2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Research on Intelligent Merging Decision-making of Unmanned Vehicles Based on Reinforcement Learning

Xue-mei Chen, Qiang Zhang, Zhen-hua Zhang, Ge-meng Liu, Jian-wei Gong, Ching-Yao Chan

Abstract—**The decision-making model of merging behavior is one of the key technologies of unmanned vehicles. In order to solve the problem of unmanned vehicles' merging decision-making, this paper presents a merging strategy based on Least-squares Policy Iteration (LSPI) algorithm, and selects the basis function which includes reciprocal of TTC, relative distance and relative speed to represent state space and discretizes action space. This study synthetically takes consideration of safety, the success of the task, the merging efficiency and comfort in setting reward function, compares the Q-learning with LSPI algorithm, and verifies its adaptability by using NGSIM data. The algorithm can ultimately achieve a success rate of 86%. This research can provide theoretic support and technical basis for the merging decision-making of unmanned vehicles.**

I. Introduction

With the in-depth development of the major research project "cognitive computing of audiovisual information" by the National Natural Science Foundation of China, the researches on technologies of unmanned vehicles have made great progress in China, which can meet the requirements of low-speed driving in a simple urban environment, and high-speed driving on inter-city highway. However, driving in real urban environments still faces many problems [1]. One of the key issues is the merging decision-making, which restricts the current unmanned vehicles' development [2].

Carnegie Mellon University [3] develops an autonomous vehicle "BOSS" which is built on the corresponding driving behavior in real time based on given knowledge and rules. The Talos unmanned vehicle moderated by MIT [4] develops a decision-making task by the navigation module and calculates the position of the next target point in real time. Ulbrich et al. [5] use some of the observability Markov processes to obtain lane-changing decisions in discrete state space. Sharifzadeh et al. [6] use deep reinforcement learning based on depth Q network to set up overtaking mode on the expressway, but the simulation of the scene is relatively simple. Li et al. [7] propose the concept of "driving brain" to analyze and formalize the cognitive ability of unmanned vehicles based on human brain processing information process, and develop the adaptive ability of decision-making in complex environment. Xu et al. [8] ensure the unmanned vehicle to avoid obstacles in the continuous state space based on the approximate strategy iterative KLSPI algorithm, but there is still limiting factor in its environment adaptive behavioral decision algorithm.

To realize the safety, efficiency and comfort in merging decision-making, this paper proposes LSPI algorithm to study the merging strategy of unmanned vehicles under continuous state space and improves the unmanned vehicles' environmental adaptability.

II. Methodology

A. Reinforcement Learning

Reinforcement learning is an interactive learning method that is tied to both supervised learning and unsupervised learning as one of the branches of machine learning. Its main features are trial and error search and delayed return [9]. The process is as follows:

(1) Agent aware of the current state of the environment S_t;

(2) According to the current state St and reward function R_t, choose an action A_t and execute;

(3) As the action A_t affects the environment, the environment shifts to the new state S_{t+1} and gives a new reward R_{t+1};

(4) The Agent calculates the return value based on feedback from the environment and updates the internal strategy.

The principle of Agent selection action is to maximize the delayed return. The traditional Q-learning [10] is to set the appropriate reward function in the predetermined state space and the action space to update the Q value table until the Q value table converges.

B. Least-squares Policy Iteration (LSPI) algorithm

The LSPI algorithm uses a set of linear basis functions ϕ_i to approximate the value function. Starting from any initial strategy π_0, the LSTD-Q algorithm is used to evaluate the current strategy in each round of iteration l (l> 0), the Q-value function Q_l^π is calculated, and then is employed to improve the strategy by $Q_{l+1}^\pi(x) = \underset{a}{\operatorname{argmax}}\, Q_l^\pi(s,a)$ until the algorithm eventually converges to the optimal strategy π^*. The value function approximate of LSPI consists of linear weights of k basis functions (features):

$$\hat{Q}^\pi(s,a,w) = \sum_{i=1}^{k} \phi_i(s,a) w_i = \phi(s,a)^T w^\pi \tag{1}$$

w is a parameter (weight) vector. The accuracy and generalization ability of LSPI algorithm are closely related to the selection of the basis function. Generally, they are selected based on knowledge of experience, usually with radial basis functions, polynomial basis functions etc. Because $k \ll |S||A|$, the above system becomes an over-constrained system.

$$\mathbf{\Phi}\omega \approx \mathcal{R} + \gamma \mathbf{P}^\pi \mathbf{\Phi}\omega \tag{2}$$

* Research supported by the Nation Natural Science Foundation of China (No. 51705021), (No. U1764261), (No.91420203).

Xue-mei Chen (corresponding author) is with School of Mechanical Engineering, Beijing Institute of Technology. (corresponding author to provide phone: +86 010 68914582; Email: chenxue781@126.com)

Qiang Zhang (the author) is with School of Mechanical Engineering, Beijing Institute of Technology. (Email: wnclyzq08@126.com)

$$(\boldsymbol{\Phi} - \gamma \mathbf{P}^{\pi} \boldsymbol{\Phi})\omega \approx \mathcal{R} \tag{3}$$

$\boldsymbol{\Phi}$ is a matrix of size $(|S||A| \times k)$. A set of w^{π} is found to make a fixed point in the value function space. That is to say, at the fixed point, the parameter of the operation value function $\mathrm{Q}^{\pi} = \boldsymbol{\Phi}\omega^{\pi}$ will not be updated in the gradient direction of entropy regularization. Assume that the columns of Φ are linearly independent, then:

$$\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{\Phi})'\boldsymbol{\Phi}^{\boldsymbol{T}}(\mathcal{R} + \gamma \mathbf{P}^{\pi}\boldsymbol{\Phi}\omega^{\pi}) = \boldsymbol{\Phi}\omega^{\pi} \tag{4}$$

$$\Rightarrow \omega^{\pi} = (\boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{\Phi} - \gamma \mathbf{P}^{\pi}\boldsymbol{\Phi}))^{-\mathbf{1}}\boldsymbol{\Phi}^{\boldsymbol{T}}\mathcal{R} \tag{5}$$

This is the standard fixed-point approximation method for linear-valued functions, which is directed to action-value functions, not state-value functions. Therefore, the solution of LSPI algorithm is:

$$\begin{cases} w^{\pi^{t}} = (\boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{\Phi} - \gamma \mathbf{P}^{\pi}\boldsymbol{\Phi}))^{-\mathbf{1}}\boldsymbol{\Phi}^{\boldsymbol{T}}\mathcal{R} = A^{-1}b \\ \pi^{t+1}(s) = \arg\max_{a \in A} \hat{Q}(s,a,w) = \arg\max_{a \in A} \phi(s,a)^{T} w^{\pi^{t}} \end{cases} \tag{6}$$

III. JOINT VIRTUAL SIMULATION PLATFORM BASED ON PRESCAN/VISSIM AND DATA PREPROCESSING

In developing and testing the adaptive merging strategy under the unmanned vehicle environment, this study builds a joint simulation platform based on traffic flow simulation software Vissim and unmanned vehicle simulation software PreScan.

A. Modeling process of PreScan

The platform PreScan is connected with MATLAB/ Simulink to control unmanned vehicles. The modeling process of PreScan is mainly composed of four steps: building the scene, building up the sensor model, adding control system and running simulation experiment. The workflow and data transmission are shown in Fig. 1.

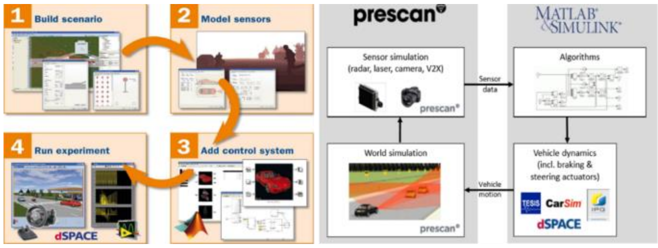


Fig. 1. PreScan workflow and data transfer diagram

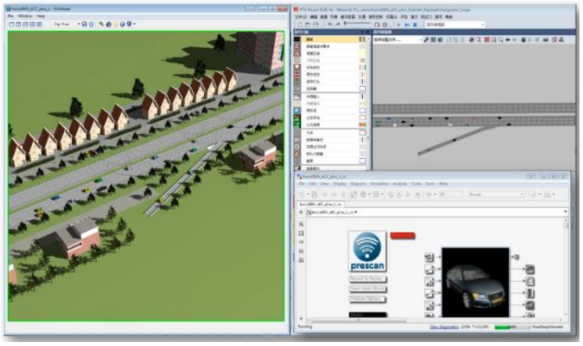


Fig. 2. Virtual traffic scene

B. Joint simulation platform construction

In this paper, Vissim is used to collect the movement information of vehicles and extract the useful behavioral information of the unmanned vehicles from PreScan. The vehicles in the Vissim simulation environment can make real-time feedback on the driving behaviors of the unmanned vehicles. The platform structure is shown in Fig. 3.

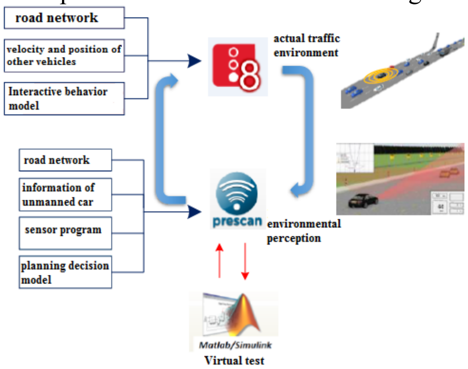
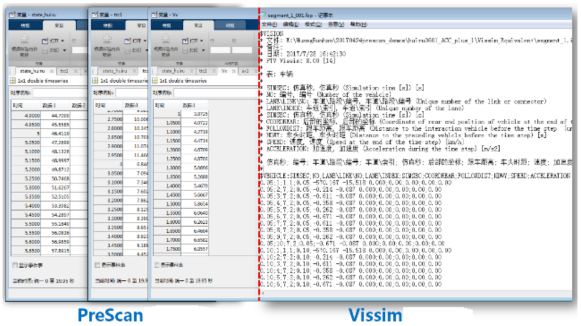


Fig. 3. Vissim + PreScan test platform

The operating status data of the unmanned vehicle and the vehicle data of the surrounding environment are extracted and the simulation processes are evaluated according to safety and efficiency, and the simulation results are shown in Fig. 4.



变量 - state_huiru

1x1 double timeseries

时间	数据1	数据2
4.9000	44.7089	
4.9500	45.5585	
5	46.4119	
5.0500	47.2689	
5.1000	48.1326	
5.1500	48.9997	
5.2000	49.8712	
5.2500	50.7468	
5.3000	51.6267	
5.3500	52.5105	
5.4000	53.3982	
5.4500	54.2897	
5.5000	55.1848	
5.5500	56.0836	
5.6000	56.9859	
5.6500	57.8915	

当前时间: 统一 0 至 19.95 秒

变量 - ttc1

1x1 double timeseries

时间	数据1
2.7500	10.006
2.8000	10.345
2.8500	10.701
2.9000	11.074
2.9500	11.464
3	6.844
3.0500	7.094
3.1000	7.346
3.1500	7.602
3.2000	7.862
3.2500	8.125
3.3000	8.390
3.3500	8.657
3.4000	8.922
3.4500	9.186
3.5000	9.453

变量 - Vx

1x1 double timeseries

时间	数据1
1	3.8725
1.0500	4.0722
1.1000	4.2719
1.1500	4.4714
1.2000	4.6709
1.2500	4.8703
1.3000	5.0696
1.3500	5.2688
1.4000	5.4679
1.4500	5.6667
1.5000	5.8654
1.5500	6.0640
1.6000	6.2623
1.6500	6.4604
1.7000	6.6582
1.7500	6.8557

当前时间: 统一 0 至 19.95 秒

segment_1_001.fzp - 记事本

```
$VISION
* 文件: E:\HuangRunhan\20170424prescan_demos\huiru0601_ACC_plus_1\Vissim_Equivalent\segment_1.
* 备注:
* 日期: 2017/7/28 16:42:30
* PTV Vissim: 8.00 [14]
*
* 表: 车辆
*
* SIMSEC: 仿真秒, 仿真秒 (Simulation time [s]) [s]
* NO: 编号, 编号 (Number of the vehicle)
* LANE\LINK\NO: 车道\路段\编号, 车道\路段\编号 (Unique number of the link or connector)
* LANE\INDEX: 车道\索引, 车道\索引 (Unique number of the lane)
* SIMSEC: 仿真秒, 仿真秒 (Simulation time [s]) [s]
* COORDREAR: 后部的坐标, 后部的坐标 (Coordinate of rear end position of vehicle at the end of t
* FOLLOWDIST: 跟车距离, 跟车距离 (Distance to the interaction vehicle before the time step  (un
* HDWY: 车头时距, 车头时距 (Distance to the preceding vehicle before the time step) [s]
* SPEED: 速度, 速度 (Speed at the end of the time step) [km/h]
* ACCELERATION: 加速度, 加速度 (Acceleration during the time step) [m/s2]
*
* 仿真秒; 编号; 车道\路段\编号; 车道\索引; 仿真秒; 后部的坐标; 跟车距离; 车头时距; 速度; 加速度
*
$VEHICLE:SIMSEC;NO;LANE\LINK\NO;LANE\INDEX;SIMSEC;COORDREAR;FOLLOWDIST;HDWY;SPEED;ACCELERATION
0.05;1;1;1;0.05;-570.167 -15.518 0.000;0.00;0.00;0.00;0.00
0.05;2;7;2;0.05;-0.214 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;3;7;2;0.05;-0.611 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;4;7;2;0.05;-0.358 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;5;7;2;0.05;-0.262 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;6;7;2;0.05;-0.671 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;7;7;2;0.05;-0.611 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;8;7;2;0.05;-0.358 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;9;7;2;0.05;-0.262 -0.087 0.000;0.00;0.00;0.00;0.00
0.05;10;7;2;0.05;-0.671 -0.087 0.000;0.00;0.00;0.00;0.00
0.10;1;1;1;0.10;-570.167 -15.518 0.000;0.00;0.00;0.00;0.00
0.10;2;7;2;0.10;-0.214 -0.087 0.000;0.00;0.00;0.00;0.00
0.10;3;7;2;0.10;-0.611 -0.087 0.000;0.00;0.00;0.00;0.00
0.10;4;7;2;0.10;-0.358 -0.087 0.000;0.00;0.00;0.00;0.00
0.10;5;7;2;0.10;-0.262 -0.087 0.000;0.00;0.00;0.00;0.00
0.10;6;7;2;0.10;-0.671 -0.087 0.000;0.00;0.00;0.00;0.00
```

Fig. 4. Simulation data output

C. Data acquisition and preprocessing

The microscopic driving data under the condition of steady flow (no queuing phenomenon) on the North Third Ring Road in Beijing have been recorded using the camera, as shown in Fig.5. The data are then extracted and analyzed for result evaluation.

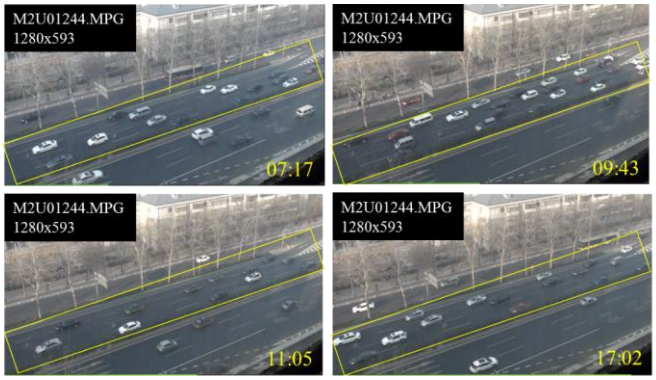


Fig.5. Merging scene in actual road

The collected field traffic data are employed to analyze the characteristics of traffic flow, calibrate the simulation parameters, and make traffic simulation closer to the real environment. Table I is the part of the trajectory data for merging vehicle.

Table I the part of trajectory data

Time(s)	X-axis (m)	Y-axis(m)	Speed(m/s)	Acceleration(m/s^2)
0.1	18.00	211.65	9.84	-0.21
0.2	17.88	212.63	9.82	0.06
0.3	17.77	213.61	9.83	0.04
0.4	17.65	214.60	9.83	0.06
0.5	17.53	215.58	9.83	-0.15
0.6	17.42	216.57	9.76	-0.99
0.7	17.30	217.55	9.61	-2.04

Fig. 6 shows the US101 observation section and Table II shows the US101 data (the unmanned car is in lane 6 of Fig. 6).

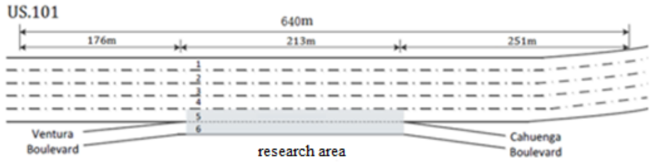


Fig. 6. US101 observation section

Table II US101 data

X(m)	Y(m)	V(m/s)	a(m/s^2)	D(m)	TTC(s)
59.17	859.3	55.18	9.79	71.6	1.30
58.84	864.9	56.08	8.90	71.4	1.27
58.43	870.5	56.82	6.17	71.1	1.25
57.93	876.3	57.38	4.33	70.9	1.24
57.44	882.0	57.87	4.90	70.7	1.22
56.94	887.8	58.44	6.72	129.8	2.22

We also employ the trajectory data acquired from the US101 dataset for result evaluation. The data are extracted from custom video analysis software with systematic errors and calibration errors. In order to reduce the noise of the data and obtain the smoother and more precise trajectory data, we use the symmetric exponential moving average method [11] to deal with the original data.

$$\begin{cases} x'_a(t_i) = \sum_{k=i-D}^{i+D} x_a(t_k) e^{-|i-k|/\Delta} \,/\, \sum_{k=i-D}^{i+D} e^{-|i-k|/\Delta} \\ D = min\{3\Delta, i-1, N_a - i\} \end{cases} \quad (7)$$

Where $x_a(t_i)$ and $x'_a(t_i)$ are the original measured values and the smoothed values of the vehicle at time t_i respectively, $i = 1 \cdots N_a$ and N_a represent the point position of trajectory data. D is the smoothing window width considering the boundary conditions, Δ= T/dt is the smoothing width, Δ = 10T because the track data dt is 0.1 second. Processing results are shown in Fig. 7:

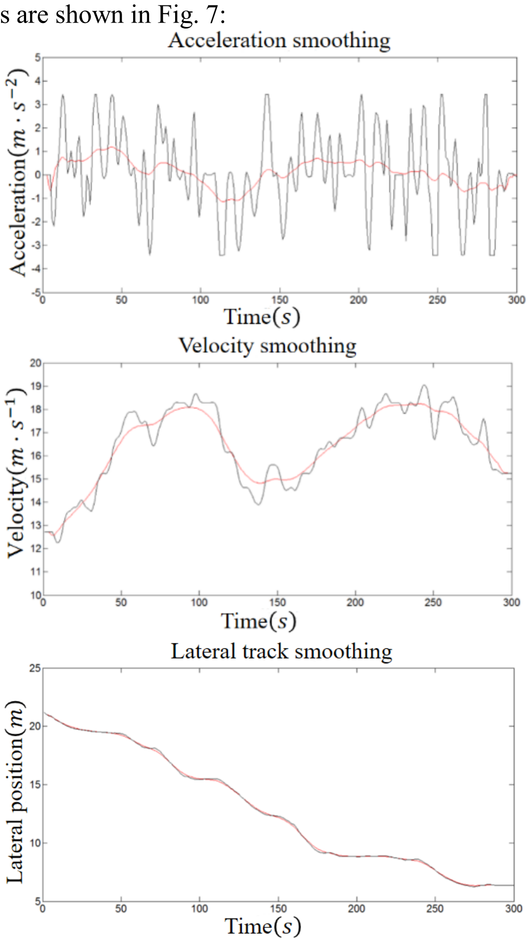


Fig. 7. Smoothing the data illustration

IV. Intelligent Merging Decision Modeling Based on LSPI Algorithm

In this paper, the least-squares policy iteration, LSPI, is employed to derive the optimal policy for merging process. The core of this paper is to replace the value function in Q-learning with the linear function approximation.

The LSPI algorithm optimizes the merging policy in a small number of iterations through the selection of basis functions and samples generated randomly. Typical merging scenario diagram is shown in Fig. 8.

A. State space

The MDP model of its unit merging system could be applied to any candidate clearance. When the unmanned vehicle Car-0 enters the acceleration lane, its own state information was obtained through the GPS sensor and the state information of the surrounding vehicles was perceived through AIR sensor. $\{v_0, x_0, y_0\}$, $\{v_1, x_1, y_1\}$and$\{v_2, x_2, y_2\}$ represent their velocity, longitudinal position, and lateral position respectively. The horizontal coordinate of the target lane centerline is 3.5 m. Unmanned vehicles starting position coordinates (0,0), under the premise of safety, the success sign of merging is the horizontal coordinates y_0 equals 3.5. This article assumes that the target lane vehicles do not change lane, and keep tracking during the whole merging process. Therefore, $y_1 = y_2 = 3.5$ remains unchanged.

In summary, the unit state space of LSPI algorithm is described as a seven-dimensional vector space$(x_0\ y_0\ v_0\ x_1\ v_1\ x_2\ v_2)$, where $(x_0\ y_0\ v_0)$is the position coordinate and speed information of the imported vehicle, $(x_1\ v_1\ x_2\ v_2)$represents the vertical position coordinate and velocity information of the target lane in the process of simulation of the leading vehicle and the succeeding vehicle.

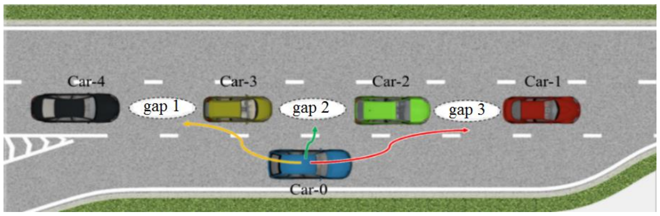


Fig. 8. Typical merging scene

B. *Determination of basis function*

Compared with Q learning, which is a classical reinforcement learning algorithm, the LSPI algorithm uses a basis function (BF) to characterize the state. Therefore, this paper selects a part of unmanned vehicle's own state information as a part of the basis function. In summary, the basis function of the driverless vehicle merging policy contains 14 dimensions, which can be calculated according to the state space variables, as shown in Table III.

Table III Determination of basis function

factors	sign	unit	Dimensions
Reciprocal of collision time	$\left[\frac{1}{\mathrm{ttc}_{10}}, \frac{1}{\mathrm{ttc}_{12}}, \frac{1}{\mathrm{ttc}_{20}}\right]$	1/s	3
Front time	$[gt_{10}, gt_{12}, gt_{02}]$	s	3
Relative distance	$[dx_{10}, dx_{12}, dx_{02}]$	m	3
Relative velocity	$[dv_{10}, dv_{12}, dv_{02}]$	m/s	3
Movement state	$[y_0, v_0]$	[m, m/s]	2

C. *Action space*

The setting of the motion space includes lateral lane change decisions and longitudinal acceleration decisions. When the horizontal decision is 0, the target lane is speeding lane and keep straight. When the transverse decision is 1, the lane change is performed, then the unmanned vehicle performs the lane changing action according to the reference lane change track.

To simplify the motion space of the model and ensure the comfort, the vertical acceleration is discretized into five action values of rapid deceleration, deceleration, constant velocity, acceleration and rapid acceleration, corresponding to (-4, -2,0,2,4). As a result, the action space contains 10 actions, as shown in Table IV.

Table IV Setting of action space

Horizontal action	Vertical action	Action space	Horizontal action	Vertical action	Action space
Lane keeping	Rapid deceleration	(-4,0)	Lane changing	Rapid deceleration	(-4,1)
Lane keeping	Deceleration	(-2,0)	Lane changing	Deceleration	(-2,1)
	Constant velocity	(0,0)		Constant velocity	(0,1)
	Acceleration	(2,0)		Acceleration	(2,1)
	Rapid acceleration	(4,0)		Rapid acceleration	(4,1)

D. *Reward function setting*

Considering the real-world urban expressway ramp-in behavior, we must observe the traffic rules, and also take into account the safety and comfort and efficiency of the driving process. Thus, this article takes speed and comfort into the evaluation index. Based on the above considerations, this paper establishes a linear weighted comprehensive reward model, as shown in the formula.

$$\mathrm{R}(s,a) = \mu_1 R_{safety}(s,a) + \mu_2 R_{task}(s,a) + \mu_3 R_{time}(s,a) + \mu_4 R_{rule}(s,a) + \mu_5 R_{comfort}(s,a) \quad (8)$$

$R_{safety}(s,a)$ represents a safety reward, $R_{task}(s,a)$is the reward indicating success or failure of the task $R_{time}(s,a)$ refers to the merging efficiency reward $R_{rule}(s,a)$ represents the reward of limit speed $R_{comfort}(s,a)$represents comfort reward. The reward value of each index needs to be normalized and the reward value is also mapped to (0,1).

1) Security rewards function

When an unmanned vehicle is driving in an acceleration lane, the collision problem is not taken into consideration. In this case, it is in a safe area and therefore the value of the safety reward function equals to zero. When the unmanned vehicle is in the state of another situation, that is, when coincident with the lane line, the safety of the target lane vehicle is started to be considered. At this time, there are three states: “easy to collide”, “collided or failed to merge” and “safe”.

$$R_{safety}(s,a) = \begin{cases} \frac{\mathrm{dis} - \min(dx_{10}, dx_{02})}{\mathrm{dis}} & 5 < \min(dx_{10}, dx_{02}) < \mathrm{dis} \text{ easy to collide} \\ 1 & \min(dx_{10}, dx_{02}) \leq 5 \text{ collided or failed to merge} \\ 0 & \text{safe} \end{cases}$$
(9)

Among them, dis is the safety threshold of the relative distance between the merging vehicle and the front vehicle in the target lane, by analyzing the real driving data, setting the dis as 5 meters.

2) Mission success rewards function

The task reward is the reward value that is fed back when the task is completed safely and efficiently. there are:

$$R_{task}(s,a) = \begin{cases} 1 & dx_{10} > \text{dis}_1, dx_{02} > \text{dis}_1, y_0 \geq 3 \text{ successfully merged} \\ & 0 \end{cases} \quad (10)$$

dis_1 is threshold of the safety distance. When the unmanned vehicle has merged successfully, a larger positive reward would be given, so the weight μ_2 is a large positive value.

3) Merging efficiency reward function

Merging behavior requires to complete the merging task of lane change efficiently with space constraints. Therefore, this article designs the merging efficiency reward value according to the timeliness of the completion of the task.

$$R_{time}(s,a) = \begin{cases} \frac{65-step}{65} & step \leq 100 \end{cases} \quad (11)$$

$step$ represents the current cycle. According to the analysis of the data of the real merging behavior, merging tasks are completed between 2.8 seconds to 13.8 seconds, and the average lane change time is 6.5 seconds; if the unmanned vehicles merged successfully within 6.5 seconds, then give a positive reward.

4) Speed limit reward function

In the driving process, we need to obey the traffic laws and regulations, this paper introduces the speed limit reward value to regulate the speed of unmanned vehicles within a reasonable range.

$$R_{rule}(s,a) = \begin{cases} \frac{v_{limit}-v_0}{v_{limit}} & v_0 > v_{limit} \text{ overspeed} \\ 0 & \end{cases} \quad (12)$$

5) Comfort reward function

Comfort during driving includes both longitudinal and lateral acceleration and impact characterization indicators. Therefore, the comfort reward value mainly considers longitudinal acceleration changes and is normalized as follows.

$$R_{comfort}(s,a) = \frac{|\Delta a|}{|a_{max}-a_{min}|} \quad (13)$$

Where $|\Delta a|$ represents the longitudinal acceleration difference of two cycles, a_{max} represents the maximum acceleration, and a_{min} represents the maximum deceleration. When the acceleration difference is 0, the reward value is zero; in other cases, the acceleration changes continuously, driving comfort is reduced, negative reward is given.

V. VERIFICATION

After analyzing the unit merging system, the state space of Q-learning in this paper is set as five-dimensional vector:

$$(dx_{10}\ dv_{10}\ dx_{12}\ dv_{12}\ y_0)$$

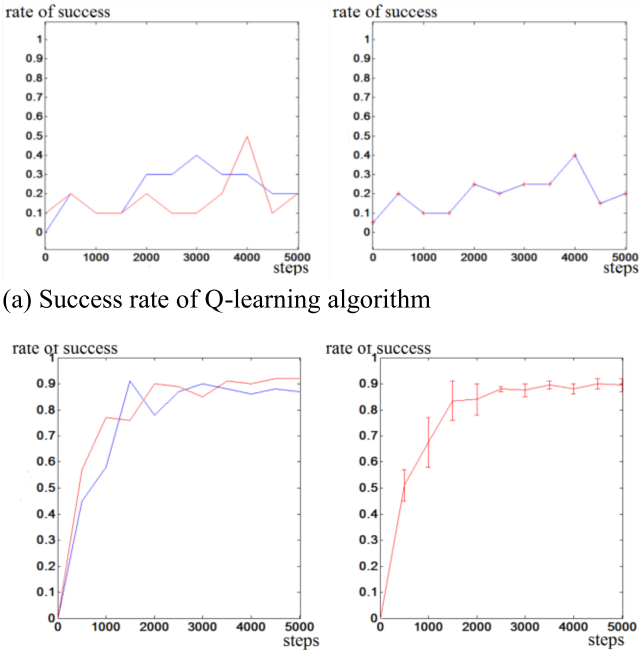
Where, $dx_{10}$and dv_{10} represent the relative distance and the relative speed of the unmanned vehicle Car-0 and the target lane Car-2745 respectively; dx_{12} and dv_{12} represent the relative distance and relative speed between Car-0 and Car-2745; y_0 is the horizontal position of the unmanned vehicle.

Comparison results with Q-learning are shown in Table V. Dispersed state space restricts the ability of generalization and promotion of Q-learning, and the sample can't be fully learnt, resulting in the loss of information.

Table V Comparison of simulation results between LSPI and Q-learning

Algorithm	Sample	Time s
Q-learning	415800	1192
LSPI	50352	403

In the co-simulation platform, a typical 3-slot inbound scenario is designed for policy optimization training. The algorithm agent continuously interacts with the simulation environment to explore and optimize the merging policy. To demonstrate the process of policy optimization more intuitively, two sets of experiments are carried out in this paper. The comparison of the results is shown in Fig. 9(a) and (b).



(a) Success rate of Q-learning algorithm

(b) Success rate of LSPI algorithm

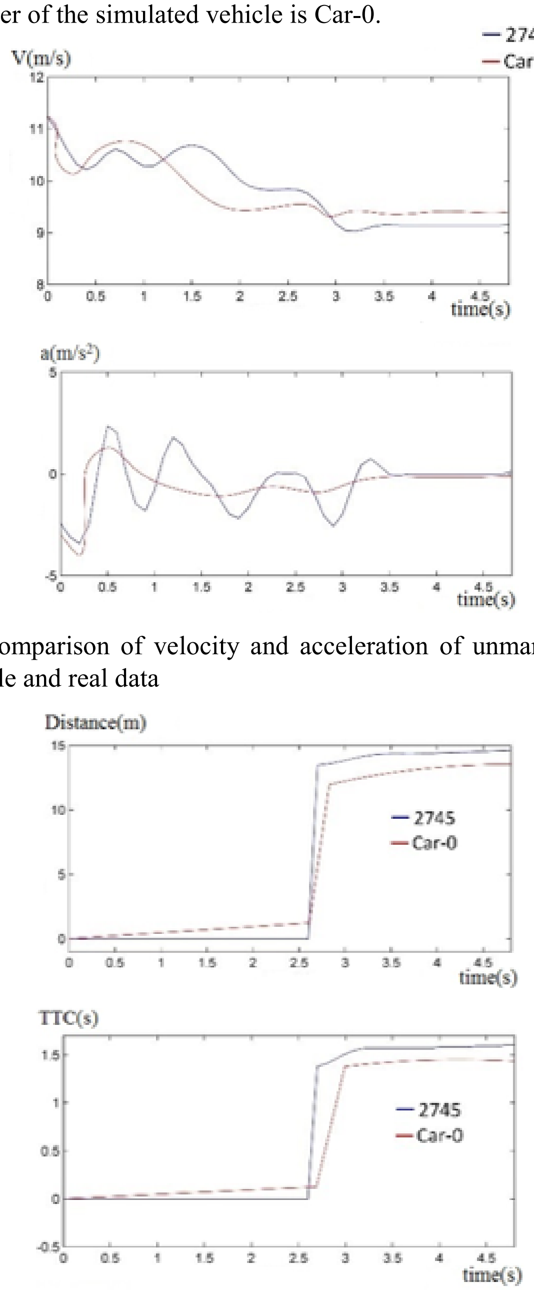
Fig. 9 Success rate comparison of Q-learning and LSPI algorithm

The maximum number of iterations in each experiment is 5000. In each iteration, the policy is evaluated and the success rate of merging behavior is recorded, and finally the success rate of 86% is reached. However, the success rate of Q-learning fluctuates at 25%, and the success rate of merging is low, so the applicability of the algorithm is not high.

In the 30 US101 datasets, as the speed of the merging vehicle is faster than the main stream, 64% of the drivers gave up the initial gap into the acceleration lane (corresponding to the second gap in this paper) and 33% of the drivers chose to re-enter the previous gap (corresponding to the first gap of this paper). While only 3% of drivers chose to import the last gap of the original gap (corresponding to the third gap in this paper).

Fig. 10 shows the real merging track of vehicle number

2745 and the lane change process of the merging policy. The number of the simulated vehicle is Car-0.



(a) Comparison of velocity and acceleration of unmanned vehicle and real data

(b) Comparison of gaps and TTC of unmanned vehicle and real data

Fig. 10. Comparison of merging data between unmanned vehicle and real vehicle dat

VI. Conclusion

(1) The convergence speed of LSPI algorithm in complex urban environments is high, and requires much less sample sets than Q-learning.

(2) After completing 5,000 iterations in the algorithm, the success rate of merging reaches 86%.

(3) In different traffic flows, the merging policy has a good environmental adaptability on different expressways.

(4) Compared with human pilots, the algorithm is relatively conservative for the choice of sink gap. Subsequent studies need to increase the type of vehicles, different ramps and the main lane after the car to slow down the line to make the import decisions.

References

[1] Chen X, Tiang G, Chan C Y, et al. Bionic Lane Driving Decision-Making Analysis for Autonomous Vehicle Under Complex Urban Environment[C]//Transportation Research Board 95th Annual Meeting. 2016 (16-4852).

[2] Chen X M, Miao Y S. Driving Decision-Making Analysis of Car-Following for Autonomous Vehicle Under Complex Urban Environment[C]//Computational Intelligence and Design (ISCID), 2016 9th International Symposium on. IEEE, 2016, 1: 315-319.

[3] Angkititrakul P, Miyajima C, Takeda K. Analysis and prediction of deceleration behavior during car following using stochastic driver-behavior model[C]//Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. IEEE, 2012: 1221-1226.

[4] Xu G, Liu L, Song Z. Driver behavior analysis based on Bayesian network and multiple classifiers[C]//Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on. IEEE, 2010, 3: 663-668.

[5] Ulbrich Simon, Maurer Markus, Probabilistic online POMDP decision making for lane changes in fully automated driving[M]: IEEE, 2013: 2063-2067.

[6] Sharifzadeh Sahand, Chiotellis Ioannis, Triebel Rudolphet al. Learning to Drive using Inverse Reinforcement Learning and Deep Q-Networks[J]. arXiv preprint arXiv:1612.03653, 2016.

[7] Zhang Xinyu, Gao Hongbo, Guo Muet al. A study on key technologies of unmanned driving[J]. CAAI Transactions on Intelligence Technology, 2016, 1（1）: 4-13.

[8] Wang Jian, Xu Xin, Liu Daxueet al. Self-learning cruise control using kernel-based least squares policy iteration[J]. IEEE Transactions on Control Systems Technology, 2014, 22（3）: 1078-1087.

[9] Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning[C]//Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017: 3357-3364.

[10] Wei Q L, Song R, Li B, et al. A Novel Policy Iteration-Based Deterministic Q-Learning for Discrete-Time Nonlinear Systems[M]//Self-Learning Optimal Control of Nonlinear Systems. Springer, Singapore, 2018: 85-109.

[11] Mukherjee A. Distribution-free phase-II exponentially weighted moving average schemes for joint monitoring of location and scale based on subgroup samples[J]. The International Journal of Advanced Manufacturing Technology, 2017: 1-16.