SCIENCE CHINA
Technological Sciences

•Article•

May 2021 Vol.64 No.5: 984–994
https://doi.org/10.1007/s11431-020-1729-2

An actor-critic based learning method for decision-making and planning of autonomous vehicles

XU Can, ZHAO WanZhong*, CHEN QingYun & WANG ChunYan

Department of Vehicle Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Received June 26, 2020; accepted September 27, 2020; published online March 19, 2021

In order to improve the agility and applicability of trajectory planning algorithm for autonomous vehicles, this paper proposes a novel actor-critic based learning method for decision-making and planning in multi-vehicle complex traffic. It is the coupling planning of vehicle's path and speed thus to make the trajectory more flexible. First, generations from the decided action to the planned trajectory are described by the end-point of the trajectory. Then, the actor-critic based learning method is built to learn an optimal policy for the decision process. It can update the policy by the gradient of the current policy's advantage. In this process, features of the real traffic are carefully extracted by time headway (TH) and speed distribution. Reward function is built by the safety, efficiency and driving comfort. Furthermore, to make the policy network have better convergency, the policy network is modularized in two parts: the lane-changing network and the lane-keeping network, which decide the optimal end-point of the path and speed candidates respectively. Finally, the curved overtaking scenario and the interaction process with human driver are conducted to illustrate the feasibility and superiority. The results show that the proposed method has better real-time performance and can make the planned coupling trajectory more continuous and smoother than the existing rule-based method.

trajectory planning, decision-making, actor-critic, feature extraction, autonomous driving

Citation: Xu C, Zhao W Z, Chen Q Y, et al. An actor-critic based learning method for decision-making and planning of autonomous vehicles. Sci China Tech Sci, 2021, 64: 984–994, https://doi.org/10.1007/s11431-020-1729-2

1 Introduction

1.1 Motivation

Nowadays, the researches on autonomous driving still raise much attention and many companies are putting efforts into high-level autonomous driving system. Within these technologies, the ACC, LDW, CWS have come into the markets. However, the level 3 or 4 according to the SAE classification that can realize skillful autonomous driving still remains deep exploration. Previous work about vehicle motion and dynamics control includes the model predictive control (MPC), the robust H-infinite controller, like refs. [1–3], which provide sufficient guarantee to the tracking accuracy. Based on this, the decision-making and planning system play an important role and are worthy of further study.

1.2 Related works

The existing studies on decision-making and planning are mainly classified in two theories: (1) the machine learning (ML) based method; (2) the model-based method. Within these two theories, the ML methods, like the artificial neural network (ANN), the deep learning and the Bayesian network, etc. [4–8], are popular these days with the repaid development of artificial intelligence (AI) and computing capacity. It can study the huge offline driving data from the experienced drivers by deep neural network and improve the network online to adapt to personal style. The online-offline

*Corresponding author (email: zwz@nuaa.edu.cn)

combined learning method was developed for the complex driving scenarios in refs. [9,10], which mainly built the general regression neural network (GRNN) to formulate the policy model, thus to realize the human-like personal driving. This method can be applied for a majority of scenarios, like the highway scenario, the urban condition or the intersection, but the error rate is still hard to eliminate and the driving style is relatively defensive. On the other hand, for the model-based method, the existing research mainly focused on low level autonomous driving, which is limited to specific scenario and the precondition is critical [11,12]. The scenario model predictive control was built for lane change assistance in ref. [13]. The APF model was widely built to realize path planning with multi constraints [14], which decided the optimal trajectory according to the gradient of the potential field force. In summary, the model-based method is reliable but the applicable scenarios are restricted. On the contrary, the ML method is flexible but not stable enough. In this circumstance, this paper combines the reliability of the model-based method and the flexibility of the ML algorithm. Meanwhile, the safety and efficiency in the decision process are ensured in a prominent position.

For the motion-planning model [15,16], the existing methods include the path candidate model, the lattices, Bezier, etc., [17,18]. These algorithms can address the path-planning task well, but the speed in the trajectory is roughly specified [19], which makes the formed trajectory not smooth enough. Besides, most of these planning methods simply solve the easy path-finding problem to optimize a cost function and short of the ability to handle long-horizon decision making, which lead to the deficiency of tackling more complex conditions. In this paper, we rebuild the trajectory model, which includes the yaw angle and speed information decided by the normal and tangential acceleration respectively. Then the traversal trajectories model is built by coupling these two profiles, which derives the transition from the action to trajectory.

To decide the optimal trajectory, the reinforcement learning process is built to replace the common search method [20]. Thus, the applicability and real-time performance can be improved to some extent. Many optimization algorithms are presented to cope with the learning problem. The Q-learning or deep Q-learning method is widely adopted [21,22], but the training process will be hard when the state space is enormous. Therefore, the direct policy search method is used in this paper that can convergence to the optimal policy rapidly. Besides, since the traversal trajectories increase a dimension compared to the path candidate model, the learning process gets more complicated. The existing learning methods mainly determine the policy for car-following maneuvers [23] or lane changing behaviors [24]. For the path and speed coupling condition, the optimal state values in different traffic scenarios are born different and the learning process is nearly impossible to converge to a stable value for these scenarios. Considering of that, we modularize the network to the lane-changing network and the keeping network since training these two networks respectively is more possible to converge. Then the decision task will be specific and can handle the multi-vehicle complex scenario well.

1.3 Contributions

The structure of this paper is shown in Figure 1, which mainly includes the generation of the traversal trajectories and the learning process of the optimal trajectory. The main contributions are listed below.

(1) The traversal trajectories model is built, which can reflect all the possible local trajectories the ego vehicle can reach. It is the coupled trajectory that contains path and speed profile and can be expressed by a series of end-points.

(2) The learning framework is constructed for the decision process that includes the actor-critic based learning method, the novel feature extraction method, and the reward function, which can make the decision system have good applicability to multi-vehicle complex scenario.

(3) The policy network is modularized into the lane-changing network and the lane-keeping network. The learning process is executed respectively, which makes the policy network have good convergence.

The remainder of this paper is organized as follows. Section 2 gives the generation of the coupled trajectory, which can infer the action space of the autonomous vehicle. In Section 3, the actor-critic based learning framework is built to learning the optimal action by the extracted environment state. Finally, in Section 4, the validation of the learning-based decision algorithm is implemented in the curved overtaking scenario and the human-vehicle interaction process. The conclusion is discussed in Section 5.

2 Generation of the traversal trajectories

As to the trajectory candidates, the existing researches mainly give limited path candidates to search the optimal one. However, the given candidates rarely consider the speed profile and cannot cover the whole accessible driving range, which will make the searched trajectory suboptimal. In consideration of that, this part gives the generation of the traversal trajectories which are the coupling planning of path and velocity.

2.1 Description of the trajectory

The local trajectory or the motion equation over the next horizon $[t, t+N_p]$ can be expressed as follows:

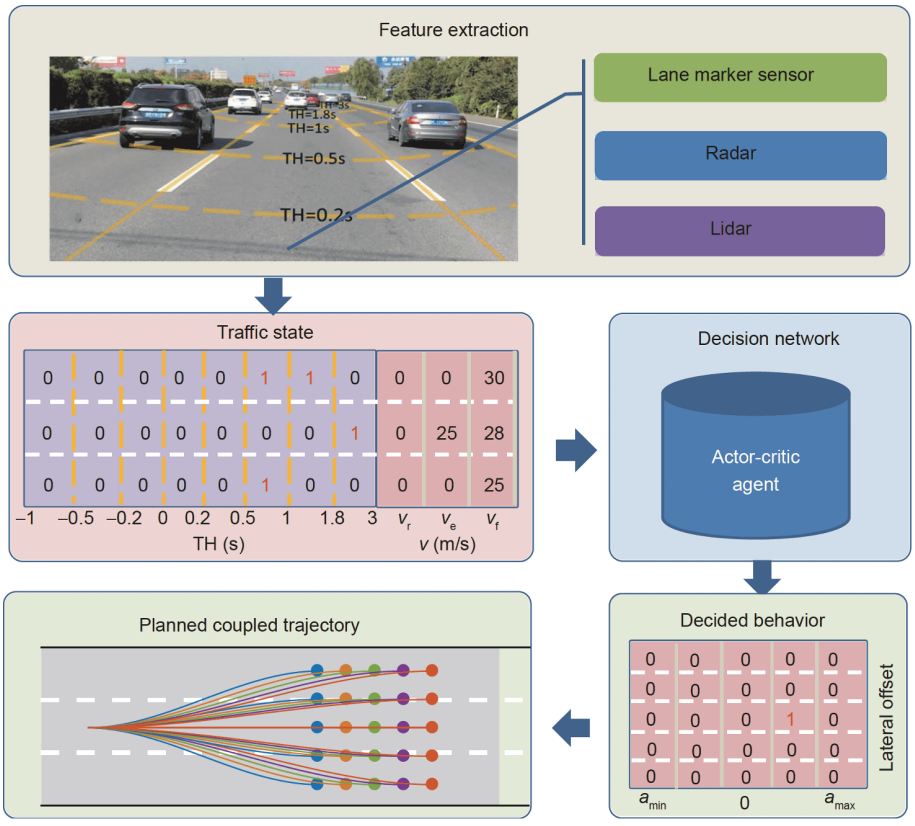

Figure 1 (Color online) Architecture of the decision-making system. TH, time headway.

$$\begin{bmatrix} s_{t+1} \\ l_{t+1} \end{bmatrix} = \begin{bmatrix} s_t \\ l_t \end{bmatrix} + T \begin{bmatrix} v_t * \cos\varphi_t \\ v_t * \sin\varphi_t \end{bmatrix} + \frac{T^2}{2!} \begin{bmatrix} \cos\varphi_t & -\sin\varphi_t \\ \sin\varphi_t & \cos\varphi_t \end{bmatrix} \begin{bmatrix} a_t^{\tau} \\ a_t^{n} \end{bmatrix}, \quad (1)$$

where s_t and l_t are the station and lateral position in the road coordinate respectively; v_t and φ_t are the velocity and yaw angle respectively; a_t^{τ} and a_t^{n} are the acceleration in the tangential and normal direction; T is the planning period.

Among these parameters, the position (s_t, l_t), velocity v_t and yaw angle φ_t are the state variables that will not mutate. The acceleration is the input variable. Therefore, the local trajectory can be described by determining the acceleration sequence $A(t)$:

$$A(t) = \{[a^{n}(t|t), a^{\tau}(t|t)], [a^{n}(t+1|t), a^{\tau}(t+1|t)], \cdots, [a^{n}(t+N_p-1\,|\,t), a^{\tau}(t+N_p-1\,|\,t)]\}^{\mathrm{T}}, \quad (2)$$

where the normal acceleration $a_t^{n} = Cv_t^{2}$; C is the curvature of the trajectory.

2.2 Coupling of path and speed candidates

It can be seen that the normal acceleration decides the curvature of the planned trajectory and the tangential acceleration determines the speed profile of the trajectory. When those two profiles are obtained, the local trajectory can be coupled. The following is the coupling of input acceleration candidates, or the path and speed candidates.

2.2.1 *The normal acceleration-path candidates*

The normal acceleration sequence is hard to obtain directly, since the high dispersity of the sequence. However, when the velocity is determined, the path curvature will be easy to solve since the driving path is always smooth and converges to the road direction. Therefore, the normal acceleration sequence is transformed to solve the path candidates or the lateral-station (L-S) function.

For the current moment t, the path candidates will be subject to the current position, yaw angle. For the terminal moment $t+N_p$, the vehicle usually has avoided the surrounding obstacles and tends to be stable, then the yaw angle and the normal acceleration will be zero. In this process, when the end-points are given, the local path candidates can be quartic polynomial fitted by five constraints in the initial and terminal point refer [25]. Since the lateral offset l_{t+N_p} is limited to the road boundary, it can be expressed as follows:

$$l_{t+N_p} = l_{\min} : \Delta l / C_p : l_{\max}, \quad (3)$$

where $l_{\min}$ and $l_{\max}$ are upper and lower limit of the road

boundary; $\Delta l=l_{\max}-l_{\min}$; C_p is the number of path candidates; i is the ith path candidate.

Then, the path candidates can be formulated as follows:

$$l = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4,$$
$$\text{s.t.}\begin{cases} l_t = l|_{s_t},\ \tan\varphi_t = \left.\dfrac{\mathrm{d}l}{\mathrm{d}s}\right|_{(s_t,l_t)}, \\ l_{t+N_p} = l|_{s_{t+N_p}},\ 0 = \left.\dfrac{\mathrm{d}l}{\mathrm{d}s}\right|_{(s_{t+N_p},l_{t+N_p})},\ 0 = \left.\dfrac{\mathrm{d}^2 l}{\mathrm{d}s^2}\right|_{(s_{t+N_p},l_{t+N_p})}, \end{cases} \tag{4}$$

where (s_{t+N_p}, l_{t+N_p}) is the ith end-point of the path candidate.

The normal acceleration in moment $t+k$ can also be acquired as follows:

$$a^n_{t+k} = C_{t+k} v^2_{t+k} = \left.\frac{l''(s)}{(1+l'^2(s))^{3/2}}\right|_{s_{t+k}} v^2_{t+k}, \tag{5}$$

where C_{t+k} is the curvature of the path candidate.

With the path function in the road coordinate system, the corresponding vehicle attitude in the global coordinate in Figure 2(a) can be transformed according to the coordinate of road centerline (x_s_t, y_s_t, ψ_s_t) as follows:

$$\begin{cases} x_t = x\_s_t - l\cdot\sin(\varphi_t), \\ y_t = y\_s_t + l\cdot\cos(\varphi_t), \\ \phi_t = \varphi_t - \psi\_s_t, \end{cases} \tag{6}$$

where ϕ_t is the yaw angle in the global coordinate. The path candidates of different driving postures are vividly shown in Figure 2(b).

2.2.2 *The tangential acceleration-speed candidates*

As to the tangential acceleration sequence a_t^τ, it determines the speed profile of the trajectory. Usually, the slope and the second derivatives of the station-time (s-t) function can reflect the speed and tangential acceleration respectively. The following is the solution of the s-t function according to the constraints in the initial and terminal moment.

The same as the path candidates, the normal acceleration in this process is temporarily ignored and set to zero. For the terminal moment $t+N_p$, the vehicle is stable and the tangential acceleration will tend to zero. In addition, the velocity is usually an ascending and descending process for the common overtaking or car-following process. Therefore, the speed in the terminal moment is set to the same as the initial moment. Finally, when the end-station is given, the s-t function can also be quartic polynomial fitted as follows:

$$s(t) = p_0 + p_1 t + p_2 t^2 + p_3 t^3 + p_4 t^4,$$
$$\text{s.t.}\begin{cases} s_t = s|_t,\ v_t = \left.\mathrm{d}s/\mathrm{d}t\right|_t, \\ s_{t+N_p} = s|_{t+N_p},\ v_{t+N_p} = \left.\mathrm{d}s/\mathrm{d}t\right|_{t+N_p},\ 0 = \left.\mathrm{d}^2 s/\mathrm{d}t^2\right|_{t+N_p}, \end{cases} \tag{7}$$

where v_{t+N_p} is the end-speed of ego vehicle decided by the end-station s_{t+N_p}.

The solution process is similar to that of the path candidates and is ignored there. For the end-station, it is influenced by the current velocity v_t and the performance of the vehicle's accelerator and brake, specifically as follows:

$$\begin{cases} s_{\max} = s_t + v_t T + a_e T^2/2, \\ s_{\min} = s_t + v_t T + d_e T^2/2, \end{cases} \tag{8}$$

where a_e and d_e are the acceleration and deceleration of the common driving respectively. Here a_e takes 2 m/s^2; d_e takes -3 m/s^2.

Then, the speed candidates can be got by the end-station as

$$s_{t+N_p} = s_{\min} : \Delta s / C_s : s_{\max}, \tag{9}$$

where $\Delta s=s_{\max}-s_{\min}$, C_s is the number of speed candidates.

With the s-t curvature function, the tangential acceleration candidates in moment k can be derived as

$$a^\tau_{t+k} = s''(t)|_{t+k}. \tag{10}$$

2.2.3 *Coupling of tangential and normal acceleration*

In this paper, to make the trajectories traversal and smooth, the path and speed are coupled-planned by the built kinematic equation. The tangential and normal acceleration sequences are given respectively above. When those two directions work simultaneously, the traversal action candidates are coupled as the matrix $A(t)$, specifically as follows:

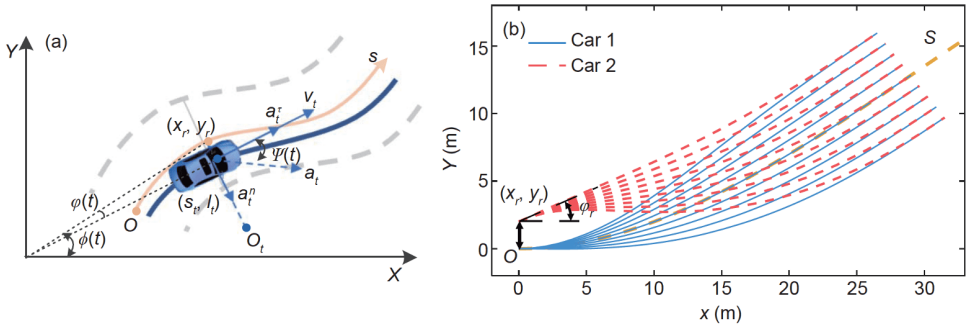

Figure 2 (Color online) (a) Vehicle position in the Frenet coordinate; (b) the path candidates for different driving postures.

$$A(t)=\begin{bmatrix} A_{11}(t) & A_{12}(t) & \cdots & A_{1C_s}(t) \\ A_{21}(t) & A_{22}(t) & \cdots & A_{2C_s}(t) \\ \vdots & \vdots & A_{ij}(t) & \vdots \\ A_{C_p1}(t) & A_{C_p2}(t) & \cdots & A_{C_pC_s}(t) \end{bmatrix}, \tag{11}$$

where

$$A_{ij}(t)=\left\{[a_i^n(t|t),a_j^\tau(t|t)],[a_i^n(t+1|t),a_j^\tau(t+1|t)],\cdots,[a_i^n(t+N_p-1|t),a_l^\tau(t+N_p-1|t)]\right\}^{\mathrm{T}},$$

i is the ith path candidate and j is the jth speed candidate.

Then the decision process can be transformed to the reinforcement learning process. The corresponding traversal trajectories in the straight, curved road and the turning intersection are graphically shown in Figure 3, which can reflect all the end-points the vehicle can arrive in the upcoming planning horizon. It can be seen that the end-points form a two-dimensional feasible region. When the optimal end-point is determined from the region, the coupled trajectory can be finally decided.

3 The actor-critic based learning process

It reveals that the coupled trajectories can be expressed by the end-points of the trajectories in Section 2. Usually, the optimization algorithm can be adopted to get the optimal trajectory from the trajectory candidates [20]. However, when there are too many candidates, the real-time performance will deteriorate. Therefore, this paper takes the learning method to get the optimal end-point of the trajectory, which can learn a sophisticated decision network that can handle the real complex traffic.

3.1 The learning framework

Firstly, we need to build the policy approximator to represent the policy. Usually, the neural network is taken to proximate the decision policy, which can output the action probability according to the input state. It will be introduced in Section 3.3.

With the policy representation, the policy gradient method is used to get the optimal policy, but the learning variance is big, which will result in slow convergence. Thus, the actor-critic method is proposed by adding the baseline to reduce the learning fluctuation, as shown in Algorithm 1.

First, the parameters of policy network are randomized and an experience trajectory is generated by interacting with the vehicle road environment as follows:

$$\tau_t=\{z_t,a_t,r_t,z_{t+1},a_{t+1},r_{t+1},...,z_{t+N_d}\}, \tag{12}$$

where N_d is the decision horizon.

The purpose is to max the expected cumulative discount reward of the trajectory.

Algorithm 1 The actor-critic learning algorithm

1. Initialize the actor with stochastic parameter values θ
2. Initialize the critic with stochastic parameter values θ_v
3. **for** episode =1,2,3… **do**
4. Randomize the initial state z_t
5. Sample N_d experiences from the state with the current policy θ: $z_t,a_t,r_t,z_{t+1},a_{t+1},r_{t+1},\cdots,z_{t+N_d}$
6. Calculate the return: $R(\tau_t)=\sum_{k=t}^{\infty}\gamma^{k-t}r_k=\sum_{k=t}^{t+N_d}\gamma^{k-t}r_k+\gamma^{N_d+1}V^{\theta}(z_{t+N_d})$
7. Calculate the Advantage: $A_t^{\theta}=R(\tau_t)-V^{\theta}(z_t)$
8. Gradient of the actor policy θ: $\nabla U(\tau_t,\theta)=\sum_{k=t}^{t+N_d}\nabla\log p(z_k,\theta)A_k^{\theta}$
9. Gradient of the critic θ_v: $\nabla V^{\theta_v}=\sum_{k=t}^{t+N_d}\nabla(V_k^{\theta}(z_k)-R_k)^2$
10. Update the policy parameter: $\theta=\theta+\alpha\nabla U(\tau_t,\theta)$
11. Update the critic parameter: $\theta_v=\theta_v+\beta\nabla V^{\theta_v}$
12. **end for**

$$\theta^*=\operatorname{argmax}(U(\tau_t,\theta)), \tag{13}$$

where θ is the policy parameters; $U(\tau_t,\theta)=E_{\tau_t\sim\theta}[R(\tau_t)]$, $R(\tau_t)=\sum_{k=t}^{\infty}\gamma^{k-t}r_k$.

Usually, the gradient descent method is adopted to update the policy and the importance sampling is used to get the policy gradient as follows [26]:

$$\nabla U(\tau_t,\theta)=E_{\tau\sim\theta}[\nabla\log p(\tau,\theta)R(\tau_t)]. \tag{14}$$

The baseline-value function is added to the reward to reduce the variance fluctuation as

$$\nabla U(\tau_t,\theta)=E_{\tau\sim\theta}[\nabla\log p(\tau,\theta)A_t^{\theta}], \tag{15}$$

where A_t^{θ} is the advantage function: $A_t^{\theta}=R(\tau_t)-V^{\theta}(z_t)$, $V^{\theta}(z_t)$ is the value function of the current policy θ.

Then, the policy gradient can be expressed by the experience trajectory as

$$\nabla U(\tau_t,\theta)=\sum_{k=t}^{t+N_d}\nabla\log p(z_k,\theta)A_k^{\theta}. \tag{16}$$

With the policy gradient, we can further update the policy network parameters.

$$\theta=\theta+\alpha\nabla U(\tau_t,\theta), \tag{17}$$

where α is the learning rate of the policy network, or the actor network.

At the same time, the approximator of the value function or the critic network is also built the same as the policy network. It can be updated by the gradients of mean square error

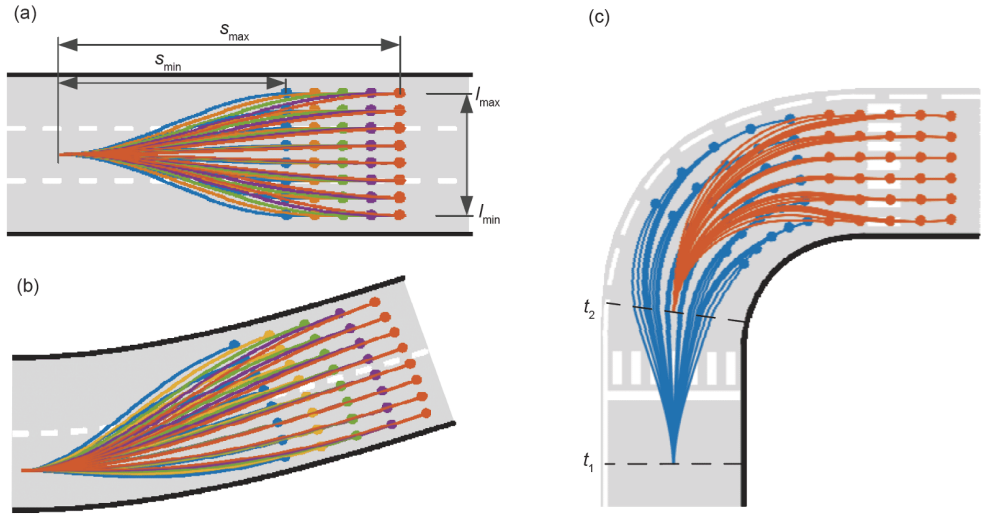

Figure 3 (Color online) The traversal trajectories for autonomous vehicles in the next driving horizon for (a) the straight road, (b) the curved road, and (c) the turning intersection.

loss between the estimated value $V^{\theta}(z_t)$ and the computed accumulated reward R_t.

$$\nabla V^{\theta_v} = \sum_{k=t}^{t+N_d} \nabla (V_k^{\theta}(z_k) - R_k)^2, \tag{18a}$$

$$\theta_v = \theta_v + \beta \nabla V^{\theta_v}, \tag{18b}$$

where β is the learning rate of the value function network, or the critic network.

3.2 State and reward in this process

The learning architecture has been built above, and then we need to determine the traffic state. The state of ego vehicle is the position (s_t, l_t), velocity v_t and yaw angle φ_t, which has been introduced in Section 2.1. The action can be represented by the end-points of the local trajectories. However, the decision state should not only include the state of ego vehicle, but also include that of surrounding vehicles. Usually, the nearest vehicles in the same lane will decide the speed of ego vehicle. The adjacent vehicles of two sides will influence the lane-changing behaviour. Therefore, the input state should cover the velocity and position distribution of ego vehicle and surrounding vehicles.

As shown in Figure 4, to guarantee the state space consistent for each decision period, the time headway (TH) is preferable than the space headway, since the effective space will diminish as speed changes. Then the observe window is discretized to simplify the decision states, thus to make the training process more efficient. The specific input state is shown in Figure 4(b), mainly including the position distribution and velocity of the front and rear vehicle.

For the longitudinal position, the surrounding vehicles can be expressed by the nearest neighbour since the grids are

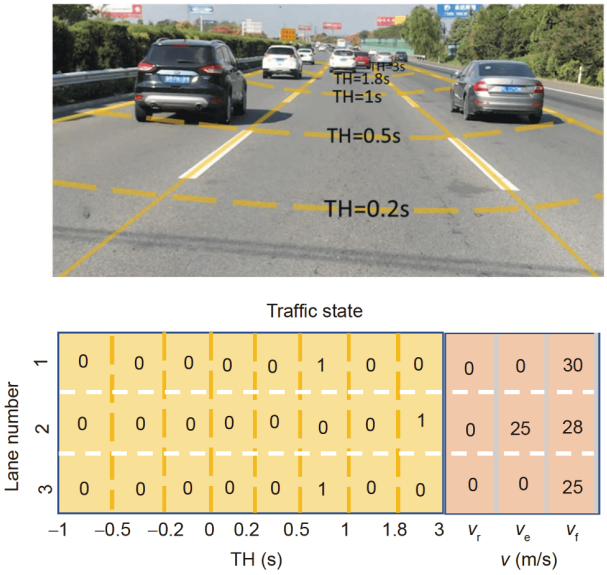

Figure 4 (Color online) (a) The real multi-vehicle traffic; (b) the extracted traffic state, where the relative positive and velocity can be got by vehicular radar and lidar, the ground truth can be got by camera.

given sufficient.

$$Th_k = Th_{\xi_i} \text{ for } \xi_i = \operatorname{argmin}\|k - \xi\|, \tag{19}$$

where Th_k is the time headway of the surrounding vehicle k, ξ is the set of the discrete time-headway, defined as $\xi = \{-1, -0.5, -0.2, 0, 0.2, 0.5, 1, 1.8, 3\}$.

For the lateral position, when the surrounding vehicle is changing lane and straddling two lanes, it will cause threat risk to the following vehicles on these two lanes. When the lane changing task is completed, the original lane will be spared. In this process, it will occupy two lanes if the edge of the vehicle passes the lane boundary and can be expressed as follows:

$$l_k = l_{\zeta_i}, \text{ if } \left\| l_k - l_{\zeta_i} \right\| < \frac{L}{2} + b, \tag{20}$$

where l_k is the lateral position of vehicle k, ζ is the set of the lane number ζ={1, 2, 3…}; L is the lane width, b is the distance from the vehicle side to CG.

The output action space has been shown in Section 2.2.3, which can be expressed by the end-points of the trajectory candidates. When calculating the reward r_t of the taken action a_t, the common reward items are mainly the driving safety, the efficiency and the comfort as

$$r_t = F_t + \lambda_e E_t + \lambda_c C_t, \tag{21}$$

where F_t is the safety assessment function; E_t is the efficiency; C_t is the riding comfort; λ_e and λ_c denote the weight on efficiency and comfort respectively.

Firstly, the existing indicators about safety include the time to collision (TTC), TH and time to brake (TTB) [27,28]. But they can only reflect the threat tendency and are hard to measure the actual risk. To solve this problem, the safe distance model is built to measure the risk as

$$d_{sf}(t) = v_e^t t_r + \frac{(v_e^t)^2}{2d_m^t} - \frac{(v_s^t)^2}{2d_m^t} + G, \tag{22}$$

where v_e and v_s are the velocity of ego vehicle and surrounding vehicle respectively; t_r is the react time; d_m is the maximum deceleration, here takes 5 m/s^2; G is the safety gap when they stopped, about 0.5 m.

We define the safety as "1", when the actual distance exceeds the safe distance as eq. (23):

$$F_t = \begin{cases} 1, & d_{es}(t) \geq d_{sf}(t), \\ 2 - \left(\dfrac{d_{sf}}{d_{es}} \right)^2, & d_{es}(t) < d_{sf}(t). \end{cases} \tag{23}$$

For the vehicle's efficiency, it is not the faster the vehicle drives, the more efficient it will be. When the driving speed goes beyond the reference speed of the road, it will be hard for the vehicle to avoid some imminent collision. Meanwhile, many traffic accidents may happen owing to the furious driving. Accordingly, the efficiency is defined by the closeness to the reference speed v_{ref} as follows:

$$E_t = 1 - \left| v_t - v_{\text{ref}} \right| / v_{\text{ref}}. \tag{24}$$

As to the comfort, it is relatively easy to represent. When the vehicle stays at the current lane and keeps the constant speed, the comfort is maximal to "0". On the contrary, the accelerating and lane-changing behaviour will reduce the comfort. The reduced comfort can be expressed as follows:

$$C_t = -\sqrt{(S_t - S_r)^2 + (L_t - L_r)^2}, \tag{25}$$

where S_t, L_t are the decided end-point of the longitudinal and lateral position; S_r, L_r are the end-point of the uniform motion and lane-keeping behavior.

3.3 The training process

The decision process from the traffic state to the action has been built above. The following is the learning process to obtain an optimal policy network.

3.3.1 *Modularization of the policy network*

As we know, driving in a smooth traffic is already very profitable and the reward is quite high. On the contrary, the reward in terrible traffic will be unsatisfying, even if taking the favourable driving behavior. In addition, the transfer from the busy traffic to the smooth traffic will be quite a long time. For a training episode, it is hard to experience all those complex scenarios and the policy can only be locally optimal for a specific scenario.

In view of that, this paper modularizes the network to the lane-changing network and the keeping network respectively. Then the decision task will be specific, where the lane keeping policy is responsible for safety and efficiency, the lane-changing policy handles safety and comfort. With these distributed decision networks, the coupled end-point of the local trajectory can be obtained. The policy network is shown in Figure 5, which has three hidden layers and a softmax layer, the neurons of the hidden layers are all 100. The input is the extracted environment state and the output is the probability of end-points. Finally, the most probable end-point of the lane-changing network and the lane-keeping network are taken as the decided behavior.

3.3.2 *Learning of the policy network*

The learning parameters of these policy networks are shown in Table 1. To make the policy explore adequate scenarios,

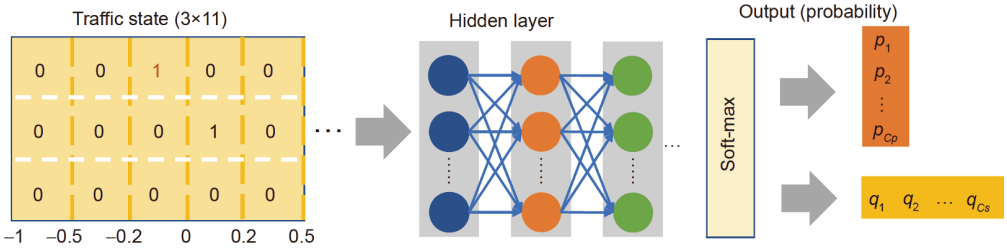

Figure 5 (Color online) The decision network that consist of the lane-changing network and the lane-keeping network, where p_i (i=1–Cp) is the probability of path candidates, q_j (j=1–Cs) is the probability of the speed candidates.

Table 1 Training parameters

Variables	Value	Description
T_d	0.5 s	Decision period
H_d	50 s	Decision horizon
Cp	3	Number of path candidates
Cs	3	Number of speed candidates
α	0.001	Learning rate of actor network
β	0.001	Learning rate of critic network
γ	0.9	Discount factor

the experience horizon takes longer to 50 s. The following is the training results.

For the lane-changing network, the decision states are mainly the position distribution of the surrounding vehicles. The purpose of the network is to choose a proper lane that can greatly improve the driving safety and make the cost of comfort worthy. The result of the lane-changing network is shown in Figure 6(a). It can be seen that the reward maintains a low level in the initial training phrase. After a thousand of episodes, the reward function converges to a high level that can handle complex scenarios.

For the lane-keeping network, the decision states are the relative distance and speed of the front or rear vehicle in the same lane. The purpose is to improve the efficiency as much as possible while ensuring the safety. This scenario has been studied by many researchers and the training process is relatively simple. The results are shown in Figure 6(b).

4 Validation and discussion

To verify the feasibility of the decision and planning algorithm, this section gives the validation and test. First the overtaking scenario is simulated in the curved road with walking pedestrians and parked vehicles. Furthermore, to validate the interaction performance with human driver, the driver in loop experiment is carried out in multi-vehicle scenario with contrast to the existing rule-based method.

4.1 Performance in the curved overtaking scenario

This scenario includes the walking pedestrians in the speed of 1.2 m/s, and two separate parked vehicles in the S-shaped curved road. Since the road is curved and the low speed of traffic participants, the expected speed of ego vehicle is set to 7 m/s. Figure 7 gives the planned trajectory of three styles including: the cautious style, the moderate style and the agile style.

The set parameters are shown in Table 2. For the decision process, the global coordinate is transformed to the Frenet coordinate to extract the traffic state. Then the policy network provides the optimal end-point to decide the corresponding trajectory.

It can be seen that the agile style maintains a high speed in this process, but it causes many detours thus makes the efficiency of the curved road unprofitable. On the contrary, the cautious style can adjust the speed well to get through the winding road. However, the speed in this process fluctuates a

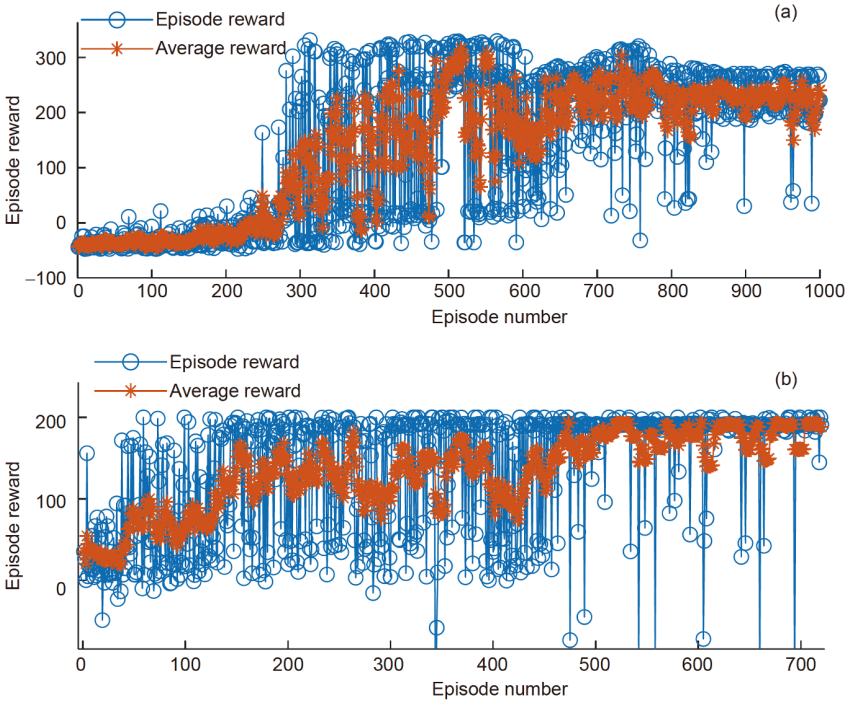

Figure 6 (Color online) The training results of the lane changing network (a) and the lane keeping network (b).

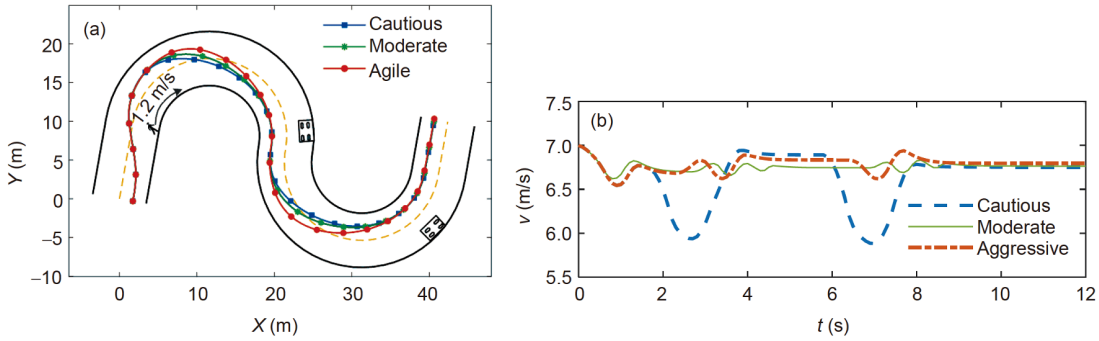

Figure 7 (Color online) The planned trajectory in curved scenario. (a) The decided personal trajectories; (b) the corresponding speed profile.

Table 2 Simulation parameters

Variables	Value	Description
T_p	0.1 s	Planning period
N_p	20	Planning horizon
L	3.5 m	Lane width
b	0.95 m	Distance from the vehicle side to CG
Cp	2	The number of path candidates
Cs	3	The number of speed candidates
γ	0.9	The discount factor
v_e	7 m/s	Initial speed of ego vehicle
λ_e	0.2/0.5/0.7	Weight on efficiency for three styles
λ_c	0.8/0.5/0.3	Weight on comfort for three styles

lot and the riding comfort is also influenced by the frequent acceleration and deceleration behavior. It reveals that the agile or cautious style is not always advantageous for the overtaking process. In contrast, the moderate style can make a compromise and behaves well in efficiency and comfort. In the following, we will make interaction with the human driver in the moderate style.

4.2 Interacting with human driver for multi-vehicle complex scenario

In this section, the driver in loop experiment is built to validate the performance when interacting with real human driver in multi-vehicle complex scenario. The experiment scenario is shown in Figure 8, where the traffic environment

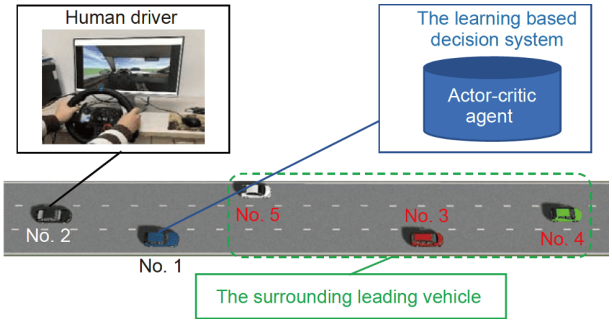

Figure 8 (Color online) The multi-vehicle interactive scenario.

is built by Prescan 8.4 and Matlab 2020a, the black vehicle (No. 2) is controlled by the Logitech G29 steering wheel and pedals. Specifically, this scenario contains three preceding vehicles that decide the tendency of the traffic flow, the blue autonomous vehicle (No. 1) that follows the red vehicle (No. 3) and the black human driver (No. 2). The initial setting of the traffic is shown in Table 3.

Since the low driving speed of front red vehicle (No. 3), the ego vehicle will seek chances to change to the middle lane. This will result in the interaction with human driver and the ego vehicle has to decide in real-time based the human driver's behavior. In this process, the style of human driver and ego vehicle are all moderate. The proposed learning-based decision algorithm is contrast with the common rule-based method when the driving input keeps consistent, thus to evaluate the advantage of the learning-based method. The results are shown in Figure 9, which contains the human

Table 3 Traffic setting

Vehicle ID	Initial position (m)	Initial lane	Initial speed (m/s)	Expected speed (m/s)
Ego vehicle (No. 1)	(5, −3.5)	3	12	13
Human driver (No. 2)	(−18, 0)	2	10	Decided by human
Red vehicle (No. 3)	(32, −3.5)	3	10	10
Green vehicle (No. 4)	(54, 0)	2	13	13
White vehicle (No. 5)	(10, 3.5)	1	15	15

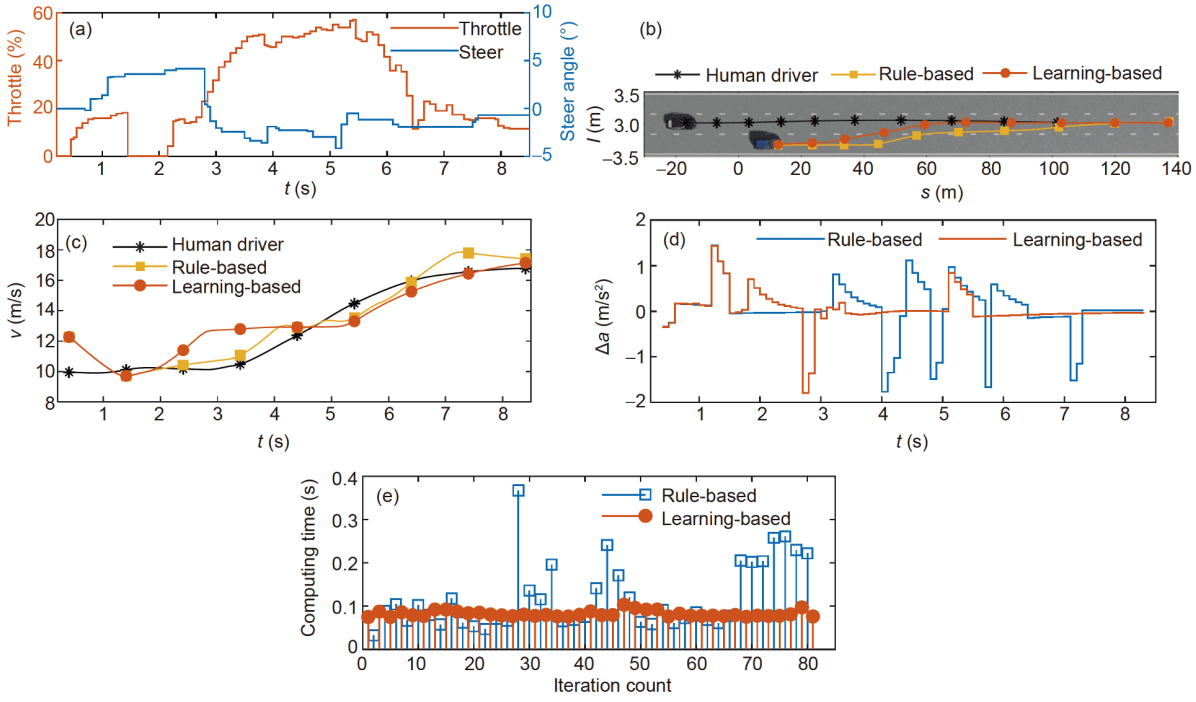

Figure 9 (Color online) The interaction results contrast to the rule-based method. (a) The human driver's throttle and steer signal; (b) path in this process; (c) speed in this process contrast to the existing algorithm; (d) the action increment in this process; (e) the calculated time contrast in this process.

driver's throttle and steer signal, the ego vehicle's decided path and speed profile. The acceleration increment and the computing time are also given in this process. The following is the discussion.

The interaction process can be divided into three phases: the deceleration and car-following phase, the lane-changing and accelerating phase, the lane-keeping phase. For the first phase in 0–2 s, since the lower speed of front red vehicle (No. 3), it will be risky to change lane directly. Then the ego vehicle decelerates to keep the driving safety and attempts to change lane according to the rear human driver's reaction.

When the human driver perceived ego vehicle's lane-changing tendency, it deaccelerates a little to cooperate. Thus, the policy network decides the lane changing behavior and then ego vehicle accelerates to improve the efficiency in the second phase from 2 to 5 s. For the last phase, the human driver also accelerates since the long gap to ego vehicle shown in Figure 9(c). From then on, the ego vehicle and the human vehicle continue the interactive car-following process. The speeds of these two vehicles tend to consistent eventually.

For the rule-based method, the decided behavior is generally corresponding to the learning method. But the driving posture is constantly adjusted to make the reward of each moment always optimal, which results the planned path and speed fluctuate a lot shown in Figure 9(b) and (c). However, the behavior of the learning-based method is pretrained by the actor-critic method and the decided behavior is determined when facing the relevant environment state. The action increment in Figure 9(d) shows that the learning method has a better driving comfort than the rule-based method. In addition, Figure 9(e) reveals that computing time of the learning method is also advantageous and can increase by over 50% in some moments since the policy network is rapid and stable. On the contrary, the rule-based method will hesitate for these complicated moments.

5 Conclusions

In this paper, an actor-critic based learning method for path and speed coupling planning in multi-vehicle complex scenario is proposed. It mainly includes generation of the traversal trajectories by a series of end-points and building of the learning framework. Besides, the policy network is modularized into the lane-changing network and the lane-keeping network to make it have good convergence. The actor-critic based learning method is adopted to make the policy network learn rapidly. Thus, the algorithm can decide an optimal driving behaviour and plan the coupled trajectory in real-time for multi-vehicle complex scenario.

However, the proposed algorithm fairly relies on the pretrained network and lacks the ability to handle some emergency condition that has not been experienced. It remains further study to enhance the policy network with wider applicability.

This work was supported by the Jiangsu Key R&D Plan (Grant No. BE2018124), the National Natural Science Foundation of China (Grant Nos. 51775007 and 51875279), and the Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYCX19_0157).

1 Cheng S, Li L, Chen X, et al. Model-predictive-control-based path tracking controller of autonomous vehicle considering parametric uncertainties and velocity-varying. IEEE Trans Ind Electron, 2020, 1
2 Cesari G, Schildbach G, Carvalho A, et al. Scenario model predictive control for lane change assistance and autonomous driving on highways. IEEE Intell Transp Syst Mag, 2017, 9: 23–35
3 Cheng S, Li L, Liu C Z, et al. Robust LMI-based H-infinite controller integrating AFS and DYC of autonomous vehicles with parametric uncertainties. IEEE Trans Syst Man Cybern Syst, 2020, 1–10
4 Kasper D, Weidl G, Dang T, et al. Object-oriented bayesian networks for detection of lane change maneuvers. IEEE Intell Transp Syst Mag, 2012, 4: 19–31
5 Xie Z W, Zhang Q, Jiang Z N, et al. Robot learning from demonstration for path planning: A review. Sci China Tech Sci, 2020, 63: 1325–1334
6 Ulbrich S, Maurer M. Probabilistic online POMDP decision making for lane changes in fully automated driving. In: Proceedings of the IEEE Conference on Intelligent Transportation Systems. The Hague, 2013. 2063–2067
7 Li L, Ota K, Dong M. Humanlike driving: Empirical decision-making system for autonomous vehicles. IEEE Trans Veh Technol, 2018, 67: 6814–6823
8 Yang D G, Jiang K, Zhao D, et al. Intelligent and connected vehicles: Current status and future perspectives. Sci China Tech Sci, 2018, 61: 1446–1471
9 Lu C, Wang H, Lv C, et al. Learning driver-specific behavior for overtaking: A combined learning framework. IEEE Trans Veh Technol, 2018, 67: 6788–6802
10 Ngai D C K, Yung N H C. A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers. IEEE Trans Intell Transp Syst, 2011, 12: 509–522
11 Noh S, An K. Decision-making framework for automated driving in highway environments. IEEE Trans Intell Transp Syst, 2018, 19: 58–71
12 Huang Z, Chu D, Wu C, et al. Path planning and cooperative control for automated vehicle platoon using hybrid automata. IEEE Trans Intell Transp Syst, 2019, 20: 959–974
13 Feng G, Wang W, Feng J, et al. Modelling and simulation for safe following distance based on vehicle braking process. In: Proceedings of the IEEE Conference on E-business Engineering. Shanghai, 2010. 385–388
14 Ji J, Khajepour A, Melek W W, et al. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. IEEE Trans Veh Technol, 2017, 66: 952–964
15 Wang J F, Zhang Q, Zhang Z Q, et al. Structured trajectory planning of collision-free lane change using the vehicle-driver integration data. Sci China Tech Sci, 2016, 59: 825–831
16 Gonzalez D, Perez J, Milanes V, et al. A review of motion planning techniques for automated vehicles. IEEE Trans Intell Transp Syst, 2016, 17: 1135–1145
17 McNaughton M, Urmson C, Dolan J, et al. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: Proceedings of the IEEE Conference on Robotics and Automation. Shanghai, 2011. 4889–4895
18 Park B, Lee Y C, Han W Y. Trajectory generation method using Bézier spiral curves for high-speed on-road autonomous vehicles. In: Proceedings of the IEEE Conference on Automation Science and Engineering. Taipei, 2014. 927–932
19 Li X, Sun Z, Cao D, et al. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. IEEE/ASME Trans Mechatron, 2016, 21: 740–753
20 Li P, Duan H B. Path planning of unmanned aerial vehicle based on improved gravitational search algorithm. Sci China Tech Sci, 2012, 55: 2712–2719
21 Yu L, Shao X, Yan X. Autonomous overtaking decision making of driverless bus based on deep Q-learning method. In: Proceedings of the IEEE Conference on Robotics and Biomimetics. Macau, 2017
22 Tram T, Jansson A, Grönberg R, et al. Learning negotiating behavior between cars in intersections using deep Q-learning. In: Proceedings of the IEEE Conference on Intelligent Transportation Systems. Maui, 2018
23 Zhu M, Wang X, Wang Y. Human-like autonomous car-following model with deep reinforcement learning. Transpation Res Part C-Emerging Technologies, 2018, 97: 348–368
24 Wang P, Chan C Y, Fortelle A. A reinforcement learning based approach for automated lane change maneuvers. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV). Changshu, 2018. 1379–1384
25 Wnag C Y, Zhao W Z, Xu Z J, et al. Path planning and stability control of collision avoidance system based on active front steering. Sci China Tech Sci, 2017, 60: 1231–1243
26 Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning. New York, 2016
27 Hillenbrand J, Spieker A M, Kroschel K. A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs. IEEE Trans Intell Transp Syst, 2006, 7: 528–540
28 Ward J R, Agamennoni G, Worrall S, et al. Extending Time to Collision for probabilistic reasoning in general traffic scenarios. Transpation Res Part C-Emerging Technologies, 2015, 51: 66–82