressiveness is an abstract concept. Eq. (15) actually defines a measure of aggressiveness, or the resulting payoff function due to driver aggressiveness. The penalty function is defined in Eq. (15) below so that it converges to the value of one when the acceleration action to take is close to the current acceleration, the anticipated final velocity is close to the desired velocity, and the lane change maneuver is almost complete (i.e. $T_{cl} \approx 0$).

$$f_w = e^{-\left(T_{cl}^2 * \frac{(a-a_0)^2}{w_1} + \frac{(v+a*T_{cl}-v_{desired})^2}{w_2}\right)} \tag{16}$$

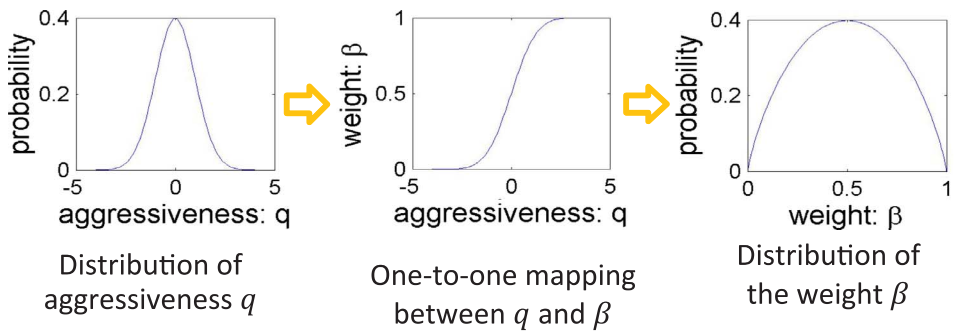

Fig. 5. Distribution of aggressiveness.

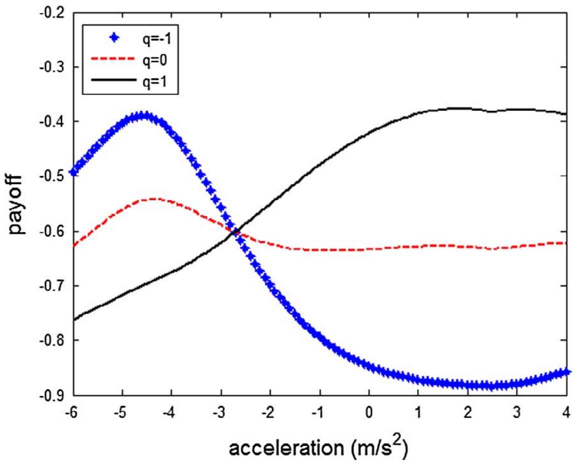

Fig. 6. An example of the payoff of the following car.

where v is the current velocity of the vehicle, $v_{desired}$ is the desired speed of the vehicle, w_1 and w_2 are penalty parameters and T_{cl} is the time needed to change lanes. The velocity of the competing vehicle before the interaction starts (e.g., before the turn signal is turned on) is regarded as its desired speed. The desired speed of the autonomous vehicle is preset, which is similar to the ACC system.

As an example, Fig. 6 illustrates how U_{payoff} could vary with aggressiveness state q and action a for a scenario where the velocity of the leading and following vehicles are 10 m/s and 15 m/s, with their initial distance 15 m and their initial accelerations zeros. It can be seen that an aggressive following car tends to accelerate to get high payoffs. Cautious cars tend to decelerate to make sure the distance between two cars is large enough.

2.5. Solution of the game

The estimated safety payoff and space payoff are brought into a GT-based model to search for the optimal strategy. In the lane-changing process, the host vehicle and the competing vehicle follow a 2-person Stackelberg game. This is a bilevel optimization problem, as shown below. Positions and marks of cars are shown in Fig. 2.

$$(a_1^*,c_l^*) = \underset{a_1,c_l}{argmax}\Big(\underset{a_2\in\gamma^2(a_1,c_l)}{min} U_1(a_1,c_l,a_2,q_1)\Big), \tag{17}$$

$$\gamma^2(a_1,c_l)\triangleq = \{\xi \in \Gamma^2\colon U_2(a_1,c_l,\xi,q_{12}) \geqslant U_2(a_1,c_l,a_2,q_{12}) ,\forall\ a_2 \in \Gamma^2\}, \tag{18}$$

subject to

$$v_i \geqslant 0, \quad i = 1,2,$$

$$a_{min} \leqslant a_i \leqslant a_{max}, \quad i = 1,2,$$

$$\begin{cases} P_1 < P_A - \frac{1}{2}(L_A + L_1) & after\ \ Car\ \ 1\ \ changes\ \ lanes \\ P_1 < P_B - \frac{1}{2}(L_B + L_1) & when\ \ Car\ \ 1\ \ stays\ \ in\ \ the\ \ current\ \ lane \end{cases},$$

$$SP_1 > K,$$

where U_i denotes the total payoff, a_i denotes the possible acceleration, c_l shows if Car 1 is changing lanes, a_1^* is the optimal acceleration, c_l^* shows if changing lanes is beneficial for Car 1, $\gamma^2(a_1,c_l)$ denotes the optimal action candidates of Car 2 given the actions of Car 1, Γ^2 indicates the action candidates of Car 2, SP_1 is the safety payoff of Car 1, a_{min} and a_{max} are the minimum and maximum acceleration that a vehicle can reach, q_1 is the aggressiveness of Car 1, q_{12} is the estimated Car 2's aggressiveness, P_i is the positions of Cars, L_i denotes the length of cars, v_i is velocity of cars.

The solution of this game is the strategy that maximizes the lower limit of the payoff from the leader's viewpoint, while considering the follower's reacting strategy. In other words, the solution indicates the pair that maximizes the worst-case payoffs subject to constraints. Constraints include non-negative velocities and the normal range of acceleration. In particular, since the predicted reactions of the competing vehicle may not be perfect, the minimum safe distance between cars should be added to constraints. In this study, d_{safe} is defined as

$$d_{safe} = 1*v_{following}, \tag{19}$$

where $v_{following}$ is the velocity of the following vehicle. Eq. (19) means the minimum acceptable time headway between cars is one second. The corresponding safety factor can be easily deducted, as shown below.

$$d_{safe} = 1 \times v_{following} \Leftrightarrow SP_1 \geqslant -\frac{1}{3}. \tag{20}$$

Even if all the candidate solutions for changing lanes do not meet the constraints, a feasible solution for staying in the current lane will still exist. In other words, it is safe to stay in the current lane by appropriately adjusting the speed. If high accuracy of strategies is not required, this bilevel optimization problem can be solve by extensively searching the discrete payoff matrix. For example, one decimal place is accurate enough for vehicle acceleration in this game. A payoff matrix that consists of a set of discrete acceleration can be built, which is similar to Table 1. Each element of the matrix represents the payoff of the combination of one of Car 1' strategies and one of Car 2's strategies. This matrix only has finite elements and the Stackelberg equilibrium can be found easily. If the accuracy of strategies is particularly important, this bilevel optimization problem can also be solved by the Bilevel Evolutionary Algorithm (BLEAQ) (Sinha et al., 2013). It should be noticed that the solution of the game is calculated in every instant. It may change when the game is played. For example, the solution may change from changing lanes to staying in the current lane after getting the latest traffic information, which acts like human drivers.

2.6. Estimate aggressiveness

The GT-based model simulates the reasoning process of the leader and follower during lane changes. If the host vehicle knows the aggressiveness of the competing vehicle, it can predict the future reactions of the competing vehicle. Based on the predicted reactions, the host vehicle can find its optimal strategies. However, there is still one unknown, the aggressiveness of the competing vehicle. In order to find the optimal strategy accurately, the host vehicle needs to estimate the aggressiveness of the competing vehicle, as shown in Fig. 7. The host vehicle starts to interact with the competing vehicle by using the turn signal or making a small lateral move. Initially, the host vehicle knows nothing about the competing vehicle. It is reasonable to assume the competing vehicle is driven by a normal driver, whose aggressiveness is zero. The host vehicle predicts how a normal driver will react to its future actions and finds its own optimal strategy. In the next instant, the host vehicle is able to observe the real action of the competing vehicle. The real action is compared with the predicted action and the difference is used to update the estimated aggressiveness.

The estimated aggressiveness is only updated when the difference is bigger than a threshold. It is updated by solving the following inequality.

$$U_{payoff}(\hat{q}_2, Real_{a_2}) \geqslant U_{payoff}(\hat{q}_2, Predicted_{a_2}), \tag{21}$$

where U_{payoff} is the total payoff of Car 2, $\hat{q}_2$ is the estimated aggressiveness of Car 2, $Real_{a_2}$ is the real acceleration of Car 2 and $Predicted_{a_2}$ is the predicted acceleration of Car 2 based on $\hat{q}_2$. Assuming we have the correct form of payoff for the computing vehicle,

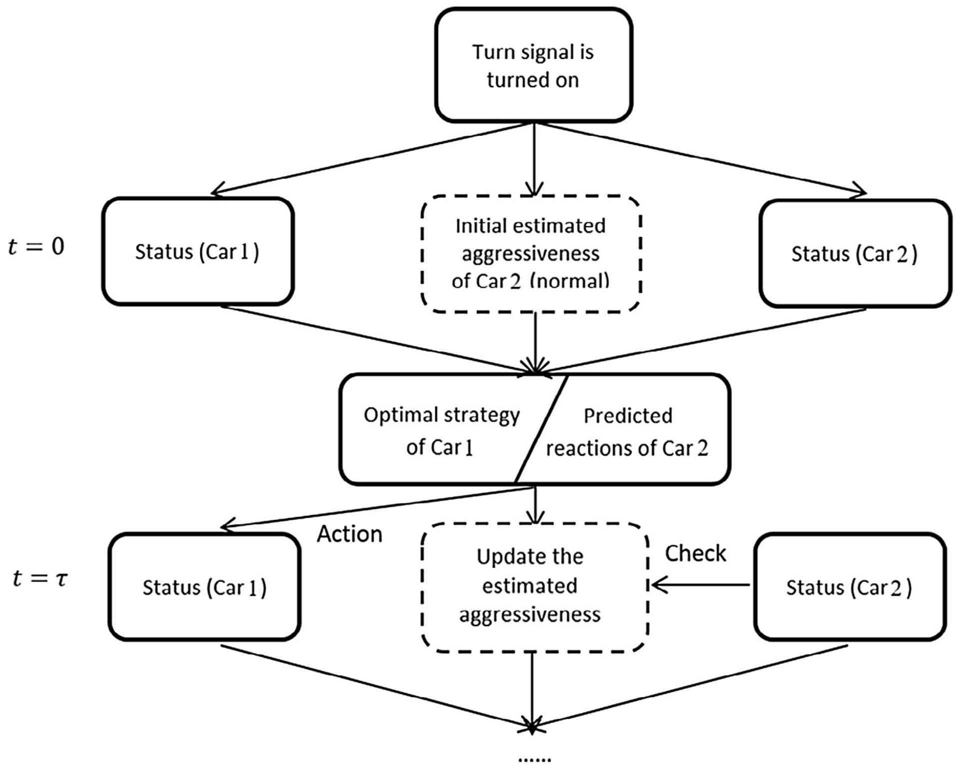

Fig. 7. Estimate aggressiveness.

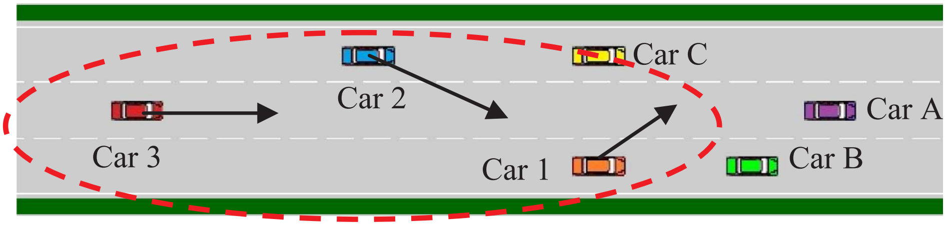

Fig. 8. Multiplayer game.

we can estimate the only unknown q_2. Eq. (21) means the payoff of the real strategy is always bigger than or equal to the predicted strategy. The range of the aggressiveness of Car 2 can be estimated by solving the inequality, which is similar to the deduction process of a human driver.

2.7. Multiplayer game

As discussed in Section 2.1, though most vehicle interactions can be treated as two at a time, the host vehicle may need to interact with multiple vehicles at the same time in certain scenarios. One example is shown in Fig. 8. Both Car 1 and Car 2 want to move to the center lane, which is occupied by Car 3. In this case, the GT-based model needs to be extended to a multiplayer game. Car A, Car B and Car C are still treated as moving obstacles and constraints. There are only three players (i.e. Car 1, Car 2 and Car 3) in this game. Car 1 is the lead vehicle. The problem is formulated as a three-player Stackelberg game, as shown below.

$$(a_1^*,c_{l1}^*) = \underset{a_1,c_{l1}}{argmax}\Big(\underset{a_2,c_{l2}\in\gamma^2(a_1^*,c_{l1}^*)}{min}\ \underset{a_3\in\gamma^3(a_1^*,c_{l1}^*,a_2,c_{l2})}{min} U_1(a_1,c_{l1},a_2,c_{l2},a_3,q_1)\Big), \tag{22}$$

$$\gamma^2(a_1,c_{l1}) \triangleq \{\xi\in\Gamma^2: \underset{a_3\in\gamma^3}{min} U_2(a_1,c_{l1},\xi,a_3,q_{e2}) \geqslant \underset{a_3\in\gamma^3}{min} U_2(a_1,c_{l1},a_2,c_{l2},a_3,q_{e2}),\forall\ a_2,c_{l2}\in\Gamma^2\}, \tag{23}$$

$$\gamma^3(a_1,c_{l1},a_2,c_{l2}) \triangleq \{\xi\in\Gamma^3: U_3(a_1,c_{l1},a_2,c_{l2},\xi,q_{e3}) \geqslant U_3(a_1,c_{l1},a_2,c_{l2},a_3,q_{e3}),\forall\ a_3\in\Gamma^3\}, \tag{24}$$

subject to

$$v_i \geqslant 0,\quad i = 1,2,3,$$

$$a_{min} \leqslant a_i \leqslant a_{max},\quad i = 1,2,3,$$

$$\begin{cases} P_1 < P_A - \frac{1}{2}(L_A + L_1) & after\ \ Car\ \ 1\ \ changes\ \ lanes \\ P_1 < P_B - \frac{1}{2}(L_B + L_1) & when\ \ Car\ \ 1\ \ stays\ \ in\ \ the\ \ current\ \ lane \\ P_2 < P_A - \frac{1}{2}(L_A + L_2) & after\ \ Car\ \ 2\ \ changes\ \ lanes \\ P_2 < P_C - \frac{1}{2}(L_C + L_2) & when\ \ Car\ \ 2\ \ stays\ \ in\ \ the\ \ current\ \ lane \end{cases},$$

$$SP_i > K,\quad i = 1$$

where U_i is total payoff, a_i denotes the possible acceleration, c_{l1} shows if Car 1 is changing lanes, c_{l2} shows if Car 2 is changing lanes, γ^2 denote the optimal action candidates of Car 2 given the actions of Car 1, γ^3 denote the optimal action candidates of Car 3 given the actions of Car 1 and Car 2, Γ^2 and Γ^3 indicate the action candidates of Car 2 and Car 3, SP is the safety payoff, a_{min} and a_{max} are the minimum and maximum acceleration that a vehicle can reach, q_1 is the aggressiveness of Car 1, q_{e2} and q_{e3} are the estimated aggressiveness of Car 2 and Car 3, P_i denotes the longitudinal position, v_i denotes the velocity and L_i is the vehicle length.

When there are multiple players in the game, the total space payoff of a car is the average of space payoffs with respect to other cars. For example, the space payoff of Car 1 is shown below.

$$U1_{space} = \frac{1}{2}(U12_{space} + U13_{space}) \tag{25}$$

where $U12_{space}$ and $U13_{space}$ are the space payoffs of Cars 1 with respect to Car 2 and Car 3. $U1_{space}$ is the total space payoff of Car 1. Since Car 3 is in the adjacent lane, $U13_{space}$ is still calculated by Eq. (14). In contrast, there is one lane between Car 1 and Car 2. $U12_{space}$ is calculated in a different way, which is discussed below.

If Car 1 changes lanes while Car 2 does not change lanes, Car 1 gets more space than Car 2. The space factor of Car 1 at instant t is calculated by

$$RP_{123}(t) = \begin{cases} -1 & t_{12}(t) \leqslant -3 \\ \frac{2}{3}t_{12}(t) + 1 & -3 < t_{12}(t) \leqslant 0, \\ 1 & t_{12}(t) > 0 \end{cases} \tag{26}$$

If Car 1 does not change lanes while Car 2 changes lanes, Car 1 loses some space. The space factor of Car 1 at instant t is calculated by

$$RP_{123}(t) = \begin{cases} -1 & t_{12}(t) \leqslant 0 \\ \frac{2}{3}t_{12}(t){-}1 & 0 < t_{12}(t) \leqslant 3, \\ 1 & t_{12}(t) > 3 \end{cases} \tag{27}$$

If Car 1 and Car 2 change lanes or stay in their current lanes together, the space factor of Car 1 at instant t is calculated by

$$RP_{123}(t) = \begin{cases} -1 & t_{12}(t) \leqslant -3 \\ \frac{t_{12}(t)}{3} & -3 < t_{12}(t) \leqslant 3. \\ 1 & t_{12}(t) > 3 \end{cases} \tag{28}$$

3. Tests and discussion

This section presents the test results of the proposed algorithm. Two tests were conducted. The first test was run in Simulink. The GT-based controller was compared with a rule-based controller in a lane-merging scenario. The classic bicycle model has proved to be effective in vehicle control (Jazar, 2013). It was adopted in this study. The state space model contains six states (i.e. x, y, ψ, v_x, v_y and ω) and two inputs (i.e. F_x and δ), where v_x, v_y and ω represent the longitudinal velocity, lateral velocity and yaw rate of the vehicle, x,y and ψ are longitudinal position, lateral position and yaw angle, δ is the steering angle and F_x denotes the longitudinal force. The second test was performed in the dSPACE environment. dSPACE allowed a human driver to use analog steering wheels and pedals to control a virtual vehicle in real time and made tests more realistic. The objective of the second test was to study if the GT-based controller could interact with a human driver. Besides the demonstration in this paper, test videos could be viewed at http://people.tamu.edu/~hzy5046.

3.1. Interact with virtual drivers

The first test scenario is lane merging, as shown in Fig. 9. The orange car was the host vehicle. It wanted to change lanes because it was approaching the end of the merging lane. However, there were multiple cars moving in the adjacent lane. The orange car needed to find the appropriate timing and acceleration to change lanes while preventing accidents. Initial conditions of the test are shown in Table 2. The blue car and the red car were relatively aggressive while the purple car was comparatively cautious. At the beginning, the orange car was slower than vehicles in the adjacent lane.

The rule-based controller uses a gap acceptance model. The host vehicle assumes the competing vehicle has constant velocities and acceleration over a period of time. The vehicle changes lanes when the minimum required acceleration for changing lanes is smaller than 4 m/s^2. The minimum required acceleration is calculated by

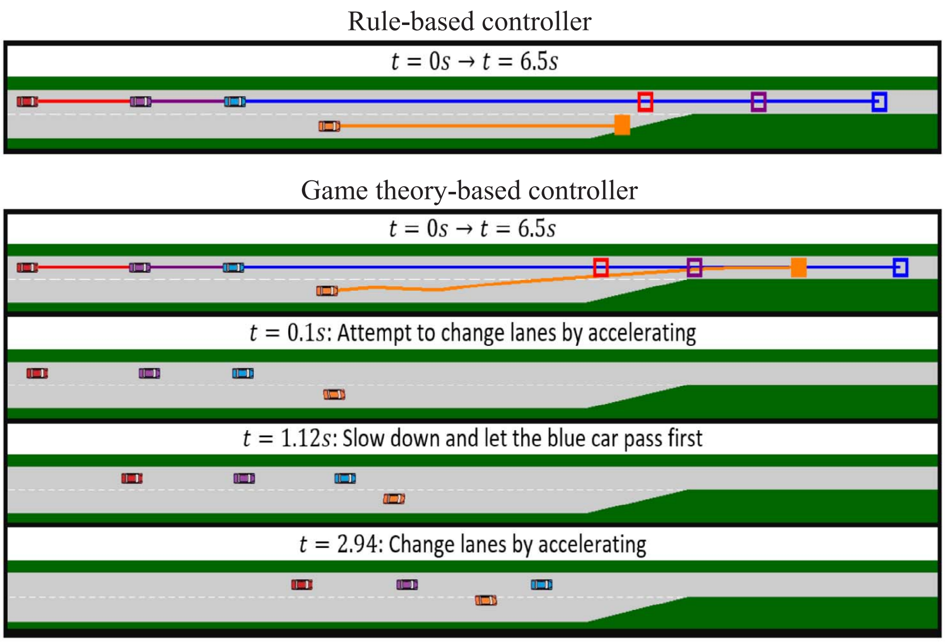

Fig. 9. A comparison between the rule-based controller and the GT-based controller.

Table 2
Initial conditions.

	Aggressiveness	Initial velocity (m/s)
Orange car	0 (used for the host vehicle)	10
Blue car	0.5	15
Purple car	−0.4	15
Red car	0.6	15

$$a_1 = \frac{v_2 T_{cl} + \frac{1}{2}a_2 T_{cl}^2 + d_{safe} - v_1 T_{cl} - d_{12}}{\frac{1}{2}T_{cl}^2}, \tag{29}$$

where v_i denotes the velocity of the vehicle, T_{cl} is the time needed to complete the lane change, d_{12} is the initial distance between the host vehicle and the competing vehicle, and d_{safe} is the minimum safe distance between two cars, which is defined by Eq. (19). Trajectories of vehicles are illustrated in Fig. 9.

Squares in the figure indicate the final positions of vehicles. The test result shows that the vehicle controlled by the rule-based controller always thought the required acceleration for changing lanes safely was too large. The host vehicle failed to change lanes and finally stopped at the end of the merging lane, which was dangerous. In contrast to the rule-based controller, the vehicle controlled by the GT-based controller was able to interact with surrounding vehicles and change lanes successfully, as shown in Fig. 10.

In phase one, the orange car assumed the blue car was driven by a normal driver. It predicted how a normal driver would react to its future actions according to the distribution shown in Fig. 5. After evaluation, the blue car thought it was able to change lanes successfully by accelerating. Consequently, it turned on the turn signal and attempted to change lanes. However, it found that the blue car was more aggressive than 0.4990 after a while. It was dangerous to further compete with the blue car. Therefore, the orange car changed its strategy and moved into phase two. It slowed down and let the blue car pass first. Then the orange car started to interact with the purple car, which is shown in phase three. It found that the purple car was less aggressive than −0.3149 during the interaction. As a result, the orange car tried to change lanes by accelerating. Finally, the orange car changed lanes successfully and reached phase four. It adjusted the time headway by changing its velocity. It can be seen that the GT-based controller outperformed the rule-based controller. Though the estimated aggressiveness was still different from the actual aggressiveness, meaningful information was extracted from the interaction, which was still helpful for finding the optimal strategy.

Another test was conducted for a multiplayer game, as shown in Fig. 11. Both the orange car and the blue car wanted to move to the center lane. Initial conditions of the test are shown in Table 3.

It can be seen that both the orange car (i.e. Car 1) and the blue car (i.e. Car 2) changed lanes successfully. The estimated aggressiveness of Car 2 and Car 3 is illustrated in Fig. 12. The true aggressiveness of Car 2 was 0.3. During the interaction process, it was estimated that the aggressiveness of Car 2 was bigger than 0.2933. The true aggressiveness of Car 3 was −0.5. It was estimated that the aggressiveness of Car 3 was less than −0.255. The host vehicle learned from the interaction and made reasonable decisions accordingly.

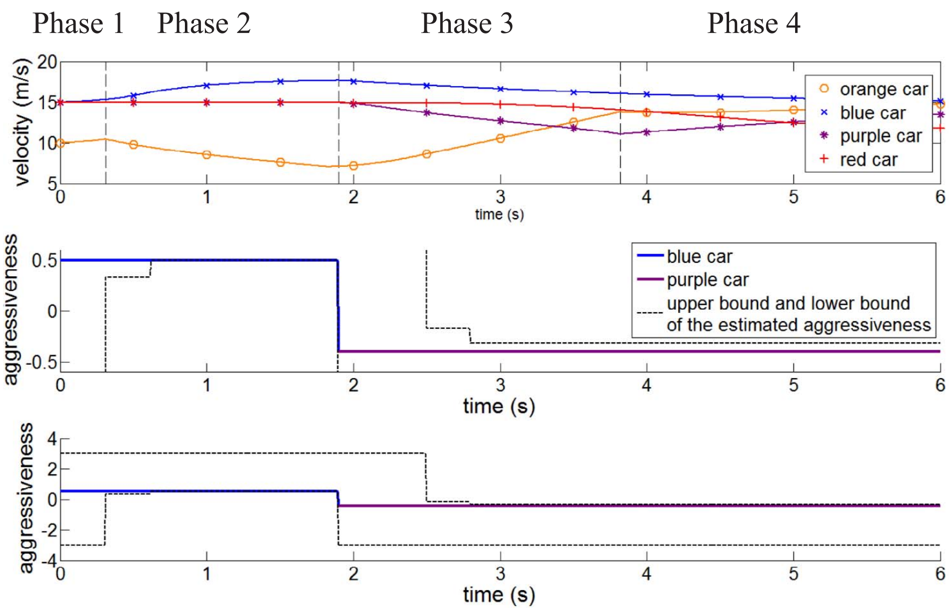

Fig. 10. Interactions between vehicles.

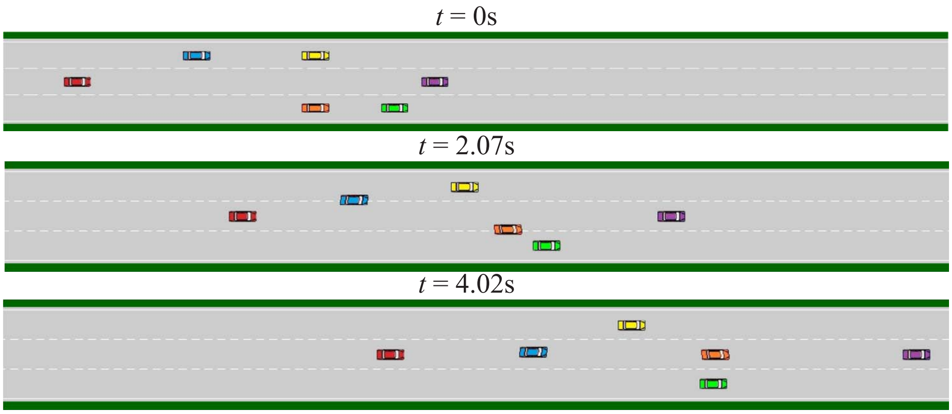

Fig. 11. Multiple cars want to move to the same spot.

3.2. Computing time

In this paper, the game was formulated by a discrete table (i.e. acceleration is not continuous in the decision-making table) for all the simulation. Fig. 13 shows the computing time for the lane merging scenario. The exemplary single-threaded MATLAB implementation was run on an Intel Core i7-6820 M at 2.7 GHz. The decision of the GT-based controller is updated every 0.3 s (i.e. the threshold). It can be seen that the computing time is always below the threshold. Therefore, the algorithm could be run in real time. Real-time experiments were conducted in dSPACE environment, as shown in Fig. 14.

3.3. Interact with human drivers

In the second test, a human driver participated in the interaction between vehicles, as shown in Fig. 14. The blue car was driven by a human driver while the orange car was controlled by a GT-based controller. Trajectories of cars are shown in Fig. 15. Orange circles indicate the positions of the host vehicle and blue squares indicate the positions of the human-driven car. At first, the human driver played the role of an aggressive driver. He tried to accelerate and pass the orange car. The test result shows that the estimated aggressiveness gradually increased and GT-based controller let the human driver pass and then changed lanes safely. In another trial, the human driver acted as a cautious driver. When he noticed the turn signal of the host vehicle, he pressed the brake pedal and let the host vehicle move to his lane. The test result shows that the estimated aggressiveness gradually decreased and the Car 2 ontrolled by the GT-based controller changed lanes successfully. This test validated that the GT-based controller could run in real time and its ability to interact with human drivers. However, other scenarios are needed to further study the interaction between the controller and human drivers.

3.4. Interact with connected vehicles

Besides interacting with human drivers, the proposed system could also cooperate with connected vehicles. If V2V technology is available, connected vehicles can share their levels of aggressiveness with each other. In such a way, the problem is turned into a game with complete information. The structure of the Stackelberg game works like a protocol and a vehicle can predict the actions of other connected vehicles accurately. Once the motion of other vehicles is known, the host vehicle can find its optimal strategy easily.

3.5. Discussion of the GT-based controller

A big difference between the GT-based controller and the rule-based controller is whether the future reaction of the competing vehicle is considered. A rule-based controller assumes the competing vehicle has constant velocities and acceleration. Then it tries to

Table 3
Multiplayer game: initial conditions.

	Aggressiveness	Initial velocity (m/s)
Orange car (Car 1)	0	10
Blue car (Car 2)	0.3	10
Red car (Car 3)	−0.5	15
Purple car	0	15
Green car	0	15
Yellow car	0	15

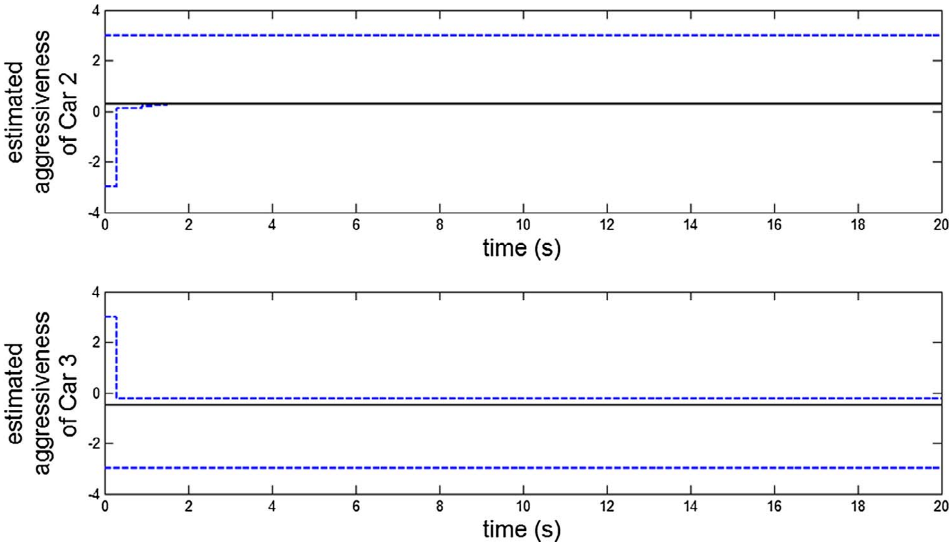

Fig. 12. Multiplayer game: estimated aggressiveness.

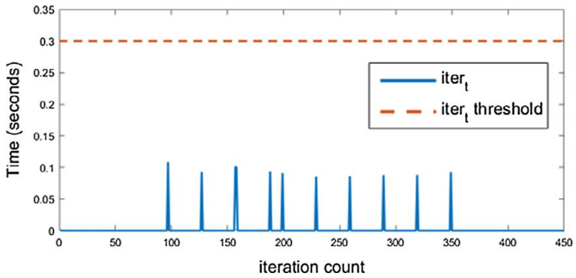

Fig. 13. Computing time.

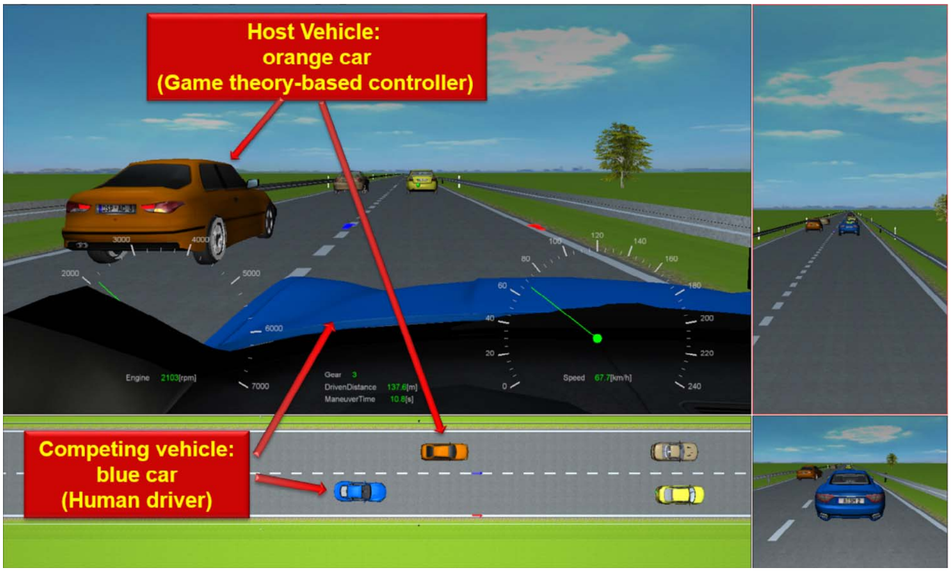

Fig. 14. dSPACE simulator.

find the optimal strategy based on that assumption. In contrast, the GT-based controller predicts the future reactions of the competing vehicle and finds the optimal strategy accordingly. The future strategy of the host vehicle and the reaction of the competing vehicle are calculated at the same time.

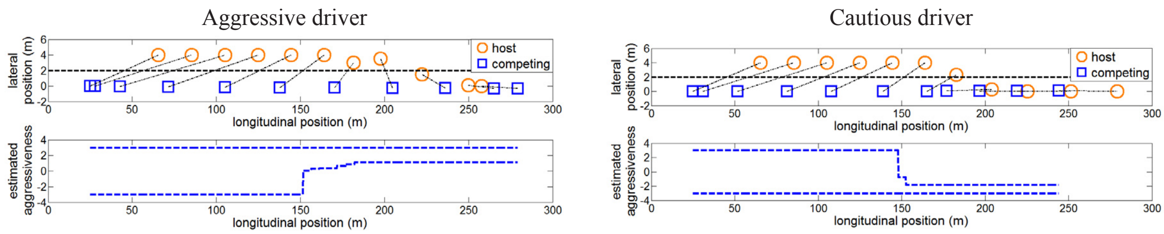

Fig. 15. Interact with a human-driven car.

In addition, the host vehicle (i.e. the car controlled by the GT-based controller) works by interacting with vehicles nearby in previous tests. However, there may not be cars within the detection range of sensors such as a LiDAR sensor sometimes. In this case, the GT-based controller of the host vehicle can assume there exists a vehicle following far behind (e.g., 1000 m away) in the adjacent lane. It tries to interact with this imaginary car. Evidently, this car hardly affects the payoffs and decision of the host vehicle. The host vehicle is able to find the optimal acceleration for changing lanes by calculating its total payoff. Since the total payoff function has a penalty for changing acceleration, the host vehicle will change lanes smoothly.

Another important point is that the estimated aggressiveness and the acceleration of the competing vehicle are not always negatively correlated. An example is shown in Fig. 16. Orange circles represent the positions of the host vehicle while blue squares represent the positions of the competing vehicle. It can be seen that though the competing vehicle decelerated at the beginning, the estimated aggressiveness jumped to a positive value. The reason is that the host vehicle thought a normal driver would decelerate faster than the competing vehicle in that situation. Therefore, the competing vehicle was still relatively aggressive.

Finally yet importantly, it may be doubted that the distribution of aggressiveness is possibly not accurate. Currently, it is assumed the aggressiveness of drivers obeys normal distribution, as shown in Fig. 5. However, the aggressiveness' distribution actually depends on the environment. For example, the aggressiveness of drivers in big cities is probably higher than that of drivers in small towns. It follows that the distribution is left-skewed in big cities and right-skewed in small towns.

A test was conducted to study the influence of the distribution of aggressiveness. Let q_c denote the aggressiveness of the competing vehicle. The distribution of aggressiveness is left-skewed, as shown in Fig. 17. The host vehicle wanted to change lanes when there was a normal vehicle (i.e. the aggressiveness was 0.67, which was the mean value) following in the adjacent lane, as shown in Fig. 18.

The upper figure shows how the host vehicle moved when the real left-skewed distribution was unknown. A normal distribution was used to generate an initial guess at q_2, which was zero (i.e. the mean value). If the host vehicle could change lanes when $q_2 = 0$, it could also change lanes successfully when $q_2 < 0$. Therefore, if the GT-model showed a vehicle could change lanes when $q_2 = 0$, there was a 50% probability that $q_2 \leqslant 0$ and the host vehicle could change lanes successfully. However, the real distribution was left-skewed in this test. It meant that even if the host vehicle could change lanes when $q_2 = 0$, there was still an 89.6% probability that $q_2 > 0$ and the host vehicle could not change lanes. As a result, it can be seen from Fig. 18 that the host vehicle failed to change lanes at the first attempt. In contrast, the lower figure shows how the host vehicle moved when the real distribution was known. The host vehicle only changed lanes after the competing vehicle passed. In summary, there may be unnecessary lane-changing attempts when the distribution is left-skewed. Reversely, the host vehicle will lose some opportunities of changing lanes if the distribution is right-skewed.

Nevertheless, it should be noted that the assumed distribution is only used to guess how a normal driver will react in the first step.

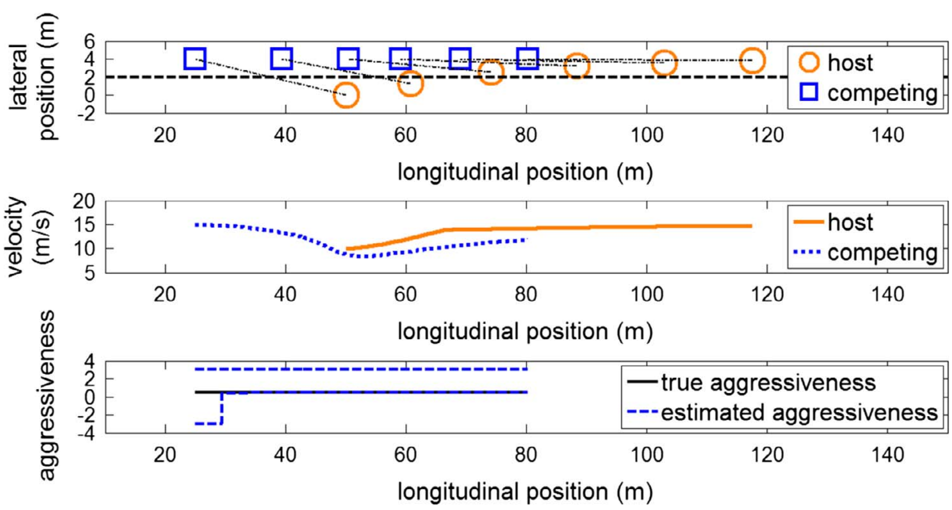

Fig. 16. The estimated aggressiveness increases when the competing vehicle is decelerating.

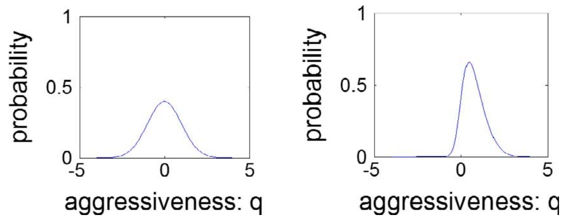

Fig. 17. Real distribution may be different from the assumption.

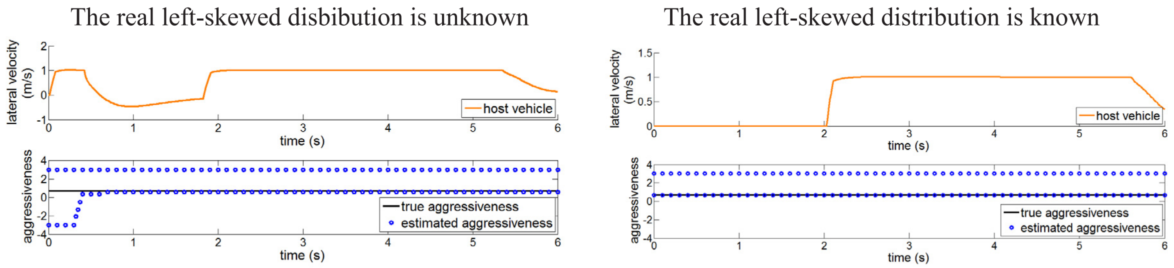

Fig. 18. The influence of the distribution of aggressiveness.

Only the mean of the distribution of aggressiveness is used. Even if it is not accurate, the estimated aggressiveness will be updated according to the real reactions of the competing vehicle.

3.6. Gap selection model

The proposed GT-based model evaluates if the host vehicle is able to get into a gap in the target lane. However, a human driver may evaluate several gaps in the target lane at the same time (Hidas, 2005), as shown in Fig. 19. The orange car needs to merge to the left lane. In this merging scenario, a human driver tends to evaluate Gap 1 and Gap 2 and the same time. Since Gap 2 is much bigger than Gap 1, the human driver will possibly slow down and move into Gap 2 directly. In contrast, the original GT-based model will let the orange car interact with the red car (e.g., turn on the turn signal) and evaluate the possibility of moving into Gap 1 first no matter how big Gap 2 is. Unfortunately, when the orange car is at Position A, the purple car may not be able to see and interact with the orange car since the red car is between them. The GT-based model cannot be used to evaluate multiple gaps in this case since the interaction between the orange car and the purple car may not exist when the orange car is at Position A. In this paper, it is assumed that the following car will only react to the lead car when there is no car between them. In order to further mimic human logic, a gap selection model was developed and combined with the GT-based model. The gap selection layer compares the payoff of getting into Gap 1 at Position A (i.e. current position) and the payoff of getting into Gap 2 at Position B (i.e. a little behind the red car). If Gap 2 has a higher payoff than Gap 1, the system will let the orange car slow down and enter Gap 2 by interacting with the purple car at Position B. The interaction process is a two-player game. Since the aggressiveness of the red car and the purple car are not known in the beginning, it is reasonable to assume both the purple car and the red car are driven by normal drivers while calculating payoffs. The payoff of a gap is shown below.

$$U_1 = \alpha_1 SP_{t=T_{cl}} + \alpha_2 \Delta D + \alpha_3 \Delta V, \tag{30}$$

where $SP_{t=T_{cl}}$ is the safety factor for changing lanes, ΔD is the distance between Position A and Position B, α_1, α_2 and α_3 are weights, ΔV is the change of velocity. The model will choose Gap 2 if Gap 2 is safer than Gap 1 and the host vehicle does not need to slow down a

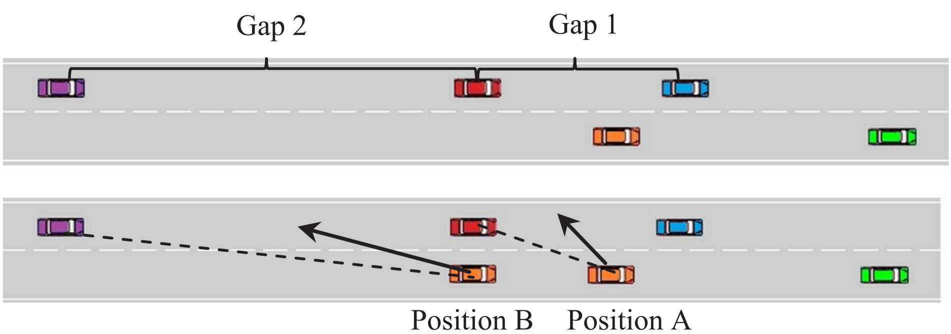

Fig. 19. Gap selection.

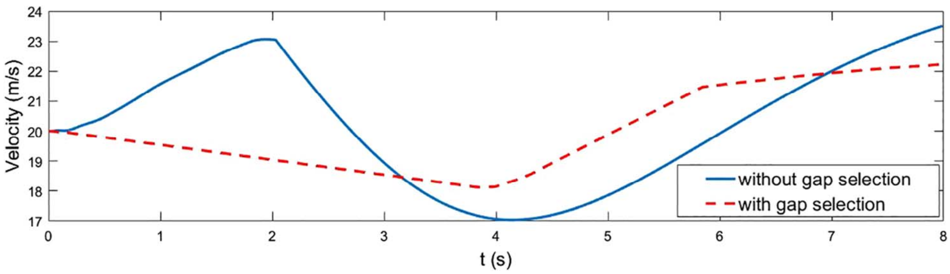

Fig. 20. Test of the gap selection model.

lot. This model can be further extended to compare more than two gaps. A test was conducted to compare the original model with the extended model, as shown in Fig. 20. The deceleration for moving from Position A to Position B was chosen to be $-0.5\ m/s^2$.

The velocity of the host vehicle (i.e. the orange car) is illustrated. The blue curve shows the change in velocity when the gap selection layer was not implemented. The orange car tried to get into Gap 1 first by speeding up. After a short period of time, it found the red car was more aggressive than expected. As a result, the orange car slowed down and waited for the opportunity to get into Gap 2. It finally entered Gap 2 successfully. The red curve shows the velocity when the gap selection layer was utilized. The system chose Gap 2 directly since it had a higher payoff than Gap 1. As you can see, the whole process was comparatively smooth after the gap selection model was added. An unnecessary lane-changing attempt was avoided.

4. Conclusions

This paper proposes a lane-changing controller based on game theory. The controller can find the situation-dependent optimal timing and acceleration for changing lanes by interacting with surrounding vehicles, estimating their aggressiveness and predicting future reactions, which imitates the reasoning process of human drivers. An advantage of this controller is that it considers the potential reactions of surrounding vehicles to the future actions of the host vehicle. In such a way, the vehicle could adjust its strategies in advance. It is similar to MPC, which considers the future actions of the host vehicle and changes inputs beforehand. If the real reaction is different from the predicted reaction, the estimated aggressiveness will be updated to improve the prediction. It should be noted that the estimated aggressiveness might not converge to the real value due to the short duration of lane changes. However, as long as the updating direction of the estimated aggressiveness is correct, the extracted information will help the system find superior strategies. In order to future mimic human logic, a multiplayer game and a gap selection model were developed for special scenarios such as several cars want to move to the same spot simultaneously. The GT-based controller was tested in both Simulink and dSPACE. Test results show that it outperforms a rule-based controller and is able to interact with human drivers in real time.

In the present study, a PID controller is used for lateral control and the lateral movement is almost linear. In the future, the lateral control could also be integrated into the game. As a result, the dimension of the game will increase. Furthermore, the current model only predicts one-step ahead. It is possible to extend the model so that it could predict multiple steps ahead, similar to a longer prediction horizon in MPC. In addition, the GT-based model could be extended to traffic simulation. The traffic model will be more realistic when the interaction between drivers is considered. Most importantly, the structure of our presented approach can be further extended to other complex scenarios, including those that involve decision makers interacting with each other. For example, it can be used for studying the interaction between vehicles and pedestrians at an intersection.

Acknowledgment

This work was supported by Ford Motor Company.

Appendix A. Comparison between the GT-based controller and MPC

In this section, the GT-based controller is compared with MPC. A lane merging scenario is selected, as shown in Fig. A1. Car 1 is the host vehicle. It wants to merge to the left lane and two gaps are available. A multiphase MPC (Anil et al., 2010; Darby et al., 2011) was chosen to control the host vehicle. It calculates the costs of merging into Gap 1 and merging into Gap 2. The gap that has a lower cost will be chosen. The cost function of MPC is shown below.

$$minimize\ J = tf_2 + \int_0^{tf1} (\gamma_1 \dot{a}^2 + \gamma_2 \dot{\delta}^2)dt + \int_{tf1}^{tf2} (\gamma_1 \dot{a}^2 + \gamma_2 \dot{\delta}^2)dt, \tag{1}$$

subject to

$$\forall\ i = 1{,}2$$

$$\dot{\zeta}_i = f(\zeta_i, u_i),$$

$$\zeta_{min} \leqslant \zeta_i(t) \leqslant \zeta_{max},$$

$$u_{min} \leqslant u_i(t) \leqslant u_{max},$$

subject to

$$\zeta_2(tf_2) = \zeta_f,$$

$$\zeta_1(0) = \zeta_0,$$

$$\zeta_1(tf_1) = \zeta_2(tf_1),$$

$$x_{carA}(t) \leqslant x_2(t) \leqslant x_{car2}(t), \text{ when gap 1 is selected}$$

$$\text{or } x_{car2}(t) \leqslant x_2(t) \leqslant x_{car3}(t), \text{ when gap 2 is selected}$$

where J is the cost function of the host vehicle, a is the acceleration of the vehicle, δ is the steering angle, tf_1 is the duration of phase one of the lane change, tf_2 is the duration of the lane change, ζ_i is the state vector of phase i, ζ_{min} and ζ_{max} are the lower bound and upper bound of the state vector, u_i is the control vector of phase i, u_{min} and u_{max} are the lower bound and upper bound of the control vector, γ_1 and γ_2 are weights, ζ_0 is the initial value of the state, ζ_f is the final value of the state, u_{min} and u_{max} are the minimum value and the maximum value of the input, x_2 is the longitudinal position of Car 1 of phase two, x_{carA}, x_{car2} *and* x_{car3} are the longitudinal positions of Car A, Car 2 and Car 3 in the adjacent lane.

Several tests were conducted in Simulink. Initial conditions and test results are shown in Table A1. $x0_{carA}$, $x0_{car1}$, $x0_{car2}$ and $x0_{car3}$ are initial positions of Car A, Car 1, Car 2 and Car 3 respectively. $v0_{carA}$, $v0_{car1}$, $v0_{car2}$ and $v0_{car3}$ denote initial velocities. $aggressiveness_{carA}$,$aggressiveness_{car2}$,$aggressiveness_{car3}$ denote aggressiveness of cars. In test 1, Gap 1 is big enough. Both the GT-based controller and MPC let the host vehicle get into Gap 1. In test 2, Gap 1 is relatively small. Gap 2 is a better choice than Gap 1. Both the GT-based controller and MPC let the host vehicle get into Gap 2. In test 3, Both Gap 1 and Gap 2 are small. MPC cannot find a solution. In contrast, the GT-based controller let the host vehicle interact with other cars. At first, it finds that Car B is more aggressive than 0.05 and gives up getting into Gap 1. After that, it interacts with Car C and finds that it is less aggressive than -0.24 and gets into Gap 2 successfully. Test results show that the GT-based controller outperforms MPC in certain scenarios.

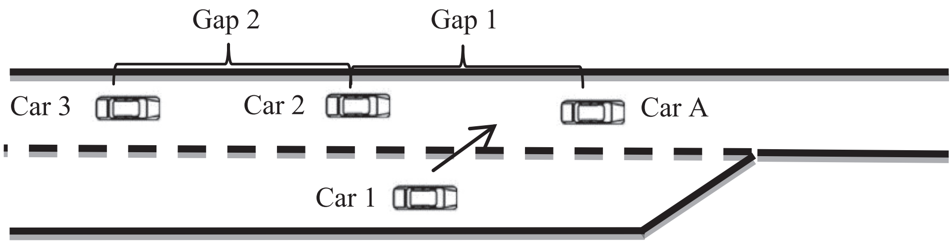

Fig. A1. Lane merging.

Table A1
Initial conditions and test results.

	Test 1	Test 2	Test 3
Position of the end of the merging lane (m)	50	50	50
$x0_{carA}$ (m)	10	10	10
$v0_{carA}$ (m/s)	15	15	15
$aggressiveness_{carA}$	0	0	0
$x0_{car2}$ (m)	−20	−8	−8
$v0_{car2}$ (m/s)	15	15	15
$aggressiveness_{car2}$	0.3	0.3	0.3
$x0_{car3}$ (m)	−30	−25	−18
$v0_{car3}$ (m/s)	15	15	15
$aggressiveness_{car3}$	−0.3	−0.3	−0.3
$x0_{car1}$ (m)	0	0	0
$v0_{car1}$ (m/s)	10	10	10
GT-based controller	Get into Gap 1	Get into Gap 2	Get into Gap 2
MPC	Get into Gap 1	Get into Gap 2	Infeasible

References

Ahmed, K.I., 1999. Modeling Drivers' Acceleration and Lane Changing Behavior. Massachusetts Institute of Technology.

Alin, A., Butz, M.V., Fritsch. J., 2012. Incorporating environmental knowledge into Bayesian filtering using attractor functions. In: Paper presented at the Intelligent Vehicles Symposium (IV); 2012 IEEE.

Anil, V., David, A., Christopher, D., 2010. GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method. ACM T Math Software 37, 22–39.

Bahram, M., Lawitzky, A., Friedrichs, J., Aeberhard, M., Wollherr, D., 2016. A game-theoretic approach to replanning-aware interactive scene prediction and planning. IEEE Trans. Vehicular Technol. 65 (6), 3981–3992.

Beck, K.H., Wang, M.Q., Mitchell, M.M., 2006. Concerns, dispositions and behaviors of aggressive drivers: what do self-identified aggressive drivers believe about traffic safety? J. Saf. Res. 37 (2), 159–165.

Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., Hrovat, D., 2005. MPC-based approach to active steering for autonomous vehicle systems. Int. J. Vehicle Autonomous Syst. 3 (2–4), 265–291.

Brand, M., Oliver, N., Pentland, A., 1997. Coupled hidden Markov models for complex action recognition. In: Paper presented at the Computer vision and pattern recognition, 1997. Proceedings, 1997 IEEE computer society conference on.

Darby, C.L., Hager, W.W., Rao, A.V., 2011. An hp-adaptive pseudospectral method for solving optimal control problems. Optimal Control Appl. Methods 32 (4), 476–502.

Hidas, P., 2005. Modelling vehicle interactions in microscopic simulation of mergingand weaving. Transportation Research Part C: Emerging Technologies 13 (1), 37–62.

Huang, S.-N., Chan, S.C., Ren, W., 1999. Mixed traffic control involving manually-controlled and automatically-controlled vehicles in IVHS. In: Advances in Intelligent Autonomous Systems. Springer, pp. 507–528.

Huang, S., Tan, K.K., Lee, T.H., 2005. Autonomous cruise control using neural networks in platooning. Adv. Robotics 19 (2), 169–189.

IEEE News Releases. (2012, Nov 1). Look Ma, No Hands! Expert Members of IEEE Identify Driverless Cars As Most Viable Form of Intelligent Transportation, Dominating the Roadway by 2040 and Sparking Dramatic Changes In Vehicular Travel. Retrieved from < http://www.ieee.org/about/news/2012/5september_2_2012.html > .

Jazar, R.N., 2013. Vehicle Dynamics: Theory and Application. Springer Science & Business Media.

Kato, S., Tsugawa, S., Tokuda, K., Matsui, T., Fujii, H., 2002. Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. IEEE Trans. Intell. Transport. Syst. 3 (3), 155–161.

Katrakazas, C., Quddus, M., Chen, W.-H., Deka, L., 2015. Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions. Transport. Res. Part C: Emerging Technol. 60, 416–442.

Khodayari, A., Ghaffari, A., Ameli, S., Flahatgar, J., 2010. A historical review on lateral and longitudinal control of autonomous vehicle motions. In: Paper presented at the 2010 International Conference on Mechanical and Electrical Technology.

Kim, K.-D., Kumar, P.R., 2014. An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic. IEEE Trans. Automatic Control 59 (12), 3341–3356.

Kita, H., 1999. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. Transport. Res. Part A: Policy Practice 33 (3), 305–312.

Lawitzky, A., Althoff, D., Passenberg, C.F., Tanzmeister, G., Wollherr, D., Buss, M., 2013. Interactive scene prediction for automotive applications. In: Paper presented at the Intelligent Vehicles Symposium (IV), 2013 IEEE.

Lefèvre, S., Vasquez, D., Laugier, C., 2014. A survey on motion prediction and risk assessment for intelligent vehicles. Robomech . 1 (1), 1.

Likhachev, M., Gordon, G.J., Thrun, S., 2003. ARA*: Anytime A* with provable bounds on sub-optimality. In: Paper presented at the Advances in Neural Information Processing Systems.

Liu, C., Tomizuka, M., 2016. Enabling safe freeway driving for automated vehicles. In: Paper presented at the 2016 American Control Conference (ACC).

Liu, H.X., Xin, W., Adam, Z., Ban, J., 2007. A Game Theoretical Approach for Modelling Merging and Yielding Behaviour at Freeway On-ramp Sections. Elsevier, London, pp. 197–211.

Morris, P., 2012. Introduction to Game Theory. Springer Science & Business Media.

Nash, J., 1951. Non-cooperative games. Annals of mathematics 286–295.

Nilsson, J., Sjöberg, J., 2013. Strategic decision making for automated driving on two-lane, one way roads using model predictive control. In: Paper presented at the Intelligent Vehicles Symposium (IV), 2013 IEEE.

Oliver, N., Pentland, A.P., 2000. Graphical models for driver behavior recognition in a smartcar. In: Paper presented at the Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE.

Ozguner, U., Stiller, C., Redmill, K., 2007. Systems for safety and autonomous behavior in cars: the DARPA grand challenge experience. PROCEEDINGS-IEEE 95 (2), 397.

Paden, B., Cap, M., Yong, S.Z., Yershov, D., Frazzoli, E., 2016. A survey of motion planning and control techniques for self-driving urban vehicles. arXiv preprint arXiv:1604.07446.

Qiao, F., Rahman, R., Li, Q., Yu, L., 2016. Safe and environment-friendly forward collision warning messages in the advance warning area of a construction zone. Int. J. Intell. Transport. Syst. Res. 1–14.

Sadigh, D., Sastry, S., Seshia, S.A., Dragan, A.D., 2016. Planning for autonomous cars that leverages effects on human actions. In: Paper presented at the Proceedings of the Robotics: Science and Systems Conference (RSS).

Shinar, D., Compton, R., 2004. Aggressive driving: an observational study of driver, vehicle, and situational variables. Accident Anal. Prevention 36 (3), 429–437.

Sinha, A., Malo, P., Deb, K., 2013. Efficient evolutionary algorithm for single-objective bilevel optimization. arXiv preprint arXiv:1303.3901.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Geyer, C., 2008. Autonomous driving in urban environments: boss and the urban challenge. J. Field Robotics 25 (8), 425–466.

Vanlaar, W., Simpson, H., Mayhew, D., Robertson, R., 2008. Aggressive driving: a survey of attitudes, opinions and behaviors. J. Saf. Res. 39 (4), 375–381.

Varaiya, P., 1993. Smart cars on smart roads: problems of control. IEEE Trans. Automatic Control 38 (2), 195–207.

Von Stackelberg, H., 2010. Market Structure and Equilibrium. Springer Science & Business Media.

Wang, D., Qi, F., 2001. Trajectory planning for a four-wheel-steering vehicle. In: Paper presented at the ICRA.

Wang, M., Hoogendoorn, S.P., Daamen, W., van Arem, B., Happee, R., 2015. Game theoretic approach for predictive lane-changing and car-following control. Transport. Res. Part C: Emerging Technol. 58, 73–92.

Yoo, J.H., Langari, R., 2012. Stackelberg Game Based Model of Highway Driving. In: Paper presented at the ASME Dynamic Systems and Control Conference (DSCC).

Yoo, J,H., Langari, R., 2013. A Stackelberg Game Theoretic Driver Model for Merging. In: Paper presented at the ASME 2013 Dynamic Systems and Control Conference.

Zhang, S., Deng, W., Zhao, Q., Sun, H., Litkouhi, B., 2013. Dynamic trajectory planning for vehicle autonomous driving. In: Paper presented at the 2013 IEEE International Conference on Systems, Man, and Cybernetics.