

Cruise Control Method for Intelligent Vehicles Based on Reinforcement Learning

Hongyi Wang

School of Mechatronics and Vehicle Engineering, East China Jiaotong University,
Nanchang, 330013, China

3288197674@qq.com

Abstract. This paper explores the application of Q-learning in Adaptive Cruise Control (ACC). Environment Modeling and Decision-Making: The radar detects signals from obstacles ahead and transmits them to the ACC system. The Q-learning algorithm makes decisions based on environmental changes, such as the speed and distance of the vehicle in front. Through trial and error and continuous learning, it optimizes the following strategy. Reinforcement learning is a computational method that automates decision-making and learning processes toward goal-oriented tasks. It is framed within the Markov Decision Process (MDP), using states, actions, and rewards to define the interaction between a learning agent and the environment. Reinforcement learning emphasizes the agent's interactive learning with the environment, without the need for imitative supervision signals or complete environment modeling. The Markov Decision Process provides a theoretical framework for achieving goals through interactive learning. All reinforcement learning problems are considered an MDP process, serving as the fundamental framework for modeling reinforcement learning. By updating the state-action value function $Q(s, a)$, the Q-learning algorithm selects actions in each state that maximize long-term returns, enabling intelligent control. Experimental results show that the method applied in this paper ensures driving safety and comfort, and it can be practically implemented.

Keywords: intelligent vehicle, reinforcement learning, cruise control, Markov Decision Process.

1. Introduction

With the development of automotive intelligence technology, intelligent cruise control systems, as an advanced driver assistance system (ADAS), can automatically control vehicle acceleration, deceleration, or braking, adjusting the distance or relative speed with the vehicle in front to maintain a safe range. This greatly reduces the burden on drivers and improves driving safety and comfort. The research and application of such systems not only reflect considerations for driver safety and comfort but also represent the advanced stage of automotive collision avoidance radar technology, signaling a future development trend. Intelligent vehicle cruise control systems, such as Adaptive Cruise Control (ACC), are an advanced active safety feature. Through radar, ultrasonic, and infrared sensors, these systems detect obstacles, pedestrians, and vehicles ahead, and have automatic braking functions to ensure precise distance control. Such systems not only maintain a stable speed during high-speed driving, avoiding frequent acceleration and braking, thus reducing driver fatigue, but can also realize automatic queuing

and vehicle-following functions at low speeds (e.g., below 40 km/h), completely freeing the driver's right foot for a simpler, more relaxed, and enjoyable driving experience[1].

In 2024, Onur Cihan, Zeynep Musul, adopted a new system architecture of intelligent adaptive cruise control for safety[1]. In 2024, J. S. Saputro, et al adopted design of an intelligent cruise control system using fuzzy PID control on an autonomous electric vehicle prototype[2]. In 2023, Milad Andalibi, et al adopted optimization of intelligent autonomous cruise control system based on design and simulation[3]. In 2024, Tianchen Ruan, et al adopted general hierarchical control system model ACC system[4]. In 2020, Faezeh Farivar, et al adopted Security of network control system in intelligent vehicle and adaptive cruise control[5]. In 2004, Chan Yuen Fong and Zhang Zhendong adopted PID control theory in the design of ACC systems[6]. In 2009, Zhang Lei et al. analyzed car-following behaviors and abnormal behavior characteristics based on experimental data and established a driver characteristic self-learning method using recursive least squares with a forgetting factor. Although driving characteristics were considered in the adaptive cruise control system, the complexity of the algorithm's self-learning process affected speed and accuracy[7]. In 2012, Shakouri et al. optimized the vehicle's nonlinear characteristics into linear ones, allowing for the use of a linear time-invariant model to control tracking indicators[8]. In 2010, Li and Kural conducted real-time research and observation on vehicle dynamics models and optimized ACC algorithms using the least-tracking method[9]. In 2013, Bifulco et al. proposed a full-speed ACC system capable of adapting to various real-world traffic conditions and the driver's actual attitudes and preferences. However, these attitudes and preferences may fluctuate, leading to excessive calculation deviations in the algorithm, resulting in instability under extreme conditions[10]. In 2010, Yi et al. analyzed the ACC system using transitional state variables and obtained the desired acceleration[11]. In 2003, Mobus et al. studied the motion characteristics of ACC systems using a nonlinear method with finite time constraints[12]. Fuzzy Control primarily uses fuzzy reasoning to control the system, then applies expert experience to achieve precise control. In 2006, José E et al. applied fuzzy control theory to study ACC systems[13]. In 2007, Naranjo et al.[14] designed a longitudinal controller based on fuzzy logic, similar to the underlying control algorithm for ACC systems designed using reinforcement learning. Hou Haiyang[15], combining rolling optimization over a finite time horizon with feedback correction, built an MPC prediction model based on the longitudinal motion state equation of a vehicle, while establishing objective functions and constraints. Using the minimum two-norm method, they optimally solved for the desired acceleration and output performance indicators. Li Peng et al designed a car-following control strategy using linear optimal quadratic theory, with vehicle distance and relative speed as state variables, selecting the desired acceleration of the vehicle as the output, and the deceleration of the preceding vehicle as the disturbance, establishing the state-space equation of the car-following process[16]. The acceleration error controlled the acceleration of the vehicle, and the optimal solution was used as the desired acceleration. Zhang Lei[17] proposed a longitudinal driving assistance system based on a driver car-following model, integrating ACC and FCW/FCA (Forward Collision Warning/Forward Collision Avoidance). Using the recursive least squares method with a forgetting factor, they identified driver car-following model parameters online and built a BP neural network model to classify abnormal driving behaviors and adaptively match algorithm parameters.

This paper applies Q-learning in ACC through environment modeling and decision-making: radar detects signals from obstacles ahead and transmits them to the ACC system. The Q-learning algorithm makes decisions based on environmental changes, such as the speed and distance of the vehicle ahead. Through trial and error and continuous learning, the car-following strategy is optimized. Intelligent Control: By updating the state-action value function $Q(s, a)$, the Q-learning algorithm selects actions in each state that maximize long-term returns, enabling intelligent control and ensuring driving safety and comfort.

2. Background Knowledge

ACC integrates onboard radar for detecting vehicles ahead, allowing for vehicle following while retaining the constant speed cruise functionality. The ACC system is generally divided into four main

components: the sensor unit, which collects signals including the vehicle's status, driving environment, and input information from the human-machine interface; the ACC controller, the core unit of the system, responsible for processing driving information and determining vehicle control commands; the actuator, mainly comprising the brake pedal, accelerator pedal, and vehicle transmission actuators, used to realize vehicle acceleration and deceleration; and the human-machine interface, which enables the driver to make decisions about activating the system, setting parameters, displaying system status, and alerting in emergencies.

The ACC system is an intelligent automatic control system developed based on existing cruise control technology. During vehicle operation, distance sensors (radar) installed at the front of the vehicle continuously scan the road ahead, while wheel speed sensors collect speed signals. It has four operating conditions. In the cruise condition, there are no target vehicles ahead of the self-vehicle. The driver maneuvers the experimental vehicle to accelerate or decelerate from one speed to another, maintaining that speed for a period. The driver's speed change process during this condition can provide a reference for vehicle acceleration or deceleration in the CC mode of the personalized ACC control strategy. In the approaching target vehicle condition, there is a target vehicle ahead. The self-vehicle approaches the target vehicle from a distance at a speed greater than that of the target vehicle and maintains a stable follow for a period. The initial relative distance in this experiment ranges from 30m to 60m. This condition is used to analyze the driving characteristics of the driver during the deceleration process while approaching the target vehicle, as well as the driver's stability in maintaining a following distance. It provides a reference for the personalized ACC's approaching process. In the following target vehicle condition, the self-vehicle first follows the target vehicle for a stable distance. Subsequently, the target vehicle accelerates or decelerates to a certain speed at a fixed acceleration and then travels at a constant speed. During this process, the driver operates the accelerator or brake pedal according to their habits, accelerating or decelerating while maintaining a certain distance from the target vehicle.

The speed changes of the target vehicle differ by more than 20 km/h, aimed at analyzing the characteristics of the driver's acceleration or deceleration behavior to maintain following during significant speed changes of the target vehicle. This also captures the variation in relative distance throughout the process, providing a reference for the construction of following strategies in the personalized ACC control strategy. In the target vehicle cutting out condition, the self-vehicle initially follows the target vehicle for a stable distance. When the target vehicle changes lanes and the self-vehicle loses the target, the driver then accelerates to maintain a certain speed. This process is used to analyze the transition from following decision-making to cruising decision-making when the target vehicle cuts out. It provides a reference for mode switching in the personalized ACC control strategy.

The Q-learning algorithm is a method for solving reinforcement learning problems using temporal difference learning. This type of reinforcement learning does not rely on model-based methods nor does it require a state transition model of the environment. The control problem is solved by action-value iteration, where the action-value function is updated through learning, leading to policy updates, and eventually, the action-value function and policy converge.

MDP is characterized by a tuple of five elements: state set (s), action set (a), state transition probability (p), reward function (r), and discount factor (γ). Q-learning Algorithm Principles:

- 1) State (s): Each reachable grid in a grid map is defined as a state s , where t represents the time step.
- 2) Action (a): In each state (s), the agent (e.g., a robot) can choose actions such as moving up, down, left, or right.
- 3) Reward (r): The immediate feedback the agent receives after performing an action. A positive reward is given upon reaching the goal, while otherwise, it may be zero or a negative reward, denoted as $R+1$.
- 4) Q-Table ($Q(s,a)$): The core of the algorithm is the Q-function, which records the cumulative expected reward after taking action (a) in state (s). This is initialized with arbitrary values and updated through learning.

5) Q-learning updates the Q-values using the Bellman equation. Its core idea is that the expected return of selecting the optimal action should equal the current reward plus the maximum expected return from following the optimal policy from the next state onward. The update rule is as follows: α is the learning rate, determining the extent to which new knowledge is integrated with old knowledge, where $0 < \alpha < 1$. γ is the discount factor, used to evaluate the importance of future rewards, $0 \leq \gamma < 1$.

6) $R+1$ is the reward obtained immediately after taking action a . $\max_{a'} Q(s+1, a')$ is the maximum Q-value of all possible actions in the next state $s+1$, reflecting the expected return of following the most favorable strategy. Q-learning path planning steps involve initializing the Q-table by assigning initial Q-values to all state-action pairs (s, a) . The process follows a loop for each time step t until the termination condition is met. In this process, the action A_t is chosen based on the current state s and the ϵ -greedy policy. After selecting an action, it is executed, leading to state $s+1$, and the agent receives an immediate reward $R+1$. The Q-value ($Q(s, a)$) is then updated according to the Q-learning update rule. Following this, the state transitions to $s \leftarrow s+1$, and the process continues to the next time step. Ultimately, starting from the initial point, the path is selected based on actions that maximize the Q-value until the goal is reached. The Q-value computation in Q-learning is as follows:

$$Q(s, a) = R(s, a) + \gamma \max_{\tilde{a}} \{Q(s, \tilde{a})\} \quad (1)$$

ACC adaptive cruise control, as an intelligent automatic control system, assists the driver in operating the steering wheel and brakes effectively. Its parameters include speed and distance. If there is no vehicle ahead, the system drives at the speed set by the driver. If the vehicle ahead decelerates, preventing the set speed from being reached, the ACC system maintains a safe following distance and applies brakes or adjusts speed as necessary.

The Q-learning algorithm can control the vehicle's driving or braking systems to automatically adjust speed, ensuring the vehicle maintains a safe following distance. This process can be viewed as the agent continuously exploring the environment and learning the optimal behavior strategy to adapt to different driving scenarios and driver needs.

3. ACC Combined with Reinforcement Learning

The pseudocode for the intelligent cruise method based on reinforcement learning applied in this paper is as follows:

Step 1: Select an action. In the state s derived from the current Q-value estimation, choose an action a . However, if all Q-values are initially zero, what action should we select? This highlights the importance of the exploitation trade-off.

Step 2: The number of actions is set to 6, and the exploration is performed 1000 times with the target room being Room 5, the discount rate is 0.8. Before each exploration, the current position and action of the control body are randomized by generating a random number.

Step 3: As the control body progresses towards the destination, at each step, the control body iterates the reward function R matrix, checking whether the R -value is greater than zero, i.e., determining whether the action is optimal.

Step 4: If the action is appropriate, the control body records this action in the experience function Q matrix. During each iteration, different action values are compared, and after comparison, the Q-value is iterated according to the formula. This process repeats until the Q-table is fully updated.

Step 5: Once the Q-value iteration is complete, test the control body's movement from the initial coordinate to the endpoint coordinate. End.

The flowchart is shown in Figure 1:

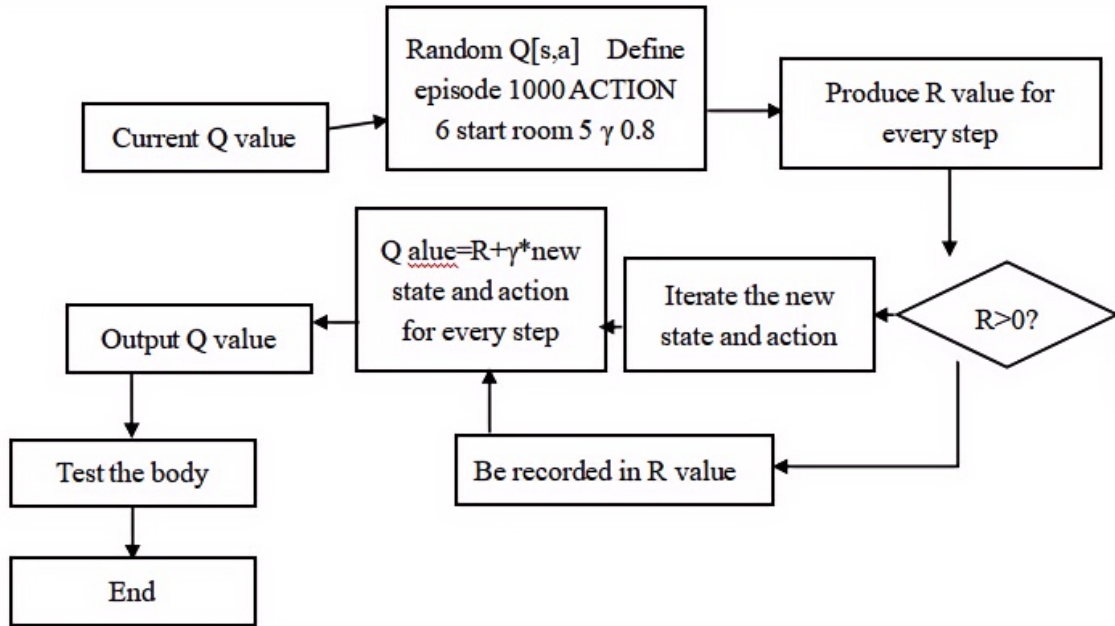


Figure 1. Algorithm Flowchart

4. Experiment

The hardware environment for running the algorithm in this paper is as follows: CPU 12th Gen Intel(R) Core(TM) i5-1240P, GPU Orayldd Driver Device, Memory 16GB. The software environment is: OS Windows 11, Visual Studio 2022. The results of the algorithm are shown in Figures 2, 3, 4, and 5. The relevant data during the operation of the algorithm in this paper are shown in Table 1,2,3,4.

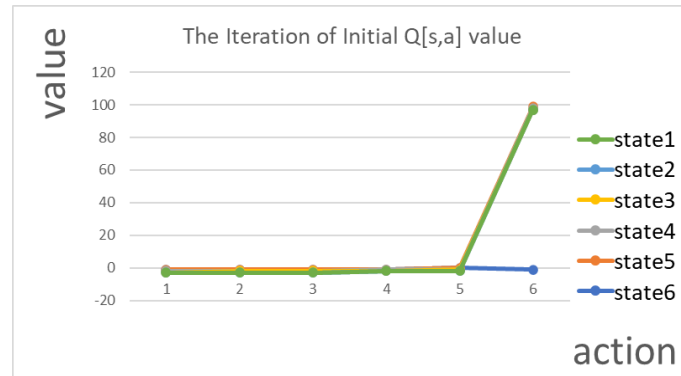


Figure 2. At the beginning of the iteration, the Q-values are distributed around 0, with only one value being 100.

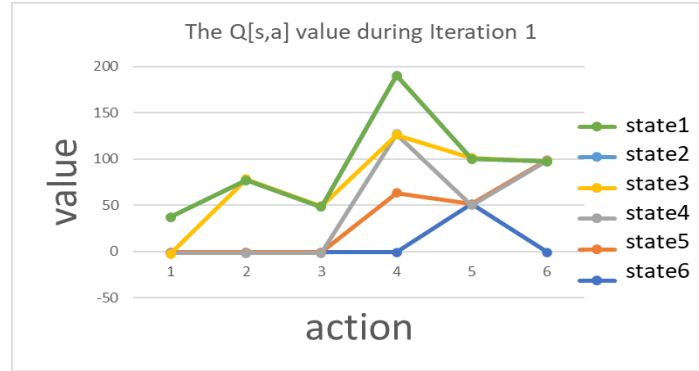


Figure 3. During the iteration, the Q-values continuously iterate and randomize.

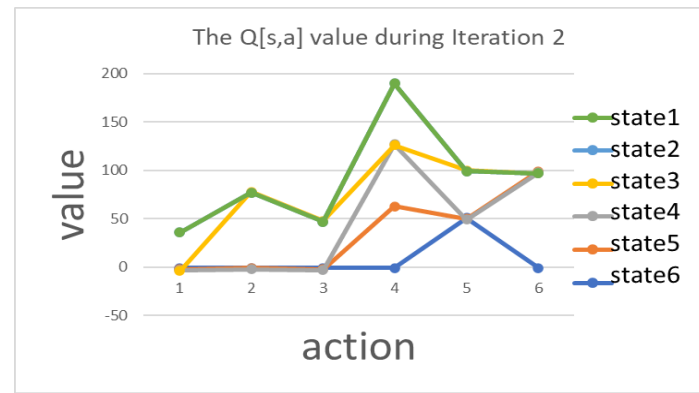


Figure 4. In the process of iteration, the changes in Q-values are not significant, with only some values changing from 0 to -1.

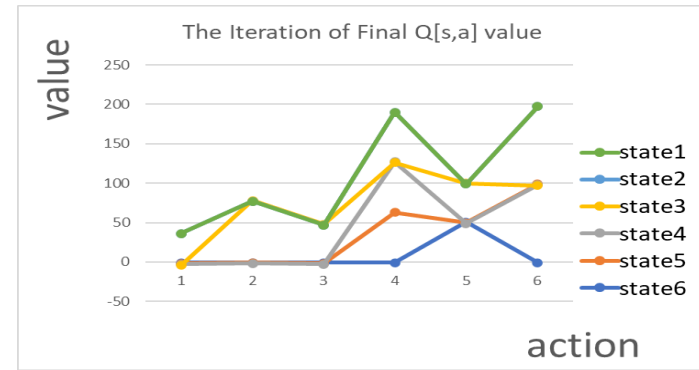


Figure 5. At the end of the exploration, the Q-value distribution curve becomes diversified and randomized.

Table 1. At the beginning of the iteration, the Q-values are distributed around 0, with only one value being 100.

	S1	S2	S3	S4	S5	S6
A1	-1	-1	-1	-1	0	-1
A2	0	0	0	0	0	100
A3	-1	-1	-1	0	-1	-1
A4	-1	0	0	-1	0	-1
A5	0	-1	-1	0	-1	0
aA6	0	0	0	0	0	0

Table 2. During the iteration1, the Q-values continuously iterate and randomize.

	S1	S2	S3	S4	S5	S6
A1	-1	-1	-1	-1	51	-1
A2	-1	0	-1	64	-1	100
A3	-1	-1	-1	64	-1	-1
A4	-1	80	51	-1	51	-1
A5	40	-1	-1	64	-1	100
A6	0	0	0	0	0	0

Table 3. During the iteration2, the Q-values continuously iterate and randomize.

	S1	S2	S3	S4	S5	S6
A1	-1	-1	-1	-1	51	-1
A2	0	0	0	64	0	100
A3	-1	-1	-1	64	-1	64
A4	-1	80	51	-1	51	-1
A5	40	-1	-1	64	-1	0
A6	0	0	0	0	0	0

Table 4. At the end of the exploration, the Q-value distribution curve becomes diversified and randomized.

	S1	S2	S3	S4	S5	S6
A1	-1	-1	-1	-1	51	-1
A2	-1	0	-1	64	-1	100
A3	-1	-1	-1	64	-1	-1
A4	-1	80	51	-1	51	-1
A5	40	-1	-1	64	-1	0
A6	0	0	0	0	0	0

5. Conclusion

The paper demonstrates the role of Q-learning in ACC. Environmental modeling and decision-making: Signals detected by radar sensors are transmitted to the ACC system, and the Q-learning algorithm makes decisions based on environmental changes (such as the speed and distance of the vehicle ahead). Through continuous trial and error and learning, the algorithm optimizes the car-following strategy. Reinforcement learning is a computational method that understands and automates decision-making and learning processes aimed at achieving specific goals. It utilizes the MDP as a formalized framework to define the interaction between the learning agent and the environment using states, actions, and rewards. Reinforcement learning emphasizes the agent's interactive learning process with the environment, without requiring imitable supervisory signals or complete modeling of the cyclic environment. The MDP provides the theoretical framework for interactive learning to achieve objectives. All reinforcement learning problems are modeled as an MDP, which serves as the foundational framework for reinforcement learning. By updating the state-action value function $Q(s,a)$, the Q-learning algorithm selects actions in each state that yield the highest long-term reward, enabling intelligent control. The experiments show that the method used in this paper can ensure both driving safety and comfort. The method can be practically applied to solve the difficulties drivers face, such as judging the distance to the vehicle ahead in real-time. The research reveals that intelligent cruise control includes scenarios such as constant-speed cruising, approaching the vehicle ahead, and situations where the vehicle ahead cuts in or cuts out. The calculation principle is based on the onboard sensors receiving signals from obstacles ahead, which are reflected into the computing system. The reinforcement learning algorithm iteratively derives the optimal experience value and transmits it to the control center, which then adjusts the power

output of the transmission system. This ensures that speed and distance meet safety and comfort standards.

References

- [1] Onur Cihan,Zeynep Musul.(2024)A novel system architecture of intelligent adaptive cruise control for safety aspects.Computer Science, Environmental Science, Engineering,10.18245/ijaet.1406829.
- [2] J. S. Saputro,Miftahul Anwar,F. Adriyanto,Agus Ramelan.(2024).Design of intelligent cruise control system using fuzzy-PID control on autonomous electric vehicles prototypes.Journal of Mechatronics, Electrical Power, and Vehicular Technology.10.55981/j.mev.2024.877.
- [3] Milad Andalibi,Alireza Shourangizhaghighi,Mojtaba Hajihosseini.(2023).Design and Simulation-Based Optimization of an Intelligent Autonomous Cruise Control System.Computers.10.3390/computers12040084.
- [4] Tianchen Ruan,Hao Wang,Rui Jiang,Xiaopeng Li,Ning Xie(2024).A General Hierarchical Control System to Model ACC Systems.10.1109/TITS.2023.3306349.
- [5] Faezeh Farivar,Mohammad Sayad Haghighi,Alireza Jolfaei(2020).On the Security of Networked Control Systems in Smart Vehicle and its Adaptive Cruise Control.Arxiv
- [6] Das, L. (2021). SAINT-ACC: Safety-aware intelligent adaptive cruise control for autonomous vehicles using deep reinforcement learning. arXiv, 2104.06506v2 [cs.RO].
- [7] Cihan, O., & Musul, Z. (2024). A novel system architecture of intelligent adaptive cruise control for safety aspects. International Journal of Automotive Engineering and Technologies.
- [8] Pananurak, W., Thanok, S., & Parnichkun, M. (2009). Adaptive cruise control for an intelligent vehicle. 2008 IEEE International Conference on Robotics and Biomimetics.
- [9] Zhang, J. D., & Zhong, H. T. (2023). Curve-based lane estimation model with lightweight attention mechanism. Signal, Image and Video Processing, 17, 2637–2643.
- [10] Ruan, S., Ma, Y., & Yan, Q. (2021). Adaptive cruise control for intelligent electric vehicles based on explicit model predictive control. Lecture Notes in Electrical Engineering.
- [11] Zhang, J., & Dou, J. (2024). An adversarial pedestrian detection model based on virtual fisheye image training. Signal, Image and Video Processing, 18, 3527–3535.
- [12] Kılıç, İ., Yazıcı, A., Yıldız, Ö., Özçelikors, M., & Ondoğan, A. (2015). Intelligent adaptive cruise control system design and implementation. 2015 10th System of Systems Engineering Conference (SoSE).
- [13] Zhang, Y., Lin, Y., Qin, Y., Dong, M., Gao, L., & Hashemi, E. (2024). A new adaptive cruise control considering crash avoidance for intelligent vehicles. IEEE Transactions on Industrial Electronics, 688–696.
- [14] Jin, J. Y., & Zhang, J. D. (2024). 3D multi-object tracking with boosting data association and improved trajectory management mechanism. Signal Processing, 218, 109367.
- [15] Vahidi, A., & Eskandarian, A. (2004). Research advances in intelligent collision avoidance and adaptive cruise control. IEEE Transactions on Intelligent Transportation Systems, 143–153.
- [16] Desjardins, C., & Chaib-draa, B. (2011). Cooperative adaptive cruise control: A reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 1248–1260.
- [17] Krishnamurthy, V. (2015). Reinforcement learning: Stochastic approximation algorithms for Markov decision processes. arXiv, 1512.07669.