Car Following Model and Algorithm Design based on Reinforcement Learning

Jie Ren*
College of Transport and Communications, Shanghai Maritime University, Shanghai, 201306, China.

* Corresponding author: 202030610123@stu.shmtu.edu.cn

Abstract. Based on reinforcement learning technology, this paper establishes a new driverless car following model. DQN algorithm and traffic simulator are mainly used to train the agent, and the following model is finally obtained. Under the precise and controllable experimental environment, the preset optimization targets can achieve the expected assumption and complete the following behavior. This study will contribute to the development of unmanned vehicles in the future.

Keywords: Driverless car following model, DQN, traffic simulator.

1. Introduction
For many years, the following model is the basic model in the field of traffic engineering, has been widely used, with the development of science and technology, the following model is also developing. M Zhou [1] proposed a micro-car following model based on cyclic neural network, which can accurately capture and predict traffic oscillations and is superior to the classical car following model in predicting traffic oscillations with different driver characteristics. S Hao [2] proposed a data-driven car following model based on rough set theory, which can well simulate the follower's micro traffic behavior. M Klawtanong [3] uses a random following model with modified optimal speed to study the dynamic characteristics of traffic flow on a circular road, which can improve the overall speed and traffic flow of the system and delay traffic congestion. GK Kamath [4] studied the impact of DAF on the classic car following model (CCFM) and reported a related application in which DAF reduced the performance of several metrics of practical interest. G Ma [5] proposed a following model that considered the aftersight effect and memorized the change of headway, which could stabilize the traffic flow better.

Car following behavior contains various factors, which are stochastic and complex. Therefore, according to the actual driving situation, this paper adjusted the reinforcement learning algorithm, optimized the constraint conditions and objective function, and established the reinforcement learning following model. Finally, the numerical simulation method was used to analyze the characteristics of different driving behaviors, and random factors or human factors were introduced to ensure the realism of simulation.
2. Car following model

2.1. Model Introduction
In 1958, Chandler proposed CA model, which mainly focused on how to avoid vehicle collisions. The model can be expressed as follows:

\[
\Delta \chi(t - 0.5) = -0.00028 \dot{\chi}(t - 0.5) + 0.00028 \ddot{\chi}(t) + 0.585 \dot{\chi}(t) + 4.1
\]  

(1)

Since human driving behavior is not optimal, this paper hopes to jump out of this research mode and realize self-training of more efficient following behavior without human driver's driving behavior as the basis.

The premise of this study is that the trajectory of the preceding vehicle within T seconds from now is known, but the speed is unknown. The main content of this paper is how to control the acceleration of the rear vehicle so as to ensure that there is no safety problem within T seconds. The time range of control is from 0 to T seconds after the current time.

2.2. Restraint condition
(1) The control variable of the vehicle is the acceleration \( a(t) \) of the vehicle behind it. The acceleration of the vehicle must meet the mechanical properties of the vehicle

\[
-d_{\text{max}} \leq a(t) \leq a_{\text{max}}
\]

(2)

Among them: \( a(t) \) is the function of the acceleration of the vehicle behind with time, \( a_{\text{max}} \) represents the maximum acceleration of the vehicle, \( -d_{\text{max}} \) represents the minimum deceleration of the vehicle.

(2) Speed is the first integral of acceleration. In addition, the scenario of this study requires that vehicles cannot retreat. Therefore, the speed must meet

\[
0 \leq v(0) + \int_0^t a(\tau)d\tau \leq v_{\text{max}}
\]

(3)

Among them: \( v(0) \) is the speed of the car behind at the initial moment, \( v_{\text{max}} \) is the maximum speed allowed by vehicles and roads.

(3) In order to ensure the safety of the vehicle, at any time, the front distance between the rear car and the front car should be kept at least a safe distance

\[
L + g_{\text{min}} \leq s_f(t) - s(t)
\]

(4)

Among them: \( L \) is the length of the vehicle in front, \( g_{\text{min}} \) is the minimum safe interval, \( s_f(t) \) is the position of the vehicle in front at time \( t \), \( s(t) \) is the position of the vehicle behind at time \( t \).

2.3. Boundary condition
There are two types of boundary conditions for following the vehicle. One is that the trajectory of the vehicle ahead is known within the optimized time range. Second, the trajectory of any vehicle is not stipulated in advance, and the operation of vehicles is limited by road or traffic control conditions.

2.4. Optimization objectives
(1) An appropriate expected headway should be selected for the speed of the rear vehicle, that is, the expected headway is a function of the speed. The optimization objective of the desired headway is that the headway of the rear vehicle is as close as possible to the desired headway during the whole driving process.
Among them: \( \text{abs}[] \) is an absolute value function. In general, the expected headway is a linear function of the speed of the rear vehicle.

(2) Reduce fuel consumption and improve vehicle comfort. Fuel consumption is usually a function of vehicle speed and acceleration. The commonly used fuel consumption function can be expressed as follows

\[
F(t) = \begin{cases} 
\sum_{p=0}^{3} \sum_{q=0}^{3} W_{p,q} \cdot v(t)^p \cdot a(t)^q & \text{for } a(t) \geq 0 \\
\sum_{p=0}^{3} \sum_{q=0}^{3} M_{p,q} \cdot v(t)^p \cdot a(t)^q & \text{for } a(t) \leq 0 
\end{cases}
\]

Among them: \( W_{p,q} \) and \( M_{p,q} \) are pre-selected weighting coefficients.

The premise of pursuing the lowest fuel consumption is that the fleet must drive forward, and the objective function of optimization is

\[
\max (1 - \gamma) \int_{t_0}^{t_d} \text{abs}(s_i(t) - s(t)) - f(v(t)))dt \\
- \gamma \int_{t_0}^{t_d} F(t)dt,
\]

Among them, \( 0 \leq \gamma < 1 \) is the weight coefficient of fuel consumption, the greater the \( \gamma \) means the lower the fuel consumption, the smaller the \( \gamma \) means the longer the driving distance.

(3) Ease traffic congestion and reduce speed fluctuation of vehicles in traffic flow. The absolute value of acceleration is a major factor reflecting velocity fluctuation

\[
A_i(T) = \int_{t_0}^{t_d} |a(\tau)| d\tau.
\]

The objective function of optimization is

\[
\max (1 - \gamma) \int_{t_0}^{t_d} \text{abs}(s_i(t) - s(t)) - f(v(t)))dt \\
- \gamma \int_{t_0}^{t_d} |a(\tau)| d\tau,
\]

3. Algorithm design

3.1. DQN algorithm design

The modeling of Deep Q Learning mainly realizes five elements: state, behavior, state transfer, reward and reduction factor.

The behavior adopts the acceleration \( a(t) \) of the car behind at each moment. The state is the speed of the vehicle in front, \( v_i(t) \), the speed of the vehicle behind, \( v(t) \), and the headway of the vehicle behind \( s(t) \). Since the distance between the vehicle and the starting point basically has no effect on the acceleration, vehicle position \( s(t) \) is not applied here as the state value, which can ensure the universality and robustness of DQN. Before modeling, time should be discretized and the time interval is \( \Delta t = 0.05s \).

The state transition equation is expressed as follows

\[
v(t + T) = v(t) + a(t) \cdot T
\]

\[
s(t + T) = s(t) + v(t) \cdot T + \frac{1}{2} a(t) \cdot T^2
\]

\[
\text{space}(t) = s_i(t) - s(t)
\]
Assuming that all vehicles are precisely controlled, the above state transition function can be used. If it is assumed that the environment has random factors and the vehicle is not precisely controllable, the value of the above state function can be used as the expected value, and then an appropriate probability model can be selected to randomly generate a new state according to the actual situation.

Reward is determined according to constraint conditions and objective function. If the following vehicle violates a constraint condition after completing the state transfer, it will return a very small Reward, which is -10 in this study. Return the value of the target function if no constraint is violated after the state transition. The reduction factor is usually 0.9.

### 3.2. Design of traffic simulator

The traffic emulator is to simulate the environment of reinforcement learning. At the beginning of the simulation, the traffic emulator will initialize all the data according to the requirements, and then return the state value of the initial state to the agent. The agent calculates the best behavior \(a(0)\) using Deep Q Network according to the state value \(S_{\text{state}}\). This behavior is then sent to the traffic emulator, which calculates the new state \(S_{\text{next state}}\) according to the state transition function, calculates the current Reward \(R(0)\) according to the Reward function, and returns the new state value and Reward to the agent. The agent stores all the information for this step \([S_{\text{next state}}, a(0), S_{\text{state}}, R(0)]\), selects the new action \(a(1)\) with the new state \(S_{\text{state}}\), and repeats until a turn attempt is completed or the turn is terminated by violating the constraint. Therefore, state transitions, Reward functions and constraints are implemented in the traffic simulator.

### 3.3. The objective function

1. The driving of the car behind has two main objectives: First, do not rear-end with the car in front. Second, as close as possible to follow the car in front. In the initial simulation, we calibrated an expected headway from the actual trajectory data

\[
Space(t) = 8 + 1.5 \times v
\]  

(13)

The objective is expected headway, that is, to ensure that the integral of the desired headway of the rear car is minimized throughout the driving process. As a discrete algorithm is used, the above objective function should be modified as

\[
\min \sum_{t} \text{abs}[(s_{r}(t) - s(t)) - f(v(t))]
\]  

(14)

The value of reward is not greater than 0, and the maximum reward value is 0. Therefore, the above objective needs to be rewritten as a function for the maximum value

\[
\max \sum_{t} \text{abs}[(s_{r}(t) - s(t)) - f(v(t))]
\]  

(15)

2. In order to reduce the fuel consumption and exhaust emissions of vehicles, the fuel consumption function needs to be taken into account. Referring to the research results of literatures, the fuel consumption function is as follows

\[
F(t) = \begin{cases} 
0.01 \times v(t) + 0.025 \times a(t) + 0.01 \times a(t)^2 & \text{for } a(t) \geq 0 \\
0.006 \times v(t) + 0.01 \times a(t)^2 & \text{for } a(t) \leq 0 
\end{cases}
\]  

(16)

If \(\gamma = 0.5\), the optimization objective is

\[
\max \sum_{t} \left(-\frac{1}{2} \text{abs}[(s_{r}(t) - s(t)) - f(v(t))] - F(t)\right)
\]  

(17)
In optimization, the previous coefficient 1/2 omitted does not affect the result of the solution. Therefore, in each step of operation, the reward function value is

\[ \text{Reward} = -\text{abs}[(s_{1}(t) - s(t)) - f(v(t))] - F(t) \] (18)

(3) In order to reduce the turbulence in the process of vehicle driving, it is necessary to adjust the vehicle with reduced acceleration. The smaller the acceleration is, the smoother the vehicle runs. The optimization objective of reducing acceleration is formula (2-8).

If \( \gamma = 0.5 \), the optimization objective is

\[ \text{max} \sum_{i=1}^{n} \frac{1}{2} \text{abs}[(s_{1}(t) - s(t)) - f(v(t))] - \text{abs}[a(t)] \] (19)

In optimization, the previous coefficient 1/2 omitted does not affect the result of the solution. Therefore, in each step of operation, the reward function value is

\[ \text{Reward} = -\text{abs}[(s_{1}(t) - s(t)) - f(v(t))] - \text{abs}[a(t)] \] (20)

3.4. The simulation results

This section analyzes the three following driving behavior goals identified above: expected headway, energy saving optimization, and comfort optimization. The experimental scene of numerical simulation is: the initial state of the front vehicle: acceleration 0, speed 10m/s, position 50m. When the car in front runs for 10s, it decelerates at the acceleration of -3m/s² until the speed decreases to 0, and keeps in a stop state until 16s, when it accelerates to 14m/s at the acceleration of 2m/s², and then runs at a uniform speed until the end.

First, the initial state and the running track of the vehicle in front were set in the traffic simulator, and the expected headway was taken as the objective function. Then, the vehicle behind was allowed to learn 30,000 steps. The initial speed of the rear car is the same as that of the front car, and the initial position and initial acceleration are both 0. Deep Q Learning algorithm is used to calculate the optimal driving trajectory of the rear vehicle with different driving targets.

1) The desired headway is taken as the objective function. The simulation results are as follows

![Figure 1. Vehicle track diagram](image1.png)

![Figure 2. Headway diagram](image2.png)

![Figure 3. Speed change diagram](image3.png)
(2) Using the optimization function of eco-driving as the objective function. The simulation results are as follows

![Vehicle track diagram](image1)

**Figure 4. Vehicle track diagram**

![Headway diagram](image2)

**Figure 5. Headway diagram**

![Speed change diagram](image3)

**Figure 6. Speed change diagram**

(3) Using soft-driving optimization function as the objective function. The simulation results are as follows

![Vehicle track diagram](image4)

**Figure 7. Vehicle track diagram**

![Headway diagram](image5)

**Figure 8. Headway diagram**

![Speed change diagram](image6)

**Figure 9. Speed change diagram**

The main advantage of soft-driving is that the driving is smoother, the speed fluctuation is smaller, and the speed curve is obviously gentler than the first two driving modes.
4. Example analysis

The study was carried out by cross experiments with reaction time of 0.1, 1.0 and 2.0. Randomness is carried out by means of random noise. After the simulator receives the control variable, a random variable is added on the basis of the control variable as the actual acceleration. Therefore, the agent cannot accurately control the vehicle. The uniform distribution model was adopted for random variables, and two schemes were adopted: [-1,1] and [-2,2]. A total of 6 experiments were carried out.

The objective of optimization is the desired headway. Intuitively, if the reaction time is larger or the random factor is larger, the desired parameter $\alpha$ of the headway is larger. The binary method was used to calibrate the parameter $\alpha$. 

\[ Space(t) = 8 + \alpha * v \]  

(21)

The calibration steps are as follows:

- Step1. Take $\alpha$ equal to 0 and train 5000 times. In order to ensure the accuracy of training, random term is not added during training, but added during running. Then run for 5 times. If the rear car can run smoothly without crashing, the parameter can be considered appropriate. If the car crashes, the value of $\alpha$ is considered too small. As the minimum $a$;
- Step2. Take a fairly large parameter to ensure the successful operation of the rear vehicle. As the maximum $b$;
- Step3. Take the appropriate point $c$ between $a$ and $b$ to carry out the experiment. If successful, replace $b$ with $c$; if failed, replace $a$ with $c$;
- Step4. Repeat Step 3 until the distance between $a$ and $b$ is less than 0.1.

4.1. The experimental results

The experiment uses the shortest response time and slight random variables in the most adverse case. After 6 experiments, $\alpha$ results were summarized, as shown in the table below.

| Number of experiments | Reaction time | Random interval | $\alpha$ |
|-----------------------|---------------|-----------------|----------|
| 1                     | 2.0           | [-2,2]          | 2.1      |
| 2                     | 2.0           | [-1,1]          | 1.7      |
| 3                     | 1.0           | [-2,2]          | 0.8      |
| 4                     | 1.0           | [-1,1]          | 0.8      |
| 5                     | 0.1           | [-2,2]          | 1.0      |
| 6                     | 0.1           | [-1,1]          | 1.0      |

4.2. Results analysis

When the response time is 2.0 seconds, the greater the randomness, the further the following distance is required, which is consistent with common sense.

When the reaction time is reduced to 1.0 seconds, the following distance is significantly reduced. It shows that shortening reaction time can improve road efficiency.

When the reaction time is 1.0 seconds, the randomness has almost no effect on the value of $\alpha$. In the two random scenarios, the experiment failed when $\alpha$ was 0.7 and succeeded when $\alpha$ was 0.8. It shows that the shorter reaction time can compensate for the randomness of control.

When the reaction time is 0.1 seconds, the value of $\alpha$ increases. This feature is quite countinuitive, mainly due to the influence of the characteristics of Deep Q Learning, the accuracy of neural network is not enough, resulting in the failure of car-following simulation. In practical application, the reaction time and the scale of neural network can be selected according to the needs.
5. Conclusions
Experiments show that the car-following model based on reinforcement learning designed in this paper can train good agents to complete car-following behavior in an accurate and controllable environment according to three feasible optimization objectives.
Although the work in this paper has made some progress in the field of unmanned driving, there are still deficiencies and limitations. For example, the preset environment is precise and controllable, the simulation scale is small, and the training time is limited, which increases the difficulty of implementing the model in practice, and is also the work direction that needs to be made up and improved in future research.

References
[1] Zhou M, Qu X, Li X. A recurrent neural network based microscopic car following model to predict traffic oscillation [J]. Transportation Research Part C Emerging Technologies, 2017, 84:245-264.
[2] Hao S, Yang L, Shi Y. Data-driven car-following model based on rough set theory [J]. IET Intelligent Transport Systems, 2018, 12(1):49-57.
[3] Klawtanong M, Limkumnerd S. Dissipation of traffic congestion using autonomous-based car-following model with modified optimal velocity [J]. Physica A: Statistical Mechanics and its Applications, 2020, 542.
[4] Kamath G K, Jagannathan K, Raina G. Impact of delayed acceleration feedback on the classical car-following model [J]. IMA Journal of Applied Mathematics, 2020, 85(4):584-604.
[5] Ma G, Ma M, Liang S, et al. Nonlinear analysis of the car-following model considering headway changes with memory and backward looking effect. 2021.