An Integrated DRL Framework for Autonomous High-Speed Cruising Control

Liang, Jinhao; Feng, Jiwei; Tan, Chaopeng; Zhou, Chaobin

DOI
10.1109/ISCTech63666.2024.10845357

Publication date
2024

Document Version
Final published version

Published in
2024 12th International Conference on Information Systems and Computing Technology, ISCTech 2024

An Integrated DRL Framework for Autonomous High-Speed Cruising Control

1st Jinhao Liang
Department of Civil and Environmental Engineering
National University of Singapore
Singapore
jh.liang@nus.edu.sg

2nd Jiwei Feng
School of Automotive Engineering
Liaocheng University
Liaocheng, China
jweifcheer@gmail.com

3rd Chaopeng Tan*
Department of Transport and Planning
Delft University of Technology
Netherlands
c.tan-2@tudelft.nl

4th Chaobin Zhou
College of Mechano-Electronic Engineering
Lanzhou University of Technology
Lanzhou, China
230198789@seu.edu.cn

2024 12th International Conference on Information Systems and Computing Technology (ISCTech) | 979-8-3503-7986-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCTech63666.2024.10845357

***Abstract*—Complex traffic scenes greatly challenge the road safety of automated vehicles (AVs). Recent work only provides an independent perspective from the fundamental modules. This paper integrates the decision-making and path-planning modules to ensure the autonomous driving performance in the high-speed cruising scenario. First, to guarantee deep exploration of the reinforcement learning method, a Bootstrapped deep-Q-Network (BDQN) is proposed to address the adaptive decision-making of AVs. Then, quantifying the multi-performance requirements of AVs under high-speed cruising can be complex. We employ an inverse reinforcement learning (IRL) approach to learn path-planning ability from skilled drivers, generating a reference path for executing lane changes. The simulation results demonstrate the proposed framework can ensure the autonomous cruising performance with safety guarantees.**

Keywords-Autonomous vehicle, Deep reinforcement earning, Inverse reinforcement learning, high-speed cruising.

I. INTRODUCTION

Autonomous vehicles (AVs) are increasingly becoming a mainstream technique aimed at enhancing traffic flow, reducing congestion, and optimizing energy usage within the intelligent transportation system [1]-[2]. AVs can perceive the environment and navigate paths independently. This plays a critical role in high-speed cruising scenarios characterized by frequent acceleration and lane-change maneuvers. The conventional method decomposes the high-speed cruising task into the vehicle-following and lane-change control, respectively [3]-[4].

From the perspective of the microscopic traffic simulation context, vehicle-following models are always designed based on driving tactics of the real traffic phenomenon [5]. Classical vehicle-following models typically consider factors such as the relative speed, headway, and acceleration/deceleration rates, which include the Gipps model [6], the Intelligent Driver Model (IDM) [7], and the Krauss model [8]. Recently, data-driven based vehicle-following models are attracted much attention from the world, including the supervised learning method [9], time-series prediction method [10], and the deep reinforcement learning method [11]. These data-driven models offer advantages in capturing complex and diverse driving behaviors.

Concerning the lane-change maneuver, researchers concentrate on designing robust decision-making algorithms [12]-[14]. The end-to-end architecture also enhances the system robustness and real-time performance compared with traditional optimal control methods [15]. Recently, some DRL algorithms such as DQN [16], Deep Deterministic Policy Gradient (DDPG) [17], and Proximal Policy Optimization (PPO) [18] are widely used to model the lane-change behavior.

However, recent research reveals the coupling between vehicle-following and lane-changing behaviors [19]-[20]. Hence, we propose a framework to realize the integrated longitudinal and lateral motion control. The main contributions are as follows.

1. Deep reinforcement learning is introduced into the autonomous decision-making process of AVs. BDQN combines deep exploration with deep neural networks, enabling exponential learning speed instead of relying on any dithering policies [21]. Specifically, the acceleration/ deceleration behavior during lane-keeping driving, as well as lane-change decision, are provided by the output of BDQN for the high-speed cruising scenario.

2. Considering AV's safety during lane-change maneuvers, we initially incorporate a polynomial trajectory generation method. To quantify the multi-objective requirements of AVs under high-speed cruising scenarios, including vehicle safety, and driving comfort, an inverse reinforcement learning (IRL) framework is employed to learn path-planning ability from experienced drivers and determine the optimal path.

The reminder is summarized as follows. The systematic framework is given in section II. The decision-making module developed by the BDQN is presented in section III. Section IV proposes the path-planning module. The test results are shown in Section V. Finally, Section VI concludes the paper.

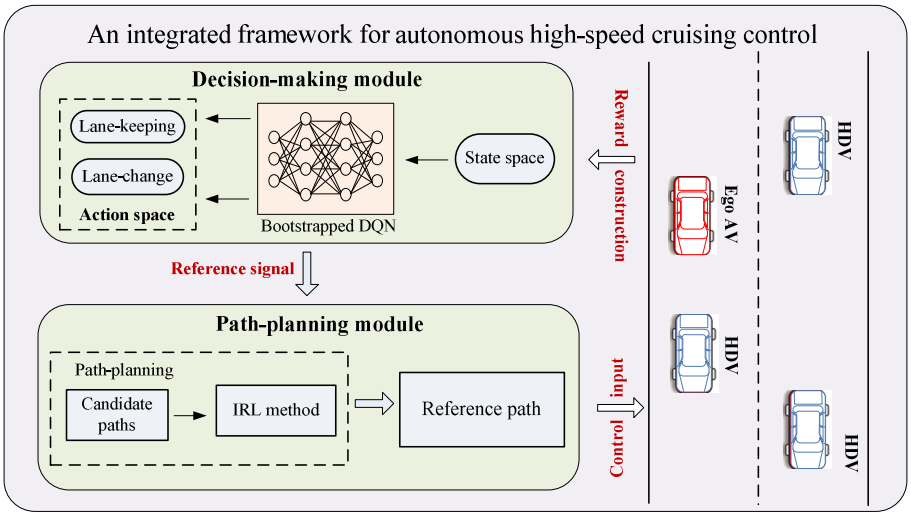

Fig. 1. The proposed holistic framework.

II. PROBLEM FORMULATION AND SYSTEMATIC FRAMEWORK

The conventional autonomous high-speed cruising task encompasses several crucial modules, including decision-making, path-planning, and motion-control, which are effectively addressed through a well-structured holistic framework (see Fig. 1).

In the decision-making process (Section III), an AV faces critical choices between executing lane-keeping or lane-change maneuvers during high-speed cruising scenario. Considering surrounding traffic conditions, a deep reinforcement learning (DRL)-based method is introduced to determine the appropriate actions to be executed. For lane-keeping maneuvers, the RL algorithm generates an acceleration control, ensuring the AV maintains a high cruising speed while avoiding potential collisions. On the other hand, for lane-change maneuvers, the path-planning process is required.

The path-planning stage is designed to achieve multi-objectives of the intelligent transportation system, notably emphasizing vehicle safety and driving comfort. To achieve this, an imitation learning algorithm is employed to learn the lane-change behavior from experienced drivers. In this scenario, surrounding vehicles are treated as human-driven vehicles (HDVs). To accurately depict HDVs' behavior, the Intelligent Driver Model (IDM) and Minimizing Overall Braking Induced by Lane Change Model (MOBIL) have been adopted within the framework [22].

III. BDQN-BASED DECISION-MAKING FOR THE AUTONOMOUS VEHICLE

The decision-making module holds significance in governing the vehicle's motion, which relys on a precise and high-fidelity vehicle model. In this section, we introduce a widely adopted bicycle model [23]-[24] to capture the dynamic characteristics of the vehicle. Subsequently, we provide a detailed explanation of the DRL decision-making process for potential lane-keeping and lane-change maneuvers in high- speed cruising scenarios.

A. Vehicle System Mode

Referring to Fig. 2, we adopt a bicycle model to describe the vehicle dynamics characteristics. The model parameters are obtained from a C-level car in the software Carsim. The longitudinal, lateral, and yaw motions can be represented as follows [25]-[26].

$$\dot{\chi} = \tilde{A}\chi + \tilde{B}\upsilon \tag{1}$$

where $\chi=\left[V_x,\ V_y,\ \phi,\ \gamma,\ X,\ Y\right]^T$, $\upsilon=\left[a_x,\ \delta_f\right]^T$.

$$\tilde{A}=\begin{bmatrix} 0 & 0 & 0 & V_y & 0 & 0 \\ 0 & \frac{2\left(c_r+c_f\right)}{mV_x} & 0 & \frac{2\left(-c_rL_r+c_fL_f\right)}{mV_x}-V_x & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{2\left(-c_rl_r+c_fl_f\right)}{I_zV_x} & 0 & \frac{2\left(c_rL_r^2+c_fL_f^2\right)}{I_zV_x} & 0 & 0 \\ 1 & 0 & -V_y & 0 & 0 & 0 \\ 0 & 1 & V_x & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{B}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-2c_f}{m} & 0 & \frac{-2c_fL_f}{I_z} & 0 & 0 \end{bmatrix}^T.$$

where c_f and c_r are the cornering stiffness of front and rear tires, respectively. ϕ and γ are the vehicle yaw angle and yaw rate, respectively. The lateral and longitudinal velocities are denoted by V_y and V_x, respectively. Y and X represent the vehicle lateral and longitudinal global positions, respectively. L_f and L_r are the distances from the center of gravity (CG) to the front and rear axles, respectively. m and I_z are the vehicle mass and inertia moment of yaw motion. a_x and δ_f are the vehicle longitudinal acceleration and front-wheel steering input, respectively.

B. RL Methodology Introduction

The RL algorithm aims to make a decision for an AV in a high-speed cruising scenario, choosing between lane-

keeping and lane-change maneuvers. To enhance the AV's deep exploration ability, we introduce the BDQN, which provides exponential learning speed without relying on dithering policies. The training process can be described as a Markov Decision Process (MDP), and further details can be found in references [33]-[34]. Through extensive interactions with the environment, the AV agent can learn a set of optimal parameters that enable the evaluation of long-term rewards $Q(s_i, a_i, \theta)$.

1) State Space

An AV would guide the path at an unknown environment by gathering the traffic information. In this study, the simulation environment is a structured highway. Consequently, the state space is composed of surrounding vehicle states, determined by their relative positions to the ego AV.

$$s_i = \begin{bmatrix} \vartheta_i \\ Y_{ev}-Y_1,\ X_{ev}-X_1,\ ,V_{ev}-V_1 \\ \vdots \\ Y_{ev}-Y_l,\ X_{ev}-X_l,\ V_{ev}-V_l \\ \vdots \\ Y_{ev}-Y_n,\ X_{ev}-X_n,\ V_{ev}-V_n \end{bmatrix} \tag{2}$$

where $\vartheta_i = [Y_{ev},\ X_{ev},\ V_{ev}]$ denotes the state information of the ego AV at the decision-making time i. Y_{ev}, X_{ev}, and V_{ev} are the longitudinal position, lateral position and velocity, respectively. Y_l, X_l, and V_l refer to the corresponding state information of the surrounding HDV $l(l=1,2,...,n)$. Eq. (2) serves as the state space for the BDQN.

2) Action Space

The RL agent is responsible for computing the global reward based on potential maneuvers of lane-keeping and lane-change in a high-speed cruising scenario. As shown in Eq. (1), a standard vehicle model incorporates acceleration and steering control inputs. Therefore, in this study, the decision-making module produces acceleration/deceleration behaviors for lane-keeping maneuvers and determines the ego AV's lane-change decision. To account for vehicle dynamics performance and road adhesion limitations, the acceleration behavior is defined by a value of 2m/s².

C. Reward Construction

The construction of rewards holds a critical role in shaping the agent's preferences and guiding its decision-making process. A well-designed reward system can expedite the training and lead to effective decision-making. In high-speed cruising scenarios, the ego AV is encouraged to maintain a faster speed while staying within safety limitations. This speed control intention can be represented as follows.

$$r_1 = \frac{V_{ego,i} - V_{x,\mathrm{mi}}}{V_{x,\mathrm{ma}} - V_{x,\mathrm{mi}}} \tag{3}$$

where $V_{x,\mathrm{mi}}$ and $V_{x,\mathrm{ma}}$ indicate the minimum and maximum velocities, respectively. $V_{ego,i}$ is the velocity of ego AV at decision time i.

Furthermore, to endow the collision-avoidance ability of the agent in a high-speed cruising scenario, an artificial potential function (APF) is introduced in the reward construction. The agent employs the APF to proactively perceive the surrounding traffic conditions. The specific definition of the APF reward is as follows.

$$r_2 = \sum_{m=1}^{n} \exp\left[\kappa_1 (Y_{ev}-Y_m)^2 + \kappa_2 (X_{ev}-X_m)^2\right] \tag{4}$$

Finally, the reward for the RL agent is constructed as:

$$r = \omega_1 r_1 + \omega_2 r_2 \tag{5}$$

The weight factors κ_1 and κ_2 play a crucial role in determining the potential collision risk for the ego AV. During the training process, the weight factors are tuned to minimize collision. After tests, κ_1 and κ_2 are set by -1.5, and -0.5 respectively. ω_1 and ω_2 are set by 15, and -2 respectively.

1) Training Process

DQN has demonstrated its effectiveness in various domains, including robotics and traffic signal control. It is particularly suitable for problems with discrete actions, like the Markov Decision Process problem presented in this work. where the action set is obviously discrete.

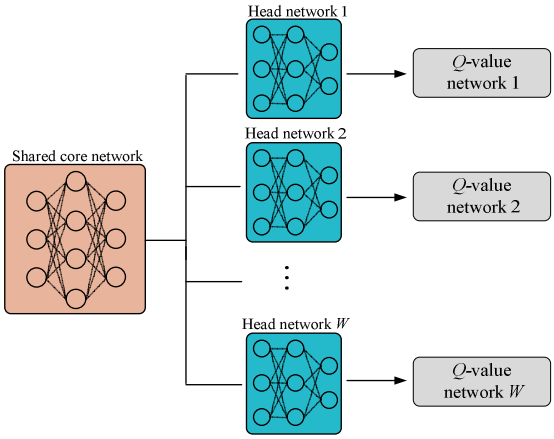

Fig. 2. The network structure of the BDQN.

The network structure of the BDQN is illustrated in Fig. 2, consisting of a shared core network and *w (w*=1,2,…,*W)* independent heads branches. Each head can be seen as a *Q*-value network combined with the core network. The essence of the BDQN lies in its utilization of the bootstrapped mask, denoted as *m*. When the vehicle takes an action from the action space, a vector *m* would be generated to acquire the bootstrapped subsamples corresponding to *W* independent heads branches. The augmented tuple of MDP defined by $(s_i, a, r, s_{i+1}, m_i)$ is introduced to replace the conventional tuple (s_i, a, r, s_{i+1}), and then store in the experience buffer at time *i*. *m* is a binary vector with a length of *W*, indicating which network should learn from the data. To achieve this masking process, we employ the masking distribution *M*

to generate each m. The gradient of the w-th Q-value network is given as follows.

$$G_{w,i} = m_{w,i}\left(y_i^{Q_w} - Q_w\left(s_i, a_i; \theta_i\right)\right)\nabla_\theta Q_w\left(s_i, a_i; \theta_i\right) \quad (6)$$

$$y_i^{Q_w} = r_i + \lambda \max_a Q_w\left(s_{i+1}, a\ ; \theta^-\right) \quad (7)$$

where $m_{w,i}$ represents the mask of w-th network, which can modulate the gradient and guide the bootstrapped behavior.

During the training process, gradient normalization is employed in this work to enhance training effectiveness at a lower cost to the training speed in the early stages.

$$G_{w,i}^{nor} = G_{w,i} / W \quad (8)$$

When an RL agent initiates exploration of potential rewards within the environment, the training process follows the fundamental DQN approach. A Q-value network is randomly selected at the start of each episode. Upon completion of training, the optimal policy can generate a series of actions. Given the variety inherent in the Q-value network within the BDQN architecture, a voting mechanism is introduced for the W networks to determine the optimal action. The principle dictates that the action with the highest number among the W different Q-value networks will be executed. Algorithm 1 presents the pseudo-code for the BDQN.

Algorithm 1: BDQN

Input: Q-value network Q^w and masking distribution D.

Initialize: Network parameter θ .

For each episode T=1,2, …, N, do

Get the state s_i through the environmental interaction.

Utilize a Q-value network Q^w to perform the action using w~ uniform{1,2, …, W}.

For each time step i=1,2…, do

Execute the action based on $a_i \in \arg\max_a Q^w(s_i, a)$.

The lane-change decision would be regarded as the reference to transmit to the lower layer of the path-planning module.

Obtain the state s_{i+1} and the reward r_i from the environment.

Sample the mask $m_i \sim M$.

Store the tuple $(s_i, a_i, r_{i+1}, s_{i+1}, m_i)$ to the experience buffer E.

End for

If the experience buffer E is full.

Then update θ

End if

End for

where N is the number of episodes during the training process. The path-planningmoiton module would describe in section IV.

IV. PATH-PLANNING MODULE

To quantify the multi-objective requirements of AVs under high-speed cruising scenarios, we propose an inverse reinforcement learning method, which can learn steering behavior from experienced drivers during lane-change maneuvers.

A. Path-planning with IRL Method

In this study, an AV travels on the structured road of a highway. A polynomial representation is used to construct candidate paths. The IRL method then selects the optimal reference from these candidate paths.

$$Y_{ref} = \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \sigma_3 X^3 + \sigma_4 X^4 + \sigma_5 X^5 \quad (9)$$

where Y_{ref} is the reference lateral position. σ_0, σ_1, σ_2, σ_3, σ_4, and σ_5 are the coefficients of the polynomial.

Assuming the initial position of the AV is (X_0, Y_0), the lane- change process involves lateral and longitudinal distances represented as L and C, respectively. If $X_0 = 0$, and $Y_0 = 0$, the terminal position (X_t, Y_t) can be determined as (L, C). Both the lateral velocity and lateral acceleration at initial and terminal positions are assumed to be zero. The reference path can be further written as follows.

$$Y_{ref} = \frac{10L}{C^3}X^3 - \frac{15L}{C^4}X^4 + \frac{6L}{C^5}X^5 \quad (10)$$

where L also indicates the width of the lane. 、 The lane-change time is denoted as t_{lac}. In real applications, it serves as an optimization variable to be determined by the IRL method. The longitudinal distance of the lane-change maneuver can be further expressed as $C = V_x t_{lac}$. The candidate paths are generated by configuring the time interval of 0.1s for the lane-change time t_{lac}.

Subsequently, the IRL method is employed to evaluate the overall performance of different settings in the intelligent transportation system. The performance indices encompass vehicle safety and driving comfort. Vehicle safety is quantified by considering the vehicle stability state and the potential collision risk, in which the sideslip angle β is used to describe the vehicle stability state and is calculated by V_y / V_x. Referring to eq. (4), the potential collision risk is constricted. Driving comfort is indicated by the change rate of steering input. To feature the performance indices, a vector is constructed as follows.

$$F = \left[F_1(s_k'), F_2(s_k'), F_3(s_k')\right] \quad (11)$$

where s_k' is the AV's state in the IRL structure and used to calculate the feature vector.

$$\begin{cases} F_1(s_k') = \sum_{k=1}^{\partial} \beta_k^2, F_3(s_k') = \sum_{k=1}^{\partial} \dot{\delta}_{f,k}^2, \\ F_2(s_k') = \sum_{k=1}^{\partial}\sum_{m=1}^{n} \exp\left[\kappa_1\left(Y_{ev} - Y_m\right)^2 + \kappa_2\left(X_{ev} - X_m\right)^2\right] \end{cases} \quad (12)$$

where ∂ denotes the total number of samples taken at each candidate path. $\dot{\delta}_{f,k}$ represents the change rate of the front- wheel steering angle input at the sampling point k. The total sampling number for each candidate path varies due to different values of t_{lac}. To facilitate further representation, we introduce a new feature vector with the normalized process as follows.

$$\bar{F}=\left[\frac{F_1(s_k')}{\partial},\frac{F_2(s_k')}{\partial},\frac{F_3(s_k')}{\partial}\right] \tag{13}$$

The reward function of IRL is constructed by a linear combination of the feature vector.

$$\hat{r}=\varepsilon^T\bar{F} \tag{14}$$

where $\varepsilon=[\varepsilon_1,\varepsilon_2,\varepsilon_3]$. Considering that the AV traveling on a structured road, the driving style is relatively smooth. Hence, the weight coefficients ε_1, ε_2, and ε_3 are set as constant in this work.

The steering behavior of experienced drivers during lane-change maneuvers is used as expert experience to optimize the weight coefficient σ, enabling the AV to exhibit similar behavior to that of a skilled driver. To construct the reward function in equation (14), we conduct driving simulator tests to collect data. It's worth noting that a maximum entropy method [38] is introduced for training the IRL model.

$$\max_{\varepsilon}\sum_{\iota\in\Xi}\log P(\iota|\varepsilon) \tag{15}$$

$$P(\iota|\sigma)=\frac{e^{\varepsilon^T\bar{F}_\tau}}{\sum_{j=1}^{S}e^{\varepsilon^T\bar{F}_{\tau_j}}} \tag{16}$$

The objective is to maximize the likelihood of the driver's trajectory $\iota_i\in\Xi,(i=1,2,\ldots,L)$. $\tilde{\iota}_j$ is the path candidate generated by the quantic polynomial. S is the number of path candidates. To guarantee the effectiveness of the training data, the initial state when generating the trajectory $\tilde{\iota}_j$ is the same as that of ι in the simulation environment. L indicates the total driver's trajectories collected in the tests. $\bar{F}_t$ is the feature vector of the driver trajectory. Hence, the optimization function is represented by:

$$\Omega(\varepsilon)=\sum_{\iota\in\Xi}\left(\varepsilon^T\bar{F}_t-\log\sum_{j=1}^{S}e^{\varepsilon^T\bar{F}_{\tau_j}}\right) \tag{17}$$

In the IRL framework, the learning rate η and episodes ξ can impact the training effectiveness. By comparing the human-like driving results with different settings, $\eta=0.1$ and $\xi=100$ are selected. After the IRL method generates the reference path, a PID controller is designed to govern the AV's motion and track this path

V. SIMULATION TESTS

In this section, the simulation tests are conducted to verify the effectiveness of the proposed method using Simulink/ Python joint platform. The motion-control module is established in Simulink. To reduce communication load and improve test efficiency, a communication trigger mechanism is established between software. When performing the lane- change maneuver, communication experiences interruption within the lane-change time t_{lac}.

A. *Verification of the Path-planning Module*

For real applications, the AV's path-planning ability is initially trained. In the path-planning module, we introduce the IRL method to learn experience from skilled drivers during lane-change maneuvers. The driving simulator as shown in Fig. 3 is employed to collect the steering behavior. Here, we develop the steering assembly to provide a real driving feeling in the simulation environment.

In the experiments, we curate a dataset comprising 40 driver trajectories recorded under different test conditions. This dataset is utilized to facilitate the updating of the IRL weight vector ε. After finishing the training, we employ a set of 10 driver trajectories to evaluate the learning effectiveness. Moreover, a comparative test with another trajectory selection method, denoted by the technique for order preference by similarity to the ideal situation (TOPSIS) [27], is conducted. The optimal path of the proposed method (PM) is determined by the IRL weight vector ε. We first assess the human-like path-planning ability by analyzing the path- tracking outcomes as follows.

$$\frac{1}{\partial}\sum_{i=1}^{\partial}\left\|Y_{con,i}-Y_{hum,i}\right\|^2 \tag{18}$$

where $Y_{con,i}$ and $Y_{hum,i}$ are trajectories of the control strategy and human driver at sampling point i, respectively. From Fig. 4, it is clear the AV's trajectory closely tracks the driver's path with the proposed method. Notably, the maximum tracking error can be reduced by 89.62% compared to the TOPSIS approach. This indicates AV's path-planning ability using the proposed method can better emulate a skilled driver.

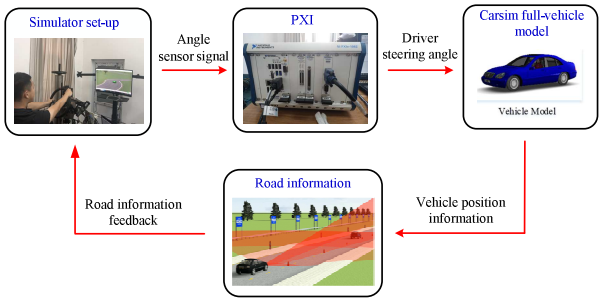

Fig. 3. Driving simulator platform

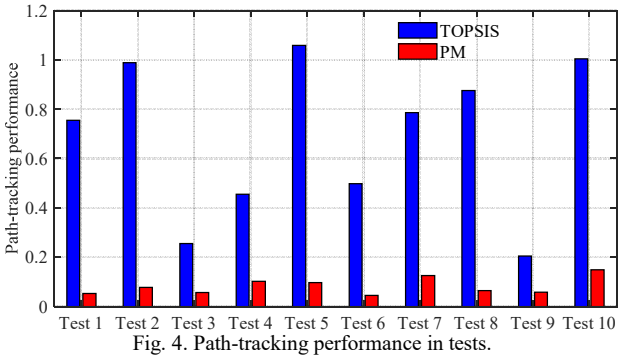

Fig. 4. Path-tracking performance in tests.

Furthermore, the feature vector $\bar{F}$ in eq. (13) is computed to provide a comprehensive evaluation of AV's performance under different control methods. The results are presented in Table I. Although some performance indices of the TOPSIS method perform well in tests, the AV's overall performance using the proposed method is superior due to its proficiency in acquiring driving skills that effectively balance various control objectives. The average enhancements for different objectives in eq. (13) across ten tests are 30.15%, 5.15%, and 17.27%, respectively, in comparison to the TOPSIS method. The test results demonstrate the effectiveness of the proposed in achieving a path-planning process of high quality.

TABLE I
TEST RESULTS

Performance index		$\bar{F}_1$	$\bar{F}_2$	$\bar{F}_3$
Test case1	PM	0.00058	406	47
	TOPSIS	0.00094	374	58
Test case2	PM	0.00049	319	40
	TOPSIS	0.00022	394	51
Test case3	PM	0.00051	280	47
	TOPSIS	0.00102	346	43
Test case4	PM	0.00047	395	46
	TOPSIS	0.00082	401	57
Test case5	PM	0.00067	327	40
	TOPSIS	0.00054	301	49
Test case6	PM	0.00052	349	56
	TOPSIS	0.00147	401	69
Test case7	PM	0.00063	302	54
	TOPSIS	0.00098	345	51
Test case8	PM	0.00059	357	56
	TOPSIS	0.00057	319	48
Test case9	PM	0.00069	328	47
	TOPSIS	0.00154	290	57
Test case10	PM	0.00064	329	53
	TOPSIS	0.00058	395	50

B. *Verification of the decision-making module*

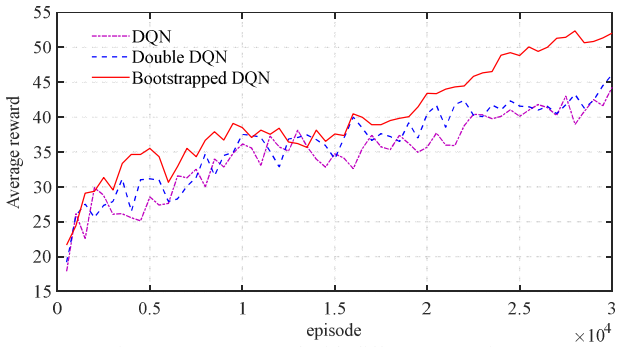

Fig. 5. Average reward with different strategies.

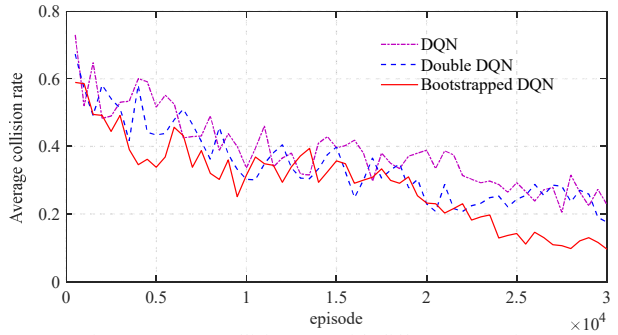

Fig. 6. Average collision rate with different strategies.

To validate the effectiveness of the decision-making module in this work, some other widely used DRL strategies DQN and double DQN are also conducted. Figs. 5 and 6 illustrate the average reward and collision rate with different strategies, respectively. From Fig. 5, as the number of training episodes increases, the average reward demonstrates a propensity to reach a state of convergence. Significantly, it becomes evident that the performance of the BDQN surpasses that of other strategies. This indicates the proposed decision-making module can better enhance the overall performance of autonomous driving. Furthermore, Fig. 6 demonstrates the bootstrapped mechanism can significantly enhance the AV's exploration capacity to find a collision-free path.

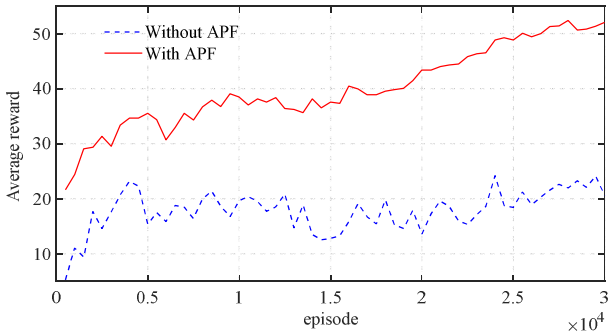

Fig. 7. Average reward with and without the incorporation of the artificial potential field.

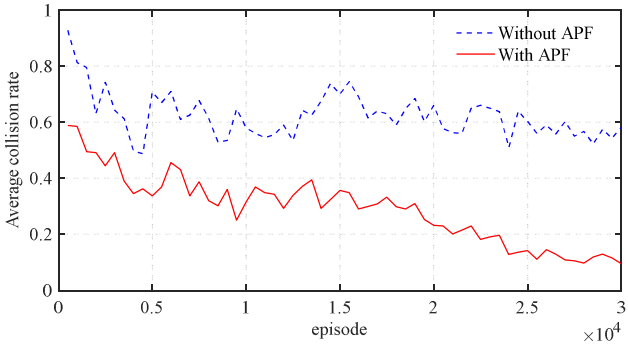

Fig. 8. Average collision rate with and without the incorporation of the artificial potential field.

Subsequently, we compare the performance of the BDQN under two conditions: with and without the incorporation of the artificial potential field. As depicted in Figs. 7 and 8, it is obvious the incorporation of the artificial potential field can improve the average reward while reducing the average collision rate. Notably, the average collision rate can be reduced by approximately 68.7% when reaching a stable state during the training progress. This demonstrates the effectiveness of the proposed decision-making module in guiding a collision-free path and ensuring the safety of the autonomous vehicle.

VI. CONCLUSION

In this work, we propose a standard development approach for integrating decision-making and path-planning modules of autonomous vehicles in high-speed cruising scenarios. To achieve this, we initially introduce a BDQN combined with an artificial potential field to enable adaptive decision-making in AVs. This approach generates reference actions for potential lane-keeping and lane-

change maneuvers. Then, to quantify the multi-performance requirements for an AV under high-speed cruising scenarios, we employ the IRL method to learn the path-planning skills of an experienced driver. This enables us to evaluate the overall performance of candidate paths and determine an optimal reference trajectory.

The test results indicate the proposed method effectively guides AVs to ensure high-speed cruising performance while taking safe actions to avoid collision. Our future work will consider the game between autonomous vehicles in the Vehicle-Road-Cloud Integration environment.

ACKNOWLEDGEMENTS

The authors Jinhao Liang and Jiwei Feng contribute equally to this work.

REFERENCES

[1] J. Liang, Q. Tian, J. Feng, D. Pi and G. Yin, "A Polytopic Model-Based Robust Predictive Control Scheme for Path Tracking of Autonomous Vehicles," IEEE Transactions on Intelligent Vehicles, vol. 9, no. 2, pp. 3928-3939, Feb. 2024.

[2] J. Betz et al., "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," IEEE Open Journal of Intelligent Transportation Systems, vol. 3, pp. 458-488, 2022.

[3] C. Wei, E. Paschalidis, N. Merat, A. S. Crusat, F. Hajiseyedjavadi and R. Romano, "Human-like decision making and motion control for smooth and natural car following," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 1, pp. 263-274, Jan. 2023.

[4] J. Liang et al., "A MAS-Based Hierarchical Architecture for the Cooperation Control of Connected and Automated Vehicles," IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 1559-1573, Feb. 2023

[5] H. U. Ahmed, Y. Huang, and P. Lu, "A review of car-following models and modeling tools for human and autonomous-ready driving behaviors in micro-simulation," Smart Cities, vol. 4, no. 1, pp. 314-335, Mar. 2021.

[6] H. Rakha, W. Wang, "Procedure for calibrating Gipps car-following model," Transp. Res. Rec,. vol. 2124, no. 1, pp. 113–124, Jan. 2009.

[7] M. Zhou, X. Qu and S. Jin, "On the impact of cooperative autonomous vehicles in improving freeway merging: A modified intelligent driver model-based approach," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 6, pp. 1422-1428, June 2017.

[8] T. Li, F. Guo, R. Krishnan, A. Sivakumar, and J. Polak, "Right-of-way reallocation for mixed flow of autonomous vehicles and human driven vehicles," Transportation research part C: emerging technologies, vol. 115, p. 102630, Jun. 2020.

[9] Y. Lin, P. Wang, Y. Zhou, F. Ding, C. Wang and H. Tan, "Platoon trajectories generation: A unidirectional interconnected LSTM-based car-following model," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 3, pp. 2071-2081, March 2022.

[10] Y. Zheng, S. He, R. Yi, F. Ding, B. Ran, P. Wang, and Y. Lin, "Categorizing car-following behaviors: wavelet-based time series clustering approach," Journal of transportation engineering, Part A: Systems, vol. 146, no. 8, p. 04020072, Oct. 2020.

[11] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," Transportation research part C: emerging technologies, vol. 97, pp. 348–368, Dec. 2018.

[12] E. Balal, R. L. Cheu, and T. Sarkodie-Gyan, "A binary decision model for discretionary lane changing move based on fuzzy inference system," Transp. Res. C, Emerg. Technol., vol. 67, pp. 47–61, Jun. 2016.

[13] Y. Dou, F. Yan, and D. Feng, "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers," in Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM), Banff, AB, Canada, 2016, pp. 901-906.

[14] G. Xiong, Z. Kang, H. Li, W. Song, Y. Jin and J. Gong, "Decision - making of lane change behavior based on RCS for automated vehicles in the real environment," in Proc. IEEE Intelligent Vehicles Symposium, Changshu, China, 2018, pp. 1400-1405.

[15] M. Bojarski et al., "End to end learning for self-driving cars," 2016, arXiv:1604.07316.

[16] C. -J. Hoel, K. Wolff and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in Proc. International Conference on Intelligent Transportation Systems, Maui, HI, USA, 2018, pp. 2148-2155.

[17] H. An and J.-i. Jung, "Decision-making system for lane change using deep reinforcement learning in connected and automated driving," Electronics, vol. 8, no. 5, p. 543, May. 2019.

[18] F. Ye, X. Cheng, P. Wang, C. -Y. Chan and J. Zhang, "Automated lane change strategy using proximal policy optimization-based deep reinforcement learning," in Proc. IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 2020, pp. 1746-1752.

[19] Y. Ye, X. Zhang, and J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," Transp. Res. C, Emerg. Technol., vol. 107, pp. 155–170, Oct. 2019.

[20] J. Zhao, T. Qu, and F. Xu, "A deep reinforcement learning approach for autonomous highway driving," IFAC-PapersOnLine, vol. 53, no. 5, pp. 542–546, Dec. 2020.

[21] J. Peng, S. Zhang, Y. Zhou and Z. Li, "An integrated model for autonomous speed and lane change decision-making based on deep reinforcement learning," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 21848-21860, Nov. 2022.

[22] T. Tan, T. Chu and J. Wang, "Multi-agent bootstrapped deep Q-network for large-scale traffic signal control," in Proc. IEEE Conference on Control Technology and Applications, Montreal, QC, Canada, 2020, pp. 358-365.

[23] J. Liang et al., "A Distributed Integrated Control Architecture of AFS and DYC Based on MAS for Distributed Drive Electric Vehicles," IEEE Transactions on Vehicular Technology, vol. 70, no. 6, pp. 5565-5577, June 2021.

[24] J. Liang et al., "An Energy-Oriented Torque-Vector Control Framework for Distributed Drive Electric Vehicles," IEEE Transactions on Transportation Electrification, vol. 9, no. 3, pp. 4014-4031, Sept. 2023.

[25] J. Liang et al., "A Hierarchical Control of Independently Driven Electric Vehicles Considering Handling Stability and Energy Conservation," IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 738-751, Jan. 2024.

[26] J. Liang et al., "Robust Shared Control System for Aggressive Driving Based on Cooperative Modes Identification," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 11, pp. 6672-6684, Nov. 2023.

[27] O. Dogan, M. Deveci, F. Canitez, and C. Kahraman, "A corridor selection for locating autonomous vehicles using an interval-valued intuitionistic fuzzy Ahp and topsis method," Soft Comput., pp. 1–17, 2019.