文章编号：0253-374X(2021)S1-0132-09

DOI：10.11908/j.issn.0253-374x.22787

基于参数描述的换道场景自动驾驶精确决策学习

张羽翔，何钢磊，李　鑫，刘奇芳，丛岩峰，王玉海

（吉林大学 汽车仿真与控制国家重点实验室，长春 130022）

摘要：为提高车辆驾驶安全性并充分考虑人类驾驶员对于自动驾驶控制系统的接受度，研究并实现了自动驾驶车辆在换道场景下的精确决策学习。汽车自动驾驶不仅需要决策是否换道，还需要决定汽车的具体微观行为，如换道时间和期望加速度的确定等，因此，车道变换的精确决策需使用3个参数来描述，并需要通过强化学习求解。这种基于参数精确决策的合理性体现在两个方面：首先是不同的决策参数值会影响规划的轨迹，如果决策不精确，将产生运动的不确定性；其次是基于真实交通数据(NGSIM)的分析，因为人类换道行为在换道时间和期望加速度上存在显著的差异性，在当前的决策研究中很少被明确考虑。此外，发现NGSIM数据中存在一些潜在的紧急情况，可以通过优化部分决策参数来提升其安全性；在强化学习算法的设计中，动作过程中加入换道时间和期望加速度；奖励函数考虑了安全性、当前驾驶员的意愿和平均人类驾驶风格；问题求解中，自定义了基函数，并通过基于核函数的最小二乘策略迭代强化学习方法学习精确的安全决策行为。仿真结果表明，使用强化学习参数决策可以实现更精确的决策，从而提高安全性能，并可在变道场景中模仿人类驾驶员的行为。

关键词：自动驾驶车辆；驾驶决策；真实交通数据；换道场景

中图分类号：U471.1　　**文献标志码**：A

Precise Decision-Making Learning for Automated Vehicles in Lane-Change Scenario Based on Parameter Description

ZHANG Yuxiang, *HE Ganglei*, *LI Xin*, *LIU Qifang*, *CONG Yanfeng*, *WANG Yuhai*

(State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China.)

Abstract: To promote safety and fully consider human drivers´ acceptance, precise decision-making is realized for automated vehicles under the lane-change scenario in this paper. More specifically, automated vehicles not only decide to change lanes or not but also decide specific microcosmic behaviors, such as lane-change time and expected acceleration. Thus, precise decisions for lane-change are described with three parameters and learned by reinforcement learning. The rationality of such parameter-based precise decisions is shown in two aspects. First, different values of decision parameters will notably influence the planned trajectory, which means other microcosmic behaviors will be a significant uncertainty when they are not precisely decided in the decision-making layer. Secondly, based on the analysis of real traffic data, NGSIM, changeable lane-change time, and expected acceleration are revealed in lane-change behaviors, which is seldom explicitly considered in the decision-making layer of current researches. The decision parameters that include lane-change time and expected acceleration are learned with kernel-based least-squares policy iteration reinforcement learning (KLSPI). Safety, current driver´s willingness, and average human driving style are considered in the reward function. Simulation results demonstrate that using reinforcement learning (RL) to learn decision parameters can realize more precise decisions, promote safety performance, and imitate human drivers´ behaviors in the lane-change scenario.

Key words: automated vehicle; driving decision; real traffic data; lane-change

Automatic control will be fully realized from the decision-making layer to the planning layer in automated vehicles[1-2]. In a higher-performance autopilot driving system, the human driver′s willingness and diverse driving preference should also be taken into consideration to improve the acceptance in either general or individual human drivers[3-5]. However, the human′s decision process is much more

收稿日期：2021-09-25
基金项目：国家自然科学基金青年基金(61803173)；吉林省中青年科技创新领军人才及团队项目(20200301011RQ)
第一作者：张羽翔(1994—)，女，工学博士生，主要研究方向为智能汽车驾驶决策与规划控制。E-mail：yxzhang16@mails.jlu.edu.cn
通信作者：王玉海(1977—)，男，教授，工学博士，主要研究方向为汽车动力学与控制。E-mail：wangyuhai@jlu.edu.cn

complicated to be captured with a mathematical model[6] and has no obvious multi-layer hierarchical framework as same as autopilot driving system[7]. Meanwhile, human behavior is influenced by driving style and skills[8-9], which is different in both the decision-making and planning layers for different drivers.

In previous researches, learning-based methods have obtained great attention to learn human-like driving behaviors[10]. Beyond numerous learning-based methods, reinforcement learning (RL), as a model-free method for sequential control, is widely researched in decision-making[11]. Generally speaking, RL learns with reward signal, which can not only be physical measurements about safety constraints, like the distance between the host vehicle and surrounding vehicles but also can be the correspondence with human′s decision.

As it is hoped that the autopilot driving system can ensure safety performance as well as respect human willingness, using parameters that can be some physical measurements to describe decisions helps to make more precise decisions[12-13] and realize such performance for the following reasons. Regarding vehicle′s motion, despite macroscopical behavior, such as lane change or not, microcosmic behaviors, such as lane-change time and expected acceleration, will also influence a vehicle′s trajectory. Specific microcosmic behaviors actually will fix the trajectory under a smaller range. If macroscopical behaviors are only used to describe a decision, decisions will be much more conservative to keep safe and different from human drivers. Meanwhile, different drivers and the driving situations will reflect microcosmic behaviors. In the driving process, a driver will make decisions with driving experience naturally to predict the influence of these microcosmic behaviors, which will lead to some specific or intelligent behaviors, like accelerate to cut in or reduce the time for lane-change when the distance gap is not sufficient, especially in heavy traffic[14].

The precise decision-making method based on parameter description in the lane-change scenario is investigated in this paper. The overall framework is shown in Fig. 1, which mainly focuses on the RL-based decision-making module and the model predictive control (MPC)-based trajectory planning module. The first contribution is to analyze the rationality of parameter-based precise decisions framework in two parts. First, the influence on trajectory under different lane-change time and expected acceleration are shown basing on the designed trajectory planning controller. Second, a comparison is made to show different decision parameters between real driving profiles. The trajectory planning controller used here is proposed in reference [15] and will be extended in this research to realize precise decision-making. The public dataset of vehicle trajectories used here is from NGSIM, a program developed by the US Federal Highway Administration[9]. The second contribution is to design the learning-based precise decision-making method. The lane-change decision-making problem is modeled as a Markov decision process (MDP) in the parameter decision framework. The reward function considers the human driver′s willingness and safety situation. Thus, the learning promotes the safety performance on the potential dangerous driving scenario and imitates actual human drivers′ behavior in NGSIM. The kernel-based least-squares policy iteration reinforcement learning (KLSPI) that is proposed in reference [16] is used to solve this problem. In the simulation, the driving scenarios are recognized and divided into three sample sets for learning and evaluation. By training, the parameter-based precise decisions are realized, and simulation results are also shown.

1　Analysis of potential performance with parameter decision framework

The trajectory planning controller is described in Fig. 1. Based on this controller, parallel simulations with different decision parameters are conducted to show its influence on the planned trajectory. The real traffic data, NGSIM, are analyzed to find out the range of decision parameters that fit the trajectory planning controller and potential promoted performance in the decision layer.

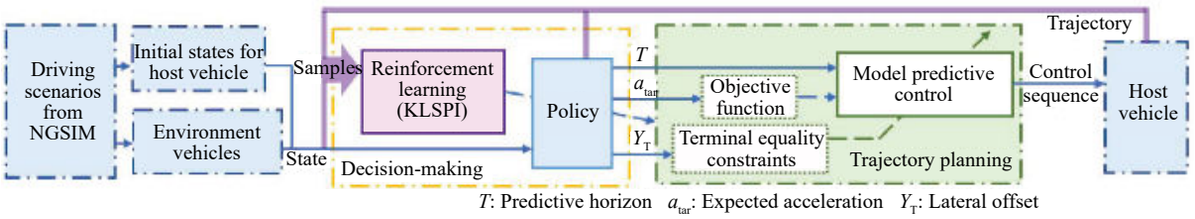


Fig.1 Diagram of framework

1.1 Optimization trajectory planning controller

The trajectory planning controller is designed with nonlinear model predictive control in the parameter decision framework[15]. The main intention is to simplify the implementation of driving behavior by only constraining terminal states to fit with the road segment. For example, on a straight road, as shown in Fig. 2, at the end time step in lane-change and lane-keeping, the vehicle's lateral position should be on the center of the target lane and will stay on the lane (the same heading angle with the target lane, zero yaw rate, and lateral acceleration). Meanwhile, the sequence of control inputs is directly optimized to obtain a feasible trajectory without a reference polynomial function or B-spline trajectory.

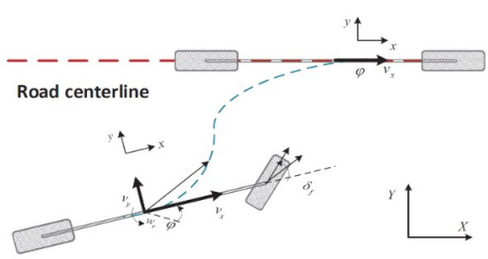


Fig.2 Diagram of the straight-road scenario

The nonlinear motion control model $\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},u)$ is established as:

$$\boldsymbol{f}(\boldsymbol{x},u)=\begin{bmatrix} v_x\cos\varphi-v_y\sin\varphi \\ v_x\sin\varphi+v_y\cos\varphi \\ \omega_r \\ a \\ \dfrac{l_f}{I_z}F_{yf}-\dfrac{l_r}{I_z}F_{yr} \\ \dfrac{1}{M}F_{yf}+\dfrac{1}{M}F_{yr}-v_x\omega_r \end{bmatrix} \tag{1}$$

where: the vector of states is $\boldsymbol{x}=[X, Y, \varphi, v_x, v_y, \omega_r]$, X and Y are the positions, φ is the heading angle, v_x and v_y are the longitudinal and lateral velocity, ω_r is the yaw rate; the vector of control inputs is $\boldsymbol{u}=[a, \delta_f]$, δ_f is the steering-wheel angle, the simple longitudinal dynamics $\dot{v}_x=a$ is considered to simplify the motion control model; M is the mass of the vehicle; I_z is the moment of inertia of the vehicle around the z-axis; l_f and l_r are the distances from the center of gravity (CoG) to the front and rear axles, respectively. The linear tire model is considered, and the tire slip angle of the front wheel α_f and rear-wheel α_r can be linearized with the small slip angle, the lateral tire force F_{yf} and F_{yr} on the front and each rear tires is written as:

$$\begin{aligned} F_{yf}&\approx -C_f\left(\frac{v_y+l_f\omega_r}{v_x}-\delta_f\right), \\ F_{yr}&\approx -C_r\frac{v_y-l_r\omega_r}{v_x}, \end{aligned} \tag{2}$$

where: C_f and C_r are the cornering stiffness values of the front and rear tires, respectively.

The trajectory planning problem in the straight lane-change scenario at decision parameters (Y_T, T, a_{tar}) is formulated after discretization as[15]:

$$\begin{aligned} \min\ & \sum_{k=0}^{T-1}\left(\|u(k)\|_Q^2+\|a_{tar}-a(k)\|_P^2\right) \\ & +\sum_{k=0}^{T-2}\|u(k+1)-u(k)\|_R^2 \\ \text{s.t. } & x(k+1)=x(k)+f(x(k),u(k))\Delta t \\ & u_{min}\leqslant u(k)\leqslant u_{max} \\ & v_y(T)=0,\ \omega_r(T)=0 \\ & \varphi(T)=0,\ Y(T)=Y_T \end{aligned} \tag{3}$$

where: Q, P, R are the weighting matrices; $Y_T\in\{0, L, -L\}$ is the lateral offset to the center of target lane, which is different in lane-keeping and lane-change; L is the distance between two neighboring lanes. The lane-change time T decides the predictive horizon. As have illustrated before, the terminal equality equation ensures that the lane-change behavior is finished in the predictive horizon without a reference trajectory.

1.2 Influence of different decision parameters

Decisions in the lane-change scenario can be precisely described as lane-keeping or lane-change to the right/ left lane in fast/ moderate/ mild mode with acceleration/maintained velocity/ deceleration. The examples represented with decision parameters are listed in Tab. 1. Such a description of the decision can provide more precise and diverse behaviors for decision-making and be more human-like.

Tab.1 Examples of decision description for lane—change at parameter representation

Description	Y_T/m	T/s
Lane keeping	0	1
Right lane—change in fast mode	$-L$	3
Right lane—change in mild mode	$-L$	5
Left lane—change in moderate mode	L	4

The action a_{tar} is not listed, while its value will decide whether to accelerate, maintain velocity or decelerate.

Then, the influence on the planned trajectory is shown with different decision parameters through parallel simulations. In the parallel simulations, the left lane-change ($Y_T=L$) is implemented with changeable decision parameters T and a_{tar} at two initial longitudinal velocities v_x, which involves different trajectory optimization problems as shown in Eq. (3). These optimization problems are solved by sequential quadratic programming (SQP) algorithm using MATLAB.

The trajectories with different action durations $T\in\{T1, T2, T3\}$, expected accelerations $a_{tar}\in\{A1, A2, A3\}$, and initial longitudinal velocities $v_x\in\{V1, V2\}$ are shown in Fig. 3. Here, the initial longitudinal velocity $V1=10$ m/s and $V2=20$ m/s. The action duration $T1=3$ s, $T2=4$ s, and $T3=5$ s. The expected acceleration $A1=-0.5$ m/s^2, $A2=0$, and $A3=0.5$ m/s^2.

As shown in Fig. 3, there is a certain distance gap between the trajectories with different microcosmic decisions, which may cause absolutely different situations when these microcosmic decisions are not considered in the decision layer.

1.3 Analysis of range of decision parameters in NGSIM

Firstly, the parameter-based decisions and the corresponding trajectories after optimization are compared with real traffic trajectories from NGSIM to verify the rationality of parameter-based decision-making. Three typical examples (C_1, C_2, C_3) are shown in Fig. 4. The start time step and end time step of lane-change in each trajectory are recognized with the method in reference [9], which is set as the action duration T. The expected acceleration a_{tar} with an interval of 0.1 is calculated to fit the actual trajectory. The optimized trajectory is compared with the actual trajectory from NGSIM, in which the optimized trajectory has a good fitting performance compared with the actual trajectory. Meanwhile, action duration T and expected acceleration a_{tar} are listed in the right side of Fig. 4. It can be seen that different lane-change maneuvers are executed with different drivers, which is a potential behavior that can be considered in the decision layer.

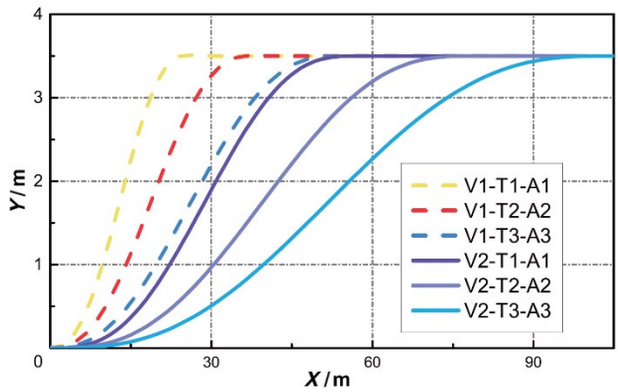


Fig.3 Trajectories at different values of decision parameters

Meanwhile, the ranges of these decision parameters are analyzed with this real traffic dataset. The lane-change trajectories from the NGSIM are selected and labeled with the method in reference [9]. The statistic values in the whole data are listed in Tab. 2. The ranges of expected acceleration and action duration in NGSIM are used as a reference in the parameter-based decision-making problem in the next section.

1.4 Driving scenarios analysis in NGSIM

In free traffic flow, there are eight potential surrounding positions around the host vehicle, which is numbered as shown in Fig. 5. The ranges of different positions are given in Tab. 3. For each labeled host vehicle, the vehicle that is in the range of positions P_1-P_8 will be added into its driving

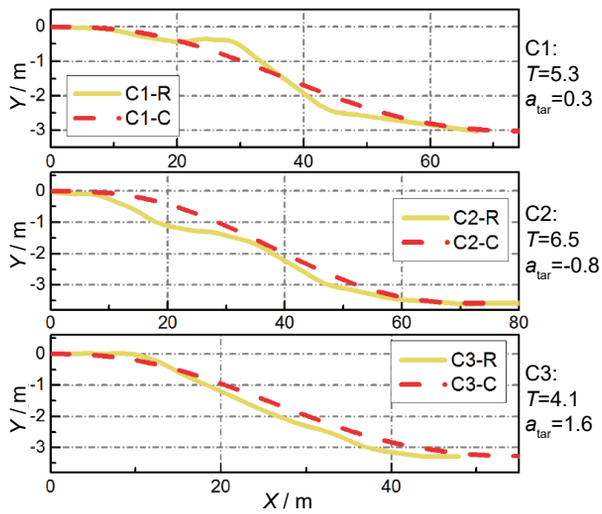


Fig.4 Comparison of actual trajectories from NGSIM (R in legend) and optimized trajectory with optimization trajectory planning controller (C in legend)

Tab.2 Statistic results in NGSIM

Action duration	T_{min}/s	T_{max}/s	$\bar{T}$/s
T/s	2.5	7.5	4.5
Expected acceleration a_{tar}/(m/s^2)	$a_{tar,min}$ −1.1	$a_{tar,max}$ 2.5	$\bar{a}_{tar}$ 0.2

scenario. The ranges of expected acceleration and action duration in NGSIM are used as a reference in the parameter-based decision-making problem in the next section.

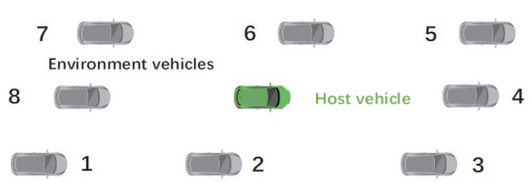


Fig.5 Diagram of host vehicle and its surrounding vehicles

Tab.3 Range of different positions

Position	Range	Position	Range
P_4	T_{Hn}/s∈[0, 3]	P_3,P_5	d_n/m∈[15, 60]
P_2,P_6	d_n/m∈[−15, 15]	P_1,P_7,P_8	d_n/m∈[−15, −60]

d_n and $T_{Hn}=d_n/v_h$ are the distance in the lane direction and timeheadway between the host vehicle and the surrounding vehicles in the position i. $v_h$is the velocity of the host vehicle.

In these driving scenarios, the relative distance between the host vehicle and the surrounding vehicles are calculated on its current lane and target lane when the host vehicle is changing the lane. 15 driving scenarios that the minimal distance is beyond 4 m are assumed as a potential emergency driving situation and can be promoted with a better decision while the driver's intention is also fully respected. Thus, the start time step of lane-change, the action duration for lane-change, and the average acceleration are recorded in these driving scenarios. The learning is done on the driving scenarios to obtain better safety situations without violating the driver's willingness. The driver's willingness includes the intention of changing lane, the action duration, and the average acceleration of lane-change.

2 MDP modeling and RL algorithm

The RL based parameter lane-change decision-making problem is established, and the kernel-based least squares policy iteration algorithm (KLSPI) that is proposed in reference [16] is applied to learn the lane-change decision parameters.

2.1 MDP modeling

The decision process for driving is modeled as a MDP, which contains the design of state space, action space, and reward function. The trajectory planning controller changes the state of the host vehicle with the action selected in the decision layer.

In the design of state, to depict each of the vehicle in the potential positions P_1–P_8, the relative velocity $\Delta v_n(k)=v_n(k)-v_h(k)$, the acceleration $a_n(k)$, the relative distance in the lane direction $d_n(k)$, and the intention of the surrounding vehicles $I_n(k)$ are considered. The intention of the surrounding vehicles $I_n(k)$ is calculated using the method from reference [9]. The subscript n indicates the position of surrounding vehicles while the subscript h indicates the host vehicle. Thus, the state vector can be expressed as:

$$\boldsymbol{s}(k)=(\underbrace{\Delta v_1(k),a_1(k),d_1(k),I_1(k)}_{P_1},\ldots,\underbrace{\Delta v_8(k),a_8(k),d_8(k),I_8(k)}_{P_8})^{\mathrm{T}} \tag{4}$$

In the parameter-based decision framework, the decision is described with parameters, which is the lateral offset, the time of lane-change and expected acceleration, respectively. The action vector is

expressed as:

$$\boldsymbol{a}(k)=(Y_L(k), T(k), a_{\text{tar}}(k))^{\mathrm{T}} \tag{5}$$

where: $Y_L(k)\in\{-L, 0, L\}$ is the target lateral offset, L is the distance between two neighboring lanes; $T(k)\in[3, 6]$ is the time of lane-change or $T(k)=1$ is the time of lane keeping; $a_{\text{tar}}(k)\in[-1, 2]$ is the expected acceleration.

The reward function is designed with the consideration of safety r_s, the intention of diver r_r, and the consistency with all drivers r_c, which can be expressed as:

$$r(k)=k_s r_s(k)+k_r r_r(k)+k_c r_c(k) \tag{6}$$

The safety reward r_s evaluates the safety situation compared with the action taken $r_{s,e}$ and the original situation in NGSIM $r_{s,o}$, which is the driver's willingness considered in this paper. The safety reward r_s is:

$$r_s=r_{s,o}-r_{s,e} \tag{7}$$

The relative distance d_i between the host vehicle and the surrounding vehicles on its current lane and target lane when the host vehicle is changing the lane is calculated during the lane change process. Assuming there are n_{ev} surrounding vehicles, the incremental equation of action taken $r_{s,e}$ and the original situation in NGSIM $r_{s,o}$ are calculated in the same way and expressed as:

$$r_{s,e/o}=\begin{cases} r_{s,e/o}+d_i-d_e & \text{for } d_i<d_e,\ i=1,2,\ldots,n_{ev} \\ r_{s,e/o}-10 & \text{for } d_i<d_c,\ i=1,2,\ldots,n_{ev} \\ r_{s,e/o} & \text{else} \end{cases} \tag{8}$$

where: $d_e=4$ is the emergency distance, $d_c=2$ is the collision distance. The reward for the intention of diver r_r and the consistency with all drivers r_c can be expressed as:

$$r_r=-k_{r,1}(T-T_o)^2-k_{r,2}(a_{tar}-a_{tar,o})^2 \tag{9}$$

$$r_c=-k_{c,1}(T-T_c)^2-k_{c,2}(a_{tar}-a_{tar,c})^2 \tag{10}$$

Here: T_o and T_c are the time for lane-change with the current driver and the average of all drivers, respectively; $a_{tar,o}$ and $a_{tar,c}$ are the average acceleration with the current driver and the average of all drivers.

2.2 KLSPI algorithm

The reinforcement learning algorithm KLSPI that is proposed in reference [16] is used to solve this lane-change decision-making problem. The detailed of this algorithm is not discussed in this paper, only some changes that can better fit this algorithm to the problem and give the pseudocode in Algorithm 1 are explained. The main equations in Algorithm 1 are from reference [16].

Algorithm 1 KLSPI for lane-change decision-making:

1) Collect sample set

$$\{\boldsymbol{s}(k), \boldsymbol{a}(k), r(k), s'(k),\ k=1, 2, \ldots, n\}$$

with random policy and trajectory planning controller.

2) Sparsification: Initialize empty dictionary Dic=$\{\}$.

For k=1 to n, **do**

Assuming that the current dictionary has t features, for feature $m(k)$ calculate

$$\begin{cases} \boldsymbol{c}=\boldsymbol{W}^{-1}\boldsymbol{w}(\boldsymbol{m}(k)) \\ \boldsymbol{\xi}=w_{kk}-\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}(k))c \end{cases} \tag{11}$$

where:

$$[W]_{i,j}=\kappa(m(i), m(j)),\ w_{kk}=\kappa(m(k), m(k))$$

$$\boldsymbol{w}(m(k))=[\kappa(m(1), m(k)), \ldots, \kappa(m(t), m(k))]^{\mathrm{T}}$$

If $\boldsymbol{\xi}<\mu$, **do** Dic = Dic $\cup\, m(k)$

else continue

end if

end for

3) Policy Iteration: Random initialize A, b

Loop for j=1 to maximum iteration n_m

For k=1 to n

Compute A, b, α with

$$\begin{cases} A=A+\boldsymbol{w}(\boldsymbol{m}(k))[\boldsymbol{w}(\boldsymbol{m}(k))-\gamma\boldsymbol{w}(\boldsymbol{m}'(k))]^{\mathrm{T}} \\ b=b+\boldsymbol{w}(\boldsymbol{m}(k))\boldsymbol{r}(k) \\ \alpha=(A)^{-1}b \end{cases} \tag{12}$$

end for

For k=1 to n

Implement policy improvement with

$$\pi'=\arg\max_{a_j'}\tilde{Q}(s', a_j') \tag{13}$$

end for

Until $\sum\|(a_j'-a_{j-1}')\|_2<\mu_e$

4) Output α

In KLSPI, first, all training samples are collected. Secondly, the sparsification procedure is done to obtain features in this sample set that are not evident linear

correlation to form a dictionary for function approximation, which is calculated withby using Eq. (11) and decided by using threshold parameter μ. Thirdly, policy iteration is carried outconducted until the termination threshold μ. Thirdly, policy iteration is carried out until the termination threshold μ_e is satisfied. These iteration equations can be directly found in reference [16]. Finally, the policy is output with the weight vector $\alpha$of estimated state-action value function $\tilde{Q}(s, a)$.

The radial basis function (RBF) network is used as the function approximation to approximate the state-action value function, which can be expressed as:

$$\tilde{Q}(\boldsymbol{s}(k), \boldsymbol{a}(k)) = \sum_{j=1}^{t} \alpha_j \kappa(\boldsymbol{m}(j), \boldsymbol{m}(k)) \quad (14)$$

Feature representation $\boldsymbol{m}(k)$ is customized, which combines the state vector $\boldsymbol{s}(k)$ and the action vector $\boldsymbol{a}(k) = [Y_T, T, a_{\text{tar}}]$ and can be expressed as:

$$\boldsymbol{m}(k) = (\boldsymbol{h}(1)\boldsymbol{s}(k), \boldsymbol{h}(2)\boldsymbol{s}(k), \boldsymbol{h}(3)\boldsymbol{s}(k), T, a_{\text{tar}})^{\text{T}} \quad (15)$$

here $\boldsymbol{h}$ is the activation vector and can be expressed as:

$$\boldsymbol{h} = \begin{cases} (1,0,0) & \text{for } Y_T = -L \\ (0,1,0) & \text{for } Y_T = 0 \\ (0,0,1) & \text{for } Y_T = L \end{cases} \quad (16)$$

The weight vector $\boldsymbol{k}_\varphi$ is manually set to normalize the feature vector with a different range and measure state and action differently. The kernel function can be described as:

$$\kappa(\boldsymbol{m}(k), \boldsymbol{m}(j)) = \mathrm{e}^{-\frac{(\boldsymbol{k}_\varphi(\boldsymbol{m}(k)-\boldsymbol{m}(j)))^2}{2\sigma^2}} \quad (17)$$

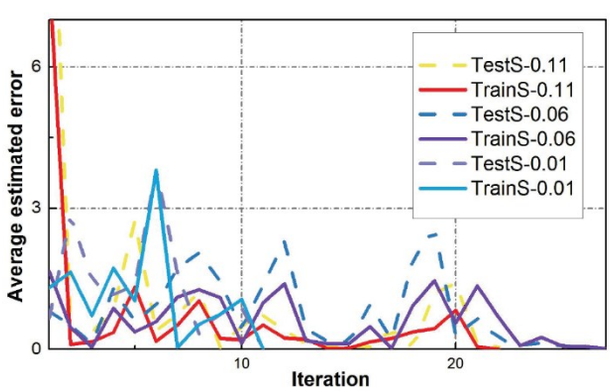
3 Simulation results

The driving scenarios in NGSIM are obtained and divided into the training set, the test set, and the cross-validation set. Sample sets are generated from the training set. The learning is implemented as illustrated in Sections 2 and 3, and influential parameters are decided. Finally, simulation results in the test set and the cross-validation set and promoted performance are verified.

3.1 Sample sets and learning process analysis

In NGSIM, 254 driving scenarios whose host vehicle execute lane-change are selected and are randomly divided into the training set, the test set, and the cross-validation set, which has 214, 30, 10 driving scenarios, respectively. In the training set, the time step that the host vehicle changes lane is found and three decision time steps that before and behind this time step are also considered whose time interval is 0.2 s. We use tThese time steps rather than the whole time are used to sample, because the behavior of other environments could be assumed to be maintained in this short time interval. Lane-change or lane-keeping decisions are both simulated in these decision points to collect sample sets in these training scenarios. Finally, 10 327 samples are obtained.

In the learning process, the threshold parameter μ will be compared with a feature linear correlation with features in the current dictionary to decides whether it will be added to the dictionary, which will be used to approximate the action-state value function. Thus, the threshold parameter μ will influence the dimension of the dictionary and function approximation. The dimension of the dictionary is 572, 170, and 91, respectively, when μ=0.01, 0.06, and 0.11. The average estimated error in the training set and test set are shown in Fig. 6. Eventually, μ= 0.01 is chosen because of its better performance in both the training set and the test set.



The error in each sample is calculated using $e_e = \gamma \tilde{Q}(s', a') + r - \tilde{Q}(s, a)$

Fig.6 Average estimated error in the training set and test set

3.2 Performance validation

First, an emergency scenario whose minimal distance between the host vehicle and the surrounding

vehicles in the whole lane-change process is only 2.5 m in the test set is simulated. As the performance shown in Fig. 7, the control policy after learning tends to change lane late with acceleration and a faster mode to obtain a safer driving situation. Thus the minimal distance in the initial stage is considerable, and the minimal distance in the whole lane-change process has increased to 3.7 m, which promotes the safety performance and retains the driver′s intention to change lane rather than keep the lane. A better-performing behavior but also acceptable for still executing the human driver′s willingness is realized.

In the cross-validation set, a common scenario is simulated, and the performance is shown in Fig. 8. During diving, the control policy after learning change lane in a shorter time, which maintains the performance in both safety, velocity control, and finishing lane-change task. The maintained performance in the cross-validation set verifies the generalization performance of the learning results.

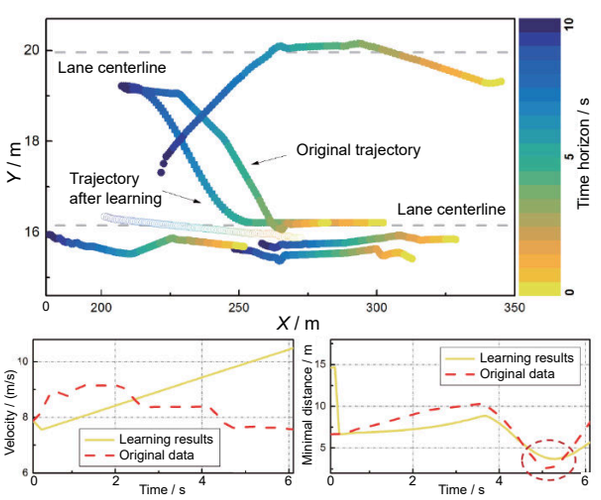


Fig.7　Results for an emergency scenario in test set

4　Conclusions and future works

Microcosmic behaviors, such as lane-change time and expected acceleration, in precise decision-making not only influence trajectory but also differ in drivers, which are shown in NGSIM but seldom considered in the decision layer. As emergency driving scenarios exist in NGSIM, a learning-based parameter decision-making method for automated vehicles to learn the precise decision-making has been

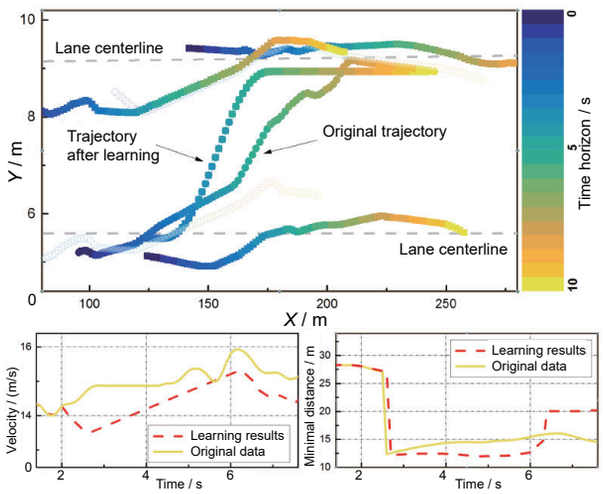


Fig.8　Results for a common scenario in cross-validation set

investigated, which can balance safety and human driving willingness. The lane-change time and expected acceleration are added to the action space. Safety, the current driver′s willingness, and the average human driving style are considered in the reward function. After training by KLSPI with driving scenarios in NGSIM, precise decision-making is realized. Safety performance is promoted in an emergency lane-change scenario of the test set, which indicates a better-performing behavior and is acceptable for still executing the human driver′s willingness. Safety performance is maintained in the cross-validation set, which verifies its generalization performance of the learning results.

In the further, other deep reinforcement learning methods will be explored in more complex and changeable driving scenarios.

Reference:

[1] PADEN B, ČÁP M, YONG S Z, *et al*. A survey of motion planning and control techniques for self-driving urban vehicles [J]. IEEE Transactions on Intelligent Vehicles, 2016, 1(1): 33.

[2] LI X H, SUN Z P, CAO D P, *et al*. Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications [J]. IEEE/ASME Transactions on mechatronics, 2015, 21(2): 740.

[3] GUO C Z, KIDONO K, TERASHIMA R, *et al*. Toward human-like behavior generation in urban environment based on Markov decision process with hybrid potential maps[C]// 2018

IEEE Intelligent Vehicles Symposium (IV). Changshu: IEEE, 2018: 2209.

[4] CHU H Q, GUO L L, YAN Y J, *et al*. Self-learning optimal cruise control based on individual car-following style[J], IEEE Transactions on Intelligent Transportation Systems, 2020, 99: 1.

[5] GINDELE T, BRECHTEL S, DILLMANN R, *et al*. Learning driver behavior models from traffic observations for decision making and planning [J]. IEEE Intelligent Transportation Systems Magazine, 2015, 7(1): 69.

[6] MARTINEZ C M, HEUCKE M, WANG F Y, *et al*. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(3): 666.

[7] GONZÁLEZ D, PÉREZ J, MILANÉS V, *et al*. A review of motion planning techniques for automated vehicles [J]. IEEE Trans. Intelligent Transportation Systems, 2016, 17(4): 1135.

[8] VALLON C, ERCAN Z, CARVALHO A, *et al*. A machine learning approach for personalized autonomous lane change initiation and control [C]// 2017 IEEE Intelligent Vehicles Symposium (IV). Los Angeles: IEEE, 2017: 1590.

[9] HE G L, LI X, LYU Y, *et al*. Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks [C]// 2019 Chinese Automation Congress (CAC), Hangzhou: IEEE, 2019: 2646.

[10] TAN Y V, ELLIOTT M R, FLANNAGAN C A C, *et al*. Development of a real-time prediction model of driver behavior at intersections using kinematic time series data [J]. Accident Analysis & Prevention, 2017, 106: 428.

[11] YOU C X, LU J B, FILEV D, *et al*. Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning [C]// 2018 IEEE Intelligent Vehicles Symposium (IV). Changshu: IEEE, 2018: 1227.

[12] SHALEV-SHWARTZ S, SHAMMAH S, SHASHUA A, *et al*. On a formal model of safe and scalable self-driving cars [DB/OL]. arXiv: 1708.06374, 2017. https://doi. org/10.48550/arXiv.1708.06374.

[13] ZHANG Y X, GAO B Z, GUO L L, *et al*. Adaptive decision-making for automated vehicles under roundabout scenarios using optimization embedded reinforcement learning [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 99: 1.

[14] ARIKERE A, YANG D, KLOMP M, *et al*. Integrated evasive manoeuvre assist for collision mitigation with oncoming vehicles[J]. Vehicle System Dynamics, 2018, 56(10): 1.

[15] ZHANG Y X, GAO B Z, GUO L L, *et al*. A novel trajectory planning method for automated vehicles under parameter decision framework[J]. IEEE Access, 2019, 7: 88264.

[16] XU X, HU D W, LU X C, *et al*. Kernel-based least squares policy iteration for reinforcement learning [J]. IEEE Transactions on Neural Networks, 2007, 18(4): 973.