Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/106/20241243

An Adaptive Cruise Control Algorithm Based on DDPG Algorithm Based on Deep Reinforcement Learning Under Variable Acceleration Conditions

Yuhao Lan

College of Automotive Sciences, Tongji University, Shanghai, China

2153511@tongji.edu.cn

Abstract. Adaptive cruise control (ACC) dynamically regulates a vehicle's speed to preserve a secure gap from the preceding vehicle, enhancing road safety. In this study, ACC is examined through the lens of deep reinforcement learning, with a focus on the Deep Deterministic Policy Gradient (DDPG) technique. The reward function takes into account the speed error, and two modes—speed control and distance control—are implemented. The proposed ACC strategy is trained and validated through simulations on the MATLAB/Simulink platform. The experimental results indicate that the reward function converges rapidly, confirming the suitability of the DDPG algorithm for automotive ACC research.

Keywords: Reinforcement Learning, DDPG, Adaptive Cruise, Intelligent Vehicle.

1. Introduction

Automobile adaptive cruise control (ACC), sometimes referred to as active cruise control, is a new system that adds the control function of keeping reasonable distance with the vehicle in front on the basis of the constant speed cruise system. Compared with cruise control, it does not require the driver to frequently cancel and set the cruise control function, so as to avoid distracting the driver during driving. In addition, adaptive cruise can automatically control the engine and brakes properly. In other words, models equipped with fixed-speed cruise can only play a role when there is very little traffic and the road conditions are very good, while the applicable scenarios of adaptive cruise are much wider. Ideally, as long as you get on the high speed, you can directly turn on the adaptive cruise, so that it can control its own deceleration and acceleration, and almost completely free the right foot. ACC further considers safety and comfort, can alleviate driving fatigue, and has a broad space for development.

Fu et al. examined three critical factors—efficiency, accuracy, and passenger comfort—and utilized the DDPG algorithm to address autonomous braking by closely analyzing both lane changes and braking processes [1]. Zhang et al. introduced the use of the Catmull-Rom spline used for modeling lane markings, which offers greater stability and simplicity in construction compared to cubic polynomials, along with enhanced accuracy [2]. Liu et al. developed the AD-GNN framework for predicting adaptive traffic flow using a neural network, which integrates gated time convolutional networks with diffusion convolutional networks to simulate spatial-temporal relationships [3]. Zhang et al. also created an adversarial pedestrian detection model, employing adversarial learning during training, which enhanced detection accuracy by 18.45% and delivered real-time processing at 318 frames per second [4]. Jin et al.

DOI: 10.54254/2755-2721/106/20241243

applied the 3D-EAFF model, along with trajectory deletion and update strategies, to enhance the tracking accuracy of 3D multi-object tracking (MOT) [5]. Desjardins et al. used function approximation and gradient descent algorithms to refine automatic vehicle control strategies, enhancing the performance of cooperative adaptive cruise control (CACC) [6]. Finally, Chen et al. designed a path planning strategy leveraging deep reinforcement learning to reduce vehicle fuel consumption while meeting task deadlines [7]. This paper focuses on adaptive cruise control, employing the DDPG algorithm from deep learning to implement speed and distance control modes by adjusting for speed and distance errors.

2. Background Knowledge

Deep reinforcement learning (DRL) is an AI method that simulates human decision-making by combining the perceptive abilities of deep learning with the decision-making mechanisms of reinforcement learning. This approach allows control to be exercised directly based on visual inputs. The remarkable achievements of deep reinforcement learning, particularly demonstrated by AlphaGo, have drawn considerable attention from the global research community. Its effectiveness in making decisions under complex conditions highlights its potential. The complexity of vehicle operation, which requires analyzing large volumes of detected data and making real-time decisions, aligns well with the strengths of deep reinforcement learning. Figure 1 illustrates the key components and processes involved in reinforcement learning.

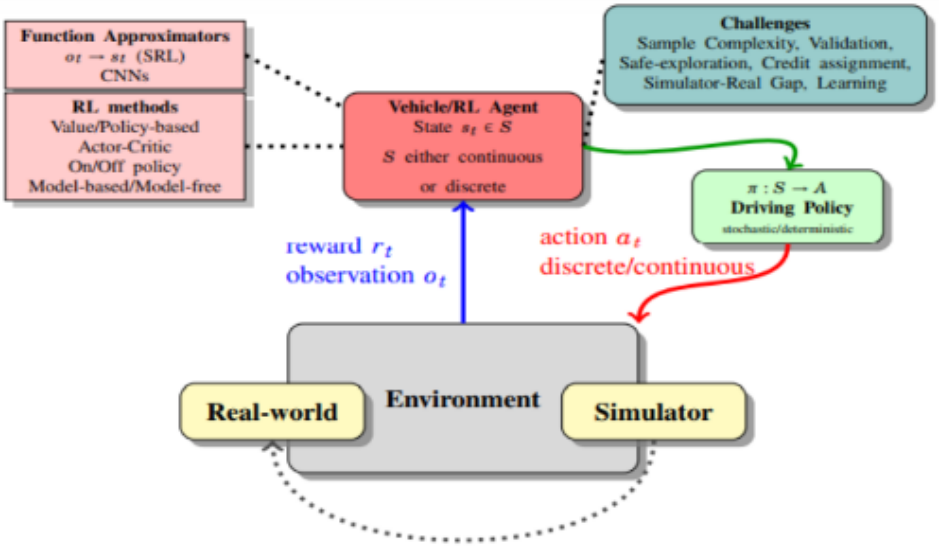

Figure 1. Component composition and process of reinforcement learning.

The agent engages with its environment by taking actions, which in turn cause the environment to change and yield a reward. Through repeated interactions, the agent acquires knowledge through a process of trial and error to develop an optimal strategy. This process, involving trial and error with delayed feedback, is referred to as a Markov decision process. A Markov decision process is characterized by a tuple $< S, A, P, r >$, where S represents the set of all potential states, A represents the set of all potential actions, P represents the probability of state transition (i.e., the likelihood of moving from state s to state s' after taking action a), and $r(s, a)$ is the reward function, which specifies the reward earned from executing action a in state s. During each interaction cycle, the agent traces an interaction trajectory, and the cumulative return at a given state is calculated as follows: $(s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T, \cdots)s_t$

$$R_t = \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (1)$$

DOI: 10.54254/2755-2721/106/20241243

Where$\gamma \in [0,1]$ represents the discount factor, which determines how future rewards are valued relative to immediate ones. A higher value of γ places more emphasis on long-term rewards, while a lower value focuses more on short-term gains. The main goal of reinforcement learning is to optimize the expected cumulative reward, expressed as:

$$\pi(a|s) = \arg\max_{a} E[R] \tag{2}$$

To determine the optimal strategy, we can introduce the value function and the state-action value function. The value function is defined as:

$$V_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right] = E_\pi[r_{t+1} + \gamma V(s_{t+1}) | s_t = s\Big] \tag{3}$$

The state-action value function is defined as:

$$Q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right] \tag{4}$$

Then the optimal state-behavior function corresponding to the optimal strategy is satisfied:

$$q^*(s,a) = \max_{\pi} q_\pi(s,a) \tag{5}$$

3. ACC Based on DDPG Algorithm

The Deep Deterministic Policy Gradient (DDPG) algorithm is a deep reinforcement learning method specifically created to address issues in continuous action spaces. It integrates deterministic strategies with the concept of experience replay. DDPG operates on a framework based on the actor-critic model, where the Actor is tasked with learning deterministic policies—outputting specific action values for a given state. Meanwhile, the Critic's role is to learn the value function and assess the current state. Unlike traditional policy gradient methods, DDPG uses deterministic strategies that output exact action values rather than probabilities, making it more effective in continuous action spaces. Before training begins, the network architecture and parameters for both the Actor and Target-Actor, as well as the Critic and Target-Critic, are identical. During training, the parameters of the Target-Actor and Target-Critic networks are updated according to formula (5):

$$\begin{cases} \theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\pi'} = \tau\theta^{\pi} + (1-\tau)\theta^{\pi'} \end{cases} \tag{6}$$

Where $\theta^{Q'}$, θ^{Q}, θ^{π}, $\theta^{\pi'}$ represent the parameters for the Target-Critic, Critic, Target-Actor, and Actor networks, respectively. Additionally, $\tau \ll 1$ governs the rate of parameter updates. The anticipated gradient of the action-value function aligns with the gradient of the deterministic policy function:

$$\nabla_\theta J(\mu_\theta) = E_{\mu_\theta}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu}(s,a)|_{a=\mu_\theta(s)}\right] \tag{7}$$

The spacing strategy employed in this paper utilizes the fixed time interval approach (CTH). In the CTH method, the time gap between vehicles is kept constant, and the distance relative to the vehicle ahead is directly related to its speed. For safety purposes, the strategy includes a minimum safe distance, outlined in the following formula t_H:

$$\Delta x_{des} = t_h v_H + x_0 \tag{8}$$

In this formula, Δx_{des} represents the desired relative distance between the vehicles, t_h is the time gap, v_H is the speed of the leading vehicle, and x_0 is the minimum safe distance to follow. The MATLAB/Simulink simulation flowchart for the algorithm utilized in this study is displayed in Figure 2.

DOI: 10.54254/2755-2721/106/20241243

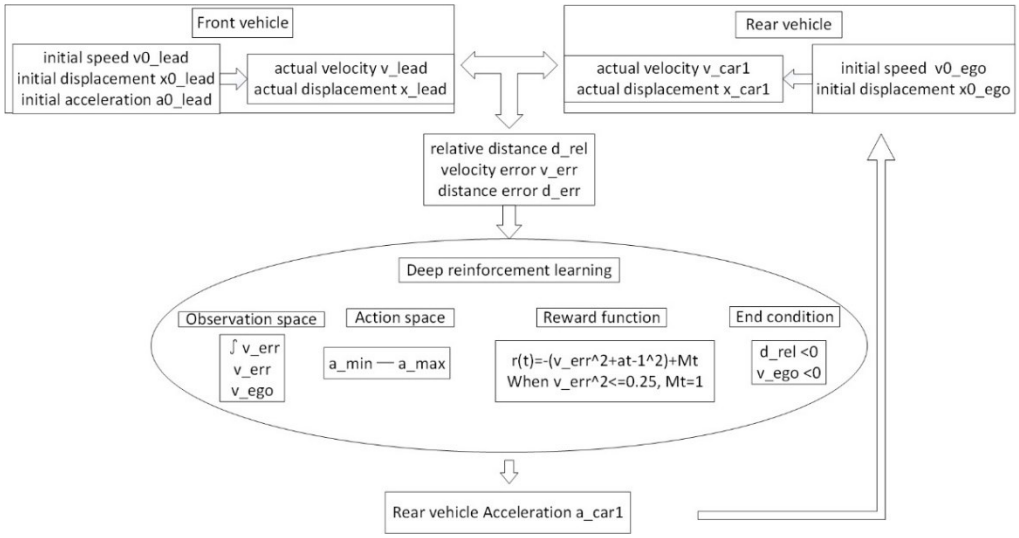

Figure 2. MATLAB/Simulink simulation flow chart.

4. Experimental Results

The hardware setup for the algorithm in this study includes an AMD R7 5800H processor with Radeon Graphics, 16GB of DDR4 3200MHz RAM (8GB × 2), an M.2 2280 512GB SSD, and an NVIDIA GEFORCE RTX 3050 GPU. The software environment is Windows 10 Home (Chinese 64-bit) running Matlab R2022a with Simulink. Figure 3 displays the safety distance and relative distance curves, while Figure 4 illustrates the displacement curve.

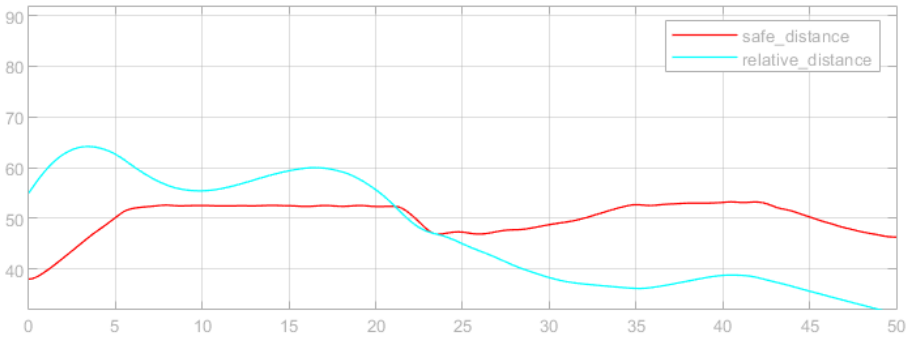

Figure 3. Safety distance and relative distance curve.

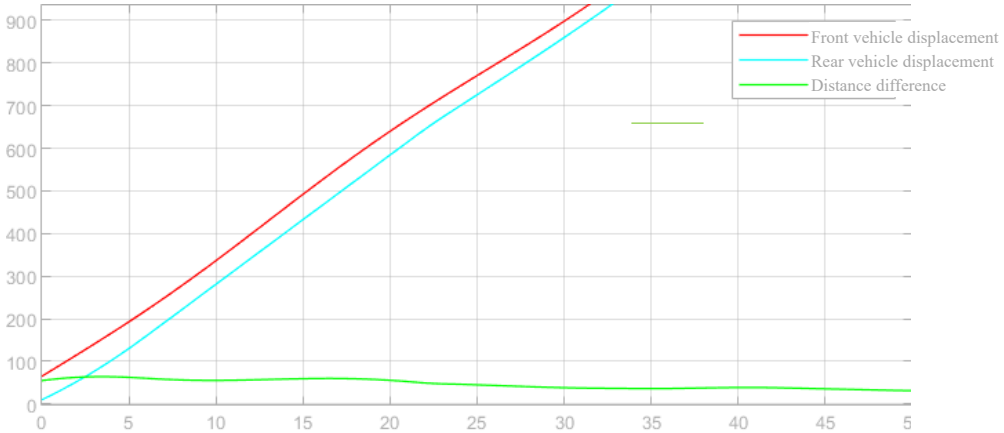

Figure 4. Displacement curve.

DOI: 10.54254/2755-2721/106/20241243

As shown in the figure, the relative distance between the two vehicles exceeds the safe distance during the first 23 seconds. After 23 seconds, the relative distance falls below the safe distance, but the gap between the two vehicles stabilizes. Figure 5 presents the acceleration curve of the leading vehicle.

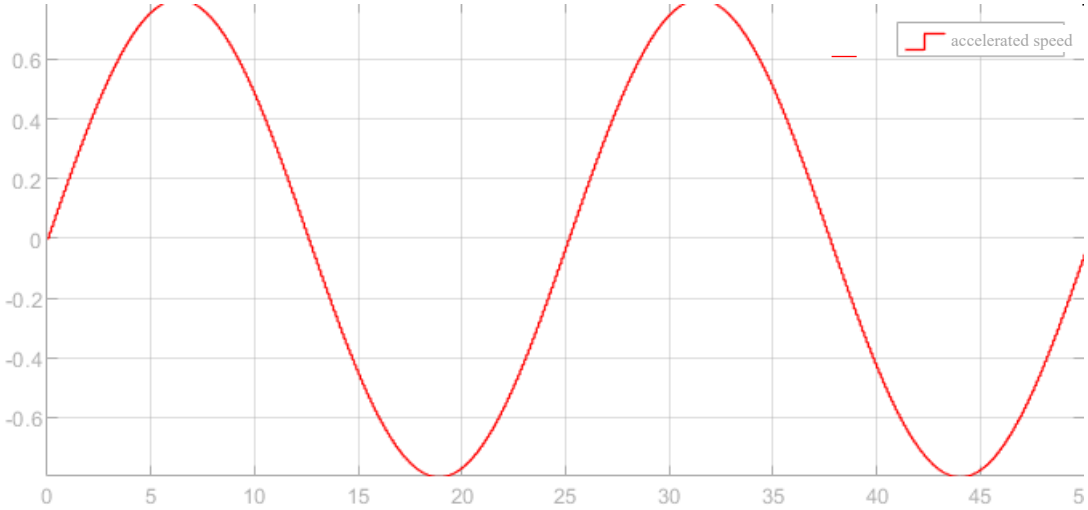

Figure 5. Acceleration curve of the front vehicle.

In this study, the initial acceleration of the leading vehicle is defined as 0.8sin (0.25ωt). The speed curve is depicted in Figure 6.

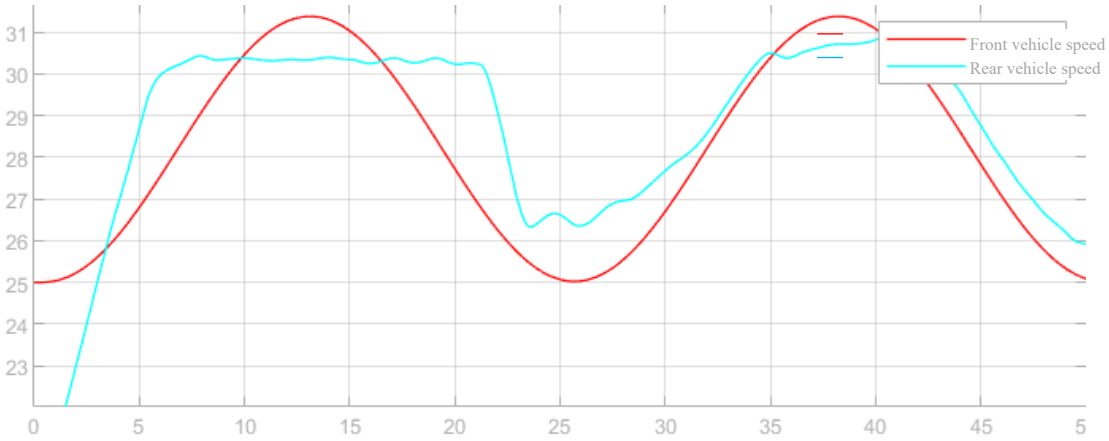

Figure 6. Speed curve.

The figure shows that before 23 seconds, the relative distance surpasses the safe threshold, and the system operates in speed control mode, maintaining the vehicle speed at around 30 m/s. After 23 seconds, when the relative distance drops beneath the safety threshold, the system switches to distance control mode, with v_ref=min {v_lead, v_set}. The acceleration curve for this vehicle is presented in Figure 7.

DOI: 10.54254/2755-2721/106/20241243

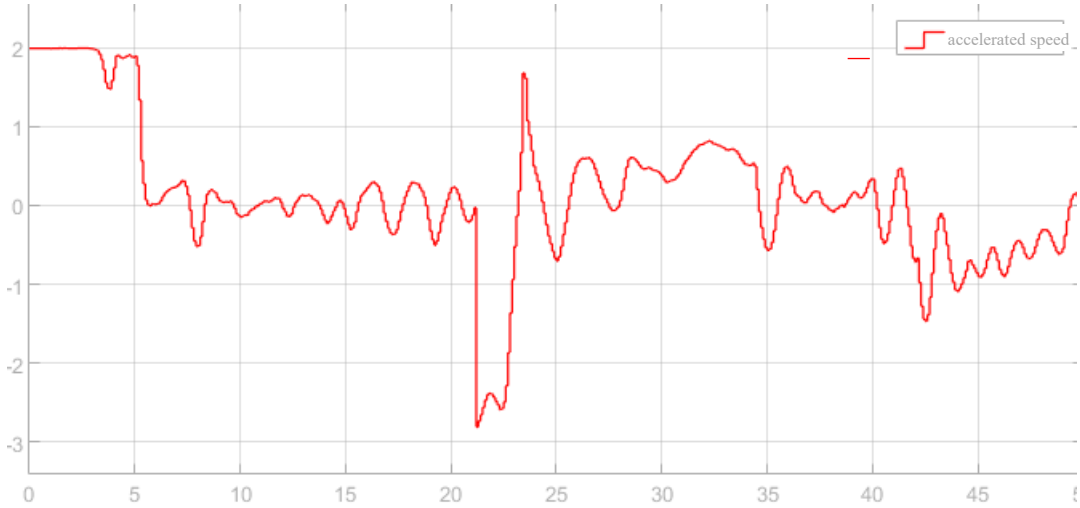

Figure 7. Acceleration curve of the vehicle.

The acceleration curve of the leading vehicle indicates that the acceleration varies over time during both the speed control and spacing control modes. Figure 8 illustrates the reward value curve, which shows rapid convergence. This demonstrates the effectiveness of using reinforcement learning to implement adaptive cruise control (ACC) efficiently.

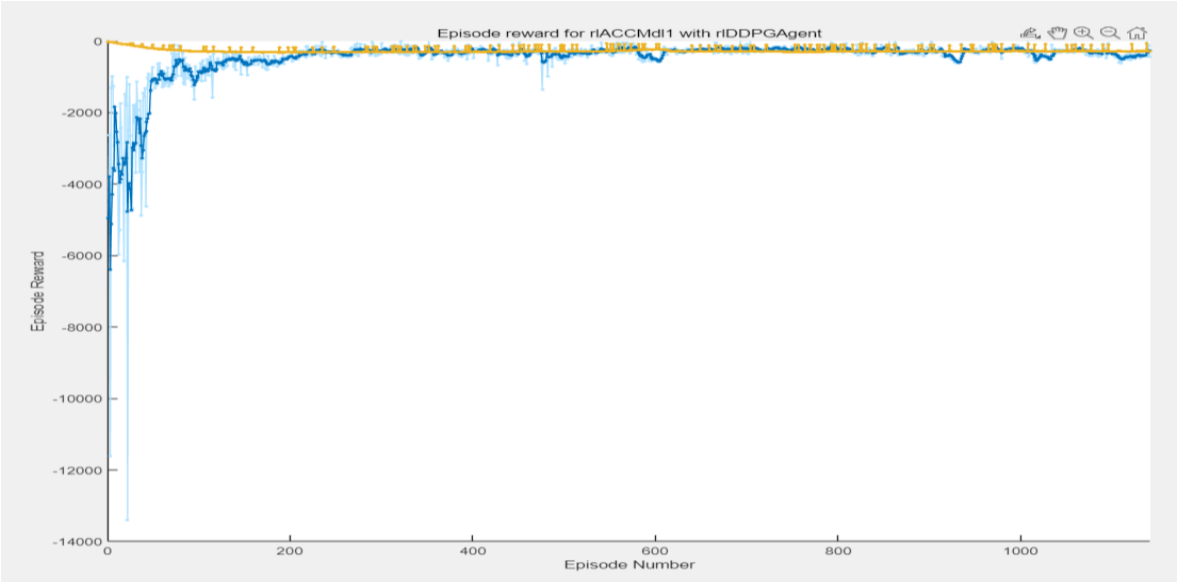

Figure 8. Reward value curve.

5. Conclusions

This paper focuses on adaptive cruise control (ACC) and implements it using the DDPG (Deep Deterministic Policy Gradient) algorithm within deep learning. By incorporating functions for speed error and distance error, two control modes—speed control and distance control—are established along with a reward value function. The algorithm is tested and validated through simulations in MATLAB/Simulink. The experimental results confirm that the DDPG algorithm successfully handles both control modes using deep reinforcement learning. While this study considers a scenario involving two vehicles, real-world traffic conditions often involve three or more vehicles. Future research will aim

DOI: 10.54254/2755-2721/106/20241243

to enhance model performance by extending the approach to multi-vehicle scenarios using deep reinforcement learning.

References

[1] Yuchuan Fu, Changle Li, Fei Richard Yu. A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6):5876-5888.

[2] Jindong Zhang, Haoting Zhong. Curve-based lane estimation model with lightweight attention mechanism[J]. Signal, Image and Video Processing, 2023, 17(5):2637-2643.

[3] Tianbo Liu, Jindong Zhang. An adaptive traffic flow prediction model based on spatiotemporal graph neural network[J]. The Journal of Supercomputing, 2023, 79(14):1-25.

[4] Jindong Zhang, Jian Dou. An adversarial pedestrian detection model based on virtual fisheye image training[J]. Signal, Image and Video Processing, 2024, 18(4):3527-3535.

[5] Jingyi Jin, Jindong Zhang, Kunpeng Zhang, et al. 3D multi-object tracking with boosting data association and improved trajectory management mechanism[J]. Signal Processing, 2024, 218(2024):109367

[6] Charles Desjardins, Brahim Chaib-draa. Cooperative adaptive cruise control: a reinforcement learning approach[J]. IEEE Transactions on intelligent transportation systems, 2011, 12(4): 1248-1260.

[7] Li, Meng, Cao Zehong, Li Zhibin. A reinforcement learning-Based vehicle platoon control strategy for reducing energy consumption in traffic oscillations[J]. IEEE Transactions on Neural Networks & Learning Systems, 2021, 32(12):5309-5322.